WHAT MAKES LEARNED KNOWLEDGE LAST IN CONTINUAL LEARNING?

Anonymous authors
Paper under double-blind review

ABSTRACT

Catastrophic forgetting limits the ability of neural networks to learn multiple tasks sequentially. Continual learning methods address this problem by reducing the interference introduced by later tasks. However, once a sample is classified correctly, existing methods treat it as securely learned, without considering how easily that prediction may later be reversed. We show that predictions that are equally correct at task completion can differ substantially in their future retention. To capture this difference, we introduce acquisition stability (AS), which measures how often a correct prediction remains correct under mild, label-preserving input changes. In our experiments, low-AS predictions are forgotten up to 7.1$\times$ more than high-AS ones. Since AS can be measured while the current-task data are still available, fragile predictions can be identified and strengthened before future interference arrives. We therefore propose AS-CL, which (i) strengthens low-AS predictions before leaving a task, and (ii) gives greater replay weight to predictions that remain fragile during later tasks. Across three benchmarks, AS-CL consistently reduces forgetting, with the largest gains on low-AS predictions.

1 INTRODUCTION

Continual learning (CL) trains a model on a sequence of tasks while preserving performance on earlier ones (Kirkpatrick et al., 2017; Wang et al., 2024). Forgetting is commonly viewed as interference from later tasks: updates for new tasks change parameters that still support earlier predictions. Accordingly, most CL methods protect past knowledge during later training through replay of stored examples, regularization of important parameters, or distillation of previous outputs or features (Riemer et al., 2019; Buzzega et al., 2020; De Lange et al., 2023; Eskandar et al., 2025). Under this view, once $f_{\theta_t}(x) = y$ at the end of task t, the prediction is treated as learned and the remaining problem is to prevent future updates from reversing it. Yet correctly learned predictions may already differ in how susceptible they are to such updates, and this difference can be observed before any future task arrives. Consider two examples that are both classified correctly at the end of task t. Mild label-preserving changes (such as brightness, a small translation, a rotation) leave the first prediction unchanged, but flip the second. The first prediction is correct in a neighborhood of the input; the second is correct at a point. Later tasks move the decision boundary, and the pointwise prediction is the one that is lost first (Gu et al., 2023; Bian et al., 2024; Le et al., 2026).

Prior work predicts forgetting from how an example was learned. Toneva et al. (2019) measured forgetting events during training, while Goldilocks (Hacohen & Tuytelaars, 2025) showed that learning speed (how early an example becomes correct) can predict later forgetting in CL. Another line of work enforces stability as a uniform training objective (Bhat et al., 2022; Eskandar et al., 2025). These approaches use training history to estimate susceptibility or promote stability broadly during training. What remains less explored is whether the stability of an already-correct prediction, can itself predict its future forgetting and identify where protection is most needed. We measure that property with a post-training probe. For a correctly classified example (x, y), we generate K mild label-preserving views and define $S_t(x)$ as the fraction that the model still classifies as y. We call $S_t(x)$ the *acquisition stability* (AS) of the prediction. Input views and future parameter updates are different perturbations, and we do not assume that one reproduces the other. Instead, we test whether the ranking induced by AS among currently correct predictions can predict forgetting as later tasks update the model. Fig. 1 illustrates this idea. This observation motivates *Acquisition-Stability Continual Learning* (AS-CL). Once a task ends, replay has access only to examples retained in memory,

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.

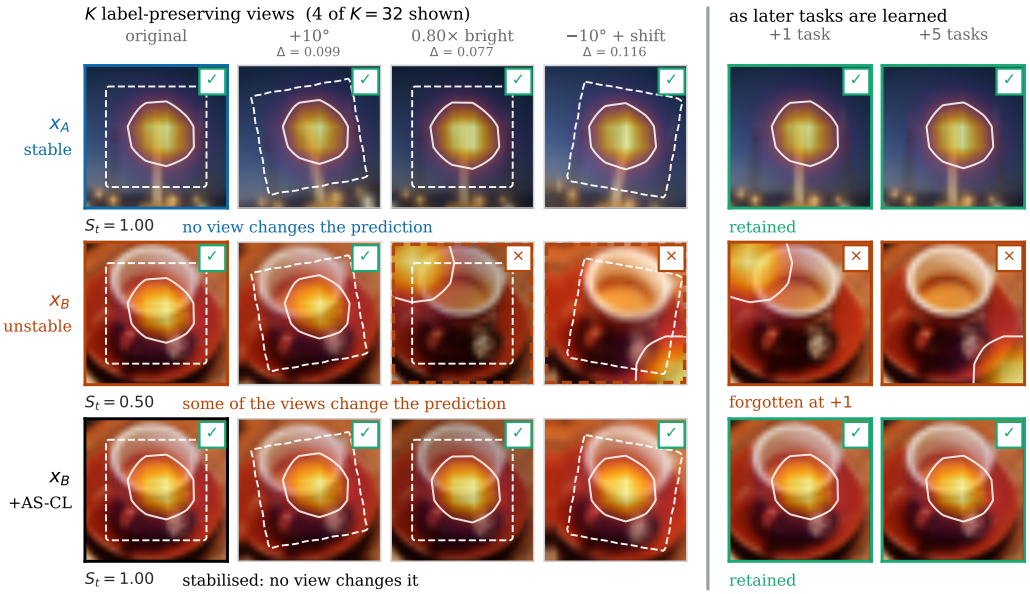


Figure 1: Problem illustration. Left: at the end of task t both predictions are correct. None of the sampled label-preserving views changes the prediction for x_A, whereas half change the prediction for x_B ($S_t = 0.50$). The dashed white square is the same reference crop under each transformation, and ✓/× indicates whether the prediction is correct. Right: after later tasks, x_A is still correct; x_B is already wrong at $+1$. Bottom: the same x_B with using AS-CL, remains correct.

whereas AS identifies vulnerable predictions while the full current-task data are still available. AS-CL acts in two stages. During the current task, Stage 1 assigns greater stabilization weight to correctly classified examples with low AS. The base method then selects its replay examples using its original rule. AS is recomputed for the stored examples, and Stage 2 assigns larger replay weights to those with greater residual susceptibility during later tasks. AS-CL therefore changes neither buffer size nor sample selection. The distinction is that AS is first used as a *diagnostic* of future forgetting and then as a signal for *selective* protection. Existing stability objectives apply their regularization uniformly (Eskandar et al., 2025), while other method use susceptibility to guide memory construction (Hacohen & Tuytelaars, 2025). AS-CL instead identifies vulnerable predictions at acquisition, strengthens them before the current data disappear, and carries their residual susceptibility into future replay. Table 4 shows that applying the same stabilization objective uniformly is less effective than directing it toward low-AS predictions. Our contributions are:

- We study forgetting among predictions that are already correct, and introduce acquisition stability, a current-time probe that predicts which of them will later be forgotten.
- We introduce AS-CL that strengthen low-stability predictions before the task leaves the stream, and protect residual instability during replay, without changing buffer selection.
- Across three image-classification benchmarks, AS-CL reduces forgetting and improves final accuracy, with the largest retention gains on low-stability predictions.

2 Related Work

Protecting knowledge after a task has ended. Continual learning has a mature set of answers to later interference. Regularization limits change to parameters that were important for past tasks (Kirkpatrick et al., 2017; De Lange et al., 2023; Wang et al., 2024), distillation preserves previous outputs or features (Li & Hoiem, 2017; Masip et al., 2024), and replay re-trains on a small buffer of old examples (Riemer et al., 2019; Buzzega et al., 2020; Eskandar et al., 2025). Constraint-based variants such as GEM and A-GEM further require that the current update not increase the loss on stored examples (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019). Recent replay methods refine what is preserved rather than *when*: BFP distils in a learned feature space (Gu et al., 2023), SARL adds semantic sparse coding (Sarfraz et al., 2025), and IDER (ER + ID) (Liu et al., 2026) enforces idempotent predictions. All of these methods act once a task is already old. They do not distinguish which of those correct predictions are already brittle.

Example-level susceptibility to forgetting. Another line of work shows that forgetting is not uniform across examples. Toneva et al. (2019) documented different forgetting rates during training, and Hacohen & Tuytelaars (2025) brought this into continual learning, showing that learning speed predicts future forgetting with a last-in, first-out pattern: examples learned later are lost first. Related rules populate memory from C-scores, learning-consistency, or forgetting counts (Krawczyk & Gepperth, 2024; Benkő, 2024), or from instantaneous interference and uncertainty (Aljundi et al., 2019; Serra et al., 2025). These methods score examples in order to decide what to store. The score is training history or current uncertainty, not the stability of a prediction that is already correct.

Stability as training objectives. Stability has also been studied in CL from several different perspectives. The stability gap describes a transient loss of old-task performance that occurs after learning a new task and may partially recover before the usual end-of-task evaluation. DER++ (Buzzega et al., 2020) and SER (Zhuo et al., 2023) match logits across time, including a forward term on the current stream. Bhat et al. (2022) add view-consistency to experience replay, perturbing the input. Closest to our work, STAR (Eskandar et al., 2025) regularizes the output under worst-case parameter perturbations on correctly classified buffer samples, and C-Flat seeks flatter minima that tolerate future updates (Bian et al., 2024). These methods apply a uniform pressure, after examples are in memory (STAR) or to every current example (Bhat, SER). They treat stability as a property to impose, not as a measurement that separates one already-correct prediction from another. AS-CL sits in the gap between these three lines. It score already-correct predictions at the end of the task by consistency under mild input views (acquisition stability), rather than summarizing training history (Toneva et al., 2019; Hacohen & Tuytelaars, 2025) or perturbing weights (Eskandar et al., 2025; Bian et al., 2024). We use that score first to repair brittle predictions before the task leaves, then to reweight replay for residual fragility. Buffer selection is left unchanged, unlike speed-based learning, which acts through buffer composition (Hacohen & Tuytelaars, 2025).

3 METHOD

A prediction that is correct when a task ends may still be be fragile and later forgotten. We separate this into two factors: the interference introduced by future tasks (which is unknown at time t), and how easily the prediction itself can be reversed (which can be estimated at time t). Our goal is to identify which acquired predictions are already fragile before future interference arrives. In this section, we first formulate forgetting at the prediction level (Secs. 3.1-3.2), then define a current-time probe to estimate this fragility (Secs. 3.3-3.4), and use the probe to build AS-CL (Sec. 3.5).

3.1 SETUP: FROM TASK-LEVEL TO PREDICTION-LEVEL FORGETTING

We consider a sequence of tasks $\{\tau_1, \dots, \tau_T\}$, with training data $\mathcal{D}_t = \{(x_i, y_i)\}_{i=1}^{N_t}$ and testing data $\mathcal{D}_t^{\text{test}}$. Let θ_t denote the model parameters after learning τ_t, then the accuracy on task τ_t is defined

$$A(t,s) = \frac{1}{|\mathcal{D}_t^{\text{test}}|} \sum_{(x,y)\in\mathcal{D}_t^{\text{test}}} \mathbf{1}[f_{\theta_s}(x) = y]\,, \tag{1}$$

and task-level forgetting is $F_{\text{acc}}(t) = A(t,t) - A(t,T)$. This average does not identify which predictions are lost. We therefore define the predictions acquired at the end of τ_t as

$$\mathcal{A}_t = \{(x,y) \in \mathcal{D}_t^{\text{test}} : f_{\theta_t}(x) = y\} \tag{2}$$

An acquired prediction is forgotten at a later stage $s > t$ if $Z_{t,s}(x) = 1$, where

$$Z_{t,s}(x) = \mathbf{1}[f_{\theta_s}(x) \neq y]\,, \qquad (x,y) \in \mathcal{A}_t. \tag{3}$$

From a decision boundary perspective, a prediction is correct if its margin, $m_\theta(x) = z_{\theta,y}(x) - \max_{c\neq y} z_{\theta,c}(x)$, is positive (Jiang et al., 2019). Every prediction in $\mathcal{A}_t$ satisfies $m_{\theta_t}(x) > 0$, but this binary correctness does not indicate how easily that prediction can later be reversed. We therefore can fragility already be detected when the prediction is acquired?

3.2 FORGETTING AS A PERTURBATION OF ACQUIRED PREDICTIONS

Learning subsequent tasks updates the parameters from θ_t to $\theta_s = \theta_t + \Delta_{t\to s}$, so for $(x,y) \in \mathcal{A}_t$ future forgetting can be written as $Z_{t,s}(x) = \mathbf{1}[\, m_{\theta_t + \Delta_{t\to s}}(x) \leq 0\,]$. This splits forgetting into two ingredients (Appendix G.1): i) the future update $\Delta_{t\to s}$, determined by tasks not yet seen, and (ii) the fragility of the acquired prediction, i.e. how easily it can be reversed by a subsequent change to the model. Existing replay and regularization methods act on (i) by constraining $\Delta_{t\to s}$. Other methods

address (ii) through parameter perturbations (Eskandar et al., 2025; Bhat et al., 2022; Hacohen & Tuytelaars, 2025). Our focus is complementary: we estimate fragility for each already-correct prediction before future interference occurs. A first-order expansion of the margin yields

$$m_{\theta_t+\Delta}(x) \approx m_{\theta_t}(x) + \langle \nabla_\theta m_{\theta_t}(x), \Delta \rangle, \tag{4}$$

$$m_{\theta_t}(a(x)) \approx m_{\theta_t}(x) + \langle \nabla_x m_{\theta_t}(x), a(x) - x \rangle, \tag{5}$$

a prediction survives if its margin is large relative to its sensitivity (Novak et al., 2018). These sensitivities are coupled (Appendix G.2): at the first layer $h{=}W_1 x$, we have $\nabla_{W_1} m{=}g\, x^\top$ and $\nabla_x m{=}W_1^\top g$ (where $g = \partial m/\partial h$). Thus, a point that is sensitive to mild input views is also sensitive to early-layer parameter moves. Therefore, we use mild label-preserving input views at time t as a surrogate probe to estimate vulnerability to future parameter updates $\Delta_{t\to s}$ (Sec. 3.4).

3.3 Acquisition Stability

Let $\mathcal{T}$ be a family of mild label-preserving transformations (small changes in brightness, rotation, or translation; ranges provided in Sec. 4). For an example (x, y), we sample $a_1, \ldots, a_K \sim \mathcal{T}$ and set

$$S_t(x) = \frac{1}{K} \sum_{k=1}^{K} \mathbf{1}[f_{\theta_t}(a_k(x)) = y], \qquad Q_t(x) = 1 - S_t(x), \tag{6}$$

where $S_t(x) \in [0, 1]$ is the acquisition stability (AS) and $Q_t(x)$ the fragility of the prediction. An acquired prediction is secure when $S_t(x){=}1$, and fragile as S_t decreases. $S_t(x)$ is a Monte Carlo estimate under the transformation distribution $\mathcal{T}$ and requires only K forward passes ($K{=}32$; Appendix G.3). It needs no future data and no training history (unlike learning speed (Hacohen & Tuytelaars, 2025)), and it is a probe, not a training objective (Eskandar et al., 2025). Although Eq. 6 is defined for any labelled example, our forgetting analysis considers only $(x, y) \in \mathcal{A}_t$.

3.4 Does Acquisition Stability Predict Future Forgetting?

If acquisition stability captures prediction fragility, then among predictions that are correct when τ_t ends, those with lower AS should be forgotten more frequently. Formally

$$\Pr[Z_{t,s}(x) = 1 \mid (x, y) \in \mathcal{A}_t,\ S_t(x) = \sigma] \quad \text{decreases as } \sigma \text{ increases.} \tag{7}$$

Stability groups. To validate this relationship, we tracked forgetting on CIFAR-100 (10 tasks, $M{=}2000$) using standard ER and DER++ (Sec. 4.1). After each task, we recorded $S_t(x)$ for every acquired prediction without using it for training, and then tracked $Z_{t,s}$ across the next five tasks. Because many predictions share the same AS value, we divide them into five stability groups based on the number of failed views (from G_5: zero failures, to G_1: eight or more failures; Appendix B). As shown in Fig. 3, forgetting scales inversely with stability: the most fragile group (G_1) loses $5.5\times$ (ER) and $7.1\times$ (DER++) more predictions than the most secure group (G_5). Table 2 shows that AS is not a restatement of confidence or learning speed (Appendix G.4); adding AS to confidence and learning speed raises AUROC from 0.78 to 0.86 for DER++. The large difference in forgetting rates between G_1 and G_5 proves that applying stabilization uniformly across all data is inefficient. This motivates our proposed model, which isolates and repairs the most vulnerable predictions.

3.5 AS-CL: Acquisition-Stability Continual Learning

The analysis above identifies fragile predictions before future interference occurs. AS-CL uses this information at two points (Fig. 2). **Stage 1** strengthens fragile predictions while the current-task is available; **Stage 2** records the fragility that remains after stabilization and gives those predictions greater protection during future replay.

Stage 1: Intra-task stabilization. Each task is trained for E total epochs, split into E_{base} standard epochs and $E - E_{\text{base}}$ stabilization epochs (overhead analyzed in Sec. 4.5). Base epochs use $\mathcal{L}_{\text{CL}}$, produce intermediate parameters θ_t^{base}. We then isolate the successfully acquired training set $\mathcal{C}_t = \{(x, y) \in \mathcal{D}_t : f_{\theta_t^{\text{base}}}(x) = y\}$, and compute a frozen fragility score $\widetilde{Q}_t(x)$ via Eq. 6 at θ_t^{base}. Freezing $\widetilde{Q}_t(x)$ ensures the algorithm targets a fixed diagnostic metric, preventing the stabilization weights from fluctuating as the model updates. During the stabilization epochs, we isolate correctly acquired examples within each mini-batch ($\mathcal{B}^+ = \mathcal{B} \cap \mathcal{C}_t$). Unacquired examples continue training normally under $\mathcal{L}_{\text{CL}}$, while for acquired examples ($\mathcal{B}^+$), we apply a weighted stability loss

Table 1: Main notations.

Symbol / term	Meaning
$\tau_t,\ \mathcal{D}_t,\ \mathcal{D}_t^{\text{test}}$	Task t, its training set and held-out test set.
$\theta_t,\ f_{\theta_t}$	Parameters and prediction function after completing task τ_t.
$\mathcal{A}_t$	Acquired prediction: the examples of $\mathcal{D}_t^{\text{test}}$ that f_{θ_t} classifies correctly.
$\mathcal{T},\ a \in \mathcal{T}$	Family of mild class-preserving transformations (with a single instance a).
$K,\ K'$	Views drawn to estimate stability (K=32), and fresh views used in each stabilization step (K'=4).
$S_t(x)$	Acquisition stability: fraction of K views on which the prediction of x stays correct at θ_t (Eq. 6).
$Q_t(x) = 1 - S_t(x)$	Acquisition fragility; Q_t=0 means every view is classified correctly.
θ_t^{base}	Parameters after the E_{base} base epochs of τ_t, before stabilization.
$\widetilde{S}_t(x),\ \widetilde{Q}_t(x)$	Preliminary stability and fragility, measured once at θ_t^{base}.
$v(x)$	Stage-1 weight, $v(x) \propto \epsilon + \widetilde{Q}_t(x)$, normalised to mean one (Eq. 8).
$\mathcal{L}_{\text{CL}},\ \mathcal{L}_{\text{stab}},\ \mathcal{L}_{\text{acq}}$	CL method objective, the stabilization term (Eq. 8), and their sum (Eq. 9).
q_x	Residual fragility of a buffer example: $1 - S_t(x)$ recomputed at the final θ_t with fresh views, and never updated.
w_i	Stage-2 weight for the i-th buffer example, $w_i \propto 1 + \lambda_q q_{x_i}$ (Eq. 10).
$\ell_{\text{replay}},\ \mathcal{L}_{\text{replay}}^{\text{AS}}$	Replay loss of the base learner, and the reweighted version that replaces it.
$Z_{t,s}(x)$	Forgetting event: $x \in \mathcal{A}_t$ is misclassified after learning through a later task τ_s, $s > t$ (Eq. 3).
$A(t,s),\ F_{\text{acc}}(t)$	Accuracy on τ_t measured after τ_s, and task-level forgetting (Eq. 1).
$R_{\text{low}}^{(+5)}$	Retention of the fragile band: predictions in subset G_1 still correct five tasks later.

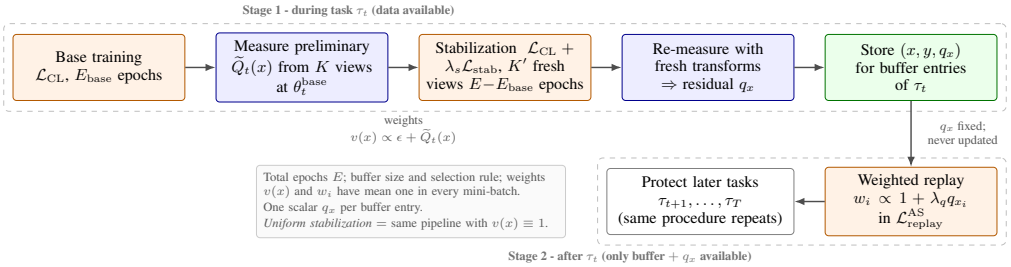


Figure 2: Overview of AS-CL. Stage 1 runs while the current task is still available, identifies fragile predictions (blue) and stabilizes them (orange) within the same epoch. The fragile that remains is stored as one scalar q_x per buffer (green), and Stage 2, uses it to reweight replay in all later tasks.

$$\mathcal{L}_{\text{stab}} = \frac{1}{|\mathcal{B}^+|} \sum_{(x,y)\in\mathcal{B}^+} v(x)\Big[\frac{1}{K'}\sum_{k=1}^{K'} \ell\big(f_\theta(a_k(x)), y\big)\Big], \qquad v(x) = \frac{\epsilon + \widetilde{Q}_t(x)}{\frac{1}{|\mathcal{B}^+|}\sum_{x'\in\mathcal{B}^+}(\epsilon + \widetilde{Q}_t(x'))}. \tag{8}$$

where $a_1, \ldots, a_{K'}$ are fresh transformations from $\mathcal{T}$, ℓ is the classification loss, and $\epsilon > 0$ maintains a baseline weight for already-secure predictions. The Stage 1 objective become

$$\mathcal{L}_{\text{acq}} = \mathcal{L}_{\text{CL}} + \lambda_s\, \mathcal{L}_{\text{stab}}. \tag{9}$$

Because the weight $v(x)$ averages to 1 across the batch, λ_s sets the stabilization budget, while v redistributes that budget toward fragile predictions (an ablation of uniform weighting $v\equiv1$ is in Table 4b, and Appendix G.5). At the end of Stage 1 (yielding final parameters θ_t), we compute the residual fragility q_x=$1 - S_t(x)$ for all examples selected for the buffer, storing the tuple (x, y, q_x).

Stage 2: Inter-task replay protection. During subsequent tasks, residual fragility determines the strength of replay protection. For a replay batch $\mathcal{B}_{\mathcal{M}}$, we apply the weights

$$w_i = \frac{1 + \lambda_q q_{x_i}}{\frac{1}{|\mathcal{B}_{\mathcal{M}}|}\sum_{j\in\mathcal{B}_{\mathcal{M}}}\big(1 + \lambda_q q_{x_j}\big)}, \qquad \mathcal{L}_{\text{replay}}^{\text{AS}} = \frac{1}{|\mathcal{B}_{\mathcal{M}}|}\sum_{i\in\mathcal{B}_{\mathcal{M}}} w_i\, \ell_{\text{replay}}(x_i, y_i; \theta), \tag{10}$$

Similar to Stage 1, the weights w_i have a mean of 1, so AS-CL does not increase the replay weight, rather, λ_q redirects the existing replay budget toward examples that remained fragile after Stage 1. Setting $\lambda_q = 0$ gives uniform replay, while $\lambda_s = \lambda_q = 0$ recovers the base CL method (Appendix G, Proposition 8). Table 1 lists the notations (and our algorithm is provided in the Appendix A).

4 Experiments

Our experiments address three questions. Q1: among predictions that are correct at the end of a task, does our model identifies those that later forget? Q2: does AS-CL improves retention and Class-IL

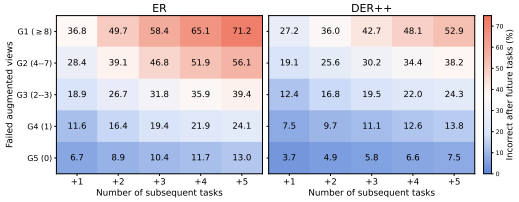


Figure 3: Predictions that are correct at the end of τ_t and incorrect after k further tasks (%), by AS group (G_1: $\geq$8 of K=32 views wrong; most fragile). Split CIFAR-100, 10 tasks, M=2000.

Method	Individual predictor			Combined predictor	
	1-Conf.	1-Speed	Q_t	Conf.+Speed	Conf.+Speed+Q_t
ER	0.63	0.72	0.81	0.75	**0.83**
DER++	0.68	0.75	0.84	0.78	**0.86**

Table 2: AUROC for $Z_{t,t+5} = 1$ on $\mathcal{A}_t$. 1−Conf. is $1 - p_{\theta_t}(y \mid x)$; Speed is the first epoch the test example is correct during τ_t (higher = learned later). Combined (logistic regression) trained on two class orders, tested on the remaining three.

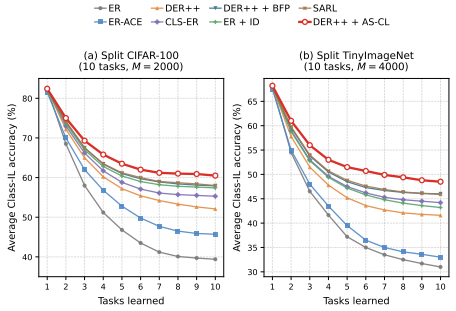


Figure 4: Average Class-IL accuracy.

Method	CIFAR-100 (10 tasks, $M = 2000$)			TinyImageNet (10 tasks, $M = 4000$)		
	FAA ↑	BWT ↑	LA ↑	FAA ↑	BWT ↑	LA ↑
ER	39.4 ± 0.62	-36.55	72.3	30.2 ± 0.58	-40.28	66.5
ER-ACE	45.7 ± 0.52	-28.01	70.9	32.4 ± 0.51	-37.12	65.8
DER++	52.1 ± 0.37	-24.32	74.0	41.6 ± 0.47	-29.58	68.2
CLS-ER	55.3 ± 0.49	-20.35	73.6	44.2 ± 0.50	-25.74	67.4
DER++BFP	57.9 ± 0.43	-18.11	74.2	46.0 ± 0.42	-24.89	68.4
ER + ID	57.4 ± 0.45	-17.78	73.4	43.3 ± 0.48	-26.70	67.3
SARL	58.2 ± 0.31	-18.09	**74.5**	45.9 ± 0.45	-25.12	**68.5**
DER++ + Goldilocks	57.2 ± 0.34	-18.90	74.2	47.1 ± 0.39	-23.55	68.3
DER++ + STAR	56.9 ± 0.38	-19.35	74.3	45.8 ± 0.47	-24.41	68.2
ER + ASCL	54.7 ± 0.47	-20.32	73.0	44.0 ± 0.49	-25.74	67.2
DER++ + ASCL	**60.3 ± 0.35**	**-15.56**	74.3	**48.6 ± 0.42**	**-21.95**	68.4

Table 3: Final Class-IL results. FAA: final avg. accuracy; BWT: backward transfer; LA: learning accuracy.

accuracy? Q3: are the gains concentrated on low-AS predictions, and do both stages contribute? We compare against ER, ER-ACE (Caccia et al., 2022), CLS-ER (Arani et al., 2022), DER++, DER++ with BFP (Gu et al., 2023), STAR (Eskandar et al., 2025), SARL (Sarfraz et al., 2025), Goldilocks (Hacohen & Tuytelaars, 2025) and IDER (ER + ID) (Liu et al., 2026). Unless stated otherwise, AS-CL is added to DER++ (**DER++ + AS-CL**).

Datasets. We use Split CIFAR-100 (10 tasks of 10 classes), Split TinyImageNet (10×20) and Split ImageNet-100 (10×10), with disjoint classes. Buffer sizes are $M \in \{500, 2000\}$ on CIFAR-100, M=4000 on TinyImageNet, and M=5000 on ImageNet-100. The main tables are Class-IL, mean ± std over five class orders (seeds 0-4; Appendix F): five permutations of the same 10-task split. Task-IL result is given in Appendix D, with $\mathcal{A}_t$ and S_t computed using only that task's classes.

Training. All methods use a ResNet-18 (lr 0.1 and weight decay 3×10^{-4}), with 32 current-task and 32 replay examples per mini-batch. For CIFAR-100 the stem is 3×3 with stride one and no max-pooling, and training runs 50 epochs per task with the learning rate multiplied by 0.1 at epochs 35 and 45; TinyImageNet and ImageNet-100 use 100 epochs with decays at 35, 60 and 75. In AS-CL the last 20% of each task's epochs are stabilization epochs ($E - E_{\text{base}} = 10$ and 20), in which $\mathcal{L}_{\text{stab}}$ is applied with $K' = 4$ views per example. We set $\lambda_s = 0.5$, $\lambda_q = 1.0$ and $\epsilon = 0.05$.

Acquisition stability. S_t uses K=32 views: brightness/contrast in $[0.8, 1.2]$, rotations in $[-10°, 10°]$, translations up to 6.25% of the image side. $\mathcal{L}_{\text{stab}}$ and the uniform-stabilization ablation use the same $\mathcal{T}$. For Q1, S_t is measured on test examples in $\mathcal{A}_t$, and for Stage 2, q_x is computed at θ_t for every example that replay stores (not only $\mathcal{C}_t$).

Metrics. With $A(t, s)$ in Eq.1, we report final average accuracy FAA=$\frac{1}{T}\sum_{t=1}^{T} A(t, T)$, learning accuracy LA=$\frac{1}{T}\sum_{t=1}^{T} A(t, t)$, and backward transfer BWT=$\frac{1}{T-1}\sum_{t=1}^{T-1}\big[A(t, T) - A(t, t)\big]$ (less negative is better). Since FAA = LA + $\frac{T-1}{T}$BWT, every FAA gain splits into a learning and a retention part, and LA shows whether stabilization costs current-task accuracy. Prediction-level metrics use a horizon $+k$ within the same 10-task sequence: after τ_t we score the same example after k later tasks. To use the same predictions for all horizons, we restrict this analysis to $t \leq T - 5$ (tasks 1-5 when T=10) and then average over those t and over the five class orders. The forgetting rate of group G_g at $+k$ is the mean of $Z_{t,t+k}$ (Eq. 3) on $\mathcal{A}_t \cap G_g$; $R_{\text{low}}^{(+5)}$ is the share of G_1 still correct at +5; and FRR_5 is the ratio of the G_1 and G_5 rates at +5; while $\text{RMRH}_5 = \mathbb{E}[\min(h, 5)]$ is the number of later tasks a G_g prediction remains correct.

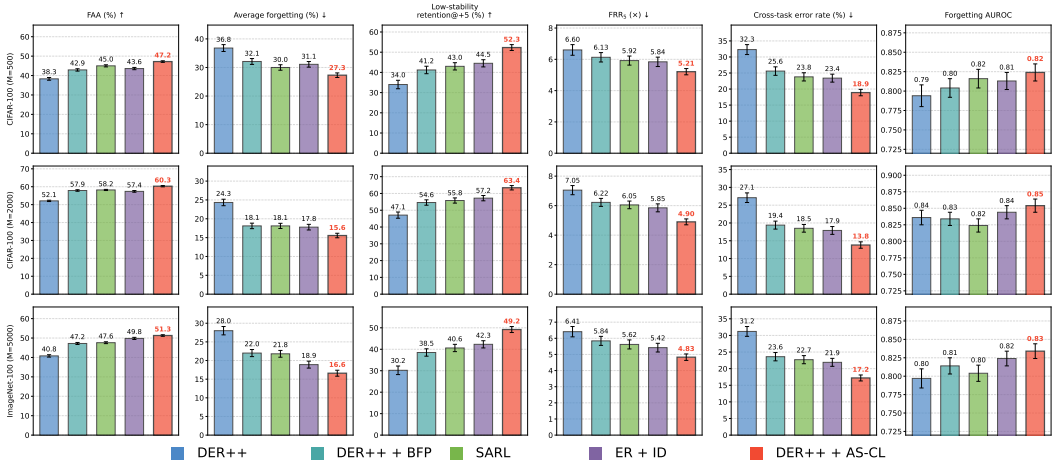


Figure 5: Results across memroy and dataset scale. Columns: average accuracy (FAA), average forgetting (−BWT), retention of the low-stability group $R_{\text{low}}^{(+5)}$, the forgetting-risk ratio FRR_5, cross-task error, and AUROC of each method's own Q_t. Columns 3-4 use the capped set $\mathcal{A}_t^{\text{cap}}$ with groups defined by DER++; the other columns use each method's own predictions.

4.1 Acquisition Stability Predicts Future Forgetting

We test whether AS identifies which *already-correct* predictions are later forgotten (Q1). After learning τ_t we compute $S_t(x)$ on every test example in $\mathcal{A}_t$, assign it to G_1-G_5 (group sizes in Appendix B), and continue training without using S_t. Fig. 3 is the fraction of each group that is incorrect after k further tasks ($Z_{t,t+k}$=1), averaged over $t \leq T-5$ so every horizon uses the same tasks. Under ER, 71.2% of G_1 is lost at +5 tasks against 13.0% of G_5 (5.5×); DER++ lowers every rate but preserves the ordering (52.9% vs. 7.5%; 7.1×). The gap is already present after one task (36.8% vs. 6.7%), so fragility shows up as soon as the next task arrives. Table 2 asks whether Q_t is only low confidence or slow learning. As a single predictor of forgetting, Q_t reaches 0.81 (ER) and 0.84 (DER++), above speed (0.72, 0.75) and confidence (0.63, 0.68); adding Q_t to a logistic predictor improves from 0.75 to 0.83. AS is therefore not a restatement of these standard signals.

4.2 Class-Incremental Learning Performance

Sec. 4.1 showed that AS ranks already-correct predictions that later become incorrect. We now test whether using that signal improves Class-IL accuracy (Q2). Fig. 4 plots average Class-IL accuracy after each task, and Table 3 reports FAA, BWT, and LA. AS-CL has the highest FAA and the least negative BWT. Over DER++, FAA rises from 52.1 to 60.3 on CIFAR-100 and from 41.6 to 48.6 on TinyImageNet, with BWT from −24.32 to −15.56 and from −29.58 to −21.95. Since $\text{FAA} = \text{LA} + \frac{T-1}{T}\text{BWT}$, 7.9 of the 8.2-point gain (on CIFAR-100) are retention (0.9×8.76), and only 0.3 are current-task learning (LA: 74.0 → 74.3). Stabilization therefore does not buy FAA by inflating $A(t,t)$. Among other methods, Goldilocks (sampling by learning speed) and STAR (uniform parameter-stability) improve DER++ by +5.1 and +4.8 FAA on CIFAR-100. AS-CL still leads: +2.1 over SARL on CIFAR-100 and +1.5 over Goldilocks on TinyImageNet.

4.3 Across Memory Budgets and Dataset Scale

In this section we analyse whether the gains persist with a smaller buffer and a larger input, and whether it is concentrated on the predictions AS marks as fragile (Q3). Fig. 5 reports three settings, one per row: CIFAR-100 with $M = 500$ and $M = 2000$, and ImageNet-100 with $M = 5000$. The same five class orders as Table 3 are used. We compare with DER++, DER++ + BFP, SARL, and ER + ID. Columns 1-2 are FAA and average forgetting (−BWT). AS-CL is best on both in every row, and its gain over the best baseline is preserved when memory drops from 2000 to 500 (+2.1 and +2.2 over SARL; +1.5 over ER + ID on ImageNet-100). Columns 3-4 compare all methods on the same predictions. Let $\mathcal{A}_t^{\text{cap}}$ be the test examples that every method classifies correctly at the end of τ_t; we then compute acquisition stability on these examples using DER++ as the reference learner and form G_1-G_5 (via the definition in Sec. 4.1). $R_{\text{low}}^{(+5)}$ and FRR_5 are then measured for each method on that same slice, $t \leq T-5$. At M=2000, AS-CL raises $R_{\text{low}}^{(+5)}$ from 47.1% (DER++) to 63.4%,

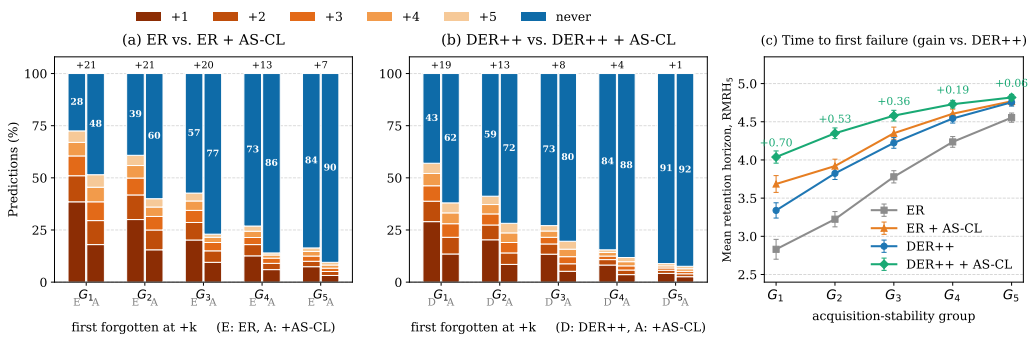


Figure 6: First-failure horizon on Split CIFAR-100 (M=2000). (a,b) Distribution of the first-failure horizon T per stability group, with and without AS-CL; numbers are the percentage never forgotten within five tasks. (c) Mean retention horizon RMRH_5; green deltas are vs. DER++.

Table 4: Ablation on Split CIFAR-100 (M=2000). FAA = LA + $\frac{T-1}{T}$BWT with T=10.

	overall			fragile band
	FAA ↑	LA ↑	BWT ↑	$R^{(+5)}_{\text{low}}$ ↑
DER++ (λ_s=λ_q=0)	52.1	74.0	−24.3	47.1
(a) Which stage does the work?				
+ stabilized acquisition (Stage 1)	58.2	73.9	-17.4	58.7
+ fragility-weighted replay (Stage 2)	56.3	74.2	-19.8	55.3
AS-CL (both)	**60.3**	74.3	-15.6	**63.4**
(b) Is it the signal, or the compute?				
uniform stabilization ($v \equiv 1$)	55.2	73.5	-20.3	51.4
confidence weighting ($1 - p_\theta$)	57.3	73.9	-18.5	53.7
margin weighting ($-m_\theta$)	57.5	73.8	-18.1	54.2
loss weighting	55.8	73.5	-19.7	51.8
AS-CL weights, permuted within batch	55.0	73.7	-20.8	50.6
inverted AS-CL ($v \propto \epsilon + \tilde{S}_t$)	52.4	73.6	-23.5	46.7
(c) Where should stability act?				
buffer selection by stability, loss untouched	54.3	73.9	-21.8	50.2
replay sampling $\propto q_x$, loss untouched	55.7	74.1	-20.4	51.9
AS-CL + stability-based selection	60.1	74.2	-15.7	62.7

and lowers FRR_5 from 7.05 to 4.90. Indeed, the G_5 error is 7.3-7.5% for every method in that row, so the FRR_5 spread is almost from G_1: stable predictions are not sacrificed. The same G_1-specific pattern appears at M=500 and on ImageNet-100 (G_5 errors ≈ 9.5% and ≈ 10.7%). Column 5 is the cross-task error at θ_T, which is the percentage of earlier-task examples ($\tau_1, \ldots, \tau_{T-1}$) misclassified into a later-task class. AS-CL lowers CTE in every setting (e.g., 27.1% → 13.8% at M=2000), indicating that fewer old predictions drift onto new heads. The last column is a different object: AUROC of each method's own Q_t for $Z_{t,t+5}$ on that method's own $\mathcal{A}_t$. Values lie in $[0.79, 0.85]$, and AS-CL is as high as every baseline, which means the probe remains informative after Stage 1 (which is what makes the stored q_x usable in Stage 2).

4.4 When Acquired Predictions Are Forgotten

Sec. 4.1 showed which already-correct predictions are later forgotten; we now ask *when* this happens. For each acquired prediction in $\mathcal{A}_t$, we define $h(x)$ as the first horizon $k \in \{1, \ldots, 5\}$ for which $Z_{t,t+k}(x) = 1$. If the prediction remains correct throughout all five future tasks, we set $h(x) = \infty$. A prediction that fails and later recovers is counted at its first failure. Therefore, the never-forgotten rate is at most the prediction that still correct at +5. We summarize the first-failure time using $\mathrm{RMRH}_5 = \mathbb{E}[\min(h, 5)]$, where predictions that are never forgotten within the observed window are censored at 5. A larger RMRH_5 therefore indicates later forgetting. As in Sec. 4.3, groups are defined using DER++ on the predictions acquired by all four methods (Fig. 6; full distributions in Appendix C), so rates differ slightly from Fig. 3, which uses each learner's own acquired

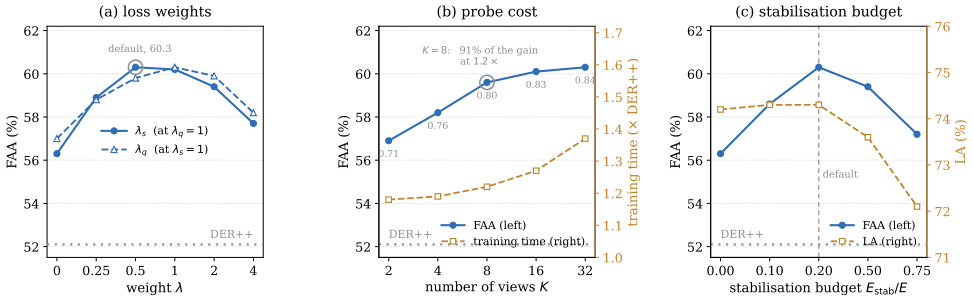


Figure 7: Hyperparameter verification (DER++ + AS-CL, CIFAR-100, M=2000). Blue is FAA, grey marks the DER++ baseline and the default setting. (a) FAA against the two loss weights. (b) FAA and training time against the number of views K, with AUROC(Q_t) in grey values. (c) FAA and LA against the stabilization budget E_{stab}/E; the dashed line is the value used throughout.

set. Under ER, 38.5% of G_1 predictions first fails at +1, compared with 7.4% of G_5. only 27.5% of G_1 remain correct throughout all five tasks, compared with 83.6% of G_5. AS-CL delays failure: for G_1, the never-forgotten rate increases from 27.5% to 48.5% with ER and from 43.0% to 62.0% with DER++. Correspondingly, RMRH_5 increases by 0.86 with ER and 0.70 with DER++. The gain becomes smaller toward G_5 (+0.70, +0.53, +0.36, +0.19, +0.06 for DER++), where the baselines are already stable. Finally, ER+AS-CL reaches $\text{RMRH}_5 = 3.69$ on G_1, above plain DER++ (3.34), showing that weighting by acquisition stability recovers fragile predictions even on a simpler replay.

4.5 Ablation

Table 4 isolates AS-CL's performance gains on Split CIFAR-100 (M=2000). To ensure a fair comparison, all variants in block (b) use the same K probe, the same K' views in $\mathcal{L}_{\text{stab}}$, and the same E_{stab}. Stage 1 alone adds +6.1 FAA, Stage 2 alone +4.2, and together adds +8.2 and boosts the retention of fragile predictions ($R^{(+5)}_{\text{low}}$) from 47.1 to 63.4. Uniform stabilization ($v \equiv 1$) acts as a regularizer, adding only +3.1 FAA and +4.3 $R^{(+5)}_{\text{low}}$. Permuting the AS-CL weights inside the batch returns 55.0, near uniform; inverting them ($v \propto \epsilon + \widetilde{S}_t$) erases the gain and degrades fragile retention below the baseline (46.7 < 47.1). Confidence and margin, recover about two thirds of the FAA gain (57.3, 57.5) but only $\approx 40\%$ of the R_{low} gain (53.7, 54.2 vs. 63.4), in line with the weaker AUROC in Table 2. Selecting the buffer by AS, or sampling replay $\propto q_x$, without changing the loss, gives +2.2 and +3.6 FAA (54.3, 55.7); worse than Goldilocks (57.2). Stacking AS selection on AS-CL does nothing (60.1 vs. 60.3), which is why AS-CL leaves buffer size untouched.

Hyperparameters. FAA is flat for $\lambda_s, \lambda_q \in [0.5, 1]$ and both curves stay within 0.5 FAA of their max (Fig. 7a), so one setting is used for every dataset. The λ_s=0 endpoint reproduces the Stage-2-only row (56.3) and the λ_q=0 endpoint (57.0) is close to Stage-1-only. As shown in Fig. 7b, the probe saturates before K=32, and eight views already recover 91% of the gain at 1.2× the training time of DER++, against 1.38× for the full setting. AUROC(Q_t) rises with K and reaches 0.80 at K=8, so the gain tracks the accuracy of the fragility estimate. The stabilization budget has an optimum at 0.20 (Fig. 7c), and LA is flat up to that point and falls only afterwards, so the retention gain is not taken out of plasticity. More ablation in Appendix E.2.

5 Conclusion

Not all predictions acquired during a continual learning are equally secure. We demonstrated that the predictions most vulnerable to catastrophic forgetting can be identified before the next task begins. We formalized this through Acquisition Stability (AS), which quantifies the fragility of learned predictions using label-preserving input perturbations, and showed that it anticipates future forgetting more accurately than confidence or learning speed. Building on this, we introduced AS-CL, which uses this signal to protect fragile predictions in two stages. First, it strengthens the most fragile predictions while the current task's data is still available (Stage 1). Second, it tags each stored memory with a residual fragility score to protect it during future replay (Stage 2). Our ablations show that the retention gain comes from assigning greater weight to more fragile examples, rather than applying the same weighting uniformly. However, our approach is limited by Stage 1's reliance on an offline, task-aware protocol and by the fact that the scalar q_x is not revised once stored in memory. Addressing these limitations is left for future work.

AI USE STATEMENT

Generative AI tools were used to improve the clarity, and readability of the manuscript. They were not used to generate research ideas or experimental results. All AI-assisted edits were reviewed and revised by the authors, who take full responsibility for the content of the paper.

REFERENCES

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *NeurIPS*, 32, 2019.

Elahe Arani, Fahad Sarfraz, and Bahram Zonooz. Learning fast, learning slow: A general continual learning method based on complementary learning system. *ICLR*, 2022.

Beatrix Benkő. Example forgetting and rehearsal in continual learning. *Pattern Recognition Letters*, 179:65–72, 2024.

Prashant Shivaram Bhat, Bahram Zonooz, and Elahe Arani. Consistency is the key to further mitigating catastrophic forgetting in continual learning. In *Lifelong Learning Agents (CoLLAs)*, pp. 1195–1212, 2022.

Ang Bian, Wei Li, Hangjie Yuan, Chengrong Yu, Mang Wang, Zixiang Zhao, Aojun Lu, Pengliang Ji, and Tao Feng. Make continual learning stronger via c-flat. *NeurIPS*, 37:7608–7630, 2024.

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

Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New insights on reducing abrupt representation change in online continual learning. *ICLR*, 2022.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

Matthias De Lange, Gido Van de Ven, and Tinne Tuytelaars. Continual evaluation for lifelong learning: Identifying the stability gap. *ICLR*, 2023.

Masih Eskandar, Tooba Imtiaz, Davin Hill, Zifeng Wang, and Jennifer Dy. Star: Stability-inducing weight perturbation for continual learning. In *ICLR*, pp. 49102–49117, 2025.

Qiao Gu, Dongsub Shim, and Florian Shkurti. Preserving linear separability in continual learning by backward feature projection. In *CVPR*, pp. 24286–24295, 2023.

Guy Hacohen and Tinne Tuytelaars. Predicting the susceptibility of examples to catastrophic forgetting. *ICML*, 2025.

Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *CVPR*, pp. 831–839, 2019.

Yiding Jiang, Dilip Krishnan, Hossein Mobahi, and Samy Bengio. Predicting the generalization gap in deep networks with margin distributions. *ICLR*, 2019.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Alexander Krawczyk and Alexander Gepperth. An analysis of best-practice strategies for replay and rehearsal in continual learning. In *CVPRW*, pp. 4196–4204, 2024.

Duc-Thanh Le, Doanh C Bui, Maï K Nguyen, and Khang Nguyen. Continual model merging with test-time adaptation for whole-slide image analysis. *arXiv preprint arXiv:2607.04755*, 2026.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

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

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *NeurIPS*, 30, 2017.

Sergi Masip, Pau Rodriguez, Tinne Tuytelaars, and Gido M van de Ven. Continual learning of diffusion models with generative distillation. *Lifelong Learning Agents (CoLLAs)*, 2024.

Roman Novak, Yasaman Bahri, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Sensitivity and generalization in neural networks: an empirical study. *ICLR*, 2018.

Matthew Riemer, Tim Klinger, Djallel Bouneffouf, and Michele Franceschini. Scalable recollections for continual lifelong learning. In *AAAI*, volume 33, pp. 1352–1359, 2019.

Fahad Sarfraz, Elahe Arani, and Bahram Zonooz. Semantic aware representation learning for lifelong learning. In *ICLR*, volume 2025, pp. 58620–58639, 2025.

Giuseppe Serra, Ben Werner, and Florian Buettner. How to leverage predictive uncertainty estimates for reducing catastrophic forgetting in online continual learning. *TMLR*, 2025.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. *ICLR*, 2019.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE transactions on pattern analysis and machine intelligence*, 46(8):5362–5383, 2024.

Tao Zhuo, Zhiyong Cheng, Zan Gao, Hehe Fan, and Mohan Kankanhalli. Continual learning with strong experience replay. *arXiv preprint arXiv:2305.13622*, 2023.

APPENDIX

A ALGORITHM

Algorithm 1 AS-CL: training on task τ_t.

Require: $\mathcal{D}_t$, $\mathcal{T}$, buffer $\mathcal{M}$ with stored q_x, epochs $E_{\text{base}} < E$, weights λ_s, λ_q.

Base epochs: for E_{base} epochs, minimise $\mathcal{L}_{\text{CL}}$ with replay term replaced by $\mathcal{L}^{\text{AS}}_{\text{replay}}$ (Eq. 10)
$\theta_t^{\text{base}} \leftarrow \theta$
for each $x \in \mathcal{D}_t$ **do**
  sample $a_1, \dots, a_K \sim \mathcal{T}$; $\widetilde{Q}_t(x) \leftarrow 1 - \frac{1}{K}\sum_k \mathbf{1}[f_{\theta_t^{\text{base}}}(a_k(x)) = y]$ ▷ Eq. 6
end for
Stabilization epochs: for $E - E_{\text{base}}$ epochs, for each mini-batch $\mathcal{B}_t$, $\mathcal{B}_{\mathcal{M}}$:
  $v(x) \leftarrow (\epsilon + \widetilde{Q}_t(x))/\text{mean}_{\mathcal{B}_t}(\epsilon + \widetilde{Q}_t)$ ▷ Eq. 8
  sample K' fresh transforms per x; take a step on $\mathcal{L}_{\text{CL}} + \lambda_s \mathcal{L}_{\text{stab}}$ (Eqs. 8-9)
$\theta_t \leftarrow \theta$
for each entry (x, y) of τ_t in $\mathcal{M}$ **do**
  sample *fresh* $a_1, \dots, a_K \sim \mathcal{T}$; store $q_x \leftarrow 1 - S_t(x)$ ▷ residual susceptibility
end for

B STABILITY GROUPS, SIZES, AND SENSITIVITY

This section gives the details behind the stability groups $G_1, \dots, G_5$ used in Sec. 3.4 and Fig. 3. At the end of task τ_t we freeze θ_t and take the acquired set $\mathcal{A}_t$ (the test examples of τ_t that f_{θ_t} classifies correctly). For each $x \in \mathcal{A}_t$ we draw K=32 fresh transformations $a_1, \dots, a_K \sim \mathcal{T}$ and count the views on which the prediction changes,

$$n(x) \;=\; \sum_{k=1}^{K} \mathbf{1}[f_{\theta_t}(a_k(x)) \neq y] \;=\; K\left(1 - S_t(x)\right), \tag{11}$$

so $n(x)$ and the acquisition stability $S_t(x)$ of Eq. 6 carry the same information. Groups are defined on the failed-view count rather than on equal-width bins of S_t, because the distribution of $n(x)$ is concentrated at zero: a correctly classified example usually survives in every view. Equal-width bins on S_t would place the large majority of $\mathcal{A}_t$ in the top group and leave the remaining groups too small. Doubling the width at each step (0, 1, 2-3, 4-7, ≥ 8) spreads the mass while keeping G_5 (predictions that survive all 32 views) as a reference group.

	G_1	G_2	G_3	G_4	G_5
failed views $n(x)$	≥ 8	4–7	2–3	1	0
stability $S_t(x)$	≤ 0.75	$(0.75,\ 0.875]$	$(0.875,\ 0.9375]$	0.96875	1

Table 5 reports how $\mathcal{A}_t$ is distributed over the five groups on Split CIFAR-100 with M=2000. Only tasks $t \leq T - 5$ are included, to align with Fig. 3. Three points matter for this grouping. First, the distribution is top-heavy: G_4 and G_5 together hold 55-60% of the acquired predictions, so most of what a replay learner acquires is already stable, and the fragile band is a minority of the acquired set. Second, G_1 is the smallest group with 8-10% of $\mathcal{A}_t$; its forgetting rate is therefore the noisiest, which is why the G_1/G_5 ratio is used together with the group sizes and averaged over five class orders. Third, merging G_2 and G_3, or moving the G_1 edge from $n \geq 8$ to $n \geq 6$ or $n \geq 12$, leaves the forgetting rate monotone in the group index and changes the G_1/G_5 ratio by less than 0.4; the ratios reported in Sec. 4.1 are 5.5 (ER) and 7.1 (DER++).

C WHEN FRAGILE PREDICTIONS FAIL, AND WHY STAGE 1 ACTS EARLY

Table 6 shows the full distributions behind Fig. 6; and RMRH_5 is computed directly from each row. Reading the rows (except the 'never' column) shows when forgetting happens, which is what motivates our two-stage design. RMRH_5 is computed directly from each row. Under both base

Table 5: Distribution of acquired predictions over the stability groups (% of $\mathcal{A}_t$, mean±std over 5 class orders, Split CIFAR-100, M=2000, tasks $t \leq T-5$). The last column is the average number of acquired predictions per task, out of 1000 test-set images.

	G_1 $n \geq 8$	G_2 4-7	G_3 2-3	G_4 1	G_5 0	$\lvert\mathcal{A}_t\rvert$ per task
ER	10.0 ± 0.8	15.0 ± 1.2	18.0 ± 1.2	20.0 ± 1.0	37.0 ± 2.1	723
DER++	8.0 ± 0.6	13.0 ± 0.9	17.0 ± 1.0	22.0 ± 1.1	40.0 ± 1.8	740

Table 6: First-failure horizon per stability group (Split CIFAR-100, M=2000), RMRH_5 computed from each row. Groups use DER++ as reference on the predictions acquired by all four methods, so rates differ slightly from Fig. 3, which uses each learner's own acquired set. Each row sums to 100.

		first forgotten at (% of group)						
Method	Group	+1	+2	+3	+4	+5	never	RMRH_5
ER	G_1	38.5	12.5	9.5	6.5	5.5	27.5	2.83
	G_2	30.0	11.8	8.1	6.0	4.9	39.2	3.22
	G_3	20.2	8.4	5.8	4.5	3.8	57.3	3.78
	G_4	12.6	5.4	3.4	3.0	2.5	73.1	4.24
	G_5	7.4	2.6	2.5	2.2	1.7	83.6	4.55
ER + AS-CL	G_1	18.0	11.5	9.0	7.0	6.0	48.5	3.69
	G_2	15.5	9.5	6.5	4.5	4.0	60.0	3.92
	G_3	9.5	5.5	4.0	2.5	1.5	77.0	4.35
	G_4	6.0	3.0	2.5	1.5	1.0	86.0	4.61
	G_5	3.2	1.9	1.6	1.5	1.3	90.5	4.77
DER++	G_1	29.0	9.8	7.4	5.9	4.9	43.0	3.34
	G_2	20.3	7.1	5.3	4.5	4.0	58.8	3.82
	G_3	13.4	4.9	3.2	3.0	2.6	72.9	4.22
	G_4	8.2	2.6	1.7	1.7	1.3	84.5	4.54
	G_5	4.2	1.5	1.1	1.0	1.1	91.1	4.75
DER++ + AS-CL	G_1	13.5	8.0	6.5	5.2	4.8	62.0	4.04
	G_2	8.5	5.5	5.0	4.6	4.6	71.8	4.35
	G_3	5.2	3.6	3.4	3.6	3.8	80.4	4.58
	G_4	3.6	2.2	2.0	2.0	2.0	88.2	4.73
	G_5	2.6	1.4	1.2	1.2	1.2	92.4	4.82

learners (ER and DER++), almost half of all failures within five tasks occur at the very next task, in every stability group (47-53% for DER++). Replay can only defend an example once it is in the buffer, and only after the next task has been learned, so for these predictions it arrives after the damage. This is the gap Stage 1 fills, by repairing fragile predictions while $\mathcal{D}_t$ is still available.

If AS-CL only delayed failure, the predictions it saves at $+1$ would reappear later. On G_1 under ER, failures at $+1$ fall from 38.5% to 18.0% while failures at $+2$ to $+5$ together stay flat ($34.0\% \rightarrow 33.5\%$), and the never-forgotten rises by almost the amount removed from $+1$ ($27.5\% \rightarrow 48.5\%$). Under DER++ the shift is the same in kind. Across groups, the failures occurring at $+1$ drops to 27-43%, so what remains is spread over horizons rather than concentrated at the first update. **Comparing the never-forgotten columns shows that, on G_1, DER++ retains 47.1% at $+5$ (comparing G_1 and G_5) but only 43.0% never fail, so 9% of its retained predictions were recovered after breaking**; for AS-CL the gap is 63.4% against 62.0%, or 2%.

D TASK-IL PERFORMANCE

Task-IL uses the same trained models as Table 3; only the evaluation changes. The task identity of the evaluated example is given, and the softmax is restricted to that task's ten classes before the argmax is taken. The stability groups are those of Sec. 4.1, defined under Class-IL by the number of views K=32, so both parts of every bar in Fig. 8 describe the same predictions. All values are averaged over the same five class orders as the main results, over tasks $t \leq T-5$ so that every horizon uses the same tasks. Table 7 gives the same ranking of methods as Class-IL. The AS-CL

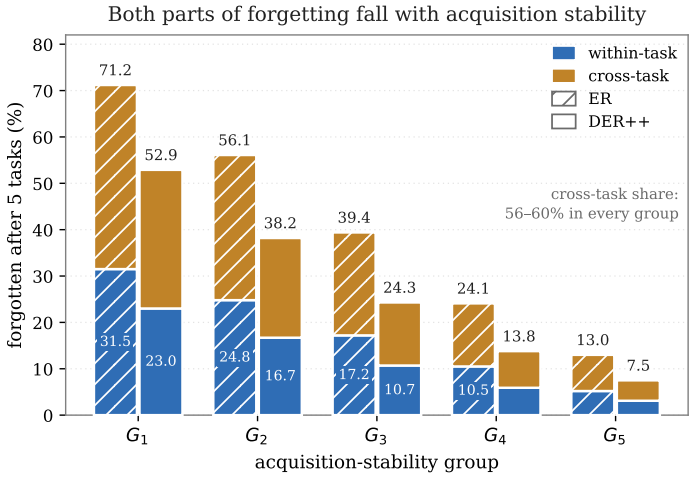


Figure 8: Each bar is the percentage of predictions correct at the end of τ_t that are incorrect five tasks later (Split CIFAR-100, M=2000). It splits into the part that survives when task identity is given (blue, the Task-IL rate) and the part that task identity removes, i.e. predictions captured by a class of another task. Hatched bars are ER, solid bars DER++. Both parts decrease with acquisition stability, and the cross-task share is 55-60% in every group, so acquisition stability is not a proxy for competition with later classes: it predicts forgetting of both kinds.

improves over DER++ from $+8.2$ to $+3.9$ under Task-IL, which matches the halving of the cross-task error rate in Fig. 5: about half of the Class-IL gain is the removal of cross-task confusion, and the other half is retention within the task itself.

Table 7: Task-IL results on Split CIFAR-100 (M=2000, 10 tasks), mean over five class orders. Same trained models as Table 3; only the evaluation protocol differs.

Method	FAA ↑	BWT ↑	LA ↑
ER	71.8	−16.31	86.5
DER++	79.0	−10.24	88.2
DER++ + Goldilocks	80.7	−8.51	88.4
SARL	81.3	−8.03	88.5
DER++ + AS-CL	**82.9**	**−6.24**	88.5

E Additional Ablation Results

E.1 Computation cost

Fig. 9 plots final accuracy against training time on Split TinyImageNet. AS-CL reaches the highest FAA in the comparison, and the K=8 setting recovers most of that gain at a lower cost. The methods that cost more than AS-CL (CLS-ER, SARL, DER++ + BFP) add work inside the backward pass at every step, whereas AS-CL's probe is forward-only and the stabilization term is applied in the last 20% of epochs. The overhead is larger here than on CIFAR-100 ($1.57\times$ against $1.38\times$ at K=32).

E.2 Sensitivity to hyperparameters

Table 8 tests the choices behind the probe and the weights. Using only rotation and translation, or only brightness and contrast, is worse than using both, in FAA and in AUROC: fragility is not tied to one kind of change, so the probe needs both to see it. Strong views change what the image shows, so the probe stops asking whether a prediction is secure and starts asking whether the object is still recognisable; weak views are too small to make a fragile prediction fail. Both lower FAA and AUROC. **The effect of ϵ.** FAA is flat up to ϵ=0.1 and falls beyond it, as expected: a larger ϵ makes $v(x)$ closer to the same value for every example, so AS-CL moves toward uniform stabilization

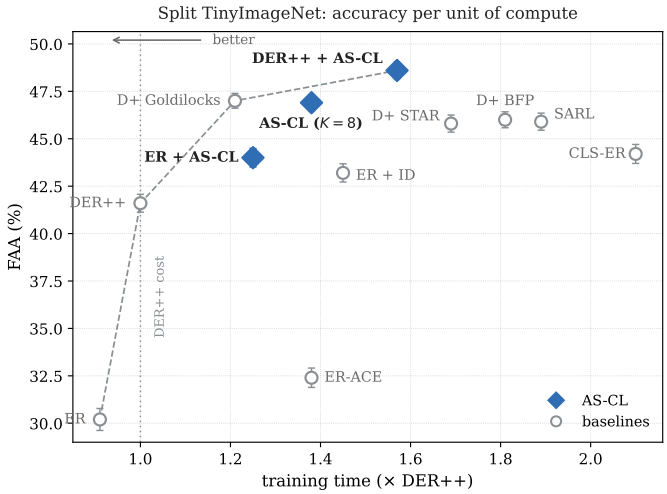


Figure 9: Accuracy vs. computation on Split TinyImageNet (M=4000). FAA values are from Table 3; training time is relative to DER++. AS-CL's overhead compared to CIFAR-100 is: $1.57\times$ vs. $1.38\times$ at K=32; because the K forward passes scale with image size.

(Table 4b). **Storing q_x.** Recomputing q_x at every replay, instead of fixing it at the end of the task, gains 0.2 FAA and costs $1.18\times$ more time, so one stored scalar per example is enough. **A second base learner.** On ER the ordering is the same as on DER++: AS-CL gains most, uniform stabilization less, and inverted weights give no gain over the base learner. The conclusions of Table 4 are therefore not specific to DER++.

Table 8: Additional ablations on Split CIFAR-100 ([M=2000]). Changing the views changes both the probe and the model trained on it, so FAA and AUROC move together.

View family	FAA	AUROC(Q_t)
geometric only (rotation, translation)	59.2	0.82
photometric only (brightness, contrast)	58.7	0.79
both (default)	**60.3**	**0.85**
stronger ($\pm 20°$, 12.5%, $[0.6, 1.4]$)	59.6	0.83
weaker ($\pm 5°$, 3.125%, $[0.9, 1.1]$)	58.4	0.80
The probe is unchanged in the following three blocks, so AUROC(Q_t) stays at the default value.		
Weight floor ϵ (default 0.05)		
0.05 / 0.1 / 0.25 / 0.5	60.3 / 60.1 / 59.4 / 58.6	-
Fragility		
q_x fixed at θ_t (default, $1.38\times$)	60.3	-
q_x recomputed at each replay ($1.63\times$)	60.5	-
Second base learner (ER)		
ER / + uniform stab. / + *inverted* / + AS-CL	39.4 / 47.2 / 39.8 / 54.7	-

F Class orders / Splitting Tasks

Mean ± std is over five class orders (seeds 0-4), generated as follow for three datasets.

F.1 Split CIFAR-100 (10 tasks × 10 classes).

The class Ids are (0-99); Seed 0; task 1 is:
`oak_tree, lamp, baby, clock, lizard, mountain, crab, butterfly, cattle, wardrobe.`

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

F.2 SPLIT-TINYIMAGENET (10 TASKS × 20 CLASSES).

Ids are the sorted TinyImageNet-200 index (0-199).

Seed 0.
T1: 103, 145, 86, 126, 37, 6, 130, 15, 46, 193, 91, 139, 8, 114, 77, 120, 30, 152, 0, 63;
T2: 119, 192, 34, 116, 55, 179, 42, 78, 159, 170, 56, 162, 166, 16, 158, 142, 191, 18, 153, 185;
T3: 58, 109, 148, 90, 29, 22, 81, 198, 112, 111, 102, 68, 123, 155, 66, 31, 59, 143, 186, 87;
T4: 24, 10, 85, 45, 107, 144, 41, 195, 131, 110, 189, 50, 177, 93, 19, 47, 125, 178, 121, 173;
T5: 14, 181, 172, 38, 132, 113, 69, 190, 188, 75, 71, 137, 65, 98, 104, 106, 61, 53, 3, 171;
T6: 124, 73, 141, 194, 122, 149, 39, 2, 84, 76, 105, 33, 99, 43, 160, 174, 127, 197, 48, 140;
T7: 146, 133, 28, 25, 51, 11, 97, 164, 157, 36, 17, 60, 150, 154, 40, 118, 88, 72, 82, 70;
T8: 187, 79, 167, 49, 74, 21, 7, 169, 175, 54, 32, 161, 9, 64, 13, 134, 136, 156, 100, 23;
T9: 57, 115, 147, 67, 128, 180, 163, 108, 44, 168, 1, 135, 199, 151, 183, 96, 165, 4, 95, 26;
T10: 12, 62, 5, 20, 52, 27, 117, 80, 182, 35, 184, 101, 196, 176, 92, 83, 94, 89, 129, 138.

Seed 1.
T1: 110, 168, 104, 155, 7, 65, 108, 173, 165, 137, 129, 12, 76, 85, 43, 145, 21, 64, 11, 45;
T2: 186, 101, 170, 135, 154, 188, 151, 29, 27, 8, 161, 24, 130, 54, 2, 17, 91, 199, 26, 47;
T3: 1, 146, 63, 153, 22, 89, 118, 30, 68, 157, 179, 20, 150, 53, 70, 114, 44, 169, 0, 19;
T4: 166, 171, 119, 172, 144, 113, 69, 61, 178, 38, 105, 190, 197, 37, 120, 100, 152, 112, 62, 194;
T5: 149, 82, 86, 131, 71, 106, 176, 125, 116, 13, 81, 102, 121, 78, 23, 6, 177, 35, 50, 90;

T6: 133, 185, 3, 95, 139, 67, 57, 41, 40, 148, 14, 142, 109, 75, 46, 163, 60, 77, 15, 52;
T7: 79, 162, 83, 51, 49, 134, 136, 66, 167, 191, 117, 159, 87, 195, 189, 141, 18, 196, 59, 183;
T8: 198, 175, 180, 124, 192, 127, 184, 122, 98, 158, 107, 140, 103, 94, 80, 111, 182, 181, 73, 16;
T9: 123, 92, 25, 147, 128, 31, 74, 115, 9, 156, 138, 10, 39, 99, 36, 164, 34, 33, 187, 55;
T10: 126, 56, 132, 72, 4, 48, 93, 88, 143, 42, 160, 84, 193, 28, 32, 96, 174, 97, 5, 58.

Seed 2.
T1: 85, 74, 15, 17, 178, 49, 22, 192, 57, 32, 6, 52, 10, 12, 19, 198, 108, 97, 122, 133;
T2: 90, 106, 196, 20, 54, 21, 142, 86, 60, 1, 95, 185, 143, 172, 137, 9, 81, 181, 173, 156;
T3: 138, 46, 63, 195, 16, 26, 166, 187, 28, 157, 160, 175, 161, 39, 7, 107, 132, 33, 171, 98;
T4: 136, 128, 84, 186, 31, 27, 56, 169, 130, 168, 135, 35, 197, 149, 114, 194, 140, 94, 34, 73;
T5: 162, 129, 65, 146, 121, 5, 134, 119, 167, 55, 120, 110, 11, 104, 62, 189, 145, 123, 88, 111;
T6: 51, 124, 179, 174, 193, 148, 67, 176, 68, 44, 147, 42, 0, 18, 100, 8, 182, 150, 89, 75;
T7: 112, 164, 29, 43, 183, 125, 13, 101, 36, 131, 47, 38, 159, 61, 23, 127, 155, 40, 139, 102;
T8: 48, 45, 93, 69, 117, 152, 72, 80, 184, 77, 151, 105, 109, 141, 96, 153, 92, 87, 91, 50;
T9: 78, 116, 191, 41, 126, 24, 71, 180, 199, 64, 177, 82, 58, 25, 14, 59, 37, 190, 30, 83;
T10: 70, 99, 2, 170, 66, 163, 158, 53, 79, 165, 115, 103, 76, 4, 188, 144, 118, 113, 154, 3.

Seed 3.
T1: 195, 147, 177, 17, 36, 70, 23, 165, 80, 121, 113, 115, 22, 67, 20, 43, 105, 56, 151, 106;
T2: 160, 194, 89, 48, 77, 12, 181, 82, 155, 124, 57, 50, 123, 164, 33, 100, 133, 161, 34, 125;
T3: 0, 18, 64, 86, 108, 99, 7, 31, 137, 158, 102, 1, 135, 81, 153, 116, 127, 87, 19, 131;
T4: 176, 44, 166, 95, 30, 101, 178, 38, 109, 173, 8, 62, 37, 61, 69, 139, 110, 60, 63, 162;
T5: 141, 180, 5, 11, 114, 146, 29, 93, 117, 52, 174, 130, 79, 184, 159, 16, 138, 170, 104, 142;
T6: 9, 28, 72, 51, 172, 83, 199, 192, 107, 54, 171, 119, 112, 148, 71, 41, 150, 145, 84, 188;
T7: 46, 74, 27, 163, 143, 103, 55, 156, 26, 175, 47, 197, 53, 65, 94, 98, 120, 122, 190, 2;
T8: 96, 111, 182, 21, 78, 39, 68, 193, 186, 189, 179, 140, 198, 49, 144, 42, 40, 169, 32, 14;
T9: 92, 25, 59, 85, 149, 88, 187, 183, 75, 191, 4, 76, 185, 15, 132, 45, 118, 134, 58, 10;
T10: 90, 73, 24, 6, 196, 126, 128, 91, 154, 157, 13, 136, 66, 97, 129, 152, 35, 3, 167, 168.

Seed 4.
T1: 176, 4, 54, 160, 119, 62, 27, 97, 139, 127, 98, 0, 48, 100, 28, 101, 52, 107, 31, 53;
T2: 80, 175, 150, 148, 59, 65, 40, 106, 69, 57, 170, 130, 163, 16, 23, 85, 143, 67, 180, 118;
T3: 129, 76, 88, 103, 181, 33, 89, 24, 46, 168, 193, 140, 178, 125, 136, 187, 172, 185, 191, 142;
T4: 104, 10, 123, 96, 91, 83, 167, 113, 124, 49, 116, 39, 138, 153, 81, 149, 74, 60, 128, 144;
T5: 30, 20, 73, 37, 19, 145, 102, 194, 8, 45, 6, 68, 70, 92, 188, 156, 135, 105, 157, 1;
T6: 36, 75, 21, 190, 195, 79, 108, 141, 109, 189, 50, 5, 26, 64, 159, 158, 47, 171, 63, 35;
T7: 82, 94, 199, 13, 25, 137, 179, 61, 2, 196, 86, 177, 134, 117, 41, 114, 169, 192, 18, 131;
T8: 78, 112, 121, 55, 95, 154, 122, 22, 72, 174, 110, 152, 146, 34, 51, 58, 161, 77, 43, 151;
T9: 165, 164, 111, 66, 183, 120, 173, 32, 87, 133, 84, 17, 147, 90, 198, 3, 99, 7, 182, 162;
T10: 42, 38, 93, 197, 14, 56, 71, 9, 15, 44, 29, 12, 126, 155, 166, 184, 115, 186, 132, 11.

F.3 Split ImageNet-100 (10 tasks × 10 classes).

We use the 100-class subset of Hou et al. (2019), indexed 0-99 in that list. The five task assignments are the same integer tables as Split CIFAR-100 (10-task) above.

G Theoretical discussion

This section provides facts used in Sec. 3.

G.1 When an acquired prediction is forgotten

For $(x, y) \in \mathcal{A}_t$ and $s > t$, writing $m_{\theta_s}(x)$ as a displacement from $m_{\theta_t}(x)$ separates what is known at time t from what is not

$$Z_{t,s}(x) = 1 \quad \Longleftrightarrow \quad \underbrace{m_{\theta_t}(x)}_{\text{margin at acquisition}} + \underbrace{\big(m_{\theta_t+\Delta_{t\to s}}(x) - m_{\theta_t}(x)\big)}_{\text{displacement caused by later tasks}} \le 0. \tag{12}$$

The prediction is retained whenever the first term dominates the magnitude of the second. The first is fixed once task t ends; the second is determined by tasks not yet seen. Bounding the displacement turns Eq. 12 into a usable sufficient condition.

Proposition 1 (Bounded parameter drift). *Fix $(x, y) \in \mathcal{A}_t$ and suppose $\theta \mapsto m_\theta(x)$ is $L_\theta(x)$-Lipschitz on a convex set containing θ_t and θ_s. If*

$$m_{\theta_t}(x) \;>\; L_\theta(x)\,\|\Delta_{t\to s}\|, \tag{13}$$

then $Z_{t,s}(x) = 0$.

Proof. $|m_{\theta_s}(x) - m_{\theta_t}(x)| \le L_\theta(x)\|\Delta_{t\to s}\|$; combined with Eq. 13 this gives $m_{\theta_s}(x) > 0$.

Eq. 13 is the shape of the whole argument: retention is decided by a margin measured at acquisition against a sensitivity measured at acquisition, and only the scale of $\|\Delta_{t\to s}\|$ is unknown at time t. What is missing is a way to estimate $L_\theta(x)$ per example without seeing $\Delta_{t\to s}$.

G.2 First-order expansions and local survival radii

Assume $m_\theta(x)$ is C^2 in (θ, x) on a neighborhood of (θ_t, x). For piecewise-linear networks m is instead locally affine on the activation region containing (θ_t, x), where Eqs. 14-15 below hold exactly with $R_\theta{=}R_x{=}0$; the C^2 statement covers smooth activations and the case where a region boundary is crossed.

Proposition 2 (Taylor with remainder). *There exist remainders R_θ, R_x such that*

$$m_{\theta_t+\Delta}(x) = m_{\theta_t}(x) + \langle \nabla_\theta m_{\theta_t}(x), \Delta\rangle + R_\theta(\Delta), \tag{14}$$

$$m_{\theta_t}(x+\delta) = m_{\theta_t}(x) + \langle \nabla_x m_{\theta_t}(x), \delta\rangle + R_x(\delta), \tag{15}$$

with $|R_\theta(\Delta)| \le \frac{1}{2}H_\theta\|\Delta\|^2$ and $|R_x(\delta)| \le \frac{1}{2}H_x\|\delta\|^2$, where H_θ (resp. H_x) bounds the Hessian operator norm in θ (resp. x) on the line segment.

Proof. Lagrange remainder for C^2 functions.

Corollary 1 (Linearized survival radii). *Suppose $\|\Delta\|$ is small enough that $|R_\theta(\Delta)| \le \frac{1}{2}m_{\theta_t}(x)$. Then $m_{\theta_t+\Delta}(x) > 0$ whenever*

$$\|\Delta\| \;<\; \rho_\theta(x) \;:=\; \frac{m_{\theta_t}(x)}{2\|\nabla_\theta m_{\theta_t}(x)\|_2}.$$

Likewise, if $\|\delta\|$ is small enough that $|R_x(\delta)| \le \frac{1}{2}m_{\theta_t}(x)$, then $m_{\theta_t}(x+\delta) > 0$ whenever

$$\|\delta\| \;<\; \rho_x(x) \;:=\; \frac{m_{\theta_t}(x)}{2\|\nabla_x m_{\theta_t}(x)\|_2}.$$

Proof. By Eq. 14 and Cauchy-Schwarz, $m_{\theta_t+\Delta}(x) \ge m_{\theta_t}(x) - \|\nabla_\theta m_{\theta_t}(x)\|_2\|\Delta\| - \frac{1}{2}m_{\theta_t}(x)$, which is positive exactly when $\|\Delta\| < \rho_\theta(x)$. The input case is identical using Eq. 15.

To first order, both parameter and input survival are therefore controlled by the same ratio, margin over local sensitivity (Novak et al., 2018). The two radii are not equal, because $\nabla_\theta m$ and $\nabla_x m$ are different maps and $\Delta_{t\to s}$ is not a random draw in the δ-ball. They are, however, coupled through the same backpropagated quantity: at the first layer $h{=}W_1x$ we have $\nabla_{W_1} m{=}g\,x^\top$ and $\nabla_x m{=}W_1^\top g$ with $g{=}\partial m/\partial h$, so an example with large $\|g\|$ is sensitive in both. This is what licenses using input views to probe ρ_θ, which we cannot measure at time t.

Proposition 3 (Input ball implies $S_t = 1$). *Suppose $x \mapsto m_{\theta_t}(x)$ is L_x-Lipschitz on a region containing $\{a(x) : a \in \mathcal{T}\}$ and $\|a(x) - x\| \le r$ for every $a \in \mathcal{T}$. If $m_{\theta_t}(x) > L_x r$, then $S_t(x) = 1$.*

Proof. $m_{\theta_t}(a(x)) \ge m_{\theta_t}(x) - L_x\|a(x) - x\| > 0$ for every $a \in \mathcal{T}$, so every view is correct.

Proposition 3 is one direction only, and this bounds what AS should be taken to mean. $S_t(x) = 1$ on K random views does not imply a lower bound on the adversarial input margin: the views in $\mathcal{T}$ are mild and class-preserving, not worst-case. Nor does Corollary 1 establish that ρ_x orders examples the same way ρ_θ does; the gradients are coupled but not proportional, and $\Delta_{t\to s}$ is structured by the future task rather than isotropic. AS estimates a nonlinear, distributional analogue of ρ_x under $\mathcal{T}$, and whether that ranking tracks the drops of m produced by the true $\Delta_{t\to t+1}$ is an empirical question, tested in Sec. 3.4. AS is a cheap diagnostic, not a robustness certificate.

G.3 What S_t estimates

Define the population acquisition stability

$$s_t(x) \;:=\; \mathbb{P}_{a\sim\mathcal{T}}\big(f_{\theta_t}(a(x)) = y\big) \;=\; \mathbb{E}_{a\sim\mathcal{T}}\big[\mathbf{1}[f_{\theta_t}(a(x)) = y]\big]. \tag{16}$$

Proposition 4 (Unbiased Monte Carlo)**.** *If $a_1, \dots, a_K$ are i.i.d. from $\mathcal{T}$, then $\mathbb{E}[S_t(x) \mid x] = s_t(x)$ and, for every $\varepsilon > 0$,*

$$\mathbb{P}\big(|S_t(x) - s_t(x)| \geq \varepsilon\big) \;\leq\; 2e^{-2K\varepsilon^2}. \tag{17}$$

Proof. Each indicator is $[0, 1]$-valued and independent given x; Hoeffding's inequality gives Eq. 17.

Eq. 17 is distribution-free and correspondingly loose: at K=32 it gives a 95% radius of 0.24, and attaining ε=0.15 at that level would require K=82. Because the summands are Bernoulli, the binomial standard deviation is the sharp statement (for K=32 it is 0.053 at s_t=0.9, 0.059 at s_t=0.875 and 0.088 at s_t=0.5), and these are below the width of the cuts in Sec. 3.3. Two remarks. First, groups are assigned from the observed failure count $n(x)$ rather than from a thresholded estimate of s_t, so S_t is exact for the views drawn; what Eq. 17 bounds is how often a re-draw would move an example across a cut. Second, Appendix B compares the measured group sizes at K=32 and K=64, which speaks to the stability of the partition directly. Throughout, $Q_t = 1 - S_t$ is the Monte Carlo rate of view-flips.

G.4 AS is not a function of pointwise confidence

Let $p_\theta(y \mid x) = \mathrm{softmax}(z_\theta(x))_y$.

Proposition 5 (Distinct objects)**.** *There exist networks, inputs, and transform families $\mathcal{T}$ for which $p_{\theta_t}(y \mid x)$ (or $m_{\theta_t}(x)$) is identical at two points x, x' while $s_t(x) \neq s_t(x')$. Conversely, s_t may coincide while $p_{\theta_t}(y \mid x)$ differs.*

Proof. Both $p_{\theta_t}(y \mid x)$ and $m_{\theta_t}(x)$ are functions of $z_{\theta_t}(x)$ alone, whereas $s_t(x)$ depends on z_{θ_t} over the whole orbit $\{a(x) : a \in \mathcal{T}\}$. For an explicit witness, let $x \in \mathbb{R}$, let $\mathcal{T}$ be translations $x \mapsto x + \delta$ with $\delta \sim \mathcal{U}[-1, 1]$, and let $m(u) = 1$ for $u \leq 2$ and $m(u) = 1 - 2(u - 2)$ for $u > 2$, which a single hidden layer with two ReLU units realizes exactly. At x=0 and x'=2 we have $m(x) = m(x') = 1$, hence identical margin and identical confidence, while $s_t(x) = 1$ and $s_t(x') = \mathbb{P}(\delta > 0.5) = 0.75$. For the converse, rescale the logits at x without changing the sign pattern of m on the orbit: p moves and s_t does not.

This is why Sec. 3.4 reports AS within bins of confidence and of learning speed (**?**): the comparison is informative only once the pointwise quantity is held fixed.

G.5 Mean-one reweighting is a covariance, not extra loss

Let $\mathcal{B}^+ = \{x_i\}_{i=1}^n$ and write $q_i = \widetilde{Q}_t(x_i)$ and $\ell_i = \frac{1}{K'} \sum_k \ell(f_\theta(a_k(x_i)), y_i)$.

Proposition 6 (Stage-1 identity)**.** *With $v_i = (\varepsilon + q_i)/\mu$ and $\mu = n^{-1} \sum_j (\varepsilon + q_j)$, one has $n^{-1} \sum_i v_i = 1$ and*

$$\frac{1}{n} \sum_{i=1}^{n} v_i \ell_i \;=\; \bar{\ell} \;+\; \frac{\mathrm{Cov}_n(q, \ell)}{\varepsilon + \bar{q}}, \tag{18}$$

where $\bar{\ell} = n^{-1} \sum_i \ell_i$, $\bar{q} = n^{-1} \sum_i q_i$, and $\mathrm{Cov}_n(q, \ell) = n^{-1} \sum_i (q_i - \bar{q})(\ell_i - \bar{\ell})$.

Proof. $\mu = \varepsilon + \bar{q}$, so $\sum_i v_i = n$ and the weights have mean one. Then

$$\frac{1}{n} \sum_i v_i \ell_i = \frac{\varepsilon \bar{\ell} + \frac{1}{n} \sum_i q_i \ell_i}{\varepsilon + \bar{q}},$$

and substituting $\frac{1}{n} \sum_i q_i \ell_i = \bar{q}\, \bar{\ell} + \mathrm{Cov}_n(q, \ell)$ gives Eq. 18.

Uniform stabilization is $v \equiv 1$, i.e. $\bar{\ell}$; AS-CL adds a term proportional to $\mathrm{Cov}_n(q, \ell)$. If fragile points carry larger view-loss, which is the intended case, that covariance is positive and extra training is spent on high-Q points without raising the mean weight. This is what the $v \equiv 1$ ablation isolates: it sets the covariance term to zero while keeping λ_s, so the same budget is spent uniformly.

Proposition 7 (Stage-2 identity)**.** *With* $w_i = (1 + \lambda_q q_{x_i})/\nu$ *and* $\nu = n^{-1} \sum_j (1 + \lambda_q q_{x_j})$*, one has* $n^{-1} \sum_i w_i = 1$ *and*

$$\frac{1}{n} \sum_i w_i \ell_i^{\mathrm{replay}} = \bar{\ell}^{\mathrm{replay}} + \frac{\lambda_q \, \mathrm{Cov}_n(q, \ell^{\mathrm{replay}})}{1 + \lambda_q \bar{q}}.$$

Proof. Apply Proposition 6 with $\varepsilon = 1$ and q replaced by $\lambda_q q$, using $\mathrm{Cov}_n(\lambda_q q, \ell) = \lambda_q \, \mathrm{Cov}_n(q, \ell)$.

Proposition 8 (Exact recovery of the backbone)**.** *If* $\lambda_s = 0$ *then* $\mathcal{L}_{\mathrm{acq}} = \mathcal{L}_{\mathrm{CL}}$*. If* $\lambda_q = 0$ *then* $w_i = 1$ *and* $\mathcal{L}_{\mathrm{replay}}^{\mathrm{AS}}$ *is ordinary replay. If* $\lambda_s = \lambda_q = 0$*, AS-CL is the underlying method, including its buffer rule.*

Proof. Immediate from Eqs. 9 and 10. When λ_s=0 the stabilization term vanishes and no view passes are computed, so the per-step cost also reduces to that of the backbone.