RENEWAL–PERSISTENCE REPLAY FOR RETURNING CONCEPTS IN TIMESTAMPED MULTIMODAL STREAMS

Anonymous authors
Paper under double-blind review

ABSTRACT

An image–text pair may become useful again after its concept disappears and returns, but a finite replay buffer can evict the pair before its renewed value is visible. Renewal–Persistence Replay (RPR) addresses both decisions: a frozen harmful-staleness ranker updates priority from teacher disagreement, embedding migration, and stable-anchor support, while cluster sampling mass and eviction floors keep low-frequency history reachable. Recurring, low-displacement concepts account for 28.5–35.8% of future queries in our three timestamped streams, making this return regime substantial rather than exceptional. With the model, contrastive objective, buffer, update budget, and checkpoint rule held fixed, five-seed SciCapStream comparisons reduce forgetting by 0.7±0.5 points versus MIR-style replay and 0.9±0.5 versus GSS; RPR also lowers forgetting from 14.1±0.6 to 9.4±0.4 relative to strict timestamp ordering without reducing future retrieval. The ordering repeats on PMC-Quarterly and RedCaps-Monthly and transfers to ViT-L/14. Its mechanism is localized: the SciCap RPR–Recency effect expands to -6.6±0.9 points for high recurrence but contracts to -0.2±0.5 for weak recurrence; corrupting visible time erases RPR's margin over a shared timestamp-agnostic MIR trajectory; and oscillating concepts in ConceptDrift-5K gain +14.7±1.8 R@1. These results identify returning-concept streams as a consequential replay regime and show that coupling current utility to continued reachability improves replay there.

1 INTRODUCTION

An image–text concept need not remain continuously present to remain useful. A microscopy relation, visual genre, or naming convention can appear, disappear for several stream windows, and then return after the representation has moved. The historical pair is old at the moment its corrective value is renewed. Using frozen CLIP pair states (Radford et al., 2021), we find this trajectory at material scale: recurring, low-displacement families cover 28.5–35.8% of future queries across SciCapStream, PMC-Quarterly, and RedCaps-Monthly.

The corresponding replay problem is not only which available sample to retrieve. It is also whether a potentially useful sample remains available when its concept returns. We call a reversal between age order and current training value *temporal utility inversion*. Its algorithmic consequence separates into *renewal*, which updates priority using the current model state, and *persistence*, which protects the option to make that update later. Rescoring cannot recover an evicted pair; retention without rescoring cannot identify the survivor that is useful now.

Renewal–Persistence Replay (RPR) turns this consequence into a finite-buffer controller without changing the contrastive objective. A logistic model fitted once on a paper-disjoint pilot stream maps teacher disagreement, embedding migration, and stable-anchor support to harmful-staleness probability. Re-evaluation at each boundary allows a surviving pair to regain sampling weight as the representation moves. Explicit cluster mass and eviction floors simultaneously reserve capacity for low-frequency concepts. RPR changes only selection and survival: model initialization, stream order, loss, replay ratio, gradient updates, and checkpoint selection remain shared.

Adaptive replay already establishes that utility is model-dependent: MIR scores interference, GSS controls gradient geometry, ASER values class boundaries, and GRASP and SWIL use prototype structure (Aljundi et al., 2019a;b; Shim et al., 2021; Harun et al., 2025; Saxena et al., 2022). RPR addresses a different failure in raw, label-free image–text streams: a selector evaluated only on the

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.

current buffer cannot benefit from historical candidates already removed. Its technical contribution is therefore a coupled state transition in which current-model signals renew replay weight and a separately defined floor preserves the candidate set on which renewal can act.

The strongest fixed-objective test is closure against adaptive non-age selectors: on SciCap, RPR reduces forgetting by 0.7 ± 0.5 points versus MIR and 0.9 ± 0.5 versus GSS, while strict timestamp ordering moves from 14.1 ± 0.6 to RPR's 9.4 ± 0.4. The ordering repeats on PMC and RedCaps. Recurrence strata then locate the gain where concepts return, corruption of controller-visible time removes RPR's margin over invariant MIR, and ConceptDrift-5K records survival and replay at return. The same signature persists at ViT-L/14 and becomes neutral at a natural weak-recurrence boundary.

Contributions. The primary contribution is RPR, a replay operator that makes renewed utility and finite-buffer reachability jointly actionable while preserving the learner and objective. The empirical contribution is an identification sequence spanning three real streams and two backbones: paired closure against interference- and gradient-based selectors, timestamp intervention, method-independent recurrence localization, direct returned-pair re-entry, component counterfactuals, and a natural boundary. Together they establish a concrete decision rule for recurring multimodal streams: historical value should be reconsidered when concepts return, and that reconsideration is effective only over history that remains reachable.

2 Related Work

Selection, storage, and retrieval. Experience replay, reservoir sampling, iCaRL, GDumb, GEM, and A-GEM choose among admission, persistence, and retrieval operators (Chaudhry et al., 2019b; Vitter, 1985; Rebuffi et al., 2017; Prabhu et al., 2020; Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019a). MIR retrieves samples with high predicted interference, whereas GSS preserves a gradient-diverse memory (Aljundi et al., 2019a;b). These are decisive controls for RPR: MIR can renew retrieval priority without age, GSS couples model state to storage, and Reservoir preserves an age-agnostic history. RPR differs in assigning renewal and finite-buffer survival to explicit, separately testable operators.

Adaptive and temporally structured replay. ASER uses label-conditioned KNN Shapley values for both memory update and retrieval; GRASP traverses class-prototype distance from easy to hard; and SWIL allocates replay to old classes similar to the incoming class (Shim et al., 2021; Harun et al., 2025; Saxena et al., 2022). Watch Your Step systematizes such class- and sample-selective primitives in a fixed-buffer object-detection setting (Hickok & Kudithipudi, 2024). These methods make strong current-batch choices, whereas RPR's return event also depends on what survived the intervening absence in a label-free paired stream. Adaptive Memory Replay avoids that loss by allocating from a fully retained history (Smith et al., 2024). PHASER instead protects brief action phases and routes replay by task interference in VLA trajectories (Chen et al., 2026). RPR combines finite raw-pair survival, current-state renewal, and a direct intervention on timestamp validity in one testable controller.

Continual vision–language learning. Regularization, distillation, architectural expansion, contrastive objectives, and feature or generative replay preserve prior capabilities through changes to the learner, loss, or memory representation (Kirkpatrick et al., 2017; Li & Hoiem, 2016; Rusu et al., 2016; Cha et al., 2021; Douillard et al., 2022; Ni et al., 2023; Zheng et al., 2023; Frascaroli et al., 2024). CoMem, for example, stores concept graphs and rehearses generated features with additional consistency and distillation terms (Zhou et al., 2026). Those systems target end-to-end adaptation; our selector-only closure keeps raw pairs and the symmetric CLIP objective fixed. DER++-CLIP is reported separately because its logit-replay mechanism changes that objective (Buzzega et al., 2020).

Data value under non-stationarity. Curriculum and mixture methods allocate training across examples or sources (Bengio et al., 2009; Ibrahim et al., 2024; Xie et al., 2023); data valuation and subset selection estimate influence, learnability, or proxy utility (Ghorbani & Zou, 2019; Koh & Liang, 2017; Ilyas et al., 2022; Paul et al., 2021; Killamsetty et al., 2021; Coleman et al., 2020; Sorscher et al., 2022; Mindermann et al., 2022; Toneva et al., 2019). Temporal utility inversion adds

a reachability constraint absent from static valuation: a later utility estimate is actionable only if the example remains in the buffer. Appendix B compares this distinction at the operator level.

3 Replay Utility After Concept Return

3.1 Pairwise utility and controlled comparison

Data arrives in windows $W_1, \ldots, W_T$ with paired observations $x_i = (x_i^I, x_i^T)$ and controller-visible timestamps t_i. Let $u_i(t)$ denote the expected next-transition retrieval change when pair i replaces a matched candidate in an otherwise identical replay update at boundary t; the expectation is over minibatch order and matched co-replay contexts. For two available pairs with $t_i < t_j$, monotone Recency ranks j first, and $u_i(t) > u_j(t)$ is a temporal utility inversion. Our protocol operationalizes its consequences at two levels: pilot exchanges supervise harmful staleness against matched same-window controls, and real streams estimate paired selector effects localized by recurrence, timestamp intervention, and returned-pair re-entry.

Every selector-only method therefore uses the same pretrained ViT-B/16 CLIP initialization, symmetric image–text contrastive loss, AdamW optimizer, update count, new/replay ratio, buffer capacity M, temporal split, and final-checkpoint evaluation. Recency, Reservoir, MIR-style, GSS, and all RPR variants satisfy this contract. Joint training sees all windows simultaneously and is an offline upper bound.

DER++-CLIP retains the DER++ mechanism and therefore sits outside selector-only closure. For replay batch $\mathcal{R}$, it optimizes

$$\mathcal{L}_{\mathrm{DER++}} = \mathcal{L}_{\mathrm{CLIP}} + \lambda_{\mathrm{DER}} \| S_\theta(\mathcal{R}) - S_{\mathrm{stored}}(\mathcal{R}) \|_F^2 / |\mathcal{R}|^2,$$

where S is the temperature-scaled pairwise image–text logit matrix and $\lambda_{\mathrm{DER}} = 1.0$. Its comparisons test whether RPR remains competitive with an objective-changing continual-learning baseline; claims of same-objective closure refer only to the selector-only set.

3.2 Frozen harmful-staleness model

Let f_θ^I and f_θ^T be the normalized CLIP image and text projections and define the pair state

$$z_i^\theta = \mathrm{norm}\big(f_\theta^I(x_i^I) + f_\theta^T(x_i^T)\big)\,.$$

Image–text aggregation is therefore fixed before every controller operation. For a buffered sample observed at t_i and evaluated at $t > t_i$, RPR estimates

$$h(x_i, t) = \sigma(w_d d_{\mathrm{teacher}}(x_i, t) + w_m m_{\mathrm{embed}}(x_i, t) + w_a(1 - s_{\mathrm{anchor}}(x_i, t)) + b)\,, \qquad (1)$$

where $d_{\mathrm{teacher}} = \| z_i^{\theta_{\mathrm{EMA}}} - z_i^{\theta_t} \|_2$ measures teacher–student disagreement and $m_{\mathrm{embed}} = \| z_i^{\theta_t} - z_i^{\theta_{t_i}} \|_2 / \Delta t_i$ measures migration per controller-visible window, with $\Delta t_i = \max\{1, t - t_i\}$. For stable-anchor bank $\mathcal{A}_s$, support is

$$s_{\mathrm{anchor}}(x_i, t) = \max_{a \in \mathcal{A}_s} \left[\left(z_i^{\theta_t} \right)^\top z_a^{\theta_t} \right]_+ .$$

During warm-up, before $\mathcal{A}_s$ is nonempty, the maximum is taken over the recent-anchor bank. Incoming candidates use $\Delta t_i = 1$ and zero migration; their current pair state becomes the stored reference. All three scalar inputs are standardized with the frozen pilot means and variances.

The binary label is one for *harmful staleness*. SciCap-Pilot uses the same CC-BY figure–caption filters, paper locking, and deduplication as SciCapStream, but its papers and probes are disjoint from every reported split. Four ordered pilot windows provide three adjacent transitions. Each of 600 unique candidates is exchanged against a same-window control matched by pre-update contrastive-loss decile in 4 co-replay contexts, producing 2400 exchange observations. The new-data batch, co-replay set except for the exchanged pair, model and optimizer states, and minibatch order are matched. An exchange receives label one when its next-transition retrieval change is lower than its control. All four contexts for a candidate stay in the same split: two complete transitions are used for fitting and the third for validation. We minimize inverse-frequency-weighted logistic loss over

exchanges with L2 coefficient 0.01. The signs $w_d < 0$, $w_m < 0$, and $w_a > 0$ let disagreement and migration restore priority when stable support indicates a persistent relation.

Normalization, coefficients, the diagnostic threshold, and calibration are frozen before any reported table is inspected. The same fitted map is used unchanged for SciCap, PMC, RedCaps, ConceptDrift-5K, and both backbones; only features are recomputed by the current model. On the complete held-out transition, validation yields AUROC 0.742±0.018, AUPRC 0.687±0.021, and expected calibration error 0.036±0.007. Intervals use a candidate-cluster bootstrap, preserving the four contexts for each resampled candidate; downstream generalization is assessed separately by matched-seed stream results. Continuous $1 - h$ determines all rankings, while $h < 0.5$ is used only for diagnostics. Appendix F reports counts, coefficients, and freeze timing.

3.3 RENEWAL–PERSISTENCE COUPLING

At every boundary, RPR applies k-means++ to the current pair states $z_i^{\theta_t}$ and runs Lloyd updates to assignment convergence with $k = 100$; the run seed fixes initialization. The retained–incoming union is reclustered after training, so incoming pairs and surviving history are compared in one representation state. Empty clusters are removed from the active set $\mathcal{C}_t$; writing $k_t = |\mathcal{C}_t| \le k$ redistributes their uniform mass across nonempty clusters. Let $\mathcal{B}_c$ denote active buffer cluster c and $S_c = \sum_{i \in \mathcal{B}_c} (1 - h_i)$. RPR assigns cluster mass

$$\pi_c = \frac{q_{\min}}{k_t} + (1 - q_{\min}) \frac{S_c}{\sum_{j \in \mathcal{C}_t} S_j}, \qquad q_{\text{cluster}}(x_i) = \frac{\pi_c}{\max(S_c, \epsilon)},$$

and the replay weight

$$\omega(x_i, t) = (1 - h(x_i, t)) q_{\text{cluster}}(x_i). \tag{2}$$

with $\epsilon = 10^{-12}$ used only as a numerical safeguard. Consequently, normalized sample weights sum to π_c within cluster c; $q_{\min} = 0.1$ reserves ten percent of replay probability uniformly across the clusters, while the remaining mass follows renewed utility. Each replay minibatch is drawn without replacement by first sampling a cluster from π and then an item within that cluster in proportion to $1 - h_i$, renormalizing after each draw.

Persistence is enforced separately from the multiplicative score. The nominal cluster floor is $\widetilde{\ell}_c = \lfloor q_{\min} M / k_t \rfloor$, and its feasible value is $\ell_c = \min\{|\mathcal{A}_c|, \widetilde{\ell}_c\}$ for available candidates $\mathcal{A}_c$. Eviction removes the lowest-ω item only from clusters above ℓ_c. If a cluster has fewer candidates than its nominal floor, RPR retains all available items and redistributes the shortfall among feasible clusters in proportion to π_c. At each boundary RPR rescales features, scores the existing buffer, computes π_c and floors, trains with constrained replay, and updates the EMA after every optimizer step. It then forms the union of retained and incoming candidates, scores the entire union under the updated model, recomputes h, S_c, π_c, ω, and feasible floors, redistributes infeasible floor capacity, and performs insertion and eviction. Anchors refresh last. This order compares all eviction candidates under one model state while renewal ranks available history and persistence preserves future options.

Operationally, insertion and eviction are one deterministic capacity projection: repeatedly remove the lowest-ω item from the currently over-floor clusters until M items remain. Ties in all selectors are resolved by a fixed hash of normalized pair content, never by arrival order. This projection also defines capacity redistribution: a floor shortfall is assigned one slot at a time to the feasible cluster with largest remaining π_c mass. Thus every incoming and retained pair is scored once under the same post-window model before the new buffer is committed.

Both anchor banks are indices into the common M-pair buffer, not auxiliary example stores. Recent anchors point to the latest $k_r = 1000$ buffered pairs. The stable bank starts empty, admits a currently buffered pair after three surviving windows with $m_{\text{embed}} < 0.05$, and drops its entry when that pair is evicted; it therefore contains at most M references. Cached pair states are metadata, and each state is refreshed by re-encoding its buffered pair at the boundary. The teacher uses $\beta_t = \beta_0(1 - \alpha \bar{m}_t)$ with $\beta_0 = 0.999$, $\alpha = 0.5$, and buffer-mean migration $\bar{m}_t$ clipped to $[0, 1]$. Algorithm 1 gives the full update order, and Table 20 varies anchor and teacher choices in addition to M, k, and $q_{\min}$.

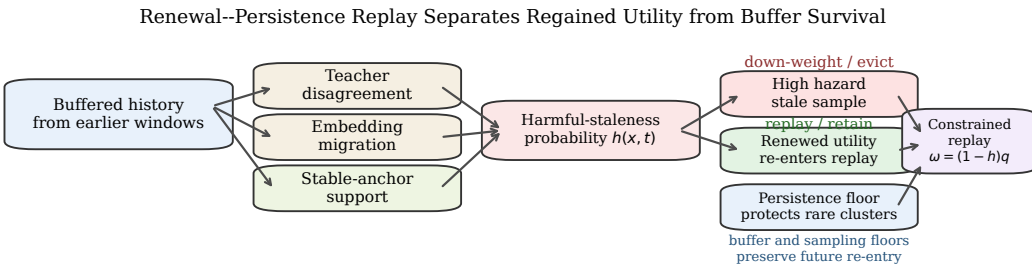

Figure 1: RPR couples utility renewal and buffer persistence. The harmful-staleness fit lets old examples regain replay weight; explicit cluster masses and eviction floors retain rare stable concepts until their value can renew.

3.4 Metrics and inference

Let $A_{u,t}$ be bidirectional retrieval R@1 on the fixed probe from W_u after training through W_t; R@1 averages image-to-text and text-to-image retrieval over the complete paper-disjoint gallery. We report

$$\mathrm{BR} = \frac{1}{T-1}\sum_{u=1}^{T-1} A_{u,T}, \quad \mathrm{F} = \frac{1}{T-1}\sum_{u=1}^{T-1}\left(\max_{v\in\{u,\dots,T\}} A_{u,v} - A_{u,T}\right),$$

and $\mathrm{FWT} = A_{T+1,T} - A_{T+1,0}$. Tables show raw Future $A_{T+1,T}$ because the common initialization subtracts the same constant. SciCapStream forgetting averages all six eligible 2018–2023 windows.

Means use five matched seeds. Intervals are two-sided 95% Student-t intervals over seed-wise metrics; paired intervals use seed-wise method differences. The primary multiplicity family consists of the four RPR contrasts against Recency, MIR-style, GSS, and DER++-CLIP within each real stream, with two-sided paired t-tests and Holm adjustment. Timestamp, recurrence, component, and transfer analyses are prespecified mechanism families and are reported with effect intervals rather than pooled into the primary family. Future competence is non-inferior when the paired interval's lower endpoint exceeds the -0.5-point margin fixed before the reported runs.

4 Experimental Design

SciCapStream contains 2.1M CC-BY arXiv figure–caption pairs in seven annual windows from 2018–2024, with 2025 as an 8421-pair future probe. PMC-Quarterly contains 8 contiguous publication quarters from 2021Q1–2022Q4, 120K open-access PubMed Central pairs per window, and 4176 queries from 2023Q1. RedCaps-Monthly uses 12 calendar months from January–December 2020, 150K public image–title pairs per month, and 5032 January 2021 queries (Desai et al., 2021). Annual, quarterly, and monthly metadata test distinct temporal resolutions. Source publication and curation filters can change recurrence prevalence, so all analyses remain stream-specific.

Each query's gallery contains every pair in its probe window; splits are paper- or post-disjoint, figure-hash deduplicated, and locked before training. Probe sizes range from 7936–8590, 3822–4176, and 4711–5032 pairs per window for SciCapStream, PMC, and RedCaps. ConceptDrift-5K assigns its 5,000 generated pairs before training to 2,000 stable, 1,500 gradually evolving, 1,000 abruptly replaced, and 500 oscillating concepts, following standard drift distinctions (Gama et al., 2014; Webb et al., 2016). Stable concepts retain their pairing, evolving concepts move continuously, replaced concepts switch once to a disjoint pairing, and oscillating concepts return after at least one complete absence window. Pair identities, assignments, ordered windows, return schedule, and probes are shared across methods and excluded from controller fitting. The re-entry denominator contains every pair buffered when an oscillating concept begins its absence. A success is such a pair, unsampled throughout the absence, selected in the first return window; eviction before return remains in the denominator as a failure.

Table 1: Fixed-objective replay selection across three streams. Cells are mean ± 95% CI over five matched seeds. Lower forgetting is better.

Stream	Selector	Future	Fgt.↓
SciCap	Recency	36.7±0.3	14.1±0.6
	Reservoir	36.2±0.4	12.0±0.5
	MIR-style	37.4±0.4	10.1±0.4
	GSS	37.1±0.4	10.3±0.5
	RPR	37.5±0.4	9.4±0.4
PMC	Recency	33.9±0.4	13.6±0.5
	Reservoir	33.5±0.5	12.4±0.6
	MIR-style	34.2±0.4	11.1±0.4
	GSS	34.0±0.5	11.3±0.5
	RPR	34.4±0.4	10.7±0.4
RedCaps	Recency	31.6±0.4	12.8±0.5
	MIR-style	32.0±0.4	10.4±0.4
	GSS	31.9±0.4	10.6±0.4
	RPR	32.1±0.4	10.1±0.4

All ViT-B/16 runs use AdamW, learning rate 10^{-5}, one 500-step warmup, batch size 512 with 256 new and 256 replay pairs, and $M = 50$K. New-pair budgets are 300K, 120K, and 150K per SciCap, PMC, and RedCaps window; the one-to-one replay ratio therefore fixes the corresponding optimizer-step budgets. ViT-L/14 uses the same buffer fraction, examples, replay ratio, and steps. The matched comparisons run on eight 80GB A100 GPUs under CUDA 12.4; selector-specific wall time and memory appear in Appendix G. Hyperparameters and approximation sizes are selected on SciCap-Pilot and then shared across reported streams.

Selector implementations. Reservoir uses standard stream-count admission and uniform retrieval. Recency assigns $r_i = (1 + \Delta t_i)^{-1}$, samples without replacement in proportion to r_i, and retains the M highest-r_i pairs after each window. Because every incoming window exceeds M, this storage rule is the literal strict-monotone policy: after each boundary its buffer contains the newest M incoming pairs. Reservoir therefore supplies the complementary age-agnostic persistence control. MIR-style draws a fresh 1,024-pair uniform candidate pool per update, takes a virtual gradient step on the 256 new pairs, scores each candidate by the increase in its symmetric CLIP loss, and replays the top 256; reservoir admission isolates retrieval from persistence. GSS computes gradients of the symmetric CLIP loss with respect to both final projection layers, admits incoming pairs with the lowest maximum cosine similarity to a 1,024-pair uniform memory candidate pool, evicts the most redundant candidate, and retrieves uniformly from the retained set. RPR scores the full buffer at boundaries and uses the sampling and capacity projection in Section 3.3. Every method sees identical new-data batches and seed-indexed minibatch order; all uniform pools are drawn from the same seed stream and all ties use the fixed content hash. Thus the strongest closure comes from MIR and GSS, while Recency measures the specific consequence of enforcing timestamp monotonicity in both retrieval and retention.

Timestamp interventions leave stream order, model updates, examples, and probes fixed. *Clean* exposes the original window index. *Coarse* maps adjacent annual windows to a shared two-year bin. *Noise* adds rounded i.i.d. $\mathcal{N}(0, 1)$ window offsets and clips to the observed range. *Shuffle* uniformly permutes controller-visible timestamps within the current buffer at every boundary. Each perturbation map is fixed within a seed and shared by RPR and Recency; MIR and GSS remain timestamp-agnostic.

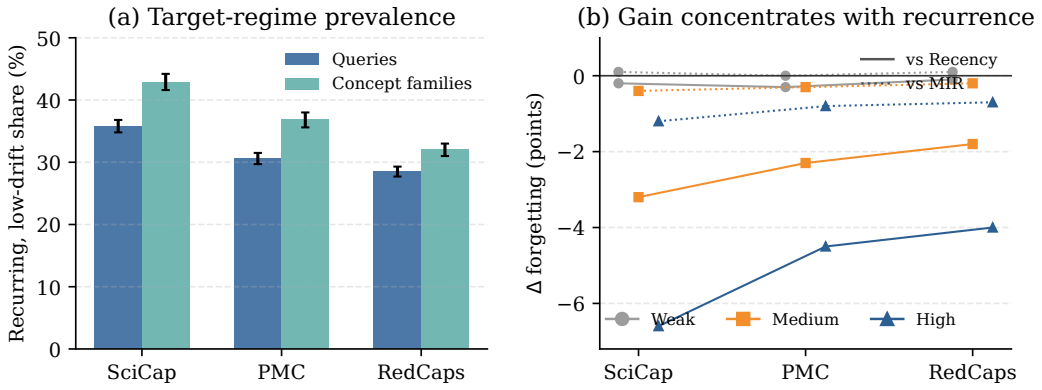

Figure 2: The RPR advantage tracks independently defined recurrence. (a) Future-query and family prevalence of recurring, low-displacement history. (b) RPR-minus-baseline forgetting by recurrence stratum; negative values favor RPR. The high-to-weak contraction repeats in all three streams. Counts, intervals, and construction details appear in Table 13.

5 Results

5.1 Cross-stream gains over adaptive selectors

The most demanding fixed-objective comparisons are the non-age selectors. Against MIR, RPR's paired forgetting effects are -0.7±0.5, -0.4±0.3, and -0.3±0.2 on SciCap, PMC, and RedCaps. Each comparison wins 4/5, 4/5, and 4/5 seeds; the Holm-adjusted p-values are 0.03, 0.04, and 0.04, respectively. Against GSS, the effects are -0.9±0.5, -0.6±0.4, and -0.5±0.3, with 5/5, 4/5, and 4/5 wins; the adjusted p-values are 0.02, 0.03, and 0.03. All six paired Future intervals satisfy the fixed non-inferiority margin (Appendix Table 12).

Strict timestamp ordering produces the larger failure: RPR reduces forgetting relative to Recency by -4.7±0.7, -2.9±0.6, and -2.7±0.5 points. The recurrence analyses below explain why the gap is largest in SciCap rather than treating it as a uniform selector advantage.

RPR also improves over the objective-changing DER++-CLIP comparator by -1.6±0.6, -0.9±0.4, and -0.7±0.4 forgetting points. Full retention anchors, denominators, loss values, adjusted tests, and paired Future effects are in Appendix C.

5.2 The gap localizes to renewable history

Recurrence is defined independently of RPR: a noun-lemma family appears in at least three windows and its consecutive CLIP-centroid displacement does not exceed 0.12. The rule covers 3012/8421 (35.8%) SciCap queries and 642/1496 (42.9%) families; the corresponding query/family shares are 30.6%/36.8% for PMC and 28.5%/32.0% for RedCaps. Per-window query prevalence ranges are 31.4–39.7%, 26.9–34.8%, and 23.7–32.6%.

Figure 2 connects prevalence to effect size. Against Recency, the SciCap effect grows from -0.2±0.5 in the weak stratum to -6.6±0.9 in the high stratum; the corresponding MIR effects are +0.1±0.3 and -1.2±0.5. PMC and RedCaps show the same contraction. Thus the aggregate gain is concentrated where old examples can plausibly regain value rather than spread uniformly across queries.

5.3 Temporal intervention isolates RPR's time dependence

The clean paired RPR−MIR effect is -0.7±0.5. Relative to that same MIR trajectory, the mean gap becomes 0.0 under coarsening, $+0.7$ under noise, and $+1.8$ under shuffling. RPR nevertheless remains 4.1 and 3.4 forgetting points ahead of Recency in the latter two conditions. Because the stream, model updates, and MIR randomness are invariant, this monotone erosion is attributable to

Table 2: SciCapStream timestamp intervention. Forgetting is mean ± 95% CI over 5 seeds. MIR is timestamp-agnostic, so one shared matched-seed MIR trajectory is the invariant reference in every row.

Condition	RPR	MIR-style	Recency
Clean	9.4±0.4	10.1±0.4	14.1±0.6
Coarse	10.1±0.4	10.1±0.4	14.5±0.6
Noise	10.8±0.5	10.1±0.4	14.9±0.7
Shuffle	11.9±0.5	10.1±0.4	15.3±0.7

Table 3: ViT-L/14 fixed-objective transfer over five matched seeds. The buffer fraction, update budget, and probes match ViT-B/16.

Stream	Selector	Future	Fgt.↓
SciCap	Recency	40.9±0.4	12.6±0.5
	MIR-style	41.7±0.4	8.9±0.4
	GSS	41.4±0.4	9.2±0.4
	RPR	41.9±0.4	8.2±0.3
RedCaps	Recency	35.8±0.4	11.5±0.5
	MIR-style	36.4±0.4	9.3±0.4
	GSS	36.2±0.4	9.5±0.4
	RPR	36.6±0.4	8.9±0.4

the information exposed to RPR's controller: its advantage over a non-age selector disappears as visible time loses structure. Future R@1 likewise falls from 37.5 to 36.0, whereas the shared MIR reference remains 37.4.

5.4 Returned concepts and backbone scale

The 428M parameter ViT-L/14 model preserves the operating signature without copying the ViT-B/16 margins. RPR−MIR forgetting is -0.7±0.4 on SciCap and -0.4±0.3 on RedCaps; RPR−GSS is -1.0±0.5 and -0.6±0.4. Appendix G reports seed wins, wall time, and peak memory.

ConceptDrift-5K records the survival-to-return transition that RPR is designed to improve. On oscillating concepts, the repeated-seed advantage over Recency is +14.7±1.8 R@1, and RPR raises returned-pair re-entry by +18.1±2.1 points; the canonical run also improves stable and replaced slices (Appendix Table 14). The component study then assigns incremental effects to the controller's two sides. Removing teacher disagreement or embedding migration raises forgetting from 9.4±0.4 to 11.2±0.5 and 10.6±0.4. Removing the explicit floor while retaining the renewal controller raises forgetting to 10.5±0.4 and lowers rare-stable R@1 from 31.0±0.8 to 24.3±1.1. Current-state signals determine renewed priority; the floor supplies the historical candidates on which that priority can act.

5.5 Mechanism-level synthesis

Figure 3 joins selector outcomes to the proposed mechanism. The RPR–Recency gap repeats across temporal resolutions, while margins over MIR and GSS show measurable room beyond their current-state selection strategies. Recurrence localization determines where the gap appears; ConceptDrift records return-time re-entry; timestamp corruption removes the advantage over invariant MIR; and floor removal isolates the floor's incremental benefit for rare-stable retrieval given the renewal controller. The weak-recurrence boundary supplies the converse. This response pattern supports the renewal–persistence explanation within the evaluated regime and distinguishes it from strict timestamp ordering and the evaluated interference- and diversity-based selectors.

Table 4: ConceptDrift-5K canonical run and five-seed summary. Canonical columns report one frozen run; five-seed columns report mean ± 95% CI. Retrieval rows are R@1; returned-pair re-entry is a percentage.

Slice	Rec. (1 run)	RPR (1 run)	Rec. (5 seeds)	RPR (5 seeds)
Overall	48.2	56.8	48.0±0.8	56.6±0.7
Oscillating	38.1	53.8	38.4±1.3	53.1±1.1
Returned-pair re-entry	6.5	25.7	6.8±1.0	24.9±1.8

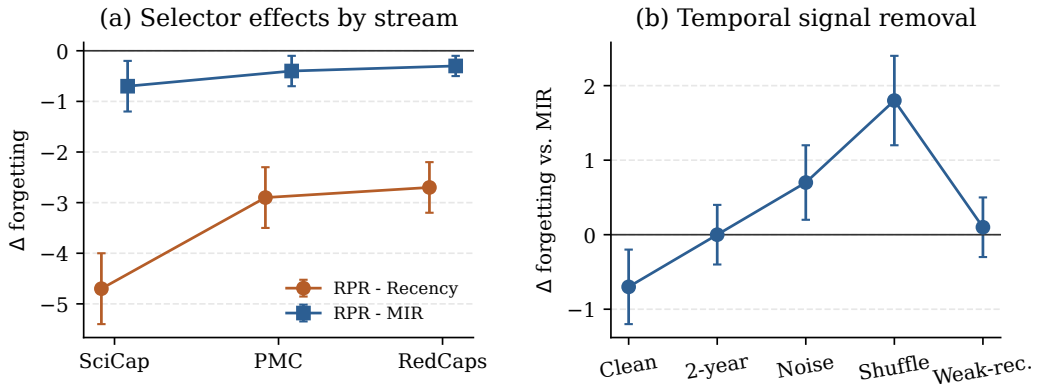

Figure 3: Mechanism-level synthesis. (a) RPR's large gap to Recency and smaller paired gap to MIR repeat across three real streams. (b) Against one invariant MIR reference, RPR's margin disappears as controller-visible time is corrupted and at the natural weak-recurrence boundary. Full results appear in Tables 1, 9, and 19.

6 Operating Regime and Compute Trade-off

RPR improves retention at the shared gradient-step budget with explicit controller overhead. On SciCapStream, it uses 49.9 total hours and 50.2 peak GB versus MIR's 42.7 hours and 48.6 GB, while moving forgetting from 10.1±0.4 to 9.4±0.4 (Appendix H). The additional 16.9% wall time comes from the one-time fit and boundary rescoring rather than extra learner updates. The resulting deployment criterion is concrete: use RPR when recurring history makes the retention improvement worth boundary-time computation; MIR remains the lower-cost point when that margin is not operationally valuable.

The same experiments define the operating boundary in one balanced view. On 1187 SciCap probes from 274 weak-recurrence or high-displacement families, RPR, MIR, and Recency record 14.0±0.4, 13.9±0.4, and 14.2±0.5 forgetting; the paired RPR–MIR effect is +0.1±0.4. The evidence supports caption-style image–text streams with meaningful annual-to-monthly timestamps and recurring, low-displacement concepts. In this regime, five-seed cross-stream transfer supports the frozen ranker as an effective controller, and the neutral boundary marks where protected renewal no longer adds leverage.

7 Conclusion

Returning multimodal concepts make replay a reachability problem as well as a ranking problem. RPR couples current-state renewal to finite-buffer persistence and improves over MIR and GSS across SciCap, PMC, and RedCaps under the shared contrastive objective; on SciCap it lowers forgetting from 14.1±0.6 under strict timestamp ordering to 9.4±0.4 while preserving future retrieval. The same ordering at ViT-L/14, concentration in recurring concepts, return-time re-entry, timestamp intervention, and neutral weak-recurrence boundary jointly support one practical conclusion: preserve historical candidates that can regain value, and rescore them as the representation evolves.

LANGUAGE-MODEL USE DISCLOSURE

The manuscript-preparation record documents use of generative AI for drafting, revising, and presenting manuscript text, tables, and figures. It does not document use of generative AI to generate experimental data, implement methods, or produce analysis results. Authors remain responsible for verifying the final submission and its claims.

AI USE STATEMENT

Generative AI tools supported language polishing and presentation refinement in this manuscript. The authors checked the edited text and remain responsible for all claims and the final submission.

REFERENCES

Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. In *NeurIPS*, 2019a.

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

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, 2009.

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

Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co2L: Contrastive continual learning. In *ICCV*, 2021.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In *ICLR*, 2019a.

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. *arXiv preprint arXiv:1902.10486*, 2019b.

Ziyang Chen, Shaoguang Wang, Weiyu Guo, Qianyi Cai, He Zhang, Pengteng Li, Yiren Zhao, and Yandong Guo. PHASER: Phase-aware and semantic experience replay for vision-language-action models. *arXiv preprint arXiv:2606.03598*, 2026. URL https://arxiv.org/abs/2606.03598.

Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. In *ICLR*, 2020.

Karan Desai, Gaurav Kaul, Zubin Aysola, and Justin Johnson. RedCaps: Web-curated image-text data created by the people, for the people. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/e00da03b685a0dd18fb6a08af0923de0-Abstract-round1.html.

Arthur Douillard, Alexandre Rame, Guillaume Couairon, and Matthieu Cord. DyTox: Transformers for continual learning with DYnamic TOken expansion. In *CVPR*, 2022.

Emanuele Frascaroli, Aniello Panariello, Pietro Buzzega, Lorenzo Bonicelli, Angelo Porrello, and Simone Calderara. CLIP with generative latent replay: A strong baseline for incremental learning. In *35th British Machine Vision Conference*. BMVA, 2024. URL https://bmvc2024.org/proceedings/863/.

Joao Gama, Indre Zliobaite, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):1–37, 2014.

Amirata Ghorbani and James Zou. Data Shapley: Equitable valuation of data for machine learning. In *ICML*, 2019.

Md Yousuf Harun, Jhair Gallardo, Junyu Chen, and Christopher Kanan. GRASP: A rehearsal policy for efficient online continual learning. In *Proceedings of the 3rd Conference on Lifelong Learning Agents*, volume 274 of *Proceedings of Machine Learning Research*, pp. 1032–1052. PMLR, 2025. URL https://proceedings.mlr.press/v274/harun25a.html.

Truman Hickok and Dhireesha Kudithipudi. Watch your step: Optimal retrieval for continual learning at scale. *arXiv preprint arXiv:2404.10758*, 2024. URL https://arxiv.org/abs/2404.10758.

Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats Leon Richter, Quentin Gregory Anthony, Eugene Belilovsky, Timothée Lesort, and Irina Rish. Simple and scalable strategies to continually pre-train large language models. *Transactions on Machine Learning Research*, 2024. URL https://tmlr.infinite-conf.org/paper_pages/DimPeeCxKO.html.

Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Predicting predictions from training data. In *ICML*, 2022.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. GLISTER: Generalization based data subset selection for efficient and robust learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(9):8110–8118, 2021. doi: 10.1609/aaai.v35i9.16988. URL https://ojs.aaai.org/index.php/AAAI/article/view/16988.

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. doi: 10.1073/pnas.1611835114. URL https://www.pnas.org/doi/10.1073/pnas.1611835114.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *ICML*, 2017.

Zhizhong Li and Derek Hoiem. Learning without forgetting. In *ECCV*, 2016.

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

Soren Mindermann, Jan M. Brauner, Muhammed T. Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Holtgen, Aidan N. Gomez, Adrien Morisot, Sebastian Farquhar, and Yarin Gal. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *ICML*, 2022.

Zixuan Ni, Longhui Wei, Siliang Tang, Yueting Zhuang, and Qi Tian. Continual vision-language representation learning with off-diagonal information. In *ICML*, 2023.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. In *NeurIPS*, 2021.

Ameya Prabhu, Philip H. S. Torr, and Puneet K. Dokania. GDumb: A simple approach that questions our progress in continual learning. In *ECCV*, 2020.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 8748–8763, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

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

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Rajat Saxena, Justin L. Shobe, and Bruce L. McNaughton. Learning in deep neural networks and brains with similarity-weighted interleaved learning. *Proceedings of the National Academy of Sciences*, 119(27):e2115229119, 2022. doi: 10.1073/pnas.2115229119. URL https://www.pnas.org/doi/10.1073/pnas.2115229119.

Dongsub Shim, Zheda Mai, Jihwan Jeong, Scott Sanner, Hyunwoo Kim, and Jongseong Jang. Online class-incremental continual learning with adversarial shapley value. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 9630–9638, 2021. doi: 10.1609/aaai.v35i11.17159. URL https://ojs.aaai.org/index.php/AAAI/article/view/17159.

James Seale Smith, Lazar Valkov, Shaunak Halbe, Vyshnavi Gutta, Rogerio Feris, Zsolt Kira, and Leonid Karlinsky. Adaptive memory replay for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 3605–3615, 2024. doi: 10.1109/CVPRW63382.2024.00364. URL https://openaccess.thecvf.com/content/CVPR2024W/ELVM/html/Smith_Adaptive_Memory_Replay_for_Continual_Learning_CVPRW_2024_paper.html.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S. Morcos. Beyond neural scaling laws: Beating power law scaling via data pruning. In *NeurIPS*, 2022.

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. In *ICLR*, 2019.

Jeffrey S. Vitter. Random sampling with a reservoir. *ACM TOMS*, 11(1):37–57, 1985.

Geoffrey I. Webb, Roy Hyde, Hong Cao, Hai Long Nguyen, and Francois Petitjean. Characterizing concept drift. *Data Mining and Knowledge Discovery*, 30(4):964–994, 2016.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. DoReMi: Optimizing data mixtures speeds up language model pretraining. In *NeurIPS*, 2023.

Zangwei Zheng, Mingyuan Ma, Kai Wang, Ziheng Qin, Xiangyu Yue, and Yang You. Preventing zero-shot transfer degradation in continual learning of vision-language models. In *ICCV*, 2023.

Heng Zhou, Jing Tang, Jusheng Zhang, Yanshu Li, Canran Xiao, Liwei Hou, Zong Ke, and Jiawei Yao. CoMem: Compositional concept-graph memory for vision–language adaptation. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=xp7wDU9JBW.

A Protocol and Dataset Details

Table 5: Evaluation construction. Counts denote paired queries and gallery items in each bidirectional retrieval direction.

Stream	Training windows	Pairs/window	Held-out/window	Future
SciCapStream	2018–2024 annual	300K	7936–8590	2025: 8421
PMC-Quarterly	2021Q1–2022Q4	120K	3822–4176	2023Q1: 4176
RedCaps-Monthly	Jan.–Dec. 2020	150K	4711–5032	Jan. 2021: 5032

Figure hashes prevent duplicate scientific images from crossing splits, and one-paper locking prevents captions from the same article entering both training and evaluation. RedCaps uses post identifiers for the analogous lock. Future probes are excluded from every fit, anchor bank, buffer, and hyperparameter decision. Retrieval candidates are complete within each probe, not sampled negatives.

Recurrence construction. Captions are lowercased and lemmatized with a frozen part-of-speech pipeline. Common and proper noun heads are retained; contiguous noun compounds are matched longest-first and exact lemma sequences define families. No manual synonym or polysemy merging is performed. A query with multiple families is assigned to its most frequent training-stream family, with the content hash breaking ties. For family g in window t, the centroid is the normalized mean of its CLIP pair states, and consecutive displacement is cosine distance between available centroids. The three-window, 0.12 high-recurrence rule and 0.25 high-displacement boundary were fixed on SciCap-Pilot before evaluation and transferred unchanged. Table 13 reports future-query denominators; stratum forgetting uses all historical probes assigned by the same rule rather than treating those future counts as the forgetting denominator.

B Prior-Work Distinction Matrix

Table 6: Operator-level distinction from representative replay lines. "Fixed" denotes an unchanged task loss within the cited method. Recency, Reservoir, GSS, MIR, DER++, and RPR are empirical comparators in our CLIP protocol.

Line	Fixed	Raw	Renew	Persist.	Time	Return
Recency	yes	yes	age only	newest	no	no
Reservoir	yes	yes	no	random	no	no
GSS	yes	yes	gradient	diversity	no	no
MIR	yes	yes	interference	reservoir	no	no
DER++	no	yes	logits	replay	no	no
ASER	yes	yes	Shapley	Shapley	no	no
GRASP	yes	yes	prototype	fixed buffer	no	no
SWIL	yes	yes	similarity	retained data	no	no
Adaptive Memory Replay	yes	yes	bandit	full history	no	no
Watch Your Step	yes	yes	selective	fixed buffer	no	no
CoMem	no	no	graph	graph	no	no
PHASER	yes	VLA frames	phase risk	phase quota	no	no
RPR	yes	yes	staleness	floor	yes	yes

C Full Real-Stream and Intervention Results

The tables below expand the primary comparison into canonical retention anchors, repeated-seed losses, and the complete time intervention under the definitions in Sections 3–4.

Table 7: Canonical SciCapStream run under the matched online budget ($M = 50$K). Retention and future columns are R@1; forgetting averages 2018–2023.

Method	Buf.	Ret.18	Ret.20	Ret.22	Fut.25	Loss	Fgt.↓
Joint (offline)	All	42.1	44.8	47.2	38.4	2.31	N/A
Sequential	0	21.3	28.4	38.1	34.8	2.89	24.1
Uniform Replay	50K	34.2	38.1	43.8	35.9	2.52	12.8
Reservoir	50K	34.8	38.4	44.1	36.1	2.49	12.1
Recency	50K	31.8	37.2	44.7	36.8	2.54	14.3
DER++-CLIP	50K	35.6	39.2	44.6	36.4	2.46	11.2
RPR	50K	37.4	40.8	45.9	37.6	2.41	9.2
RPR (no floor)	50K	36.1	39.8	45.4	37.2	2.44	10.4

SciCapStream uses 8421 future queries and six forgetting windows. RPR−MIR yields ΔFgt. -0.7±0.5 ($p = 0.03$; 4/5 wins) and Future +0.1±0.3. RPR−DER++-CLIP gives ΔFgt. -1.6±0.6 ($p = 0.01$; 5/5) and Future +0.9±0.4. Ret.19/21/23 are 38.5/42.6/46.3 for RPR, 38.1/42.0/45.9 for MIR, and 37.4/41.1/45.2 for DER++-CLIP.

Table 8: Full repeated-seed SciCapStream comparison. Selector-only methods use the identical CLIP loss; DER++-CLIP adds logit replay. Cells are mean ± 95% CI.

Method	Fut.25	Loss	Fgt.↓
Recency	36.7±0.3	2.55±0.03	14.1±0.6
Reservoir	36.2±0.4	2.50±0.03	12.0±0.5
GSS	37.1±0.4	2.46±0.03	10.3±0.5
DER++-CLIP	36.6±0.4	2.47±0.03	11.0±0.5
MIR-style	37.4±0.4	2.45±0.03	10.1±0.4
RPR (no floor)	37.2±0.4	2.44±0.03	10.5±0.4
RPR	37.5±0.4	2.42±0.03	9.4±0.4

Table 9: Full SciCapStream timestamp-validity intervention. Stream order and optimization are fixed; only controller-visible time changes. The timestamp-agnostic MIR run is shared across rows. Mean ± 95% CI.

Condition	RPR	MIR-style	Recency
Forgetting ↓			
Clean	9.4±0.4	10.1±0.4	14.1±0.6
Coarse	10.1±0.4	10.1±0.4	14.5±0.6
Noise	10.8±0.5	10.1±0.4	14.9±0.7
Shuffle	11.9±0.5	10.1±0.4	15.3±0.7
Fut.25			
Clean	37.5±0.4	37.4±0.4	36.7±0.3
Coarse	37.1±0.4	37.4±0.4	36.4±0.4
Noise	36.6±0.5	37.4±0.4	36.2±0.4
Shuffle	36.0±0.5	37.4±0.4	35.9±0.4

Sharing one MIR trajectory across conditions makes the intervention literal: only controller-visible time for RPR and Recency changes. RPR's forgetting and Future scores degrade together as timestamps lose structure.

Table 10: PMC-Quarterly replication. Retention anchors are mean R@1; Future and forgetting are mean ± 95% CI.

Method	Ret.Q1	Ret.Q4	Ret.Q7	Future	Fgt.↓
Recency	28.9	35.8	39.7	33.9±0.4	13.6±0.5
Reservoir	30.4	36.3	39.1	33.5±0.5	12.4±0.6
GSS	31.6	37.0	39.9	34.0±0.5	11.3±0.5
DER++-CLIP	31.5	36.9	39.8	34.0±0.4	11.6±0.5
MIR-style	31.8	37.2	40.0	34.2±0.4	11.1±0.4
RPR	32.3	37.8	40.4	34.4±0.4	10.7±0.4

PMC has seven forgetting windows. RPR−MIR gives ΔFgt. -0.4±0.3 ($p = 0.04$; 4/5) and Future +0.2±0.3. RPR−DER++-CLIP gives ΔFgt. -0.9±0.4 and Future +0.4±0.3.

RedCaps uses eleven forgetting windows. RPR−DER++-CLIP gives ΔFgt. -0.7±0.4 ($p = 0.03$; 4/5) and Future +0.3±0.3 ($p = 0.08$; 4/5).

Table 11: RedCaps-Monthly scope transfer. Retention anchors are mean R@1; Future and forgetting are mean ± 95% CI.

Method	Ret.M1	Ret.M6	Ret.M11	Future	Fgt.↓
Recency	26.9	32.8	35.0	31.6±0.4	12.8±0.5
GSS	29.4	34.2	36.1	31.9±0.4	10.6±0.4
DER++-CLIP	29.2	34.0	36.0	31.8±0.4	10.8±0.4
MIR-style	29.6	34.4	36.2	32.0±0.4	10.4±0.4
RPR	29.9	34.6	36.5	32.1±0.4	10.1±0.4

Table 12: Paired Future R@1 effects for same-objective selector closure. Entries are RPR minus comparator with paired 95% intervals over five matched seeds; every lower endpoint exceeds the fixed -0.5-point margin.

Stream	Δ vs MIR	Δ vs GSS
SciCap	+0.1±0.3	+0.4±0.3
PMC	+0.2±0.3	+0.4±0.4
RedCaps	+0.1±0.3	+0.2±0.3

D Regime Prevalence and Stratified Effects

Strata are assigned without any RPR score or cluster. Future-query counts quantify prevalence; forgetting effects use complete historical probe sets carrying the same family assignments.

Table 13: Forgetting effects by method-independent recurrence stratum. Counts are queries; effects are RPR minus baseline with paired 95% intervals.

Stream	Stratum	Queries	Δ vs Recency	Δ vs MIR
SciCap	Weak	2785	-0.2±0.5	+0.1±0.3
	Medium	2624	-3.2±0.7	-0.4±0.3
	High	3012	-6.6±0.9	-1.2±0.5
PMC	Weak	1457	-0.3±0.5	0.0±0.3
	Medium	1441	-2.3±0.6	-0.3±0.3
	High	1278	-4.5±0.8	-0.8±0.4
RedCaps	Weak	1874	-0.1±0.4	+0.1±0.3
	Medium	1724	-1.8±0.6	-0.2±0.2
	High	1434	-4.0±0.7	-0.7±0.3

Weak contains single-occurrence families or displacement above 0.25; high satisfies the three-window, displacement-at-most-0.12 rule; medium contains the remainder. The concentration is substantial but not identical across temporal granularities, matching the stream-specific prevalence in Figure 2.

E Mechanism and Robustness

ConceptDrift-5K isolates return behavior, component removals locate the renewal and persistence contributions, and independent slices test survival without using RPR's own cluster rule. Its 5,000 fixed pair identities are partitioned into 2,000 stable, 1,500 gradually evolving, 1,000 abruptly replaced, and 500 oscillating cases. Every selector receives the same ordered windows and probes. Oscillating concepts disappear for at least one complete window and then restore their original pairing; the re-entry endpoint follows pairs present at absence onset through the first return window, counting pre-return eviction as failure.

Table 14: ConceptDrift-5K canonical frozen run. This body is a single run; Table 4 reports the separately registered five-seed means and intervals.

Method	Overall	Stable	Evolving	Replaced	Osc.
Recency	48.2	52.1	47.8	51.4	38.1
Reservoir	51.4	54.8	50.2	48.3	42.7
RPR	56.8	58.4	55.1	57.2	53.8

Table 15: Canonical-run component counterfactual on SciCapStream. The rare-stable slice requires three-window cluster persistence and quota-free replay share below $q_{\min}$.

Variant	Fgt.↓	Fut.25	Rare-stable R@1
RPR (full)	9.2	37.6	31.2
− Teacher disagreement	11.1	36.8	29.4
− Embedding migration	10.4	37.1	30.1
− Stable anchors	10.8	37.0	27.8
− Cluster floors	10.4	37.2	24.1
− Drift-adaptive β	9.8	37.3	30.4
Recency only	14.3	36.8	22.8

Table 16: Five-seed component robustness. Cells are mean ± 95% CI and are distinct from the canonical run in Table 15.

Variant	Fgt.↓	Fut.25	Rare-stable R@1
RPR	9.4±0.4	37.5±0.4	31.0±0.8
− Teacher disagreement	11.2±0.5	36.9±0.4	29.1±0.9
− Embedding migration	10.6±0.4	37.1±0.4	29.8±0.8
− Stable anchors	10.9±0.5	37.0±0.4	27.6±1.0
− Cluster floors	10.5±0.4	37.2±0.4	24.3±1.1
− Drift-adaptive β	9.9±0.4	37.3±0.4	30.2±0.8
Recency	14.1±0.6	36.7±0.3	22.9±1.2

The repeated-seed removals separate roles: disagreement and migration primarily affect renewed priority, whereas removing floors produces the largest rare-stable loss. The next two tables replace that mechanism-aligned slice with timestamp-agnostic and independently defined recurrence slices.

Table 17: Timestamp-agnostic low-drift subset of SciCapStream's high-recurrence stratum. A separately fixed cross-window gallery-consistency rule retains 1736 probes with centroid drift below 0.12. Mean ± 95% CI.

Method	Slice R@1	Slice Fgt.↓
Recency	24.7±0.8	15.0±0.7
DER++-CLIP	28.6±0.7	12.2±0.6
MIR-style	29.0±0.7	11.8±0.5
RPR	29.7±0.6	11.1±0.5

Table 18: Independent multimodal-recurrence survival evidence. Slice definitions exclude RPR clusters, replay shares, and floor thresholds. Mean ± 95% CI.

Stream	Method	Slice R@1	Slice Fgt.↓	Future
SciCap	DER++-CLIP	28.8±0.7	12.0±0.6	37.1±0.4
	MIR-style	29.3±0.6	11.6±0.5	37.3±0.4
	RPR (no floor)	29.0±0.7	11.4±0.5	37.2±0.4
	RPR	30.1±0.6	10.8±0.5	37.5±0.4
PMC	DER++-CLIP	26.6±0.8	12.0±0.6	34.0±0.4
	MIR-style	27.1±0.7	11.7±0.5	34.1±0.4
	RPR (no floor)	26.9±0.7	11.8±0.5	34.2±0.4
	RPR	27.8±0.7	11.2±0.5	34.4±0.4

On both independent definitions, full RPR improves retention and forgetting while preserving Future; the no-floor variant loses part of that advantage. The weak-recurrence table below supplies the complementary null case.

Table 19: Natural SciCapStream boundary case: 1187 probes across 274 weak-recurrence or high-displacement families. Mean ± 95% CI.

Method	Slice R@1	Slice Fut.25	Slice Fgt.↓
Recency	22.1±0.5	33.0±0.4	14.2±0.5
DER++-CLIP	22.4±0.5	33.0±0.4	14.0±0.5
MIR-style	22.6±0.5	33.1±0.4	13.9±0.4
RPR	22.5±0.5	33.0±0.4	14.0±0.4

The boundary slice collapses the RPR–MIR difference to +0.1±0.4, matching the prediction that protected renewal has little leverage without recurring low-displacement history.

Table 20: Controller-fit and protocol sensitivity on SciCapStream. ΔFgt. is RPR minus matched MIR; negative values favor RPR. Mean ± 95% CI over five seeds.

Setting	RPR Fgt.↓	RPR Fut.25	ΔFgt. vs MIR
Alt. fit A	9.6±0.4	37.4±0.4	-0.5±0.3
Alt. fit B	9.5±0.4	37.6±0.4	-0.6±0.4
$M = 25$K	10.2±0.5	37.0±0.4	-0.4±0.3
$M = 100$K	8.9±0.4	37.7±0.4	-0.8±0.4
$k = 50$	9.7±0.4	37.3±0.4	-0.4±0.3
$k = 200$	9.3±0.4	37.6±0.4	-0.6±0.4
$q_{\min} = 0.05$	9.8±0.4	37.4±0.4	-0.3±0.3
$q_{\min} = 0.15$	9.5±0.4	37.5±0.4	-0.5±0.3
Anchor tenure 2 windows	9.8±0.4	37.3±0.4	-0.4±0.3
Anchor tenure 4 windows	9.6±0.4	37.4±0.4	-0.5±0.3
$\tau_s = 0.03$	9.7±0.4	37.4±0.4	-0.4±0.3
$\tau_s = 0.07$	9.5±0.4	37.5±0.4	-0.6±0.4
$\beta_0 = 0.997$	9.7±0.4	37.3±0.4	-0.4±0.3
$\beta_0 = 0.9995$	9.5±0.4	37.5±0.4	-0.6±0.4
$\alpha = 0.25$	9.6±0.4	37.4±0.4	-0.5±0.3
$\alpha = 0.75$	9.7±0.4	37.3±0.4	-0.4±0.3

Alternative held-out fits, buffer sizes, cluster counts, floor mass, anchor definitions, and teacher schedules all preserve the RPR–MIR forgetting sign. Alt. fit A and B rotate which of the three complete pilot transitions is held out under the same two-transition/one-transition protocol. Magnitudes vary most with buffer capacity and stable-anchor tenure, consistent with persistence changing the set on which renewal can act.

F Frozen-Fit Diagnostics

The fit diagnostics quantify the low-capacity ranker before it is transferred unchanged to every reported stream and backbone.

Table 21: Frozen harmful-staleness fit on the disjoint pilot stream. Coefficients use standardized inputs; intervals are candidate-cluster-bootstrap 95% intervals.

Quantity	Value
Pilot image–text pairs	418320
Adjacent transitions / candidate exchanges	3 / 2400
Unique candidates / contexts per candidate	600 / 4
Exchange labels: harmful / other	914 / 1486
w_d (teacher disagreement)	-0.74±0.10
w_m (embedding migration)	-0.43±0.09
w_a (missing anchor support)	+0.61±0.08
Validation AUROC / AUPRC	0.742±0.018 / 0.687±0.021
Expected calibration error	0.036±0.007

Each of 600 candidates is evaluated in 4 matched co-replay contexts, producing 2400 exchange-level labels: 914 harmful and 1486 beneficial or neutral. The two-transition fit and one-transition validation split keeps all contexts for a candidate together. Validation intervals resample candidate identities on the held-out transition and carry all four contexts with each sampled identity. The fit is frozen immediately after pilot validation, and every reported stream split, perturbation, and selector comparison occurs afterward.

G Backbone Transfer and Quality–Cost Results

Backbone transfer keeps examples and update counts fixed, while the cost columns expose the resources required by each selector on matched hardware.

Table 22: ViT-L/14 transfer with matched A100 hardware. Time is wall-clock hours per window; memory is peak per GPU in GB.

Stream	Selector	Future	Fgt.↓	Hours	Memory
SciCap	Recency	40.9±0.4	12.6±0.5	10.9	69.4
	MIR-style	41.7±0.4	8.9±0.4	11.6	71.2
	GSS	41.4±0.4	9.2±0.4	11.4	70.7
	RPR	41.9±0.4	8.2±0.3	12.1	73.8
RedCaps	Recency	35.8±0.4	11.5±0.5	5.7	69.2
	MIR-style	36.4±0.4	9.3±0.4	6.1	71.0
	GSS	36.2±0.4	9.5±0.4	6.0	70.6
	RPR	36.6±0.4	8.9±0.4	6.4	73.5

On SciCap, RPR beats MIR and GSS on forgetting in 4/5 and 5/5 seeds; on RedCaps the counts are 4/5 and 4/5. Future intervals satisfy the same non-inferiority criterion.

H ViT-B/16 Efficiency

This ledger measures selector overhead separately from the matched gradient-step budget used for the primary comparison.

Table 23: Matched-hardware ViT-B/16 quality–cost ledger on SciCapStream. Throughput, per-window time, and memory are means ± 95% CI; total hours include all windows, and RPR includes six rescoring boundaries plus its one-time fit.

Selector	Pairs/s	Hours/window	Peak GB	Total hours
Recency	14.4±0.3	5.8±0.12	47.1±0.5	40.6
MIR-style	13.7±0.3	6.1±0.15	48.6±0.6	42.7
GSS	13.9±0.3	6.0±0.14	48.1±0.5	42.0
RPR	13.0±0.3	6.4±0.17	50.2±0.6	49.9

RPR adds 18 minutes per boundary and a one-time 26 GPU-hour fit. Its 16.9% full-stream wall-time increase over MIR accompanies a 0.7-point forgetting reduction at the matched update count, exposing the quality–cost choice directly.

I RPR Update Order

Algorithm 1 makes the state boundary explicit: pre-update scores determine replay, post-update scores determine the next buffer, and anchors refresh only after capacity is committed.

Algorithm 1 Renewal–Persistence Replay

Require: Stream $\{W_1, \ldots, W_T\}$, buffer $\mathcal{B}$, frozen harmful-staleness fit, EMA teacher

Initialize f_θ on W_1; index the latest 1,000 buffered pairs as recent anchors; leave the stable index empty
for $t = 2, \ldots, T$ **do**
  form normalized pair states; standardize features and compute $h(x_i, t)$ for every $x_i \in \mathcal{B}$
  run seeded k-means to convergence; compute S_c, π_c, ω_i, and feasible floors ℓ_c
  for each batch, sample a cluster by π_c and 256 items without replacement by $1 - h_i$
  train on the 256 new and 256 sampled replay pairs
  after each optimizer step, update $\theta_{\mathrm{EMA}} \leftarrow \beta_t \theta_{\mathrm{EMA}} + (1 - \beta_t)\theta$
  score and recluster the retained–incoming union under the updated model
  recompute h, S_c, π_c, ω_i, and feasible floors; redistribute floor shortfalls
  evict the lowest-ω over-floor item until $|\mathcal{B}| = M$, breaking ties by content hash
  refresh anchor states from buffered pairs; drop evicted references and admit three-window survivors
end for

The two scoring passes serve different decisions and never compare candidates encoded by different model states. This ordering is also used for every ablation; removing a component changes only the named score or floor.