WHAT DOES LOGIT DISTILLATION PRESERVE IN PRE-TRAINED CLASS-INCREMENTAL LEARNING?

Anonymous authors
Paper under double-blind review

ABSTRACT

Distilling stored predictions is a standard remedy for forgetting in class-incremental learning. However, in stored-logit replay, many of the stored targets describe classes the model has never trained on: with a fixed classification head, every saved example carries outputs for classes not yet observed. Discarding these unsupervised constraints and distilling only over observed classes appears to be a harmless simplification. In this paper, we investigate what these constraints preserve and show that the simplification is not harmless. Removing these constraints consistently helps the model acquire new classes, but its net effect on accuracy reverses between training settings. Amplifying the reduced distillation signal to the strength of the full constraint does not recover full-distillation accuracy: the remaining loss concentrates on discrimination between classes learned at different times rather than within any single task. Probing the removed terms reveals why. The apparently unsupervised coordinates act strongly on classes learned after the targets were stored, and thereby restrict updates that later tasks depend on. Stored-logit constraints thus preserve cross-task discrimination in a way that their magnitude alone does not explain, and their value must be judged by which updates they restrict rather than by how strongly they constrain.

1 INTRODUCTION

A continual learner must preserve useful knowledge without losing the ability to learn what comes next. In class-incremental learning, this tension arises as new classes expand the classification problem (De Lange et al., 2022; Zhou et al., 2024). Distillation offers a natural response: constrain the updated model to preserve its earlier predictions (Li & Hoiem, 2018; Rebuffi et al., 2017). Yet preserving a prediction and preserving useful knowledge are not the same operation. A constraint that reduces forgetting can also make new classes harder to acquire, and this cost is rarely visible from a forgetting score alone.

Pre-trained representations sharpen this question. They support useful distinctions before incremental training begins, but adaptation can change those distinctions in ways that later tasks depend on (Zhou et al., 2025; Wang et al., 2022b). Distillation must therefore decide which outputs should remain stable and which should be allowed to change. Judging that decision requires examining the constraints themselves, asking what each of them actually preserves, rather than reading only the final forgetting score.

Stored-logit replay exposes this balance in a particularly revealing form. In this family of methods, an example is saved together with the outputs that the model produced when the example entered memory (Buzzega et al., 2020). When the classification head contains all classes from the outset, as is common in pre-trained pipelines, these stored outputs also cover classes not yet encountered in training. Such values are not supervised predictions learned from future data. They are by-products of a head that has never been trained on those classes. It therefore seems reasonable to discard these *unseen-at-storage* outputs and distill only over classes already observed. Whether this simplification is harmless, however, is not obvious: even unsupervised outputs restrict how the shared representation is allowed to move, and later tasks must be learned within those restrictions. *Does removing these constraints improve learning, or does it discard a useful source of stability?*

This paper answers this question. Doing so requires more than a direct comparison, because removing a constraint rarely comes alone. Masking the unseen-at-storage outputs removes error terms

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

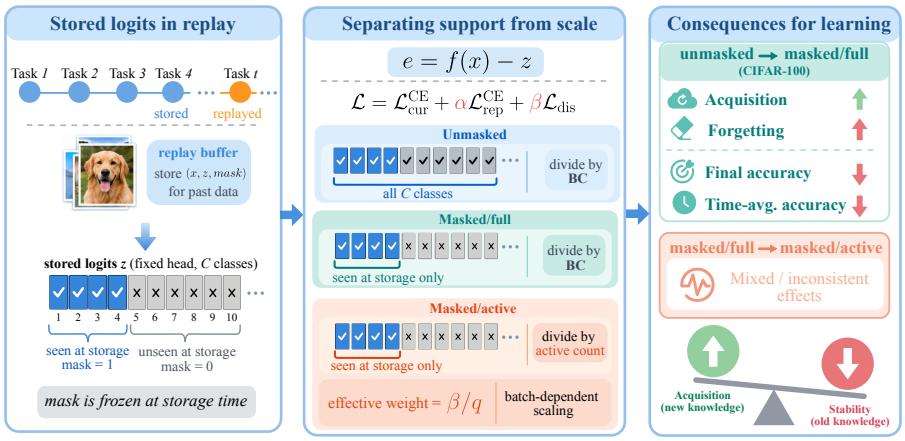


Figure 1: Separating support from scale in stored-logit distillation. Masked/full removes unseen-at-storage errors; masked/active additionally amplifies the retained terms by $1/q$, where q is the active-coordinate fraction. The right panel summarizes the CIFAR-100 results, not a universal tradeoff. Section 4 defines the objective and symbols. ImageNet-100 experiments test setting dependence and same-state gradient-norm compensation.

from the distillation loss, but the standard objective averages over the outputs that remain, so every retained term silently receives a larger weight. A comparison that changes both choices at once cannot attribute its outcome to either. We therefore decompose the objective into its support, its averaging denominator, and its nominal distillation weight, as Figure 1 illustrates, within a DER++-style (Buzzega et al., 2020) recipe that combines current-data classification, label-based replay, and stored-logit regression. This decomposition allows us to delete the unseen-at-storage constraints with or without the induced amplification of the remaining terms, and to observe what each choice costs and buys.

The results of this separation do not match the cleanup intuition. On CIFAR-100, deleting the unseen-at-storage constraints improves the acquisition of new classes but worsens forgetting and accuracy, and this tradeoff is consistent across the tested class orders and distillation weights in both pipelines. On ImageNet-100, the same deletion instead improves acquisition, retention, and both accuracy measures in one pipeline. The operation is therefore neither harmless nor universally harmful; its net value depends on the training setting. This contrast sharpens the central question: does the value of the omitted constraints come from how strongly they restrict an update, or from which update directions they constrain?

We answer this question with a compensation test. The masked distillation gradient is amplified to match the full-gradient norm at the same model state and replay batch, so that any remaining difference reflects the direction of the constraint rather than its magnitude. If the strength of the constraint were the whole story, this compensation should close the gap. It does not: across a separate set of ImageNet-100 order–seed pairs, the masked direction at matched magnitude still loses substantial final accuracy in both pipelines, and the loss concentrates in cross-task discrimination rather than within-task classification. Local probes of loss residuals and parameter gradients point to an explanation: many omitted coordinates correspond to classes learned after the targets were stored, so the removed terms constrain features that later tasks actively use, not only untouched future outputs. Together, these results support a central finding: *stored-logit constraints preserve cross-task discrimination in a way that their magnitude alone does not explain, and their value must be judged by which updates they restrict rather than by how strongly they constrain*.

Our contributions are as follows.

- We decompose stored-logit distillation into support, averaging denominator, and nominal weight, and show that changing several of these choices at once, as complete recipes do, can obscure the underlying component tradeoffs.
- We show through paired CIFAR-100 and ImageNet-100 experiments that restricting the support improves acquisition at a stability cost across class orders and distillation weights, and that this tradeoff reverses in one ImageNet-100, where all four outcomes improve.

- We reject a scalar-strength explanation with same-state gradient-norm matching across seven independent ImageNet-100 order–seed pairs, locate the accuracy loss mainly in cross-task discrimination, and show through local gradient measurements that the omitted constraints act strongly on classes learned after storage.

2 Related Work

Continual learning methods constrain parameters, predictions, or training data to reduce forgetting (Kirkpatrick et al., 2017; Li & Hoiem, 2018; Chaudhry et al., 2019), and class-incremental learning additionally requires distinguishing old and new classes without task identifiers (van de Ven et al., 2022; Zhou et al., 2024). Distillation supports this goal through prediction preservation in Learning without Forgetting and iCaRL or through feature preservation in PODNet (Li & Hoiem, 2018; Rebuffi et al., 2017; Douillard et al., 2020). Replay-based variants revisit stored examples together with their predictions: Dark Experience Replay regresses to stored logits and DER++ adds label-based replay (Chaudhry et al., 2019; Buzzega et al., 2020), X-DER handles subsequently learned classes through target updates, future-output preparation, and constraints on past and future outputs (Boschini et al., 2023), and Idempotent Experience Replay constrains predictions through an idempotence objective (Liu et al., 2026). We study the fixed-target setting, separating support deletion, averaging, and weight rather than changing the targets or how predictions are preserved.

Pre-training changes the balance between generalization and adaptation in class-incremental learning (Zhou et al., 2025). Prompt-based methods control adaptation through learned prompts (Wang et al., 2022b;a; Smith et al., 2023), while random projections and low-rank updates offer alternative ways to use a pre-trained model (McDonnell et al., 2023; Liang & Li, 2024; He et al., 2025; Wu et al., 2025). Because architecture also affects continual learning behavior (Mirzadeh et al., 2022), we estimate objective-level effects within two fixed pipelines, treating architecture, checkpoint, and training recipe as a joint setting rather than attributing differences to architecture alone.

Forgetting and intransigence distinguish retaining earlier tasks from learning later ones (Chaudhry et al., 2018), and metric and protocol studies caution against judging a continual learner through one score (Díaz-Rodríguez et al., 2018; Hsu et al., 2018; Soutif-Cormerais et al., 2023). Relative measures introduce a reference for interpreting performance (Hayes et al., 2018; Serrà et al., 2018; Michel et al., 2023). We evaluate acquisition throughout the sequence alongside forgetting, final accuracy, and time-averaged accuracy to expose the learning costs of preservation, with relative repair supplying additional effect-size context.

3 Setting and Evaluation

To determine what a constraint preserves, we measure both what the learner acquires and what remains after later updates. We study continual learning over a sequence of T disjoint class groups. At each stage t, the learner receives training examples from the current group and maintains a bounded memory for replay. Evaluation requires discrimination among all observed classes without task identifiers. Let $A_{t,k} \in [0, 1]$ denote the accuracy on group k after completing stage t. The four metrics below capture complementary aspects of this learning trajectory. We refer to the seven-condition, two-pipeline evaluation that follows as the *component study*. These metrics average task-level quantities in fraction units, with differences in percentage points (pp) unless stated otherwise. Each task contains an equal number of classes.

Acquisition, retention, and overall performance. Acquisition measures how well each group is learned when it first appears. Final average accuracy (FAA) measures the final classifier, whereas time-averaged incremental accuracy ($\mathrm{AIA}_{\mathrm{time}}$) also tracks performance along the way:

$$\mathrm{Acquisition} = \frac{1}{T}\sum_{t=1}^{T} A_{t,t}, \qquad \mathrm{FAA} = \frac{1}{T}\sum_{k=1}^{T} A_{T,k}, \tag{1}$$

$$\mathrm{AIA}_{\mathrm{time}} = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{t}\sum_{k=1}^{t} A_{t,k}.$$

We also report *final old-task accuracy*, $\frac{1}{T-1}\sum_{k=1}^{T-1} A_{T,k}$, which averages the final-stage accuracies of all but the last group, isolating retained performance from the final acquisition stage. Average forgetting (AF) measures the decline of each earlier task from its best observed accuracy:

$$\mathrm{AF} = \frac{1}{T-1}\sum_{k=1}^{T-1}\left(\max_{k\le j\le T} A_{j,k} - A_{T,k}\right). \tag{2}$$

The last task is excluded because no subsequent learning stage follows it. Including the final stage in the maximum makes this quantity nonnegative. Because an intervention can change the peak as well as the subsequent decline, a lower AF must be interpreted alongside acquisition and accuracy. We measure acquisition over the full sequence rather than substituting the final current-task accuracy $A_{T,T}$.

Controlled study. The component study splits CIFAR-100 (Krizhevsky, 2009) into ten tasks of ten classes. Five random class orders (exact sequences in Appendix I), three optimization seeds per order, and seven conditions give 210 runs across two pre-trained pipelines: ResNet-18 from ImageNet-1K and ViT-Tiny from AugReg ImageNet-21K followed by ImageNet-1K fine-tuning (Steiner et al., 2021). Both update all parameters with a randomly initialized, fixed 100-output head and 224×224 inputs. Training uses 2,000 memory examples, 20 epochs per stage, batches of 64, and SGD with initial learning rate .005. Replay reuses stored augmented inputs and fixed logits, with sampling and memory-update streams paired across conditions. Appendix A specifies storage semantics and differences from reference DER++, and Appendix B gives transformations and randomization settings. We use *retention* for preserved earlier performance, reflected in lower AF and higher final old-task accuracy.

Paired effects and uncertainty. We define each component effect as the treatment-minus-control difference within the same pipeline, class order, and optimization seed. We average paired differences over the three seeds within each order, then take the unweighted mean over five orders. Component-effect tables report pointwise 95% Student-t intervals over these order means, as do the error bars in Figures 3 and 5. A positive AF effect indicates increased forgetting, whereas positive effects on the other three outcomes indicate improvement. Appendix B gives exact two-sided sign-flip sensitivities, Holm-adjusted p-values, and leave-one-order-out (LOO) summaries. With five orders, the minimum possible two-sided exact p-value is .0625. We therefore emphasize effect sizes and directional consistency without claiming multiplicity-adjusted significance. Appendix H summarizes the design fixed in advance, the run inventory, and the reproduction requirements.

4 Disentangling Distillation Support and Scale

A stored-logit objective specifies which outputs should remain stable and how strongly to penalize their changes. Masking and averaging can alter both choices together, so we change one at a time with replay examples and classification losses fixed.

Let x_i and z_{ic} denote the input and stored logit for replay example i and class coordinate c, and let $e_{ic} = f_\theta(x_i)_c - z_{ic}$ be the deviation of the current pre-softmax output of the parameterized model f_θ from that stored value. A replay batch contains B examples, and the fixed head has C outputs. We attach a binary mask M_{ic} to each example: it equals one for classes observed when that example entered memory. The mask remains fixed even after more classes arrive. For example, an image saved at the first ten-class task retains ten active coordinates when replayed during the final task. The other outputs exist in the head but do not provide access to future examples or labels.

We compare three regression losses:

$$\mathcal{L}_{\text{unmasked}} = \frac{1}{BC}\sum_{i,c} e_{ic}^2, \quad \mathcal{L}_{\text{mask,full}} = \frac{1}{BC}\sum_{i,c} M_{ic}e_{ic}^2, \quad \mathcal{L}_{\text{mask,active}} = \frac{\sum_{i,c} M_{ic}e_{ic}^2}{\sum_{i,c} M_{ic}}. \tag{3}$$

The training objective is $\mathcal{L}_{\text{cur CE}} + \alpha\mathcal{L}_{\text{rep CE}} + \beta\mathcal{L}_{\text{dis}}$, where $\mathcal{L}_{\text{dis}}$ is one of the three regression losses in Equation 3, CE denotes cross-entropy, and $\alpha = .1$. In this notation, β always multiplies the logit-regression loss. Both cross-entropy terms use the full head during training. Table 1 summarizes the seven conditions formed by the loss variants and weights.

Support restriction. At each fixed $\beta \in \{.1, .5\}$, we compare masked/full with unmasked distillation. The denominator remains BC, so the intervention deletes precisely the unseen-at-storage error terms. This contrast measures the total deletion effect, including changes in objective and gradient magnitude.

Active-output averaging. *Active-output averaging* compares masked/active with masked/full at the same weight, changing only the reduction denominator. Let $q = \sum_{i,c} M_{ic}/(BC) > 0$ be the active fraction. On a common batch and error tensor,

$$\mathcal{L}_{\text{mask,active}} = q^{-1}\mathcal{L}_{\text{mask,full}}. \tag{4}$$

For a batch with one tenth of the coordinates active, averaging multiplies each retained error term by ten. In general, it applies an effective weight β/q to both the masked loss and its parameter gradient at the same state. Because memory contains examples stored at different stages, q varies across batches, making this an implicit weight schedule. Appendix Table 8 summarizes the measured active fractions and resulting trajectories. The identity concerns a common state and batch, not average losses along separately trained trajectories.

Static weighting. The batch-dependent scaling above motivates a comparison with explicit weight changes. For the unmasked objective, we compare $\beta = 0$ with .1, and .1 with .5, distinguishing the introduction of distillation from the strengthening of an existing constraint. The $\beta = 0$ condition supplies an experience replay (ER) control under the same implementation. These compare specified objectives with a fixed ER control, not independently tuned methods.

Table 1: Seven-condition design grouped by support, averaging denominator, and nominal weight β. All conditions share current-data and replay cross-entropy terms. Experience replay (ER) sets distillation weight $\beta = 0$.

Condition	Support	Denominator	Weight β
ER	None	—	0
Unmasked	All head outputs	BC	.1, .5
Masked/full	Seen at storage	BC	.1, .5
Masked/active	Seen at storage	$\sum_{i,c} M_{ic}$	.1, .5

Matching magnitude while retaining the masked direction. Neither a shared nominal weight nor active-output averaging equates parameter-gradient norms. To test whether magnitude alone recovers the effect of full distillation, let $g_U = \nabla_\theta \mathcal{L}_{\text{unmasked}}$ and $g_M = \nabla_\theta \mathcal{L}_{\text{mask,full}}$ on the same replay batch at the current state of the masked learner. We replace its distillation gradient by

$$\widetilde{g}_M = \|g_U\|_2 \frac{g_M}{\|g_M\|_2}, \qquad g_{\text{train}} = g_{\text{cur CE}} + \alpha g_{\text{rep CE}} + \beta \widetilde{g}_M, \tag{5}$$

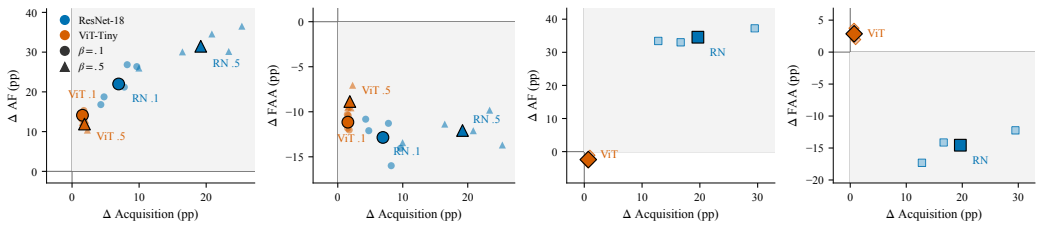
where $g_{\text{cur CE}} = \nabla_\theta \mathcal{L}_{\text{cur CE}}$ and $g_{\text{rep CE}} = \nabla_\theta \mathcal{L}_{\text{rep CE}}$ denote the parameter gradients of the current and replay cross-entropy losses, respectively. The norm covers all trainable parameters, and $\beta = .5$ in both treatment and unmasked control. We construct this vector directly, without differentiating the norm ratio or clipping its coefficient. The intervention thus retains the masked direction at the counterfactual full-gradient magnitude, matching the distillation component rather than the total optimizer update or gradients across two evolving trajectories. Appendix K specifies the numerical checks and paired evaluation.

5 Empirical Findings

We begin with the learning consequences of removing unseen-at-storage constraints, then examine why their value differs across settings. Active-output averaging and gradient-norm matching test two increasingly direct ways of compensating for the lost constraint strength. Comparisons with ER and complete method recipes place these findings in the broader acquisition–stability tradeoff.

5.1 Removing Constraints Exposes an Acquisition–Stability Tradeoff

On CIFAR-100, removing the unseen-at-storage error terms eases acquiring new classes at a substantial retention cost. Figure 2 shows the shift in both pipelines and at both weights: every order mean moves toward higher acquisition and forgetting, and final accuracy falls in every case. Notably, outputs least justified by past supervision still preserve useful performance.



(a) (b) (c) (d)

Figure 2: More acquisition, more forgetting, with a setting-dependent tradeoff. Masked/full minus unmasked effects at fixed denominator and weight, in pp. ΔAF appears in (a) and (c) and ΔFAA in (b) and (d), both against Δacquisition. Panels (a) and (b) report CIFAR-100: small points average three paired seeds per order and outlined points average five orders, and circles and triangles denote $\beta = .1$ and .5. Panels (c) and (d) report ImageNet-100 at $\beta = .5$: open markers are the three order means and filled markers their average, and squares and diamonds denote ResNet-18 (RN) and ViT-Tiny (ViT). Appendix Tables 4 and 10 give all four outcomes and intervals.

This tradeoff is consistent across class orders despite differences in magnitude between pipelines: the mean final-accuracy loss ranges from about 9 to 13 percentage points, and Appendix Table 4 shows that time-averaged accuracy also decreases. The order-level estimates agree in direction for every pipeline, weight, and outcome, and Appendix Table 7 shows that each leave-one-order-out mean retains that direction. The cost extends beyond the forgetting metric: easier acquisition accompanies worse classification during and after the sequence.

The same deletion need not produce the same tradeoff everywhere. On ImageNet-100, three class orders with three paired seeds each reproduce the ResNet-18 pattern at $\beta = .5$, but yield simultaneous improvements in acquisition, forgetting, and both accuracy metrics for ViT-Tiny, as Figures 2c and 2d show. Final-accuracy effects of -14.6 and $+2.9$ pp arise with all three orders agreeing within each pipeline (Appendix Table 10). The value of preserving these outputs thus depends on the training setting rather than an architecture label alone.

Where do the outcomes diverge? An analysis of the completed trajectories separates classification within a task from competition across tasks. Let W denote accuracy when prediction is restricted to the true task, $G = W - A$ the gap to all-class accuracy A (i.e., FAA), and T_{pred} the accuracy of the task containing the all-class prediction. At $\beta = .5$, support restriction increases W in all four dataset–pipeline settings, but it also increases G, and T_{pred} falls in three of the four settings, as Appendix Table 23 shows. In ImageNet-100 with ViT-Tiny, the within-task gain exceeds the increase in this gap, yielding higher overall accuracy and task prediction. This decomposition locates the differing outcomes in the balance between within-task classification and cross-task competition, and motivates a training-level question: can restored constraint strength preserve that balance?

5.2 Amplifying the Remaining Errors Does Not Consistently Compensate

Active-output averaging retains the same errors as masked/full regression while increasing their weight by $1/q$ on each batch (Equation 4). In the CIFAR-100 component study, the effects of this first response depend on the pipeline and the nominal weight.

ViT-Tiny at $\beta = .1$ exhibits a partial recovery: active-output averaging improves both accuracy metrics and reduces forgetting, at a small acquisition cost. At $\beta = .5$, the same operation reduces accuracy and acquisition while increasing forgetting. ResNet-18 shows small mean accuracy decreases and AF increases at both weights with little acquisition change (Figure 3). Appendix Tables 5 and 6 show that its AIA_{time} interval at $\beta = .1$ spans zero.

These mixed effects highlight why the distinction between support and scale matters: multiplying the retained errors increases their influence without restoring the omitted coordinates, so it is not a consistent substitute for retaining those constraints. Appendices C and G describe the associated trajectories and per-unit-weight effects in Tables 8 and 17.

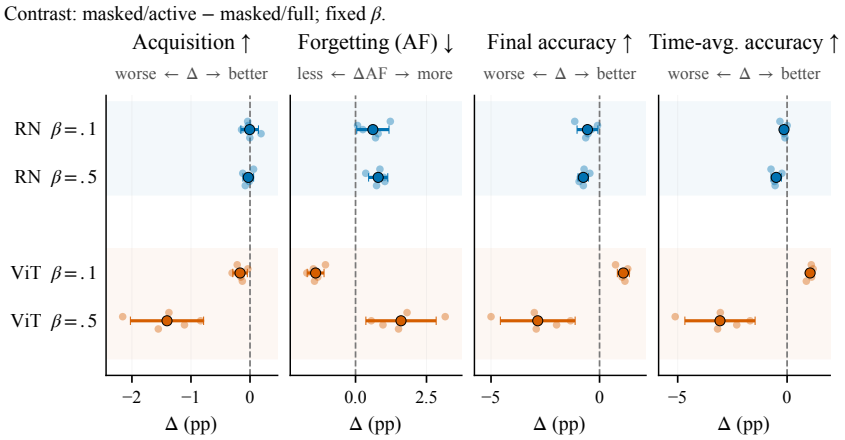


Figure 3: Rescaling does not consistently recover performance. Masked/active minus masked/full effects at fixed weight, in pp. Small points show five order means, large points their average, and bars pointwise 95% t intervals. Blue denotes ResNet-18 (RN), orange ViT-Tiny (ViT). Appendix Tables 5 and 6 give numerical estimates.

Table 2: **Same-state norm matching does not recover the full constraint.** Norm-matched masked minus unmasked effects on ImageNet-100, in pp, with pointwise 95% t intervals over seven order–seed pairs. W uses true task identity for evaluation only, T_{pred} is the accuracy of the task containing the all-class argmax, and $E_{\text{comp}} = (W - A)/W$. FAA is the primary outcome (Holm-adjusted $p = .03125$ in both pipelines), and remaining outcomes are descriptive.

Outcome	**ResNet-18**	**ViT-Tiny**
FAA	−41.10 [−45.00, −37.19]	−31.83 [−34.18, −29.47]
AIA_{time}	−24.55 [−27.33, −21.77]	−17.31 [−20.05, −14.57]
AF	+61.58 [+55.51, +67.65]	+30.78 [+29.03, +32.53]
Acquisition	+19.16 [+9.00, +29.33]	−4.12 [−5.33, −2.92]
Final old-task accuracy	−46.97 [−50.66, −43.28]	−34.66 [−37.13, −32.19]
Within-task W	−2.41 [−3.73, −1.10]	+0.34 [−0.53, +1.21]
Task prediction T_{pred}	−41.37 [−45.22, −37.51]	−31.26 [−33.42, −29.10]
Competition E_{comp}	+44.34 [+40.39, +48.29]	+37.21 [+34.41, +40.02]

5.3 Matching Gradient Magnitude Does Not Recover Cross-Task Discrimination

Active-output averaging depends on the number of retained coordinates, not on their gradients. We therefore test the more direct control in Equation 5 on ImageNet-100. Seven independently sampled class-order and optimization-seed pairs are shared across pipelines, each with a fresh unmasked control and norm-matched masked treatment, giving 28 complete runs. The intervention and the two primary FAA tests are fixed in advance, and Appendix K verifies that all training batches satisfy the specified numerical matching criterion.

Matching magnitude leaves a large performance gap: Table 2 shows final-accuracy losses of 41.10 pp in ResNet-18 and 31.83 pp in ViT-Tiny. Every pair has a negative FAA effect in both pipelines, and Appendix Table 22 shows that each leave-one-pair-out mean retains that direction. Final old-task accuracy also falls sharply, so the increased forgetting reflects loss of retained performance rather than only a higher learning peak. Acquisition rises in ResNet-18 but falls in ViT-Tiny, reinforcing the need to evaluate it separately.

The dominant loss concerns competition across tasks. Within-task accuracy changes by only −2.41 pp and +0.34 pp, whereas task-prediction accuracy falls by 41.37 pp and 31.26 pp, and the competition error share $E_{\text{comp}} = (W - A)/W$ rises by 44.34 pp and 37.21 pp, as Table 2 reports. The accuracy loss is concentrated in examples that remain classifiable given task identity: restoring the same-state gradient magnitude along the masked direction does not restore the discrimination of full distillation.

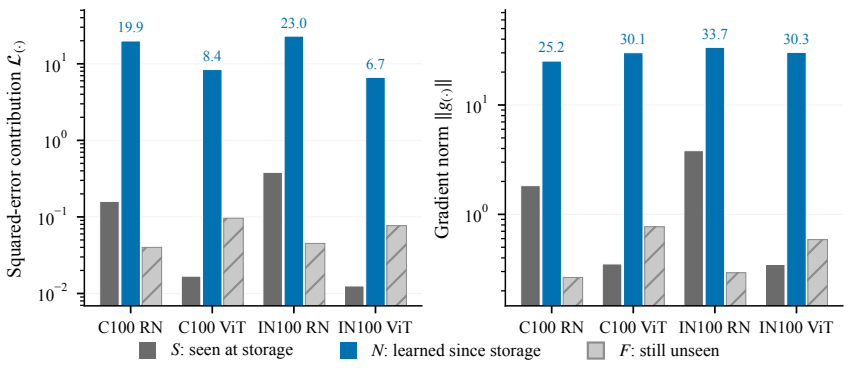


Figure 4: The omitted constraints remain active on subsequently learned classes. Squared-error contributions (left) and gradient norms over all trainable parameters (right) before the last update of task three on masked/full trajectories at $\beta = .5$. S, N, and F denote coordinates seen at storage (S), newly learned since storage (N), and future/unseen (F). Axes are logarithmic, and each bar group is one exploratory trajectory. Appendix Table 24 gives exact values.

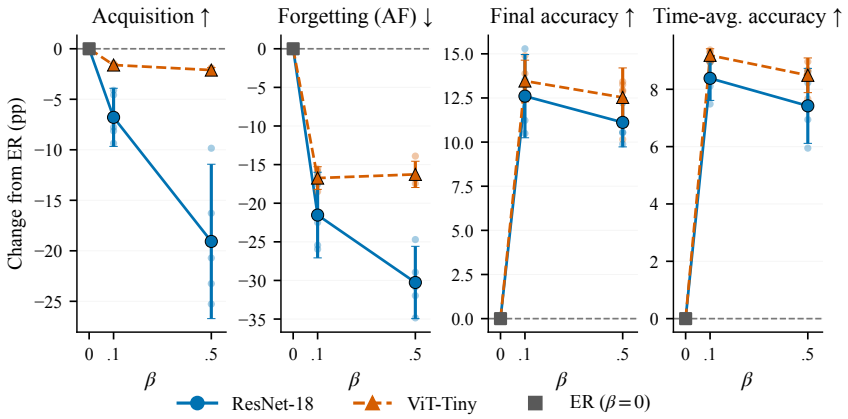


Figure 5: Stronger unmasked distillation does not improve every outcome. Changes from ER in pp: five order-level differences, their means, and pointwise 95% t intervals. The square is the shared ER reference and connecting lines are visual guides. Appendix Figure 6 isolates the adjacent increments.

Which omitted constraints remain active? Local probes clarify why "unseen at storage" is an incomplete description of their later role. We partition the coordinates of each stored example into those seen at storage (S), newly learned since storage (N), and future/unseen (F). Figure 4 shows that at the end of three-task masked trajectories, the second group has residuals and gradient norms one to two orders of magnitude larger than the third in all four dataset–pipeline settings, with most of its gradient energy in the backbone (Appendix Table 24). Thus the removed terms constrain changes involving subsequently learned classes as well as future outputs.

5.4 Adding Distillation and Strengthening It Have Different Effects

The comparisons so far modify an existing distillation objective, leaving open what the constraint contributes in the first place. We therefore compare unmasked regression with the same replay implementation at zero distillation weight, which corresponds to ER.

Introducing unmasked distillation at $\beta = .1$ improves both accuracy metrics and reduces forgetting in every order of both pipelines, at a cost to acquisition (Figure 5). Raising the weight to .5 reduces acquisition further without continuing the accuracy gains: mean FAA and $\mathrm{AIA}_{\mathrm{time}}$ fall relative to .1 with FAA intervals spanning zero, and forgetting decreases further only in ResNet-18, with each

adjacent increment reported in Appendix Figure 6. Introducing a useful constraint and strengthening it thus have distinct consequences.

5.5 A Complete Recipe Can Hide the Component Tradeoffs

Complete recipes often change several objective choices at once. A masked/active recipe at $\beta = .1$, selected on a development order and evaluated on a separate order with three paired seeds, illustrates why the control matters: against unmasked regression at $\beta = .5$ it improves acquisition but worsens forgetting and both accuracy metrics in every test seed, while against ER it improves retention and accuracy in ViT-Tiny but worsens them in ResNet-18, as Appendix Table 9 reports. A complementary analysis across the five component-study orders likewise finds no consistent ResNet-18 advantage over ER, as Appendix Table 25 shows.

The same perspective is useful beyond our implementation: under the specified recipes in the Mammoth continual learning benchmark (Boschini et al., 2023), each adjacent comparison from ER through DER, DER++, and X-DER improves final accuracy and reduces forgetting while lowering acquisition, as Table 12 in Appendix D reports, and complementary ImageNet-100 comparisons in Tables 11 and 13 cover distillation and increased replay capacity. These comparisons involve different objectives and buffer operations rather than a single increasing constraint coefficient, and their acquisition costs show why a favorable final score alone does not describe what a recipe changes.

6 What Counts as a Stability Gain?

Preserving predictions cannot be judged by forgetting alone. The CIFAR-100 tradeoffs in Figures 2 and 5 and the ImageNet-100 reversal in Appendix Table 10 show that the value of a constraint depends on what the learner acquires as well as what it retains. The norm-matching result in Table 2 sharpens this interpretation: full distillation supplies a constraint that same-state scalar compensation does not replace. Local probes reveal its temporal scope: outputs unseen at storage can later correspond to learned classes that masking no longer constrains, as Appendix M shows.

What relative repair adds. Relative repair expresses forgetting reduction as a fraction of headroom to a chosen reference, so its validity depends on that reference: the large-buffer sequential reference in Appendix Table 13 supplies positive headroom over ER in both pipelines but not over DER++ in ResNet-18. Appendices E–F develop this perspective and its statistical reliability.

Limitations. Our results concern fixed-head replay in specified pre-trained pipelines, without separating architecture, checkpoint, and recipe. Norm matching tests one same-state distillation rule rather than all scalar schedules or equal total optimizer updates, and the seven-pair experiment does not include a masked condition without norm matching. Local probes supply diagnostic evidence rather than a predictor of the sign reversal. Appendix O details these boundaries.

7 Conclusion

This paper asks what stored-logit distillation preserves in pre-trained class-incremental learning. We show that stored predictions can remain useful even when their classes had not been observed at storage time. Separating support, averaging, and weight shows how these constraints shape acquisition and stability, with net benefits that vary across training settings. A same-state gradient-norm intervention goes further: amplifying the masked direction to the full-gradient magnitude does not recover the cross-task discrimination of full distillation in the tested pipelines. Local measurements show that the omitted constraints act on classes learned since storage, beyond the classification head. These findings connect stored targets to what continual learning must preserve: not only performance within a task, but discrimination from classes learned later. The value of such preservation must be judged by which updates a constraint restricts rather than by how strongly it constrains, a judgment that becomes visible when acquisition, retention, and accuracy are evaluated together.

REFERENCES

Matteo Boschini, Lorenzo Bonicelli, Pietro Buzzega, Angelo Porrello, and Simone Calderara. Class-incremental continual learning into the extended DER-verse. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5497–5512, 2023.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: A strong, simple baseline. In *NeurIPS*, volume 33, pp. 15920–15930, 2020.

Arslan Chaudhry, Puneet K. Dokania, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *ECCV*, pp. 532–547, 2018.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K. Dokania, Philip H. S. Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. In *ICML Workshop*, 2019.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3366–3385, 2022.

Natalia Díaz-Rodríguez, Vincenzo Lomonaco, David Filliat, and Davide Maltoni. Don't forget, there is more than forgetting: New metrics for continual learning. In *NeurIPS Workshop*, 2018.

Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. PODNet: Pooled outputs distillation for small-tasks incremental learning. In *ECCV*, pp. 86–102, 2020.

Tyler L. Hayes, Ronald Kemker, Nathan D. Cahill, and Christopher Kanan. New metrics and experimental paradigms for continual learning. In *CVPR Workshops*, pp. 2031–2034, 2018.

Jiangpeng He, Zhihao Duan, and Fengqing Zhu. Cl-lora: Continual low-rank adaptation for rehearsal-free class-incremental learning. In *CVPR*, pp. 30534–30544, 2025.

Yen-Chang Hsu, Yen-Cheng Liu, Anita Ramasamy, and Zsolt Kira. Re-evaluating continual learning scenarios: A categorization and case for strong baselines. In *NeurIPS Workshop*, 2018.

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, 2017.

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

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018.

Yan-Shuo Liang and Wu-Jun Li. InfLoRA: Interference-free low-rank adaptation for continual learning. In *CVPR*, pp. 23638–23647, 2024.

Zhanwang Liu, Yuting Li, Haoyuan Gao, Yexin Li, Linghe Kong, Lichao Sun, and Weiran Huang. IDER: IDempotent experience replay for reliable continual learning. In *ICLR*, 2026.

Mark D. McDonnell, Dong Gong, Amin Parvaneh, Ehsan Abbasnejad, and Anton van den Hengel. RanPAC: Random projections and pre-trained models for continual learning. In *NeurIPS*, volume 36, pp. 12022–12053, 2023.

Nicolas Michel, Giovanni Chierchia, Romain Negrel, Jean-François Bercher, and Toshihiko Yamasaki. New metrics for analyzing continual learners. In *MIRU*, 2023. arXiv:2309.00462.

Seyed Iman Mirzadeh, Arslan Chaudhry, Dong Yin, Timothy Nguyen, Razvan Pascanu, Dilan Gorur, and Mehrdad Farajtabar. Architecture matters in continual learning. *arXiv preprint arXiv:2202.00275*, 2022.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning. In *CVPR*, pp. 5533–5542, 2017.

Joan Serrà, Dídac Surís, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *ICML*, pp. 4548–4557, 2018.

James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. CODA-Prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *CVPR*, pp. 11909–11919, 2023.

Albin Soutif-Cormerais, Antonio Carta, Andrea Cossu, Julio Hurtado, Vincenzo Lomonaco, Joost van de Weijer, and Hamed Hemati. A comprehensive empirical evaluation on online continual learning. In *ICCV Workshops*, pp. 3518–3528, 2023.

Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your ViT? data, augmentation, and regularization in vision transformers. 2021.

Gido M. van de Ven, Tinne Tuytelaars, and Andreas S. Tolias. Three types of incremental learning. *Nature Machine Intelligence*, 4(12):1185–1197, 2022.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. DualPrompt: Complementary prompting for rehearsal-free continual learning. In *ECCV*, pp. 631–648, 2022a.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *CVPR*, pp. 139–149, 2022b.

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

Da-Wei Zhou, Qi-Wei Wang, Zhi-Hong Qi, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Class-incremental learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9851–9873, 2024.

Da-Wei Zhou, Zi-Wen Cai, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *International Journal of Computer Vision*, 133(3):1012–1032, 2025.

Appendix

A Metrics and Implementation Details

This appendix specifies the evaluation conventions, the training procedure, the replay and target-storage semantics, and the relationship of our implementation to the reference DER++ recipe.

Evaluation. At each stage, predictions are restricted to classes seen so far and evaluated without task identities. Equations 1–2 use task-average accuracies. With unequal task sizes this is different from pooling examples. The current experiments have equal class-group sizes. AF uses the maximum over observations from the introduction of each task through the evaluation stage, including the evaluation stage itself. This convention measures nonnegative loss from the observed peak. A learner with a low peak may consequently have low AF without high final accuracy.

Training. Both component-study pipelines train all backbone and head parameters over ten stages with 20 epochs per stage. Current-data and replay batches each contain up to 64 examples. SGD uses momentum .9, zero weight decay, and an initial learning rate of .005, followed by epoch-wise cosine decay to zero within each stage. Optimizer and scheduler states reset at stage boundaries. Automatic mixed precision is enabled. ResNet-18 uses the torchvision ImageNet-1K V1 pre-trained checkpoint. ViT-Tiny uses the AugReg ImageNet-21K pre-trained checkpoint followed by ImageNet-1K fine-tuning (`vit_tiny_patch16_224`) (Steiner et al., 2021). The pre-trained classifier is replaced by a randomly initialized 100-output head, retained throughout the sequence. ResNet-18 uses the default linear-layer initialization, and ViT-Tiny uses the default timm initialization. The replay memory holds at most 2,000 examples. Label-based replay has coefficient .1, and Table 1 summarizes the conditions with distillation coefficients 0, .1, and .5.

Replay and stored targets. Label-based replay and logit regression share a minibatch, sampled uniformly without replacement from the buffer. Reservoir updates occur after each optimizer step during the final epoch of a stage. The inserted input is the already augmented, normalized training tensor. Its target logits come from the training-mode forward pass before that step and are saved in FP32 together with the seen-class mask. Replay reuses the stored input tensor without further augmentation, with the model in training mode. Replay terms are omitted while the buffer is empty. Once insertion begins, later minibatches can replay examples from the current stage, including during the final epoch of the first stage. Sampling and reservoir-update random streams are paired across conditions. Masking changes only logit regression. Training cross-entropy uses the full head, whereas evaluation restricts predictions to seen classes.

Relation to DER++. Our DER++-style implementation combines label-based replay with stored-logit regression. In our notation, α multiplies replay cross-entropy and β multiplies MSE. The original DER++ notation assigns these symbols to the opposite terms (Buzzega et al., 2020). The reference DER++ implementation in Mammoth draws separate replay minibatches for the two terms, whereas our implementation shares a batch and uses the buffer-update schedule specified above. The component study concerns this specified training procedure rather than an exact reproduction of the reference recipe.

B Complete Component Statistics

This section collects the complete numerical record of the component study: the randomization settings, the four-outcome support estimates, and all planned contrasts with their intervals and sensitivities. The exact order and optimization seeds appear in Table 3, and the class permutations in Table 18. For CIFAR-100 training, images are resized to 256 pixels with antialiasing, randomly cropped to 224×224, and horizontally flipped with probability .5. Evaluation uses the same resize followed by a center crop. Both use channel means $(.485, .456, .406)$ and standard deviations $(.229, .224, .225)$. This protocol uses a random crop after resizing, not a random-resized crop.

Table 4 gives the full four-outcome estimates underlying Figure 2. Tables 5 and 6 report all eight planned contrasts for every outcome and pipeline. Support contrasts are masked/full minus un-

Table 3: Randomness specification for the component study. The listed optimization seeds are shared across conditions and pipelines for each order.

Order	Order Seed	Optimization Seeds
1	247176437	235524199, 604717161, 726207697
2	561540886	1086574640, 273178405, 725278836
3	1180412353	1494926324, 1867088480, 939781018
4	1643143863	128948184, 643021955, 723797120
5	299903581	1514821603, 575672428, 934740505

Table 4: Support restriction: masked/full minus unmasked at fixed weight. Entries are mean paired differences in percentage points with pointwise 95% t intervals across five class orders. For each pipeline, **bold** indicates the more favorable effect between weights and underline indicates the secondary effect (↓ lower is better, ↑ higher is better).

Pipeline	β	Performance differences (pp)			
		ΔAcquisition (↑)	ΔAF (↓)	ΔFAA (↑)	$\Delta\text{AIA}_{\text{time}}$ (↑)
ResNet-18	.1	$\underline{+6.95}$ [+4.07, +9.82]	+21.99 [+16.41, +27.56]	$\underline{-12.85}$ [−15.52, −10.19]	$\underline{-8.63}$ [−9.33, −7.93]
	.5	+19.19 [+11.58, +26.80]	$\underline{+31.47}$ [+26.31, +36.62]	−12.08 [−14.04, −10.12]	−7.81 [−9.25, −6.37]
ViT-Tiny	.1	$\underline{+1.55}$ [+1.39, +1.72]	$\underline{+14.12}$ [+12.94, +15.29]	$\underline{-11.15}$ [−12.13, −10.18]	$\underline{-7.38}$ [−7.52, −7.24]
	.5	+1.85 [+1.48, +2.23]	+11.91 [+10.71, +13.10]	−8.86 [−10.19, −7.53]	−5.58 [−6.29, −4.87]

masked at the indicated weight. Active-output averaging contrasts, shortened to "Averaging" in the tables, are masked/active minus masked/full. Weight contrasts are higher minus lower unmasked weight. The support interaction is the support effect at .5 minus that at .1, and the averaging interaction is defined analogously. These differences of effects describe weight dependence without merging support restriction and active-output averaging into one intervention.

Each pointwise interval is $\bar{d} \pm t_{.975,4} s_d/\sqrt{5}$, where $\bar{d}$ and s_d are the mean and sample standard deviation of the five order-level paired differences. Here $t_{.975,4}$ is the .975 quantile of the Student-t distribution with four degrees of freedom. Exact sign-flip calculations enumerate the 2^5 signs under a sign-symmetry null. Holm correction uses eight contrasts separately for each pipeline and outcome. The minimum adjusted value is .5, so the tables support directional and effect-size descriptions rather than multiplicity-adjusted rejection at .05. The tables that follow constitute the complete numerical record: Tables 5 and 6 report every planned contrast for each pipeline, and Table 7 gives the underlying order-level estimates together with the range of means obtained by leaving out one order. Figure 6 reports the adjacent unmasked-weight increments that underlie the three-point comparison in Figure 5.

Table 5: Complete component effects for the ResNet-18 pipeline. Raw differences in percentage points, with pointwise 95% t intervals. Exact two-sided sign-flip p-values and Holm-adjusted p-values are reported as descriptive sensitivities.

Contrast	Outcome	Mean [95% CI]	Exact p	Holm p
Support, $\beta = .1$	Acquisition	+6.95 [+4.07, +9.82]	0.0625	0.5000
	AF	+21.99 [+16.41, +27.56]	0.0625	0.5000
	FAA	−12.85 [−15.52, −10.19]	0.0625	0.5000
	AIA_{time}	−8.63 [−9.33, −7.93]	0.0625	0.5000
Support, $\beta = .5$	Acquisition	+19.19 [+11.58, +26.80]	0.0625	0.5000
	AF	+31.47 [+26.31, +36.62]	0.0625	0.5000

Contrast	Outcome	Mean [95% CI]	Exact p	Holm p
	FAA	-12.08 [-14.04, -10.12]	0.0625	0.5000
	AIA_{time}	-7.81 [-9.25, -6.37]	0.0625	0.5000
Averaging, $\beta = .1$	Acquisition	-0.01 [-0.16, $+0.14$]	0.9375	1.0000
	AF	$+0.61$ [$+0.04$, $+1.18$]	0.0625	0.5000
	FAA	-0.56 [-1.04, -0.08]	0.0625	0.5000
	AIA_{time}	-0.14 [-0.29, $+0.01$]	0.1250	0.5000
Averaging, $\beta = .5$	Acquisition	-0.03 [-0.12, $+0.06$]	0.3750	1.0000
	AF	$+0.80$ [$+0.46$, $+1.14$]	0.0625	0.5000
	FAA	-0.75 [-0.99, -0.51]	0.0625	0.5000
	AIA_{time}	-0.50 [-0.73, -0.26]	0.0625	0.5000
Weight, $0 \to .1$	Acquisition	-6.79 [-9.67, -3.91]	0.0625	0.5000
	AF	-21.53 [-27.08, -15.99]	0.0625	0.5000
	FAA	$+12.60$ [$+10.24$, $+14.96$]	0.0625	0.5000
	AIA_{time}	$+8.38$ [$+7.61$, $+9.15$]	0.0625	0.5000
Weight, $.1 \to .5$	Acquisition	-12.28 [-19.09, -5.47]	0.0625	0.5000
	AF	-8.73 [-13.53, -3.93]	0.0625	0.5000
	FAA	-1.49 [-4.00, $+1.03$]	0.1875	0.5625
	AIA_{time}	-0.96 [-2.84, $+0.92$]	0.2500	0.5000
Support interaction	Acquisition	$+12.25$ [$+5.39$, $+19.11$]	0.0625	0.5000
	AF	$+9.48$ [$+4.14$, $+14.82$]	0.0625	0.5000
	FAA	$+0.77$ [-1.90, $+3.44$]	0.5000	0.6250
	AIA_{time}	$+0.82$ [-1.09, $+2.73$]	0.3125	0.5000
Averaging interaction	Acquisition	-0.02 [-0.20, $+0.16$]	0.7500	1.0000
	AF	$+0.19$ [-0.31, $+0.69$]	0.4375	0.5000
	FAA	-0.19 [-0.66, $+0.28$]	0.3125	0.6250
	AIA_{time}	-0.35 [-0.49, -0.22]	0.0625	0.5000

Table 6: Complete component effects for the ViT-Tiny pipeline. Raw differences in percentage points, with pointwise 95% t intervals. Exact two-sided sign-flip p-values and Holm-adjusted p-values are reported as descriptive sensitivities.

Contrast	Outcome	Mean [95% CI]	Exact p	Holm p
Support, $\beta = .1$	Acquisition	$+1.55$ [$+1.39$, $+1.72$]	0.0625	0.5000
	AF	$+14.12$ [$+12.94$, $+15.29$]	0.0625	0.5000
	FAA	-11.15 [-12.13, -10.18]	0.0625	0.5000
	AIA_{time}	-7.38 [-7.52, -7.24]	0.0625	0.5000
Support, $\beta = .5$	Acquisition	$+1.85$ [$+1.48$, $+2.23$]	0.0625	0.5000
	AF	$+11.91$ [$+10.71$, $+13.10$]	0.0625	0.5000
	FAA	-8.86 [-10.19, -7.53]	0.0625	0.5000
	AIA_{time}	-5.58 [-6.29, -4.87]	0.0625	0.5000
Averaging, $\beta = .1$	Acquisition	-0.17 [-0.29, -0.05]	0.0625	0.5000
	AF	-1.41 [-1.70, -1.11]	0.0625	0.5000
	FAA	$+1.10$ [$+0.82$, $+1.38$]	0.0625	0.5000
	AIA_{time}	$+1.05$ [$+0.89$, $+1.21$]	0.0625	0.5000
Averaging, $\beta = .5$	Acquisition	-1.41 [-2.03, -0.79]	0.0625	0.5000
	AF	$+1.60$ [$+0.36$, $+2.84$]	0.0625	0.5000
	FAA	-2.85 [-4.57, -1.13]	0.0625	0.5000
	AIA_{time}	-3.07 [-4.67, -1.46]	0.0625	0.5000
Weight, $0 \to .1$	Acquisition	-1.61 [-1.77, -1.44]	0.0625	0.5000
	AF	-16.74 [-18.21, -15.28]	0.0625	0.5000

Contrast	Outcome	Mean [95% CI]	Exact p	Holm p
	FAA	+13.46 [+12.28, +14.64]	0.0625	0.5000
	AIA_{time}	+9.18 [+8.96, +9.41]	0.0625	0.5000
Weight, $.1 \to .5$	Acquisition	−0.50 [−0.76, −0.25]	0.0625	0.5000
	AF	+0.48 [−0.79, +1.74]	0.3750	0.5000
	FAA	−0.93 [−2.23, +0.37]	0.1875	0.5000
	AIA_{time}	−0.70 [−1.16, −0.23]	0.0625	0.5000
Support interaction	Acquisition	+0.30 [−0.11, +0.71]	0.0625	0.5000
	AF	−2.21 [−3.53, −0.90]	0.0625	0.5000
	FAA	+2.29 [+0.81, +3.77]	0.0625	0.5000
	AIA_{time}	+1.80 [+1.15, +2.45]	0.0625	0.5000
Averaging interaction	Acquisition	−1.24 [−1.97, −0.50]	0.0625	0.5000
	AF	+3.01 [+1.81, +4.21]	0.0625	0.5000
	FAA	−3.95 [−5.74, −2.15]	0.0625	0.5000
	AIA_{time}	−4.12 [−5.81, −2.43]	0.0625	0.5000

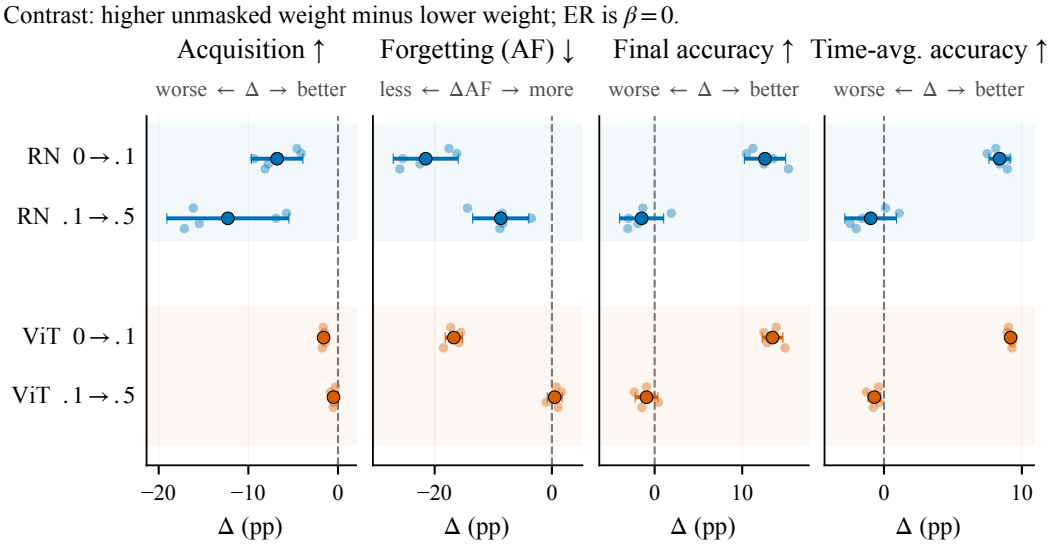


Figure 6: Adjacent unmasked-weight increments, reported as higher weight minus lower weight, in percentage points. The $\beta = 0$ condition is ER under the same implementation. Small points are the five order means, large points are their average, and bars are pointwise 95% t intervals over those five orders. These adjacent contrasts are the direct tests of introducing versus strengthening distillation. They are not obtained by adding or subtracting the intervals in Figure 5.

Table 7: Order-level differences and leave-one-order-out (LOO) range, in percentage points across five paired class orders (O_1–O_5). The LOO range denotes the minimum and maximum of the five leave-one-order-out means.

Pipeline / Contrast	Outcome	O_1	O_2	O_3	O_4	O_5	LOO Range
RN / Support, $\beta = .1$	Acquisition	+8.22	+7.79	+9.66	+4.29	+4.77	[+6.27, +7.61]
	AF	+26.84	+21.20	+26.35	+16.80	+18.74	[+20.77, +23.28]
	FAA	−15.99	−11.28	−14.06	−10.82	−12.10	[−13.36, −12.07]
	AIA_{time}	−8.99	−8.63	−9.24	−7.75	−8.54	[−8.85, −8.48]
RN / Support, $\beta = .5$	Acquisition	+25.32	+23.37	+16.46	+9.97	+20.85	[+17.66, +21.50]
	AF	+36.53	+30.17	+30.07	+25.99	+34.57	[+30.20, +32.83]
	FAA	−13.69	−9.82	−11.37	−13.42	−12.10	[−12.64, −11.68]
	AIA_{time}	−7.13	−6.29	−7.76	−8.95	−8.94	[−8.19, −7.53]

Pipeline / Contrast	Outcome	O_1	O_2	O_3	O_4	O_5	LOO Range
RN / Averaging, $\beta = .1$	Acquisition	-0.00	$+0.19$	-0.14	-0.03	-0.04	$[-0.05, +0.03]$
	AF	$+0.70$	$+0.80$	$+0.26$	$+0.07$	$+1.23$	$[+0.46, +0.75]$
	FAA	-0.64	-0.53	-0.38	-0.09	-1.14	$[-0.67, -0.41]$
	AIA_{time}	-0.10	-0.14	-0.15	$+0.00$	-0.33	$[-0.18, -0.10]$
RN / Averaging, $\beta = .5$	Acquisition	-0.08	-0.01	$+0.01$	-0.12	$+0.06$	$[-0.05, -0.01]$
	AF	$+0.74$	$+1.03$	$+1.00$	$+0.36$	$+0.86$	$[+0.74, +0.91]$
	FAA	-0.75	-0.94	-0.89	-0.45	-0.71	$[-0.82, -0.70]$
	AIA_{time}	-0.55	-0.58	-0.39	-0.24	-0.73	$[-0.56, -0.44]$
RN / Weight, $0 \to .1$	Acquisition	-8.13	-7.77	-9.36	-4.11	-4.57	$[-7.46, -6.15]$
	AF	-25.96	-22.50	-25.43	-16.22	-17.56	$[-22.86, -20.43]$
	FAA	$+15.29$	$+12.48$	$+13.53$	$+10.49$	$+11.23$	$[+11.93, +13.13]$
	AIA_{time}	$+8.95$	$+8.40$	$+8.97$	$+7.49$	$+8.11$	$[+8.24, +8.61]$
RN / Weight, $.1 \to .5$	Acquisition	-17.14	-15.48	-6.92	-5.74	-16.14	$[-13.92, -11.07]$
	AF	-8.89	-8.36	-3.51	-8.49	-14.41	$[-10.03, -7.31]$
	FAA	-3.07	-1.92	-3.00	$+1.90$	-1.34	$[-2.33, -1.09]$
	AIA_{time}	-2.01	-2.45	-1.58	$+1.11$	$+0.12$	$[-1.48, -0.59]$
RN / Support interaction	Acquisition	$+17.10$	$+15.57$	$+6.80$	$+5.68$	$+16.08$	$[+11.03, +13.89]$
	AF	$+9.70$	$+8.97$	$+3.71$	$+9.20$	$+15.83$	$[+7.89, +10.92]$
	FAA	$+2.30$	$+1.46$	$+2.69$	-2.60	-0.00	$[+0.29, +1.61]$
	AIA_{time}	$+1.86$	$+2.35$	$+1.48$	-1.20	-0.40	$[+0.43, +1.32]$
RN / Averaging interaction	Acquisition	-0.08	-0.20	$+0.15$	-0.09	$+0.10$	$[-0.07, +0.02]$
	AF	$+0.04$	$+0.24$	$+0.74$	$+0.29$	-0.37	$[+0.05, +0.33]$
	FAA	-0.11	-0.41	-0.51	-0.35	$+0.43$	$[-0.35, -0.11]$
	AIA_{time}	-0.45	-0.44	-0.24	-0.24	-0.41	$[-0.38, -0.33]$
ViT / Support, $\beta = .1$	Acquisition	$+1.76$	$+1.47$	$+1.61$	$+1.41$	$+1.52$	$[+1.50, +1.59]$
	AF	$+15.31$	$+13.09$	$+13.36$	$+13.99$	$+14.84$	$[+13.82, +14.37]$
	FAA	-12.02	-10.31	-10.41	-11.18	-11.83	$[-11.36, -10.93]$
	AIA_{time}	-7.26	-7.42	-7.33	-7.35	-7.56	$[-7.41, -7.34]$
ViT / Support, $\beta = .5$	Acquisition	$+2.03$	$+1.68$	$+1.75$	$+2.28$	$+1.53$	$[+1.75, +1.93]$
	AF	$+12.88$	$+12.33$	$+11.63$	$+10.37$	$+12.31$	$[+11.66, +12.29]$
	FAA	-9.56	-9.42	-8.72	-7.05	-9.55	$[-9.31, -8.68]$
	AIA_{time}	-5.57	-5.92	-5.70	-4.62	-6.09	$[-5.82, -5.45]$
ViT / Averaging, $\beta = .1$	Acquisition	-0.13	-0.16	-0.30	-0.04	-0.22	$[-0.20, -0.14]$
	AF	-1.44	-1.33	-1.71	-1.49	-1.06	$[-1.49, -1.33]$
	FAA	$+1.17$	$+1.04$	$+1.24$	$+1.30$	$+0.74$	$[+1.05, +1.19]$
	AIA_{time}	$+0.88$	$+1.12$	$+0.96$	$+1.19$	$+1.11$	$[+1.02, +1.10]$
ViT / Averaging, $\beta = .5$	Acquisition	-1.55	-1.11	-0.84	-2.16	-1.37	$[-1.55, -1.22]$
	AF	$+1.51$	$+0.97$	$+0.56$	$+3.16$	$+1.81$	$[+1.21, +1.86]$
	FAA	-2.92	-1.98	-1.34	-5.00	-3.01	$[-3.23, -2.31]$
	AIA_{time}	-3.18	-2.30	-1.69	-5.11	-3.05	$[-3.41, -2.56]$
ViT / Weight, $0 \to .1$	Acquisition	-1.76	-1.43	-1.67	-1.52	-1.66	$[-1.65, -1.57]$
	AF	-18.50	-15.86	-16.53	-15.56	-17.26	$[-17.04, -16.30]$
	FAA	$+14.89$	$+12.85$	$+13.21$	$+12.48$	$+13.88$	$[+13.10, +13.71]$
	AIA_{time}	$+9.28$	$+9.29$	$+9.37$	$+8.95$	$+9.03$	$[+9.14, +9.24]$
ViT / Weight, $.1 \to .5$	Acquisition	-0.52	-0.46	-0.41	-0.83	-0.29	$[-0.55, -0.42]$
	AF	$+1.05$	-0.97	-0.07	$+1.66$	$+0.72$	$[+0.18, +0.84]$
	FAA	-1.47	$+0.41$	-0.34	-2.33	-0.93	$[-1.27, -0.58]$
	AIA_{time}	-0.77	-0.33	-0.71	-1.28	-0.40	$[-0.79, -0.55]$
ViT / Support interaction	Acquisition	$+0.27$	$+0.21$	$+0.14$	$+0.87$	$+0.01$	$[+0.16, +0.37]$
	AF	-2.43	-0.76	-1.73	-3.61	-2.53	$[-2.58, -1.86]$
	FAA	$+2.46$	$+0.89$	$+1.70$	$+4.12$	$+2.29$	$[+1.83, +2.64]$
	AIA_{time}	$+1.69$	$+1.50$	$+1.62$	$+2.73$	$+1.47$	$[+1.57, +1.88]$
ViT / Averaging interaction	Acquisition	-1.42	-0.95	-0.53	-2.12	-1.16	$[-1.41, -1.02]$

Pipeline / Contrast	Outcome	O_1	O_2	O_3	O_4	O_5	LOO Range
	AF	$+2.96$	$+2.30$	$+2.27$	$+4.64$	$+2.87$	$[+2.60, +3.19]$
	FAA	-4.09	-3.02	-2.58	-6.30	-3.74	$[-4.29, -3.36]$
	$\mathrm{AIA}_{\mathrm{time}}$	-4.06	-3.42	-2.65	-6.30	-4.16	$[-4.49, -3.57]$

C Loss and Gradient Diagnostics

The diagnostics help describe how the objectives behave during training. Because the active fraction q depends only on the masks of sampled memory examples, paired conditions use matching q sequences even as their model parameters diverge. Table 8 summarizes losses, logit norms, and relative gradient magnitudes by averaging recorded measurements within runs, then across seeds within orders, then equally across orders. Measurements are taken at minibatch indices $0, 10, 20, \ldots$ of each epoch whenever the buffer is nonempty. The gradient ratio is $\beta\|\nabla_{\theta_s}\mathcal{L}_{\mathrm{dis}}\|_2/(\alpha\|\nabla_{\theta_s}\mathcal{L}_{\mathrm{rep\,CE}}\|_2)$, where θ_s includes all trainable backbone and head parameters with gradients from both losses. Norms combine squared FP32 gradients over these parameters. Undefined ratios with zero denominator are omitted. The reported value is a mean of minibatch ratios, not a ratio of mean norms. Current and stored logit norms are minibatch means of the per-example Euclidean norm over all C outputs. These diagnostics describe trajectories rather than interventions on gradient magnitude.

Equation 4 is exact on the same error tensor. Across separate trained models, both the errors and parameters differ, so the ratio of average losses need not equal the reciprocal of average q. In particular, $1/\mathbb{E}[q]$ is generally different from $\mathbb{E}[1/q]$. These summaries describe the optimization trajectories associated with each objective. The same-state norm intervention in Appendix K separately tests scalar compensation, and Appendix M examines the gradients of omitted components.

D Complementary Recipe and Dataset Comparisons

This section reports comparisons that complement the component study: a combined recipe that changes support, averaging, and weight together, cross-dataset support-restriction experiments on ImageNet-100, external-framework replay and distillation comparisons in Mammoth, and an empirical sequential reference. Each comparison uses protocols that differ from the component study, as specified in Appendix J, and their estimates remain separate from the five-order component effects.

D.1 A Combined-Recipe Comparison

The combined-recipe comparison complements the component study by examining a change that affects support, averaging, and weight together: unmasked regression at $\beta = .5$ is replaced by masked/active regression at $\beta = .1$. Selection uses a development order, followed by evaluation on a separate test order with three seeds per pipeline. Appendix J.3 specifies the candidates, selection criterion, and both orders. Table 9 reports all paired test-seed effects, and Figure 7 displays the same three seeds. Relative to unmasked regression, the selected recipe improves acquisition while reducing accuracy and retention. Its comparison with ER instead depends on the pipeline, illustrating why both the control and the outcome matter. These estimates describe the combined recipe on one test order and remain separate from the five-order component effects. The three seeds are optimization-seed differences within that order. They do not provide an order-level confidence interval.

D.2 ImageNet-100 Support Restriction

An independent ImageNet-100 experiment examines whether the support-restriction tradeoff observed in CIFAR-100 extends to a second dataset at $\beta = .5$. Three class orders, each with three paired optimization seeds and two conditions (masked/full vs. unmasked, both with full denominator), yield 36 training runs and 18 paired comparisons. Table 10 reports the order-level means and ranges for each pipeline and outcome. The protocol is specified in Appendix J.4.

In ResNet-18, support restriction produces an acquisition–stability tradeoff consistent with the CIFAR-100 component study: acquisition rises while forgetting increases and both accuracy metrics

Table 8: Descriptive loss, gradient, and logit norm summaries averaged across minibatches and class orders. The active fraction $q = .300$ reflects memory replay composition.

Pipeline	Arm	q	Weighted MSE	Gradient ratio	Logit norm	
					Current	Stored
ResNet-18	ER	0.300	0.000	0.000	30.373	30.334
	Unmasked .1	0.300	0.231	4.117	32.676	36.134
	Unmasked .5	0.300	0.457	19.313	33.312	34.671
	Masked/full .1	0.300	0.052	1.392	100.612	64.502
	Masked/full .5	0.300	0.101	3.289	103.830	65.148
	Masked/active .1	0.300	0.087	3.043	103.176	64.996
	Masked/active .5	0.300	0.212	9.029	104.305	65.196
ViT-Tiny	ER	0.300	0.000	0.000	31.177	31.668
	Unmasked .1	0.300	0.118	19.785	24.171	26.079
	Unmasked .5	0.300	0.195	77.390	25.380	26.067
	Masked/full .1	0.300	0.069	1.765	69.315	47.301
	Masked/full .5	0.300	0.089	2.597	75.364	50.104
	Masked/active .1	0.300	0.085	2.507	74.878	49.911
	Masked/active .5	0.300	0.169	3.469	78.028	51.129

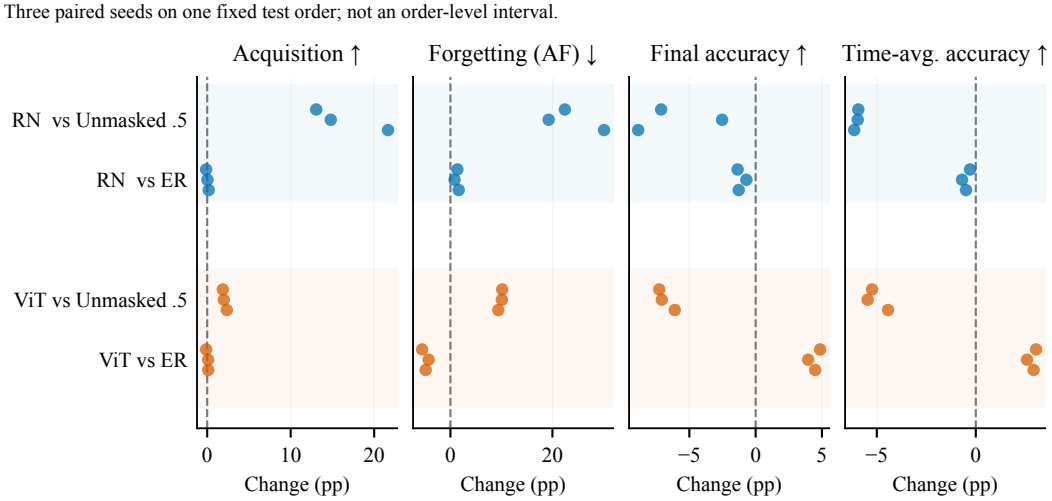


Figure 7: Combined masked/active recipe at $\beta = .1$ minus each control, in percentage points. Each point is a paired optimization-seed difference on a separate fixed test order. The three seeds describe variation within that order and are not five order means. This panel does not share the 95% order-level interval encoding of the component figures.

fall, with all three orders in agreement. The ViT-Tiny pipeline on ImageNet-100 instead improves all four outcomes in every order. With three independent orders, the minimum two-sided exact sign-flip p-value is .25, so these estimates describe effect sizes and direction rather than statistical rejection. Appendix L locates the differing accuracy effects in within-task classification and cross-task competition.

D.3 ImageNet-100 Stability Effects

Table 11 analyzes an independent ImageNet-100 experiment with No Replay, ER, and stored-logit distillation in each of two pipelines. The order is a fixed stress order, with five optimization seeds rather than five independent orders. The raw quantity is the forgetting reduction u defined in Appendix E. Shared and nested repair use structural zero with No Replay and ER as the denominator

Table 9: Combined masked/active recipe at $\beta = .1$ minus each control across three paired seeds on a separate test order, in percentage points (↓ lower AF is better, ↑ higher is better).

Pipeline	Control	Seed	ΔAcquisition (↑)	ΔAF (↓)	ΔFAA (↑)	$\Delta\text{AIA}_{\text{time}}$ (↑)
ResNet-18	Unmasked .5	25013	+21.66	+30.12	−8.86	−6.14
		60127	+14.80	+19.26	−2.53	−5.97
		88191	+13.06	+22.43	−7.13	−5.94
	ER	25013	+0.18	+1.62	−1.28	−0.49
		60127	+0.03	+0.81	−0.70	−0.70
		88191	−0.13	+1.37	−1.36	−0.29
ViT-Tiny	Unmasked .5	25013	+2.35	+9.39	−6.10	−4.43
		60127	+1.99	+10.08	−7.08	−5.47
		88191	+1.87	+10.16	−7.27	−5.24
	ER	25013	+0.12	−4.86	+4.49	+2.92
		60127	+0.12	−4.27	+3.96	+2.59
		88191	−0.13	−5.54	+4.86	+3.05

Table 10: ImageNet-100 support restriction: masked/full minus unmasked at $\beta = .5$, in percentage points across three class orders. Entries report the order mean and [min, max] range. **Bold** marks the more favorable pipeline outcome and underline marks the secondary outcome (↓ lower is better, ↑ higher is better).

Pipeline	Performance differences (pp)			
	ΔAcquisition (↑)	ΔAF (↓)	ΔFAA (↑)	$\Delta\text{AIA}_{\text{time}}$ (↑)
ResNet-18	**+19.7** [+12.8, +29.5]	+34.6 [+33.0, +37.2]	-14.6 [-17.3, -12.2]	-9.5 [-11.2, -6.8]
ViT-Tiny	+0.7 [+0.5, +1.0]	**-2.4** [-3.4, -1.1]	**+2.9** [+2.0, +3.7]	**+3.4** [+3.2, +3.7]

baseline, respectively. Relative columns report the mean of paired seed ratios. All intervals in this table are pointwise seed-level t intervals.

Table 11: ImageNet-100 stability effects under a fixed order across five optimization seeds, with pointwise 95% t intervals. Raw repair is in percentage points; shared and nested repair are dimensionless. **Bold** denotes the higher stability gain between pipelines and underline denotes the secondary value.

Pipeline	Raw Repair (pp)	Shared Repair	Nested Repair
ResNet-18	31.676 [27.411, 35.940]	0.331 [0.286, 0.375]	0.643 [0.562, 0.724]
ViT-Tiny	−0.520 [−1.999, 0.959]	−0.006 [−0.021, 0.010]	−0.011 [−0.044, 0.021]

Adding stored-logit distillation to ER does not reduce forgetting in every pre-trained pipeline. In ResNet-18, forgetting decreases by about 32 pp relative to ER, with the interval wholly above zero. In ViT-Tiny, the mean change is instead a small increase of 0.5 pp, and the interval spans zero. Here, dataset, recipe, and random-stream control differ from the component study, and support is not manipulated. The comparison therefore supplies context for adding distillation rather than evidence of cross-dataset generalization of the support effect. Its seed-level estimates remain separate from the five-order analysis.

D.4 External-Framework Replay and Distillation Comparisons

To examine acquisition and retention beyond our implementation, we run ER, DER, DER++, and X-DER in the Mammoth framework (Buzzega et al., 2020; Boschini et al., 2023) under specified sequential CIFAR-100 recipes with 50 epochs, batch size 32, and five optimization seeds. Five paired contrasts compare adjacent methods and their cumulative differences from ER. Table 12

reports the mean paired differences and 95% t intervals. Appendix J.5 specifies the framework and configuration sources.

Table 12: External-framework replay and distillation comparisons in Mammoth across five seeds, in percentage points with 95% t intervals. The DER++ − ER and X-DER − ER rows are algebraic composites of the adjacent-step contrasts and share training runs with them (↓ lower is better, ↑ higher is better).

Contrast	Outcome	Mean [95% CI] (pp)
DER − ER	Acquisition (↑)	-2.4 [-2.9, -2.0]
	AF (↓)	-17.6 [-18.3, -16.8]
	FAA (↑)	+13.4 [+12.8, +14.0]
	$\mathrm{AIA}_{\mathrm{time}}$ (↑)	+10.5 [+8.9, +12.1]
DER++ − DER	Acquisition (↑)	-1.6 [-2.0, -1.2]
	AF (↓)	-3.0 [-3.8, -2.1]
	FAA (↑)	+1.0 [+0.5, +1.6]
	$\mathrm{AIA}_{\mathrm{time}}$ (↑)	+1.8 [-0.0, +3.7]
DER++ − ER (composite)	Acquisition (↑)	**-4.0** [-4.8, -3.2]
	AF (↓)	-20.5 [-21.6, -19.4]
	FAA (↑)	+14.4 [+14.0, +14.8]
	$\mathrm{AIA}_{\mathrm{time}}$ (↑)	+12.3 [+11.1, +13.6]
X-DER − DER++	Acquisition (↑)	-17.2 [-17.6, -16.7]
	AF (↓)	-33.6 [-35.0, -32.3]
	FAA (↑)	+13.9 [+12.8, +15.0]
	$\mathrm{AIA}_{\mathrm{time}}$ (↑)	+8.6 [+7.6, +9.5]
X-DER − ER (composite)	Acquisition (↑)	-21.2 [-22.3, -20.1]
	AF (↓)	**-54.2** [-54.6, -53.7]
	FAA (↑)	**+28.3** [+27.5, +29.2]
	$\mathrm{AIA}_{\mathrm{time}}$ (↑)	**+20.9** [+20.1, +21.7]

Across the five specified contrasts, final accuracy improves and forgetting decreases while acquisition falls. X-DER has the largest accuracy and stability gains relative to ER, accompanied by the largest acquisition cost. This comparison illustrates the value of evaluating acquisition alongside final performance in an external implementation. The methods change target handling, classification losses, and buffer operations, so their ordering is not a controlled increase in a single distillation strength.

D.5 Sequential Empirical Reference

A large-buffer experiment supplies a sequential empirical reference for the forgetting reduction achievable through replay alone. This reference uses the same class order and paired seeds as the retrospective ImageNet-100 stability comparison, but expands the replay memory from 2,000 to 20,000 examples. It trains ER with the same procedure and no distillation. Table 13 compares the forgetting reduction of this 20,000-example ER reference relative to the standard 2,000-example ER and DER++ baselines. Appendix J.6 specifies the protocol.

Relative to 2,000-example ER, the large-buffer reference reduces forgetting substantially in both pipelines, with intervals wholly above zero. Relative to DER++, however, the headroom is negative in ResNet-18, indicating that ten times the replay budget without distillation does not match the stability provided by stored-logit regression at the smaller buffer size. In ViT-Tiny, the headroom relative to DER++ remains positive. The reference therefore characterizes what replay volume alone can achieve. It is not an equal-budget competitor or a theoretical lower bound on forgetting.

E Relative Repair and Empirical Scope

Let $u = \mathrm{AF}_{\mathrm{ER}} - \mathrm{AF}_{\mathrm{dis}}$ denote forgetting repaired by adding distillation to ER. Positive u means less forgetting, reversing the treatment-minus-control AF sign. For denominator baseline b and reference

Table 13: Sequential empirical reference: AF headroom from 20,000-example ER over the indicated baseline under a fixed ImageNet-100 class order (in percentage points with 95% t intervals). **Bold** marks the larger headroom within each pipeline and underline marks the smaller headroom. Supported indicates lower bound exceeding $\tau = .01$.

Pipeline	Baseline	AF headroom (pp)	95% CI	Supported
ResNet-18	ER	$+23.4$	$[+22.5, +24.3]$	✓
	DER++	$\underline{-8.3}$	$[-12.8, -3.9]$	×
ViT-Tiny	ER	$\underline{+23.0}$	$[+21.6, +24.4]$	✓
	DER++	$+23.5$	$[+22.0, +25.1]$	✓

forgetting r, define

$$R_b = \frac{u}{b - r}, \qquad b > r. \tag{6}$$

Choosing $b = \mathrm{AF}_{\mathrm{NoReplay}}$ gives *shared repair*, the additional forgetting reduction beyond ER as a fraction of No Replay headroom. Choosing $b = \mathrm{AF}_{\mathrm{ER}}$ gives *nested repair*, the fraction of remaining ER headroom repaired by distillation. A structural-zero reference sets $r = 0$, meaning zero measured forgetting, not an optimal learner. These evaluation denominators differ from the training-loss denominators in Equation 3.

What outcome conditioning changes. Table 14 reports the retrospective CIFAR-100 panel, including replay and parameter-efficient comparisons. Each row averages three paired seeds within one of three class orders. Appendix J.2 specifies the orders, seeds, and replay objectives. The endpoint rule flags positive AF repair when FAA, final old-task accuracy, or final current-task accuracy decreases. Table 15 supplies the latter two outcomes. The trajectory rule instead checks time-averaged accuracy and mean acquisition. Both use a zero-decrease threshold, so these are descriptive sign-based classifications, not practical-equivalence tests. The endpoint rule identifies tradeoffs in 12 of the 24 comparison–order rows, whereas the trajectory rule identifies 14, differing in two rows. Shared orders and baselines mean that these rows are not independent studies or an estimated population frequency. The continuous effects show how outcome choice changes the interpretation of stability gains.

Adaptation settings. The full-fine-tuning comparisons use the ResNet-18 and ViT-Tiny checkpoints described in Appendix A. No Replay denotes sequential fine-tuning on the current class group without replay, not joint training. The parameter-efficient comparisons use the same ViT-Tiny AugReg checkpoint. The Frozen baseline trains only the classification head. L2P uses a pool of ten prompts of length five, selects five prompts using class-token features and learned keys, and uses a pull-constraint coefficient of one (Wang et al., 2022b). No transformer blocks are unfrozen, and no replay is used. CL-LoRA uses rank-16 adapters with scaling coefficient 16 and zero adapter dropout on the combined query/key/value and attention-output projections (He et al., 2025). Its replay and no-replay conditions share the adaptation setup. Replay uses a 2,000-example memory and a replay cross-entropy coefficient of .05, while the no-replay condition disables that term. All panel runs use ten stages, 20 epochs per stage, batch size 64, SGD with initial learning rate .005 and momentum .9, zero weight decay, mixed precision, and the 224×224 input protocol. These fixed-recipe comparisons characterize outcome tradeoffs rather than tuned method rankings.

Table 14: Retrospective CIFAR-100 comparisons. Differences in percentage points averaged over three paired seeds per order. Acq., End., and Traj. denote acquisition, endpoint, and trajectory rules. **T** marks a forgetting reduction accompanied by an outcome tradeoff; dashes indicate no tradeoff.

Setting	Method / Baseline	Order	ΔAcq.	ΔAF	ΔFAA	$\Delta\mathrm{AIA}_{\mathrm{time}}$	End.	Traj.
RN	ER / No Replay	1	$+0.41$	-39.81	$+36.24$	$+36.59$	—	—
		2	$+0.61$	-41.58	$+38.04$	$+38.39$	—	—
		3	$+0.33$	-39.25	$+35.66$	$+37.00$	—	—
RN	DER++-style / No Replay	1	-27.42	-76.59	$+47.20$	$+44.74$	**T**	**T**
		2	-21.81	-73.84	$+47.90$	$+45.90$	**T**	**T**

Setting	Method / Baseline	Order	ΔAcq.	ΔAF	ΔFAA	$\Delta\mathrm{AIA}_{\mathrm{time}}$	End.	Traj.
		3	-8.53	-64.48	$+49.50$	$+45.22$	**T**	**T**
RN	Adapted X-DER / No Replay	1	-0.97	-52.70	$+46.45$	$+43.34$	—	**T**
		2	-1.18	-54.00	$+47.42$	$+45.53$	**T**	**T**
		3	-1.69	-52.16	$+45.25$	$+44.13$	**T**	**T**
ViT	ER / No Replay	1	$+1.46$	-43.37	$+40.49$	$+36.60$	—	—
		2	$+0.08$	-53.41	$+48.15$	$+44.70$	—	—
		3	$+0.20$	-51.57	$+46.61$	$+44.36$	—	—
ViT	DER++-style / No Replay	1	-0.54	-55.47	$+49.38$	$+43.36$	—	**T**
		2	-2.15	-67.83	$+58.89$	$+52.52$	**T**	**T**
		3	-1.64	-67.47	$+59.08$	$+52.81$	**T**	**T**
ViT	Adapted X-DER / No Replay	1	$+0.25$	-52.10	$+47.14$	$+41.60$	—	—
		2	-1.87	-63.01	$+54.84$	$+49.41$	**T**	**T**
		3	-1.69	-61.45	$+53.61$	$+48.88$	**T**	**T**
ViT	L2P / Frozen	1	$+6.62$	$+2.99$	$+3.93$	$+3.40$	—	—
		2	$+6.95$	$+3.22$	$+4.05$	$+3.10$	—	—
		3	$+5.74$	$+3.41$	$+2.67$	$+2.74$	—	—
ViT/LoRA	Replay / No Replay	1	-0.50	-48.11	$+42.80$	$+37.03$	**T**	**T**
		2	-0.51	-48.82	$+43.43$	$+35.69$	**T**	**T**
		3	-0.50	-50.18	$+44.66$	$+35.70$	**T**	**T**

Table 15: Final old-task ($A_{T,\mathrm{old}}$) and current-task ($A_{T,T}$) accuracy differences for the retrospective CIFAR-100 panel in percentage points. Together with Table 14, these determine the endpoint rule. **T** denotes a tradeoff.

Setting	Method / Baseline	Order	$\Delta A_{T,\mathrm{old}}$	$\Delta A_{T,T}$	End.
RN	ER / No Replay	1	$+40.08$	$+1.70$	—
		2	$+42.17$	$+0.80$	—
		3	$+39.51$	$+0.97$	—
RN	DER++-style / No Replay	1	$+57.47$	-45.23	**T**
		2	$+55.67$	-22.07	**T**
		3	$+56.07$	-9.67	**T**
RN	Adapted X-DER / No Replay	1	$+51.60$	$+0.17$	—
		2	$+52.75$	-0.57	**T**
		3	$+50.37$	-0.77	**T**
ViT	ER / No Replay	1	$+44.62$	$+3.33$	—
		2	$+53.44$	$+0.57$	—
		3	$+51.66$	$+1.17$	—
ViT	DER++-style / No Replay	1	$+54.81$	$+0.57$	—
		2	$+65.57$	-1.27	**T**
		3	$+65.82$	-1.60	**T**
ViT	Adapted X-DER / No Replay	1	$+52.24$	$+1.20$	—
		2	$+61.01$	-0.73	**T**
		3	$+59.63$	-0.57	**T**
ViT	L2P / Frozen	1	$+3.63$	$+6.60$	—
		2	$+3.89$	$+5.57$	—
		3	$+2.47$	$+4.47$	—
ViT/LoRA	Replay / No Replay	1	$+47.61$	-0.47	**T**
		2	$+48.35$	-0.83	**T**
		3	$+49.69$	-0.57	**T**

Connecting total and incremental repair. For baseline forgetting b, intervention forgetting a, and reference forgetting r, write the total relative repair as $\rho(a; b, r) = (b - a)/(b - r)$, with $b > r$.

With a common No Replay denominator, the shared increment in Equation 6 is

$$R_{\mathrm{AF}_{\mathrm{NoReplay}}} = \rho(\mathrm{AF}_{\mathrm{dis}}; \mathrm{AF}_{\mathrm{NoReplay}}, r) - \rho(\mathrm{AF}_{\mathrm{ER}}; \mathrm{AF}_{\mathrm{NoReplay}}, r). \tag{7}$$

The nested increment is instead $R_{\mathrm{AF}_{\mathrm{ER}}} = \rho(\mathrm{AF}_{\mathrm{dis}}; \mathrm{AF}_{\mathrm{ER}}, r)$. Thus, shared repair subtracts two total repairs with a common denominator, whereas nested repair directly treats ER as the baseline. Structural zero is a reference on the forgetting functional, not a chance accuracy or a joint-training score. Full-sequence joint training uses future data and therefore cannot supply a sequentially attainable forgetting reference. Alternative normalized accuracy measures also require their original inputs and time axes rather than substitution of an accuracy reference into an AF formula (Serrà et al., 2018; Hayes et al., 2018; Michel et al., 2023).

Tables 14 and 15 show that accuracy and acquisition change the interpretation of some stability gains, while normalization preserves rankings within comparable strata. All ratios in this panel remain estimable at headroom thresholds $\tau \in \{.005, .01, .02\}$. As Table 16 shows, even the smallest nested headroom in the replay comparisons lies well above these thresholds. The observed tradeoffs therefore call for attention to learning outcomes, not a small-denominator explanation. The large-buffer sequential reference in Appendix D.5 uses a different class order and protocol and is therefore not part of these ratio calculations. The next section uses simulation to examine ratio reliability where headroom is less informative.

Table 16: Observed shared (No Replay AF) and nested (ER AF) headroom in percentage points, reporting the mean and [min, max] range across paired seed realizations ($\tau = .01$ equals 1 pp).

Dataset	Pipeline	Shared Headroom	Nested Headroom
ImageNet-100	ResNet-18	95.71 [95.51, 95.87]	49.24 [48.53, 50.18]
	ViT-Tiny	94.64 [93.93, 94.93]	46.01 [45.33, 47.42]
CIFAR-100	ResNet-18	93.93 [93.40, 94.46]	53.71 [51.24, 56.19]
	ViT-Tiny	92.52 [77.37, 95.04]	43.07 [39.40, 51.92]

F Numerical Simulations of Ratio Reliability

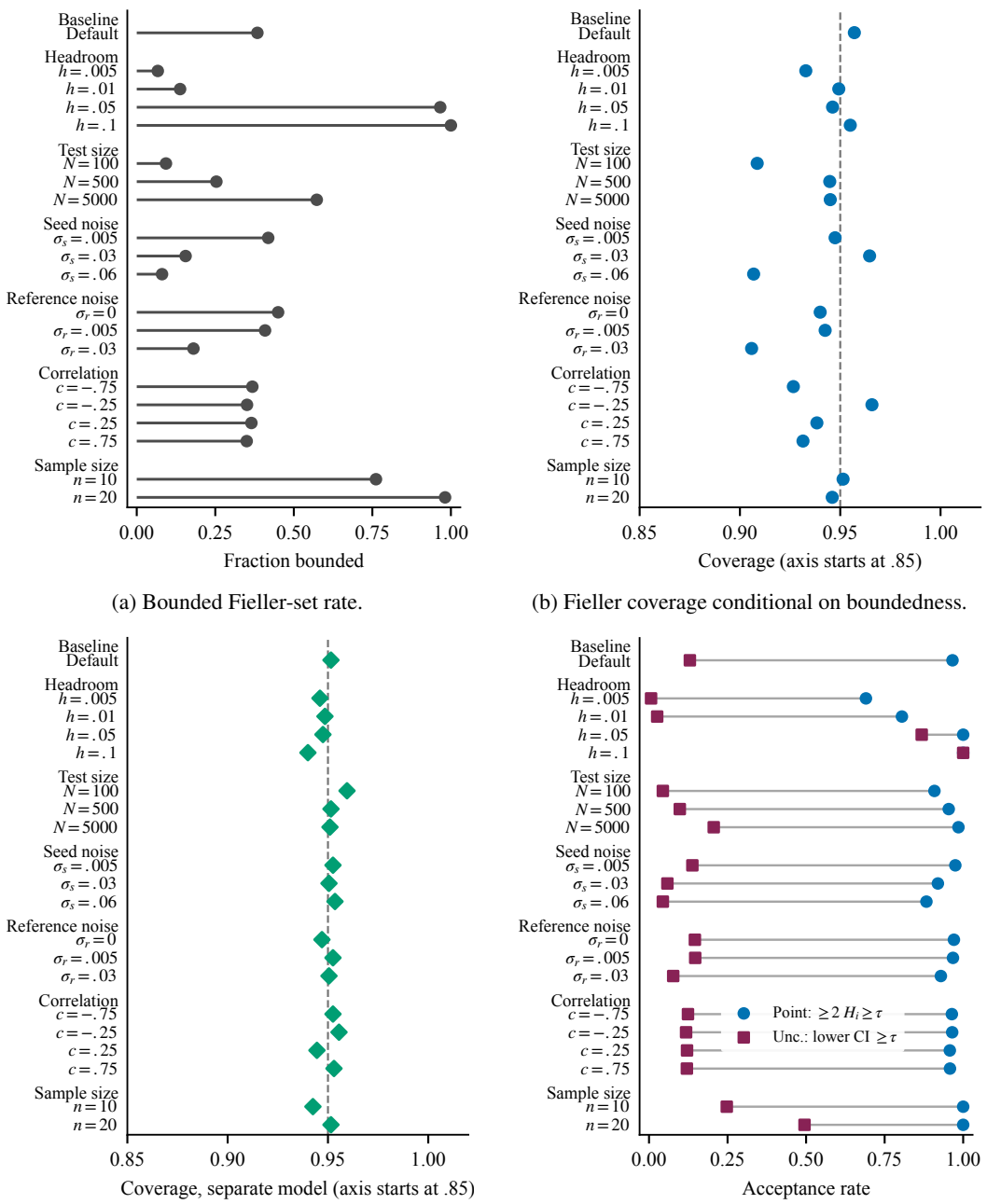
When headroom is small or uncertain, reporting a ratio requires considering both whether a finite interval is available and how reliably it covers the target. We examine these questions in 20 one-factor-at-a-time settings with 2,000 repetitions each. A point-threshold rule accepts a sample when at least two individual headroom observations satisfy $H_i \geq \tau$, with $\tau = .01$. An uncertainty-aware rule instead requires the lower endpoint of the 95% t interval for the mean headroom to reach τ. Figure 8 compares their acceptance rates with interval boundedness and coverage under the aggregate noise models specified below. This is a simulation-based characterization of the rules rather than an independently validated decision criterion for continual learners.

For each repetition, we draw n independent pairs

$$U_i = \rho h + \epsilon_{u,i} + \eta_{u,i}, \qquad H_i = h + \epsilon_{h,i} + \xi_i + \eta_{h,i}, \tag{8}$$

where $(\epsilon_{u,i}, \epsilon_{h,i})$ is zero-mean Gaussian with marginal standard deviation σ_s and correlation c. Independently, $\xi_i \sim \mathcal{N}(0, \sigma_r^2)$ and $\eta_{u,i}, \eta_{h,i} \sim \mathcal{N}(0, .25/N)$. The test-size parameter N controls this Gaussian measurement noise, rather than indexing a sampled classification test set. The ratio-of-means truth is $\mathbb{E}[U]/\mathbb{E}[H] = \rho = .4$. Defaults are $h = .02$, $n = 5$, $N = 1000$, $\sigma_s = \sigma_r = .01$, and $c = 0$. Each nondefault setting changes one factor: $h \in \{.005, .01, .05, .1\}$, $N \in \{100, 500, 5000\}$, $\sigma_s \in \{.005, .03, .06\}$, $\sigma_r \in \{0, .005, .03\}$, $c \in \{-.75, -.25, .25, .75\}$, or $n \in \{10, 20\}$. Random seeds are $20260919 + j$ for settings indexed $j = 0, \ldots, 19$ in the displayed order.

The mean-of-ratios calculation uses a separate positive-denominator model. Set $v = \log(1 + \min(\sigma_s/h, 1)^2)$, draw $\log H_i \sim \mathcal{N}(\log h - v/2, v)$, and independently draw $R_i \sim \mathcal{N}(\rho, s_R^2)$ with $s_R = \min(.2, \sqrt{.25/N}/h)$. Taking $U_i = R_i H_i$ makes $\mathbb{E}[U_i/H_i] = \rho$. Coverage is measured for the ordinary 95% Student-t interval of the n individual ratios. It is not a comparison of two interval methods on the same data-generating model.



(a) Bounded Fieller-set rate.

(b) Fieller coverage conditional on boundedness.

(c) Mean-of-ratios coverage.

(d) Acceptance of reporting rules.

Figure 8: Ratio reliability under aggregate noise, grouped by the one-factor-at-a-time setting (where "sd" denotes standard deviation). Coverage axes in (b) and (c) start at .85, and the dashed line marks nominal 95% coverage. In (d), lines join the two rules within a setting and are not confidence intervals. Subfigures (b) and (c) are not a comparison of two interval methods on the same data-generating model. High conditional coverage need not imply frequent availability of a finite interval. Each row uses 2,000 repetitions.

Figure 8 compares two interval constructions for a ratio of means. Fieller sets retain numerator–denominator covariance when targeting a ratio of means, whereas the separate mean-of-ratios analysis targets the average of individual ratios. We retain unbounded confidence sets rather than making them finite with an added constant. The figure shows the practical consequence: reasonable coverage among bounded cases can coexist with infrequent availability of a bounded interval. Both

quantities are therefore needed to describe ratio reliability. The simulation concerns noisy aggregate quantities and does not reconstruct the task-level historical maximum from summary statistics. Together with the headroom analysis in Appendix E, these simulations delimit the conditions under which a normalized stability gain can be reported reliably.

G Weight-Scaled Component Effects

To compare component effects relative to the nominal weight at which they are measured, we introduce a weight-scaled descriptor. For component $X \in \{M, D\}$, where M denotes the support-restriction effect and D the active-output averaging effect, define $S_X = X_{.5}/.5 - X_{.1}/.1$ using raw treatment-minus-control differences. This descriptor compares effects per unit nominal weight. It is distinct from the unscaled interaction $X_{.5} - X_{.1}$. Table 17 reports order-level means, pointwise intervals, and leave-one-order-out (LOO) ranges in accuracy-fraction units per unit weight. These summaries describe effect modification between the two tested weights, without specifying a response curve or distinguishing linear from nonlinear mechanisms. Under a linear response to the nominal weight, the per-unit effects would agree across weights and S_X would be zero; departures from zero therefore describe how each component effect changes per unit weight, with signs following the raw outcome directions.

Table 17: Weight-scaled component effects S_X in fraction units per unit weight, where components M and D denote support restriction and active-output averaging, respectively. Entries report order-level means with pointwise 95% t intervals and five leave-one-order-out (LOO) means.

Pipeline	Component	Outcome	Mean [95% CI]	LOO Range
ResNet-18	M	FAA	1.0435 [0.7887, 1.2982]	[0.9731, 1.1009]
		AIA_{time}	0.7067 [0.6177, 0.7957]	[0.6912, 0.7344]
		AF	−1.5692 [−2.0822, −1.0562]	[−1.6716, −1.4530]
		Acquisition	−0.3109 [−0.5710, −0.0508]	[−0.3736, −0.2294]
	D	FAA	0.0406 [−0.0062, 0.0875]	[0.0258, 0.0507]
		AIA_{time}	0.0043 [−0.0068, 0.0154]	[0.0009, 0.0066]
		AF	−0.0451 [−0.0990, 0.0087]	[−0.0565, −0.0301]
		Acquisition	0.0000 [−0.0152, 0.0152]	[−0.0036, 0.0048]
ViT-Tiny	M	FAA	0.9379 [0.8421, 1.0338]	[0.9197, 0.9617]
		AIA_{time}	0.6267 [0.6125, 0.6408]	[0.6228, 0.6296]
		AF	−1.1737 [−1.2842, −1.0631]	[−1.2015, −1.1487]
		Acquisition	−0.1184 [−0.1371, −0.0997]	[−0.1241, −0.1141]
	D	FAA	−0.1667 [−0.2146, −0.1188]	[−0.1749, −0.1509]
		AIA_{time}	−0.1666 [−0.2089, −0.1242]	[−0.1757, −0.1529]
		AF	0.1727 [0.1389, 0.2065]	[0.1630, 0.1803]
		Acquisition	−0.0113 [−0.0354, 0.0129]	[−0.0175, −0.0042]

These descriptors complement the unscaled interactions reported in Tables 5 and 6: the interaction measures how an effect changes between the two weights, whereas S_X expresses the same comparison relative to the nominal weight at which each effect was measured.

H Study Design and Reproduction

This section summarizes which design decisions were fixed before analysis and what is required to reproduce the study. The component design, contrasts, and analysis units were fixed in advance. The ImageNet-100 support-restriction and combined-recipe studies are independent experiments and are not pooled into the component estimates. The ImageNet-100 stability comparison and the CIFAR-100 panel are analyzed under pre-specified protocols, and the numerical simulations characterize ratio uncertainty under specified noise models.

The tables provide every planned component effect, its interval, order means, and inferential sensitivities. Reproduction requires the specified pre-trained checkpoints and the same class orders,

optimization seeds, replay semantics, and input transformations. Appendix A specifies the objective and training procedure, while Tables 3 and 18 provide the exact randomization settings.

The component study uses Python 3.11 and PyTorch 2.10.0 with CUDA 12.8 on an NVIDIA RTX 5070 Ti. The 210 component-study training runs take approximately 120 hours of single-GPU wall time. The ImageNet-100 support-restriction experiment contains 36 runs, the sequential reference contains 10 runs, and the external-framework comparison contains 20 runs in Mammoth. The gradient-norm experiment contains 28 complete runs, with ResNet-18 pairs on an RTX 4070 Ti SUPER and ViT-Tiny pairs on an RTX 5070 Ti. Appendix K details the software environments and execution protocol. The local probe measurements and trajectory decompositions are diagnostic exploratory analyses kept separate from these complete-run comparisons.

Reproducibility Statement

Sections 3–4 define the objectives and paired comparisons. Appendices A–C specify training, target storage, statistical aggregation, and diagnostics. Appendix F gives the simulation models, and Appendix J specifies the auxiliary comparison protocols. Appendices K–N describe norm matching, the support-effect decomposition, local probes, and the combined-recipe analysis. Exact class orders and optimization seeds accompany the complete-run comparisons.

AI use statement

In this work, we used generative AI tools to implement methods, assist with translation, create or modify scientific figures, draft parts of the paper, and edit the paper to improve readability. All AI-assisted outputs were reviewed and verified by the authors: generated text was checked against the experimental results and revised where it did not match them, and generated code was tested for correctness. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

I Exact Class Sequences

The following table specifies the five class sequences used by all seven component-study conditions. Label indices refer to the standard CIFAR-100 dataset, and task boundaries follow each consecutive group of ten labels. Each order was drawn once and then held fixed for every condition in both pipelines, so differences between conditions within an order cannot be attributed to class composition. The additional orders used by the auxiliary comparisons, including the development and test orders of the combined-recipe study and the ImageNet-100 stress order, are listed separately in Appendix J.

Table 18: Exact CIFAR-100 class orders using standard dataset label indices. Each row is one sequential group of ten classes.

Order	Stage	Class Indices
Order 1	1	94, 60, 90, 27, 4, 18, 51, 99, 98, 89
	2	64, 41, 68, 53, 82, 32, 57, 93, 91, 59
	3	63, 34, 35, 16, 87, 54, 30, 44, 42, 70
	4	26, 85, 10, 0, 45, 28, 33, 65, 52, 19
	5	67, 25, 40, 50, 49, 88, 3, 73, 6, 80
	6	2, 61, 83, 8, 86, 39, 47, 75, 11, 78
	7	62, 71, 48, 81, 77, 14, 15, 74, 37, 95
	8	20, 17, 66, 72, 38, 92, 76, 13, 29, 22
	9	21, 46, 7, 24, 96, 9, 58, 36, 69, 5
	10	23, 97, 1, 31, 79, 12, 56, 43, 55, 84
Order 2	1	82, 65, 41, 67, 83, 31, 86, 21, 28, 7
	2	89, 30, 44, 6, 78, 76, 13, 97, 98, 50
	3	15, 3, 51, 45, 0, 66, 81, 71, 93, 42

Table 18: Exact CIFAR-100 class orders (continued).

Order	Stage	Class Indices
	4	47, 57, 20, 56, 70, 75, 54, 80, 2, 69
	5	29, 96, 5, 92, 90, 32, 25, 26, 74, 49
	6	48, 1, 91, 87, 36, 10, 94, 59, 95, 88
	7	4, 55, 64, 99, 19, 79, 73, 12, 9, 77
	8	39, 58, 38, 22, 72, 63, 16, 34, 60, 33
	9	14, 85, 43, 37, 24, 40, 8, 27, 17, 61
	10	11, 52, 68, 18, 84, 35, 53, 62, 23, 46
Order 3	1	59, 66, 21, 67, 45, 16, 11, 31, 97, 24
	2	55, 57, 74, 22, 76, 73, 29, 28, 49, 25
	3	41, 69, 93, 38, 0, 89, 34, 56, 9, 53
	4	72, 4, 58, 3, 5, 94, 42, 75, 12, 77
	5	83, 46, 6, 37, 47, 81, 91, 26, 54, 51
	6	70, 10, 8, 13, 80, 90, 85, 64, 60, 36
	7	2, 87, 63, 33, 78, 14, 23, 71, 61, 39
	8	32, 30, 48, 96, 40, 82, 17, 68, 79, 92
	9	20, 50, 1, 19, 86, 95, 88, 65, 84, 43
	10	99, 18, 35, 98, 52, 7, 27, 62, 15, 44
Order 4	1	18, 40, 76, 59, 88, 12, 16, 31, 4, 51
	2	39, 63, 60, 21, 19, 24, 14, 44, 33, 5
	3	42, 22, 41, 85, 52, 73, 82, 98, 32, 25
	4	64, 1, 3, 28, 84, 7, 17, 91, 0, 79
	5	9, 43, 26, 92, 72, 48, 10, 23, 97, 70
	6	65, 68, 45, 34, 30, 69, 11, 15, 96, 81
	7	77, 56, 53, 6, 57, 47, 29, 94, 35, 50
	8	13, 46, 86, 49, 89, 95, 58, 27, 80, 90
	9	38, 37, 55, 71, 62, 75, 36, 2, 83, 74
	10	93, 78, 67, 20, 61, 99, 54, 87, 66, 8
Order 5	1	34, 52, 62, 38, 20, 3, 21, 15, 54, 69
	2	29, 11, 0, 98, 19, 77, 37, 82, 71, 79
	3	70, 31, 48, 16, 58, 59, 12, 45, 84, 60
	4	24, 99, 91, 81, 25, 76, 43, 32, 94, 53
	5	55, 67, 28, 75, 26, 17, 86, 18, 6, 13
	6	88, 61, 93, 64, 68, 63, 87, 44, 4, 9
	7	23, 92, 46, 8, 30, 2, 50, 56, 47, 10
	8	49, 96, 83, 7, 40, 35, 14, 74, 1, 97
	9	78, 90, 72, 41, 95, 85, 36, 33, 5, 66
	10	51, 42, 73, 22, 27, 89, 65, 80, 57, 39

J Protocols for Auxiliary Comparisons

This appendix specifies the protocols of the auxiliary comparisons cited in the main text: the ImageNet-100 stability comparison, the retrospective CIFAR-100 panel, the combined-recipe selection procedure, the ImageNet-100 support-restriction comparison, the external-framework comparison, and the sequential empirical reference. Each of these comparisons differs from the component study in dataset, order construction, method implementation, or random-stream control, and the relevant differences are stated in the corresponding subsection.

J.1 ImageNet-100 Stability Comparison

The ImageNet-100 comparison in Table 11 uses the 100 synsets and ten task groups listed in Table 19. The fixed order is constructed from two per-class scores computed from separate ImageNet-100 training runs: a difficulty score given by per-class reference accuracy, and a stress score given by a class-intrusion margin statistic. After reserving the final ten classes, the remaining 90 are sorted

by difficulty into ten strata of nine classes. Within each stratum, classes are assigned to the first nine stages in descending stress-score order, one class per stratum per stage. The final group is held fixed. This construction defines the stress order used here, rather than a random draw from possible class orders, and Table 19 lists the resulting sequence exactly.

Each of No Replay, ER, and unmasked stored-logit distillation is evaluated under optimization seeds 123, 456, 789, 1024, and 2026 in both pipelines. All backbone and head parameters are trained for 20 epochs per stage with SGD, initial learning rate .005, momentum .9, and zero weight decay. The learning rate follows cosine decay within each stage, and optimizer state resets at stage boundaries. Current and replay batches contain up to 64 examples. ER uses a 2,000-example memory and replay coefficient $\alpha = .1$. Distillation adds full-head mean squared error with $\beta = .5$. These runs use full precision. Images undergo random-resized cropping to 224×224 and horizontal flipping for training, and resizing to 256 followed by center cropping for evaluation, with ImageNet normalization. The backbones use the pre-training sources described in Appendix A and randomly initialized 100-output heads.

The buffer stores augmented input tensors and pre-step logits during the final epoch of each stage and reuses the stored tensors for replay. Both replay terms share a minibatch. Unlike the component study, these runs do not use separate paired random streams for replay sampling and reservoir updates. Together with the dataset and input transformation differences, this makes the comparison a setting-level illustration rather than a replication of an isolated component intervention.

J.2 Retrospective CIFAR-100 Panel

The three retrospective orders use order seeds 73129, 88241, and 96457, each with paired optimization seeds 31415, 27182, and 16180. Table 20 gives their exact permutations, separately from the five orders of the component study. The training and input settings are specified in Appendix E. The DER++-style condition uses full-head unmasked MSE with $\alpha = .1$ and $\beta = .5$, while ER sets $\beta = 0$. Current-data and DER++-style replay cross-entropy use the full head. The buffer-update and stored-input conventions match Appendix A, without the separate deterministic replay streams used in the component study.

The X-DER comparison examines target updating through an adapted objective. Current-data cross-entropy uses the full head. Replay cross-entropy is restricted to classes seen at replay time and has coefficient .5. Distillation uses $T_d^2\,\mathrm{KL}(\mathrm{softmax}(z/T_d)\,\|\,\mathrm{softmax}(f_\theta(x)/T_d))$, averaged over replay examples, with coefficient .5 and temperature $T_d = 2$. Before training each stage, an evaluation-mode forward pass on stored inputs refreshes target coordinates for the newly introduced classes, leaving other coordinates unchanged. This adaptation omits the additional future-output preparation and constraint terms of reference X-DER. It therefore characterizes a specified target-update recipe rather than reproducing or benchmarking the complete reference method.

J.3 Combined-Recipe Selection and Testing

Recipe selection uses one development order (seed 61803) and two optimization seeds (48017 and 99139) in each pipeline. The candidates are masked/full and masked/active regression, each at $\beta \in \{.1, .25, .5, 1\}$, with unmasked $\beta = .5$ and ER as controls. These ten conditions yield 40 development runs. Among masked candidates on the four-outcome Pareto frontier in both pipelines, masked/active at .1 and masked/full at .5 remain. The former is selected for its higher development FAA and lower AF in both pipelines, with an acquisition difference of at most .00045 in fraction units.

The selected recipe and the two controls are evaluated on a separate test order (seed 93479), not used for recipe selection, with optimization seeds 25013, 60127, and 88191. The resulting 18 runs supply Table 9. Development and test runs use the component-study training settings, including paired replay streams, but do not enter its five-order estimates. Table 20 supplies both class permutations.

J.4 ImageNet-100 Support-Restriction Comparison

The ImageNet-100 support-restriction experiment in Table 10 uses the same two pipelines (ResNet-18 ImageNet-1K and ViT-Tiny AugReg) with randomly initialized 100-output heads. Images use

random-resized cropping to 224×224 and horizontal flipping for training, and resizing to 256 followed by center cropping for evaluation, with ImageNet normalization. Training uses 20 epochs per stage, batch size 64, SGD with initial learning rate .005, momentum .9, zero weight decay, mixed precision, and a 2,000-example memory with replay coefficient $\alpha = .1$. Both conditions use full denominator at $\beta = .5$: the unmasked condition constrains all C outputs, while the masked/full condition constrains only outputs for classes observed when each example entered memory. Three independent class orders are drawn uniformly from the 100 ImageNet-100 synsets, each with three paired optimization seeds per pipeline. Table 21 lists the three sequences together with their order seeds and paired optimization seeds. Replay sampling and memory-update random streams are paired across conditions within each order and seed. This experiment isolates the support-restriction intervention on a second dataset.

J.5 External-Framework Comparison Protocol

The external-framework comparison in Table 12 uses the official Mammoth codebase (Boschini et al., 2023) under the sequential CIFAR-100 benchmark. Four methods are evaluated: ER, DER, DER++, and X-DER, using a ResNet-18 backbone trained from scratch, 50 epochs, batch size 32, SGD with learning rate .03 and momentum .9, ten tasks, and a 500-example buffer. Optimization seeds are 0, 1, 2, 3, and 4. DER, DER++, and X-DER use the upstream sequential CIFAR-100 best configurations for this buffer size. ER uses the default codebase hyperparameters with the X-DER dataset configuration, not an upstream best ER configuration. Because this comparison traces how acquisition, forgetting, and accuracy move together as prediction constraints strengthen across complete recipes, rather than ranking methods under equal tuning budgets, the ER reference serves as a shared starting point for the adjacent contrasts. The framework manages its own buffer updates, data augmentation, and cross-entropy formulation. In particular, reference DER++ draws separate replay minibatches for the two loss terms. These runs train all parameters without mixed precision and compare the specified complete recipes rather than isolated objective components.

J.6 Sequential Empirical Reference Protocol

The sequential empirical reference in Table 13 uses the same class order and optimization seeds as the ImageNet-100 stability comparison in Table 11, whose protocol is specified in Appendix J.1. The memory capacity is expanded from 2,000 to 20,000 examples, and the replay coefficient is $\alpha = .1$. No distillation is applied ($\beta = 0$). All other training settings, including the backbone, head initialization, optimizer, and input transformations, match the ImageNet-100 stability protocol. Five optimization seeds are run in each pipeline. The resulting forgetting is compared with the 2,000-example ER and DER++ results from the same order.

The three tables that follow list the exact class sequences referenced above: Table 19 for the ImageNet-100 stress order, Table 20 for the retrospective and combined-recipe CIFAR-100 orders, and Table 21 for the ImageNet-100 support-restriction orders. Appendix K separately specifies the seven-order gradient-norm experiment.

Table 20: Additional CIFAR-100 class sequences across ten tasks using standard indexing. Seeds: Retrospective 1–3 (73129, 88241, 96457), Development (61803), Test (93479).

Order	Task	Class indices
Development	1	76, 37, 9, 69, 61, 55, 68, 11, 3, 84
	2	60, 85, 36, 73, 62, 1, 40, 14, 26, 77
	3	96, 7, 13, 53, 90, 44, 38, 93, 27, 39
	4	43, 75, 45, 19, 91, 2, 56, 99, 95, 49
	5	0, 67, 63, 48, 86, 59, 30, 23, 94, 33
	6	97, 87, 5, 35, 92, 98, 47, 29, 17, 8
	7	81, 89, 51, 72, 20, 6, 15, 42, 32, 46
	8	88, 54, 34, 80, 78, 41, 70, 24, 71, 65
	9	21, 66, 57, 64, 16, 4, 25, 52, 74, 22
	10	12, 18, 50, 28, 10, 83, 31, 79, 58, 82
	1	66, 63, 97, 27, 17, 43, 0, 50, 40, 74
	2	3, 4, 82, 39, 60, 79, 13, 68, 45, 57

Order	Task	Class indices
	3	10, 46, 91, 70, 1, 31, 33, 69, 71, 64
	4	11, 92, 85, 87, 18, 76, 52, 26, 62, 15
	5	89, 22, 7, 94, 8, 95, 25, 59, 61, 35
	6	16, 84, 48, 12, 28, 14, 98, 96, 38, 86
	7	73, 77, 42, 29, 41, 34, 30, 47, 2, 88
	8	75, 44, 81, 80, 49, 9, 23, 51, 32, 20
	9	37, 21, 36, 56, 99, 78, 54, 65, 55, 90
	10	24, 19, 5, 58, 6, 93, 67, 53, 72, 83
Retrospective 2	1	48, 44, 60, 41, 64, 16, 82, 74, 81, 68
	2	56, 70, 73, 92, 23, 7, 63, 67, 54, 22
	3	88, 72, 33, 96, 84, 25, 18, 36, 43, 40
	4	19, 89, 37, 31, 53, 17, 42, 78, 49, 0
	5	13, 35, 6, 69, 77, 32, 26, 98, 59, 61
	6	79, 83, 75, 12, 76, 38, 65, 90, 47, 95
	7	10, 55, 15, 3, 2, 8, 28, 34, 45, 93
	8	66, 85, 99, 11, 27, 50, 21, 29, 62, 57
	9	9, 58, 86, 97, 71, 46, 80, 20, 52, 5
	10	30, 87, 51, 1, 4, 39, 94, 91, 14, 24
Retrospective 3	1	52, 83, 84, 7, 81, 25, 19, 49, 17, 10
	2	75, 87, 39, 59, 68, 95, 73, 48, 15, 92
	3	91, 55, 44, 40, 38, 67, 98, 71, 77, 32
	4	28, 54, 8, 27, 21, 88, 37, 14, 64, 29
	5	24, 31, 35, 13, 90, 62, 93, 9, 18, 4
	6	79, 72, 11, 99, 45, 89, 43, 34, 63, 46
	7	2, 41, 1, 6, 50, 78, 70, 3, 42, 94
	8	56, 61, 51, 66, 58, 60, 47, 26, 12, 65
	9	53, 36, 85, 22, 30, 16, 23, 86, 20, 0
	10	5, 96, 74, 69, 57, 97, 80, 82, 33, 76
Test	1	81, 10, 84, 19, 27, 65, 62, 80, 32, 11
	2	72, 26, 31, 12, 54, 8, 20, 56, 39, 98
	3	78, 95, 2, 68, 44, 70, 85, 94, 48, 90
	4	17, 86, 24, 15, 21, 18, 47, 33, 60, 76
	5	5, 55, 83, 3, 23, 92, 4, 99, 96, 52
	6	69, 73, 50, 67, 59, 0, 25, 43, 35, 58
	7	89, 51, 74, 40, 16, 14, 30, 77, 6, 22
	8	29, 9, 71, 87, 63, 57, 37, 28, 1, 46
	9	88, 93, 34, 82, 7, 42, 41, 36, 49, 53
	10	66, 79, 75, 38, 61, 64, 97, 45, 13, 91

Table 21: ImageNet-100 support-restriction class sequences across ten tasks, in WordNet synsets with five classes per row. Support 1 (order seed 1616461789, optimization seeds 380286043, 1046393546, 1236356477); Support 2 (order seed 1401189492, optimization seeds 416151448, 454674231, 531387718); Support 3 (order seed 1624133933, optimization seeds 1654451320, 1826329731, 1914896595).

Order	Task	Synsets
Support 1	1	n02910353, n03649909, n03498962, n02979186, n02109047, n01601694, n02037110, n06794110, n02123394, n01582220
	2	n01558993, n03481172, n02115641, n03240683, n02125311, n03891251, n03018349, n03742115, n04201297, n01537544
	3	n03016953, n04429376, n03223299, n03584254, n04033995, n09428293, n04380533, n03710721, n03041632, n02769748
	4	n01744401, n09288635, n04476259, n03047690, n02077923, n01818515, n07614500, n02342885, n02124075, n02110063
	5	n04008634, n03935335, n02006656, n02101006, n02007558, n09332890, n03376595, n04548362, n02423022, n04332243
	6	n02097474, n03995372, n07695742, n03658185, n02417914, n01616318, n01614925, n02791124, n01860187, n04049303

Order	Task	Synsets
	7	n03680355, n02840245, n02097209, n02825657, n01817953, n04099969, n01797886, n02916936, n01950731, n01608432
	8	n02963159, n01819313, n03271574, n04596742, n02095570, n04067472, n04259630, n04004767, n04392985, n03424325
	9	n02120079, n01580077, n02095314, n03337140, n02128757, n02088238, n01776313, n03929855, n02346627, n02447366
	10	n06596364, n04179913, n01796340, n01560419, n02666196, n03388183, n01592084, n03602883, n02978881, n03954731
Support 2	1	n02115641, n04004767, n01608432, n02769748, n02916936, n01776313, n03742115, n01560419, n04179913, n04596742
	2	n01537544, n02910353, n03271574, n02123394, n02109047, n01950731, n03649909, n02963159, n02037110, n03680355
	3	n01860187, n04392985, n03935335, n03584254, n04067472, n04380533, n01819313, n02840245, n01817953, n02077923
	4	n03223299, n07695742, n03929855, n02095570, n02128757, n04476259, n03602883, n03376595, n01601694, n02346627
	5	n06794110, n01797886, n02423022, n01582220, n01616318, n02095314, n02666196, n02825657, n03658185, n03016953
	6	n06596364, n02006656, n03424325, n03047690, n04429376, n03388183, n02125311, n02101006, n01796340, n03995372
	7	n01818515, n03337140, n04049303, n02342885, n04548362, n04008634, n04099969, n02791124, n09332890, n02979186
	8	n01614925, n01592084, n02120079, n01744401, n01558993, n02088238, n02007558, n03710721, n02097209, n03954731
	9	n01580077, n02124075, n02110063, n02447366, n02417914, n03240683, n04259630, n09428293, n04033995, n04201297
	10	n03498962, n04332243, n03481172, n07614500, n03891251, n02097474, n03018349, n02978881, n03041632, n09288635
Support 3	1	n06794110, n01582220, n03018349, n01817953, n03584254, n03710721, n01560419, n02840245, n03498962, n03995372
	2	n03742115, n02095314, n03047690, n02097209, n02825657, n02791124, n01776313, n02095570, n02128757, n03649909
	3	n01818515, n02037110, n03891251, n02123394, n03935335, n01616318, n02979186, n04033995, n03376595, n07614500
	4	n01608432, n03424325, n04099969, n07695742, n02007558, n04179913, n03016953, n01744401, n02417914, n03240683
	5	n03929855, n09288635, n02916936, n02447366, n01537544, n09428293, n02124075, n02088238, n03337140, n01797886
	6	n06596364, n02346627, n02963159, n02666196, n02769748, n03271574, n02006656, n02110063, n03680355, n01819313
	7	n03388183, n04548362, n04008634, n01592084, n04392985, n02077923, n01614925, n03481172, n01558993, n04004767
	8	n04596742, n02109047, n04049303, n01950731, n04380533, n01601694, n02342885, n01796340, n01580077, n03602883
	9	n02115641, n04332243, n04259630, n03658185, n04067472, n02423022, n01860187, n02097474, n02120079, n02125311
	10	n03954731, n02910353, n02101006, n04429376, n03041632, n04476259, n04201297, n02978881, n03223299, n09332890

K Gradient-Norm Matching and Cross-Task Discrimination

The experiment in Section 5.3 compares unmasked distillation with the norm-matched masked gradient in Equation 5. It uses the ImageNet-100 training settings in Appendix J.4, including mixed precision, 20 epochs per task, a 2,000-example memory, full-head classification losses, $\alpha = .1$, and $\beta = .5$. Seven class permutations and one optimization seed per permutation are sampled separately from the three-order support study. Both conditions are trained from the same initialization within each pair, with matched replay sampling, reservoir updates, and stored inputs. The two pipelines share the order–seed specification but are analyzed separately. Table 26 lists their exact class sequences and optimization seeds.

Table 19: Exact ImageNet-100 synsets in the fixed stress order across ten tasks. Synsets use WordNet identifiers with five classes per row.

Task	Synsets
Task 1	n03995372, n02840245, n03498962, n03602883, n04332243 n03271574, n04259630, n03584254, n02095570, n02417914
Task 2	n03041632, n02910353, n04179913, n04067472, n03223299 n02115641, n03891251, n02666196, n02110063, n02346627
Task 3	n03481172, n02978881, n03047690, n04476259, n01616318 n01560419, n03649909, n02791124, n01558993, n01860187
Task 4	n04392985, n02769748, n06596364, n04548362, n04380533 n04033995, n02097209, n02109047, n02120079, n02825657
Task 5	n03658185, n04099969, n04429376, n01582220, n03424325 n02097474, n02963159, n04008634, n02088238, n02007558
Task 6	n02979186, n09332890, n01797886, n04004767, n03337140 n07695742, n02037110, n02423022, n01776313, n01817953
Task 7	n02095314, n03680355, n02124075, n01580077, n03388183 n06794110, n02125311, n02101006, n01537544, n09288635
Task 8	n03376595, n09428293, n01592084, n03929855, n01796340 n07614500, n02006656, n02077923, n01818515, n02342885
Task 9	n03016953, n03018349, n02128757, n02123394, n03710721 n01819313, n01744401, n04201297, n03240683, n01950731
Task 10	n03954731, n04049303, n02916936, n01614925, n01608432 n04596742, n01601694, n03935335, n02447366, n03742115

Numerical implementation. Both candidate gradients are measured on the same forward graph using the current mixed-precision loss scale. After removing that scale, the masked direction is normalized in FP32 parameter space and multiplied by the full-gradient norm. The resulting distillation vector is added to the classification gradient with coefficient β before the optimizer step. No gradient of the norm ratio is taken, and no coefficient cap is applied. The procedure stops on nonfinite values, a masked norm at most $\max(10^{-12}, 10^{-6}\|g_U\|_2)$, or a relative vector-norm error above 5%. Every training batch satisfies these checks. They establish numerical matching of the distillation vector, not equality of the full SGD updates along different trajectories.

Statistical units and outcomes. The independent unit is an order–seed pair, giving seven units per pipeline. The primary outcome is FAA. Two-sided exact sign-flip tests enumerate the 2^7 signs of the paired differences under a sign-symmetry null, with Holm correction across the two pipeline tests. Both raw p-values are .015625 and both adjusted values are .03125. Pointwise intervals use $\bar{d} \pm t_{.975,6} s_d/\sqrt{7}$. Table 2 includes all specified secondary outcomes as descriptive estimates, without a second family of confirmatory tests. The FAA differences in Table 22 are negative in every pair. Leave-one-pair-out means range from -42.47 to -40.21 pp for ResNet-18 and from -32.23 to -30.95 pp for ViT-Tiny.

Interpreting cross-task errors. At the final stage, let A be FAA, W accuracy when the true task restricts the candidate classes, and T_{pred} the probability that the all-class argmax belongs to the true task. There is no additional task-prediction head. These quantities obey $1 - A = (1 - T_{\text{pred}}) + (T_{\text{pred}} - A)$, separating wrong-task predictions from within-task errors after correct task assignment. The gap $G = W - A$ measures how often supplying task identity recovers a correct prediction, and $E_{\text{comp}} = G/W$ is its proportion among examples classified correctly with task identity. All task identities are used for evaluation only. Because the sets of correctly classified examples can differ between conditions, differences in E_{comp} do not describe a fixed subset of examples harmed by an intervention.

Table 22: FAA effects of norm-matched masked minus unmasked distillation for each ImageNet-100 order–seed pair, in pp. Pair indices refer to the separately sampled seven-order experiment, not to the earlier three-order support study.

Pair	ResNet-18 FAA	ViT-Tiny FAA
P_1	-41.32	-37.08
P_2	-32.88	-29.40
P_3	-40.32	-30.60
P_4	-40.26	-32.48
P_5	-46.44	-30.60
P_6	-42.84	-32.12
P_7	-43.62	-30.50

Execution and scope. ResNet-18 pairs run on an RTX 4070 Ti SUPER with PyTorch 2.10.0/CUDA 12.8, and ViT-Tiny pairs on an RTX 5070 Ti with PyTorch 2.13.0/CUDA 13.0. Hardware and software are constant within pairs, so between-pipeline effect sizes are not a pure architecture comparison. All 28 runs satisfy the paired numerical checks described above before joint outcome analysis.

L Within-Task and Cross-Task Effects of Support Restriction

Table 23 analyzes the completed unmasked and masked/full trajectories at $\beta = .5$: 60 CIFAR-100 runs from the component study and 36 ImageNet-100 runs from the support comparison. Paired seed differences are averaged within each order and then equally across orders, using five orders on CIFAR-100 and three on ImageNet-100. The analysis decomposes the support effects reported in Section 5 rather than supplying an independent replication.

Table 23: Final-stage decomposition of masked/full minus unmasked effects at $\beta = .5$, in pp. $\Delta A = \Delta W - \Delta G$ by definition, and the ΔA entries round to the corresponding FAA differences in Tables 4 and 10. The gap relates to the competition share in Table 2 through $E_{\mathrm{comp}} = G/W$. The gap G and task-prediction accuracy T_{pred} characterize different events.

Dataset	Pipeline	ΔW	ΔG	ΔA	ΔT_{pred}
CIFAR-100	ResNet-18	+5.58	+17.66	-12.08	-13.32
CIFAR-100	ViT-Tiny	+3.97	+12.83	-8.86	-9.33
ImageNet-100	ResNet-18	+3.46	+18.02	-14.56	-15.35
ImageNet-100	ViT-Tiny	+9.79	+6.93	+2.86	+2.65

Both within-task accuracy and the gap increase in every order within each setting. For ImageNet-100 with ViT-Tiny, the within-task gain is large enough that all-class accuracy and task-prediction accuracy improve despite the larger gap. An increased gap alone therefore does not imply worse overall task prediction. These endpoint identities locate the observed errors without predicting the outcome of a different training setting.

M Local Measurements of Omitted Constraints

We measure two reference trajectories, unmasked and masked/full at $\beta = .5$, in each of the four dataset–pipeline settings. Each trajectory uses one development order and one seed, with three tasks and the same 20-epoch training budget per task. Probes are taken before the final update of each task and before the first update of the second task. Candidate losses and gradients are evaluated on the same current-data and replay forward graph without changing the training gradient.

Separating when a class becomes available. For each stored example, S denotes coordinates seen at storage, N those unseen at storage but seen at the current stage, and F those still unseen. Their squared-error contributions use the common full-head denominator and sum to the unmasked loss. Table 24 reports the third-task final-batch measurement on masked trajectories, where the

group N covers classes learned in up to two subsequent tasks. The same ordering of group magnitudes already appears at the end of the second task, the first probe checkpoint at which N is populated. The large N contribution shows that deleting unseen-at-storage terms also removes constraints on subsequently learned classes. Gradient energy for N is predominantly in the backbone, so this effect is not confined to the head. Each quantity is a local measurement from one exploratory trajectory per setting, so the supported conclusion is the group ordering, which is consistent across the four settings and separated by one to two orders of magnitude, rather than the exact values.

Table 24: Local masked-trajectory measurements before the last update of task three. $\mathcal{L}_S, \mathcal{L}_N, \mathcal{L}_F$ use the full-head denominator. Gradient norms cover all trainable parameters: g_S is the component retained by masking, whereas g_N and g_F are the two components it omits. Head share is $100\|g_{N,\mathrm{head}}\|^2/\|g_N\|^2$. Each row is a single development trajectory, and the group differences span one to two orders of magnitude in all four settings.

Dataset	Pipeline	$\mathcal{L}_S$	$\mathcal{L}_N$	$\mathcal{L}_F$	$\|g_S\|$	$\|g_N\|$	$\|g_F\|$	Head share (%)
CIFAR-100	ResNet-18	0.159	19.854	0.040	1.830	25.229	0.265	26.30
CIFAR-100	ViT-Tiny	0.017	8.409	0.096	0.352	30.074	0.770	8.38
ImageNet-100	ResNet-18	0.382	22.974	0.045	3.821	33.663	0.293	16.87
ImageNet-100	ViT-Tiny	0.013	6.652	0.077	0.348	30.285	0.588	4.54

Relative structure and common shifts. For example i, write $q_i = |S_i|/C$ for the fraction of coordinates seen at storage and $\bar{S}_i = N_i \cup F_i$ for its complement. The batch fraction in Equation 4 is $q = B^{-1}\sum_i q_i$. Let $m_G = |G|^{-1}\sum_{c\in G} e_{ic}$ and $v_G = |G|^{-1}\sum_{c\in G}(e_{ic} - m_G)^2$ denote the sample mean and variance of error coordinates over non-empty subset $G \subseteq \{1, \ldots, C\}$. Suppressing the example index on group means and variances, the exact decomposition is

$$\mathcal{L}_{U,i} = q_i v_S + (1-q_i)v_{\bar{S}} + q_i(1-q_i)(m_S - m_{\bar{S}})^2 + \big(q_i m_S + (1-q_i)m_{\bar{S}}\big)^2. \tag{9}$$

The terms separate within-group variation, a relative group shift, and a common shift. Empty-group terms are zero, and the decomposition is applied per example before batch averaging. A common shift leaves softmax probabilities and argmax unchanged, whereas the other terms can affect relative outputs. Masking retains $q_i(v_S + m_S^2)$, so it does not simply remove one orthogonal component of this decomposition. The recorded losses reconstruct the full loss to within 1.81×10^{-6} absolute error. Component-gradient norms are not additive because their inner products need not vanish.

The shared local patterns appear in settings with both positive and negative accuracy effects of masking. They clarify what the omitted constraints act on but do not identify a sufficient predictor of the sign reversal.

N Combined Recipe Across the Component-Study Orders

Table 25 compares masked/active regression at $\beta = .1$ with unmasked regression at $\beta = .5$ and ER across the five orders of the component study. We pair seeds, average within each order, and report the mean with a pointwise 95% t interval over five orders. These contrasts combine the support, averaging, and weight effects measured separately in that study, complementing the separate-order comparison in Appendix D.1.

Relative to unmasked regression, all five orders in both pipelines show higher acquisition and forgetting but lower final and time-averaged accuracy. Relative to ER, ViT-Tiny improves accuracy and retention with a small acquisition cost, whereas ResNet-18 has mixed order-level FAA and AF effects. Thus the same recipe can be favorable relative to one control and unfavorable relative to another, consistent with the separate-order comparison.

O Scope of the Comparisons

The experiments treat architecture, checkpoint, and recipe as a joint pipeline. ImageNet-100 is not a novel semantic domain relative to the ImageNet pre-training data. Norm matching controls the same-state distillation component, not its alignment with classification gradients, accumulated

Table 25: Combined-recipe effects over five CIFAR-100 orders: masked/active at $\beta = .1$ minus the indicated control, in pp with pointwise 95% t intervals.

Pipeline	**Control**	**Outcome**	**Mean [95% CI]**
ResNet-18	Unmasked .5	FAA	-11.92 [-13.66, -10.18]
ResNet-18	Unmasked .5	$\mathrm{AIA}_{\mathrm{time}}$	-7.81 [-9.23, -6.39]
ResNet-18	Unmasked .5	AF	+31.33 [+26.00, +36.65]
ResNet-18	Unmasked .5	Acquisition	+19.23 [+11.59, +26.86]
ResNet-18	ER	FAA	-0.80 [-2.05, +0.45]
ResNet-18	ER	$\mathrm{AIA}_{\mathrm{time}}$	-0.39 [-0.68, -0.10]
ResNet-18	ER	AF	+1.06 [-0.29, +2.41]
ResNet-18	ER	Acquisition	+0.15 [+0.10, +0.21]
ViT-Tiny	Unmasked .5	FAA	-9.12 [-10.37, -7.88]
ViT-Tiny	Unmasked .5	$\mathrm{AIA}_{\mathrm{time}}$	-5.63 [-6.21, -5.06]
ViT-Tiny	Unmasked .5	AF	+12.23 [+11.08, +13.39]
ViT-Tiny	Unmasked .5	Acquisition	+1.89 [+1.54, +2.23]
ViT-Tiny	ER	FAA	+3.41 [+2.56, +4.25]
ViT-Tiny	ER	$\mathrm{AIA}_{\mathrm{time}}$	+2.85 [+2.64, +3.07]
ViT-Tiny	ER	AF	-4.03 [-4.98, -3.08]
ViT-Tiny	ER	Acquisition	-0.22 [-0.38, -0.07]

momentum, or the full optimization path. With one seed per order, the seven-pair experiment cannot separate order variance from optimization variance. The experiment does not include a masked condition without norm matching, so a change from the earlier ViT result cannot be attributed to matching itself. We compare specified rescaling and weight choices rather than all scalar schedules or methods under equal tuning budgets. The local probes are three-task exploratory measurements, the external-framework comparison changes complete recipes, and the large-buffer reference uses additional memory and a single class order.

Table 26: Exact class sequences for gradient-norm matching. Class IDs are zero-based ImageNet-1K indices. Each order uses its listed optimization seed in both arms and pipelines.

Pair	Task	Class IDs
Pair 1, seed 801672745		
1	1	927, 159, 424, 155, 156, 436, 926, 370, 513, 541
1	2	8, 924, 158, 389, 308, 306, 833, 160, 423, 7
1	3	395, 402, 409, 408, 420, 367, 929, 373, 307, 974
1	4	978, 971, 365, 309, 394, 9, 609, 546, 925, 16
1	5	13, 511, 975, 151, 933, 369, 305, 15, 396, 154
1	6	12, 407, 372, 368, 889, 486, 671, 654, 932, 374
1	7	11, 931, 928, 366, 392, 569, 14, 973, 832, 834
1	8	977, 970, 397, 10, 579, 157, 558, 835, 301, 390
1	9	391, 979, 371, 303, 398, 153, 393, 765, 555, 846
1	10	300, 972, 302, 930, 152, 401, 304, 559, 976, 831
Pair 2, seed 1982377523		
2	1	10, 559, 834, 579, 159, 977, 392, 397, 369, 925
2	2	14, 152, 370, 933, 304, 541, 303, 300, 973, 160
2	3	154, 555, 302, 401, 972, 424, 156, 486, 835, 971
2	4	930, 301, 157, 924, 407, 391, 393, 654, 926, 546
2	5	309, 511, 927, 371, 409, 12, 831, 366, 11, 396
2	6	373, 765, 153, 931, 846, 609, 306, 372, 420, 976
2	7	558, 374, 16, 974, 408, 155, 423, 979, 978, 368
2	8	889, 367, 832, 932, 402, 151, 929, 513, 928, 8
2	9	308, 389, 158, 15, 13, 307, 671, 7, 305, 436
2	10	395, 394, 833, 569, 390, 975, 365, 9, 970, 398
Pair 3, seed 1078471058		
3	1	609, 160, 973, 302, 155, 390, 546, 765, 834, 933
3	2	654, 555, 972, 154, 157, 926, 486, 569, 373, 371
3	3	889, 308, 367, 152, 408, 971, 559, 307, 151, 158
3	4	309, 976, 12, 831, 832, 424, 931, 579, 396, 978

Pair	Task	Class IDs
3	5	671, 365, 407, 558, 13, 7, 11, 974, 10, 306
3	6	370, 928, 16, 369, 924, 374, 930, 975, 835, 368
3	7	402, 392, 153, 395, 423, 156, 397, 159, 301, 394
3	8	393, 372, 409, 401, 436, 846, 303, 929, 977, 541
3	9	391, 398, 305, 15, 513, 14, 389, 511, 979, 304
3	10	925, 932, 927, 420, 366, 300, 8, 9, 970, 833
Pair 4, seed 1213180375		
4	1	367, 152, 511, 13, 160, 423, 407, 424, 8, 308
4	2	846, 9, 372, 833, 927, 395, 373, 14, 157, 301
4	3	370, 154, 16, 831, 513, 11, 158, 558, 978, 159
4	4	155, 391, 929, 928, 979, 368, 408, 304, 569, 931
4	5	401, 7, 974, 420, 975, 371, 834, 765, 609, 12
4	6	409, 305, 555, 932, 15, 369, 151, 396, 366, 486
4	7	402, 579, 153, 398, 973, 389, 156, 307, 924, 302
4	8	972, 10, 832, 546, 309, 300, 393, 436, 374, 889
4	9	394, 654, 971, 977, 541, 976, 397, 930, 392, 671
4	10	365, 306, 835, 303, 933, 926, 925, 559, 970, 390
Pair 5, seed 1232458940		
5	1	301, 978, 300, 409, 303, 395, 569, 424, 370, 926
5	2	609, 10, 546, 671, 8, 765, 372, 832, 929, 930
5	3	160, 833, 835, 511, 392, 970, 368, 155, 14, 15
5	4	156, 307, 579, 389, 394, 305, 924, 157, 304, 7
5	5	367, 558, 928, 927, 979, 308, 13, 976, 158, 309
5	6	11, 973, 513, 391, 555, 393, 397, 486, 152, 541
5	7	154, 423, 932, 371, 408, 931, 159, 9, 153, 654
5	8	972, 365, 398, 834, 151, 401, 396, 831, 302, 559
5	9	975, 306, 16, 436, 974, 889, 933, 373, 366, 925
5	10	12, 977, 420, 402, 971, 369, 407, 390, 846, 374
Pair 6, seed 1261875866		
6	1	15, 156, 930, 374, 671, 368, 513, 300, 16, 609
6	2	970, 929, 973, 835, 366, 13, 396, 308, 370, 157
6	3	511, 7, 159, 160, 971, 834, 389, 10, 977, 436
6	4	569, 979, 541, 928, 393, 424, 976, 555, 11, 978
6	5	833, 9, 394, 831, 309, 889, 974, 304, 390, 423
6	6	559, 12, 765, 846, 924, 402, 654, 371, 486, 397
6	7	302, 306, 303, 154, 401, 155, 372, 408, 926, 151
6	8	14, 420, 305, 933, 395, 558, 365, 546, 153, 152
6	9	158, 931, 832, 8, 932, 392, 407, 398, 579, 975
6	10	927, 972, 409, 391, 373, 925, 301, 367, 369, 307
Pair 7, seed 615771557		
7	1	835, 436, 395, 408, 394, 13, 10, 513, 486, 306
7	2	372, 159, 16, 158, 975, 511, 392, 305, 931, 367
7	3	832, 424, 971, 765, 304, 671, 933, 833, 846, 156
7	4	393, 11, 157, 423, 374, 889, 307, 927, 303, 401
7	5	970, 930, 834, 12, 546, 402, 15, 154, 370, 407
7	6	924, 579, 300, 153, 974, 390, 371, 928, 926, 541
7	7	7, 151, 368, 976, 302, 8, 366, 555, 14, 559
7	8	972, 301, 609, 9, 398, 932, 569, 391, 973, 831
7	9	369, 396, 308, 929, 152, 409, 977, 979, 925, 309
7	10	978, 389, 365, 558, 420, 373, 397, 160, 654, 155