WHEN COMPLETE BECOMES PARTIAL: MULTI-LABEL ACTIVE LEARNING UNDER LABEL EXPANSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-label active learning aims to train accurate classifiers with less labeling effort by selectively querying instances that might be associated with multiple labels. However, previous studies often assume a fixed set of target categories, without considering the introduction of new categories during annotation. When new categories are introduced, annotations that were complete under the previous label set become partial, leaving unverified labels on historical instances. Therefore, learning under this expansion requires deciding both which unqueried instances to annotate and which historical instances to revisit within a limited query budget. To this end, we introduce Retrospective Multi-Label Active Learning (ReMuAL), a framework that combines first-time queries with selective revisits in a collaborative fashion. Specifically, first-time query feedback updates the model and label relationships, which are then used with verified historical labels to select revisits. Empirically, ReMuAL achieves gains up to 4.67 and 6.45 percentage points in Checklist mAP and full-trajectory AULC respectively over multiple benchmarks.

1 INTRODUCTION

In multi-label classification, a model predicts the set of labels associated with an instance rather than assigning a single class label. Training such a model with full supervision requires annotations specifying whether each target label is relevant to each training instance. However, collecting these annotations at scale can be costly and time-consuming (Durand et al., 2019). Multi-label active learning (MLAL) aims to reduce this annotation burden by selecting informative instances for annotation (Huang et al., 2015; Yan & Huang, 2018). In each round, the model guides instance selection and is then updated with the resulting annotations. For each instance, these annotations identify the relevant and irrelevant labels within the queried set.

Conventional MLAL assumes that the target label set is fixed across rounds. In practice, continued data exploration may expand this set and turn previously complete annotations into partial ones. For example, an image queried for *person* and *bicycle* retains those verified judgments after *helmet* is introduced, but its helmet label remains unverified (Fig. 1). Such gaps cannot be addressed by typical MLAL protocols that query only previously unqueried instances.

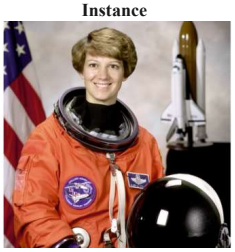


Photo: NASA, via scikit-image.

Recorded annotations

	(a) Before expansion *After the initial query*	(b) After expansion *Before any revisit*
Target set	{person, bicycle}	{person, bicycle, helmet}
person	+ Verified positive	+ Verified positive
bicycle	− Verified negative	− Verified negative
helmet	N/A Not in target set	→ ? In target set; unqueried
Coverage	**Complete**	**Partial**

Label expansion adds no new annotation; "?" means unqueried, not negative.

Figure 1: Illustrative label expansion from complete (a) to partial (b) annotations. Photo: NASA, via scikit-image.

To address the annotation gaps described above, we aim to identify positive examples of new labels among unqueried instances and unverified positives for earlier labels among previously queried instances. This leads us to a key question: how can we select such instances without complete ground-truth annotations? A line of work in MLAL uses predictive uncertainty and label correlations (Hua & Qi, 2008; Yu et al., 2024), or expected score improvements (Tan et al., 2024b; Qi et al., 2025) to assess the informativeness of candidate queries. However, identifying such instances remains challenging since these informativeness measures do not directly estimate the likelihood

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

of uncovering unverified positive labels. To address this problem, we focus on selecting unqueried instances likely to provide positive examples of new labels and historical instances likely to reveal unverified positives for earlier labels in this work.

In this paper, we introduce a retrospective MLAL paradigm and corresponding framework ReMuAL, which coordinates first-time queries and selective revisits under a shared budget. First-time queries cover the current label set and update both the classifier and label relationships, while these updates, together with each historical instance's verified labels, determine which incomplete annotations to revisit. ReMuAL preserves earlier judgments and acquires additional supervision only through explicit queries. Empirically, ReMuAL outperforms the strongest evaluated baseline by up to 4.67 and 6.45 percentage points in final Checklist mAP and full-trajectory AULC on multiple benchmarks.

2 Related Work

Multi-Label Active Learning. MLAL query selection commonly uses predictive uncertainty or expected loss/score reduction (Roy & McCallum, 2001; Yang et al., 2009; Hung & Lin, 2011; Tan et al., 2021; 2024a;b; Qi et al., 2025). Early approaches aggregate label-wise confidence (Brinker, 2006; Esuli & Sebastiani, 2009) or positive–negative margins (Li & Guo, 2013). More recent work broadens the acquisition signal beyond label-wise confidence (Vasisht et al., 2014; Ye et al., 2015; Reyes et al., 2018; Shi & Yu, 2019; Shi et al., 2021; Möllenbrok et al., 2023). EMM (Yu et al., 2024) models label correlations with Bernoulli mixtures and evidential uncertainty, BESRA (Tan et al., 2024b) combines Beta-score-based expected reduction with batch diversity, and CRAB (Qi et al., 2025) incorporates positive and negative label relationships into acquisition. To conclude, these criteria determine which instances are informative.

Query Types and Re-Annotation. Many MLAL methods annotate all target labels for a selected instance (Li & Guo, 2013; Huang et al., 2015), whereas others query individual instance–label pairs (Yan & Huang, 2018; Wu et al., 2014; Yu et al., 2022; Gui et al., 2021) or relative relevance between labels (Huang et al., 2015; Hu et al., 2019). For further annotation of labeled instances, Hua & Qi (2008) retain unannotated instance–label pairs, whereas ActiveLab (Goh & Mueller, 2023) relabels examples to improve label consensus (Lin et al., 2016). In our setting, a first-time query is complete for the current label set but becomes partial after label expansion; a revisit queries only labels that remain unverified while preserving earlier judgments.

Learning with Incomplete and Evolving Labels. Methods for incomplete multi-label supervision (Cole et al., 2021) recover missing labels from curriculum predictions (Durand et al., 2019), known-label correlations within and across images (Chen et al., 2022; 2024), or label-to-label semantic priors (Ding et al., 2023). For evolving label spaces (Abdelsalam et al., 2021; Lee et al., 2023; Muralidhara et al., 2024), DSLL (Wang et al., 2020) transfers knowledge from earlier labels and models, while LECO (Lin et al., 2022) uses old labels as partial supervision under class refinement. Unlike these approaches, we treat missing supervision as an acquisition problem: new labels accumulate without redefining earlier categories, and queries may target new or historical instances.

3 Problem Formulation

We consider multi-label active learning over a fixed image pool whose target label set expands over time. A comparison with conventional MLAL and category-discovery settings is provided in Fig. 2.

Task and Label Expansion. We consider T rounds of multi-label active learning over a fixed data pool $\mathcal{X} = \{x_i\}_{i=1}^{N}$, where the annotation process is extended to additional categories over time. Starting from an initial category set S_0, let ΔS_t denote the new categories introduced by the annotation protocol at the beginning of round t, the cumulative category set at round t is defined as:

$$S_t = S_{t-1} \cup \Delta S_t, \qquad \Delta S_t \cap S_{t-1} = \varnothing.$$

We allow $\Delta S_t = \varnothing$ in rounds without label expansion and retain the definitions of all previous categories. Introducing a category specifies an additional annotation target but does not reveal its relevance to any individual instance, since this supervision must be acquired through queries.

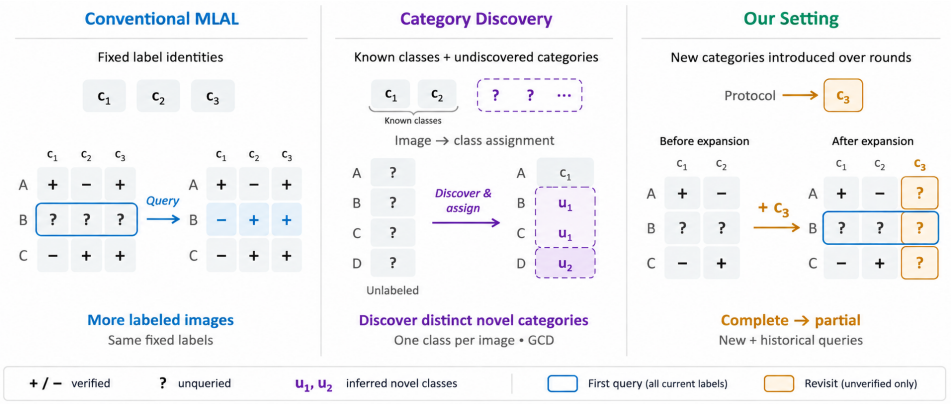


Figure 2: **Comparison of problem settings.** (a) Conventional MLAL acquires annotations for selected instances within a fixed label set with full-label queries; (b) Representative single-label generalized category discovery that uses known-class supervision to assign unlabeled instances to known classes or distinct unknown classes; (c) In our setting, the protocol introduces new category identities. Existing verified judgments remain valid, but previously complete annotations become partial. First-time queries cover all current labels, whereas selective revisits request only still-unverified judgments on previously queried instances.

Acquiring annotations for a newly introduced label block can span several active learning rounds. At round t, let Γ_t denote the most recently introduced nonempty label block, where

$$\Gamma_t = \begin{cases} \Delta S_t, & \Delta S_t \neq \varnothing, \\ \Gamma_{t-1}, & \Delta S_t = \varnothing, \end{cases} \qquad t \geq 2,$$

with $\Gamma_1 = \Delta S_1$. Throughout these rounds, the prediction task covers the cumulative category set S_t, not only the current block. For each instance $x \in \mathcal{X}$ and category $c \in S_t$, the ground-truth label $y_c(x)$ is 1 if c is relevant to x, and 0 otherwise. At the end of round t, the learned model f_t assigns each instance x a vector of relevance scores $f_t(x) \in [0,1]^{|S_t|}$, with one score for each category in S_t. Annotations acquired before a label expansion, however, do not specify an instance's relevance to the newly introduced categories and therefore provide only partial supervision.

Observed Supervision and Query Protocol. Before any queries in round t, let $B_t(x) \subseteq S_t$ denote the set of categories already annotated for instance x. We denote the observed positive labels by $Y_t^+(x) \subseteq B_t(x)$, while $B_t(x) \setminus Y_t^+(x)$ contains the observed negative labels. The unobserved label set $U_t(x)$ for x and the candidate instance sets for first-time queries and revisits are given by

$$\begin{aligned} U_t(x) &= S_t \setminus B_t(x), \\ \mathcal{X}_t^{\text{new}} &= \{x \in \mathcal{X} : B_t(x) = \varnothing\}, \\ \mathcal{X}_t^{\text{hist}} &= \{x \in \mathcal{X} : B_t(x) \neq \varnothing,\, U_t(x) \neq \varnothing\}. \end{aligned}$$

The annotations for x are complete with respect to S_t if and only if $U_t(x) = \varnothing$. To avoid querying the same instance on a same label set twice within a round, we keep $\mathcal{X}_t^{\text{hist}}$ fixed in round t, excluding instances first queried in that round from revisits.

A query $q = (x, L_q)$ specifies an instance x and a set of categories L_q to annotate. A first-time query for $x \in \mathcal{X}_t^{\text{new}}$ covers all current categories, with $L_q = S_t$. For a previously queried instance $x \in \mathcal{X}_t^{\text{hist}}$, a revisit requests annotations only for labels that have not yet been queried for x, with $\varnothing \neq L_q = U_t(x) = S_t \setminus B_t(x)$. The oracle returns $y_c(x)$ for each $c \in L_q$, and the resulting positive and negative labels are added to the existing annotations, while previously observed labels remain unchanged. Both query types share a per-round budget b_t, with each query counted as one unit regardless of $|L_q|$, while queries to the same instance in different rounds are counted separately. Within the shared budget, we aim to coordinate first-time queries and revisits to learn an accurate multi-label classifier over the current label set S_t.

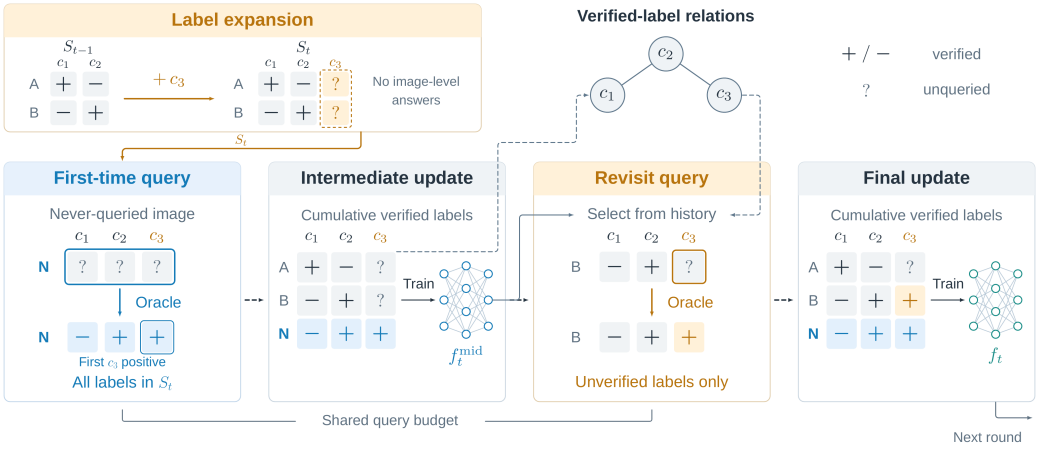


Figure 3: **Overview of ReMuAL.** Introducing c_3 makes the annotations of A and B incomplete. Querying N over S_t supplies annotations for intermediate training and label-relationship estimation. The updated predictions and relationships guide B's revisit for c_3, followed by the final update. Symbols $+$, $-$, and ? denote verified positive, verified negative, and unqueried entries, respectively.

4 Retrospective Multi-Label Active Learning (ReMuAL)

In this section, we present the ReMuAL framework for multi-label active learning under label expansion, as illustrated in Figure 3. We first describe the overall learning procedure of ReMuAL (Section 4.1). Then, we detail the selection strategies for first-time queries on unqueried instances (Section 4.2) and selective revisits to historical instances with incomplete annotations (Section 4.3).

4.1 Overall Learning Procedure

ReMuAL coordinates first-time queries and selective revisits to acquire supervision for newly introduced labels and complete the annotations of historical instances under a shared query budget. First-time queries annotate selected instances in $\mathcal{X}_t^{\text{new}}$ for all labels in S_t, providing new-label annotations and co-occurrence evidence with earlier labels. Together with previous query feedback, the newly acquired annotations are used to train f_t^{mid} and estimate label relationships. These relationships connect each historical instance's verified positives to its unobserved labels and, with updated predictions, guide subsequent revisits. Then, the revisit feedback is used for the final update to f_t.

Budget Allocation. To coordinate first-time queries and revisits, we use the incompleteness of historical annotations to guide the budget split. For all previously queried instances $\mathcal{H}_t = \{x \in \mathcal{X} : B_t(x) \neq \varnothing\}$, their average fraction of unobserved labels under S_t is

$$\rho_t = \frac{\sum_{x \in \mathcal{H}_t} |U_t(x)|}{|\mathcal{H}_t|\,|S_t|}. \tag{1}$$

For a budget of b_t queries in round t, we round $b_t \rho_t$ to the nearest integer to obtain the provisional revisit count $\widetilde{b}_t^{\text{hist}} = \lfloor b_t \rho_t + \frac{1}{2} \rfloor$. The final allocation is

$$b_t^{\text{hist}} = \min\left\{\bar{b}_t, \max\left\{\underline{b}_t, \widetilde{b}_t^{\text{hist}}\right\}\right\}, \qquad b_t^{\text{new}} = b_t - b_t^{\text{hist}}, \tag{2}$$

where $\underline{b}_t$ and $\bar{b}_t$ are the lower and upper bounds on the revisit budget, respectively.

Model Update. After each query stage, we train the classifier f on all accumulated annotations. For each instance, let m_{ic} indicate whether label c is annotated, then the supervised loss $\mathcal{L}_{\text{sup}}(f)$ is binary cross-entropy between y_{ic} and $p_{ic} = f_c(x_i)$, evaluated only where $m_{ic} = 1$.

To preserve previously learned knowledge, we complement this objective with logit matching. Before each training stage, we keep a fixed copy of the classifier as the teacher. For earlier labels

$P_t = S_t \setminus \Gamma_t$, we match the current logits to the corresponding teacher outputs:

$$\mathcal{L}_{\text{train}}(f) = \mathcal{L}_{\text{sup}}(f) + \lambda_{\text{anc}} \mathbb{E}_{x_i,\, c \in P_t} \left[d(z_{ic}, \bar{z}_{ic}) \right], \tag{3}$$

where z_{ic} is the classifier logit for label c on x_i, $\bar{z}_{ic}$ the corresponding teacher target, and d the Smooth-L_1 discrepancy with weight λ_{anc}. The expectation averages over training instances used in the current update and labels in P_t.

4.2 First-Time Query Selection

First-time query selection aims to acquire positive examples of newly introduced labels and improve recognition of existing labels. Although label expansion creates an immediate need for new-label supervision, existing labels may still have few positive examples or remain difficult for the classifier to recognize. We select instances from $\mathcal{X}_t^{\text{new}}$ using accumulated annotations and current predictions within b_t^{new}. An instance can carry both label types, so a query over S_t can serve both goals.

Candidate Scoring. To seek positive examples of newly introduced and underrepresented existing labels, we evaluate candidate predictions in relation to the positive annotations already collected. Let $n_{t,c}^+$ count verified positives of c, we evaluate both labels targeted for positive-example acquisition and labels with verified positives. For each such label set $L \subseteq S_t$, we compute

$$W_i(L) = \frac{\sum_{c \in L} p_{ic} (1 + n_{t,c}^+)^{-1/2}}{\sum_{c \in L} (1 + n_{t,c}^+)^{-1/2}}, \qquad H_i(L) = \frac{1}{|L|} \sum_{c \in L} h_2(p_{ic}), \tag{4}$$

where $h_2(p) = -p \log_2 p - (1-p) \log_2 (1-p)$ is binary entropy. After separate min–max normalization (indicated by hat), we combine the two terms as $s_t^{WH}(x_i; L) = \lambda_{t,L}^W \widehat{W}_i(L) + \lambda_{t,L}^H \widehat{H}_i(L)$ in the initial score $s_t(x_i)$, with $\lambda_{t,L}^W, \lambda_{t,L}^H$ the corresponding weights. The two terms in this formula favor positives of under-represented labels and uncertain label assignments, respectively.

Batch Selection. Selecting candidates independently by their initial scores may favor several instances with nearly identical features and similar predictions for the same labels. Although each instance may be useful on its own, annotating several such instances can yield near-duplicate positive examples or provide largely overlapping information for resolving uncertain label predictions. Therefore, after each selection, we rescore the remaining candidates by their additional label relevance and feature separation from current data batch, rather than rely on their initial scores alone.

Let $\mathcal{V}_t$ be the candidate pool and $\mathcal{B}_m$ the selected batch after m steps. For each $c \in L$, let $q_{t,c}^{(m)}(L) = \max_{x_j \in \mathcal{B}_m} p_{jc}$ denote its highest predicted relevance among the selected instances, the corresponding label-relevance gain is

$$G_{t,m}(x_i; L) = \frac{\sum_{c \in \mathcal{T}_{i,m}(L)} [p_{ic} - q_{t,c}^{(m)}(L)]_+}{|\mathcal{T}_{i,m}(L)|}, \tag{5}$$

where $\mathcal{T}_{i,m}(L)$ indexes the top-k gains over L, and $[a]_+ = \max(a, 0)$. The feature separation is formulated by $d_{t,m}(x_i) = \min_{x_j \in \mathcal{R}_t \cup \mathcal{B}_m} (1 - \boldsymbol{\phi}_i^\top \boldsymbol{\phi}_j)/2$, where $\boldsymbol{\phi}_i$ is the normalized feature and $\mathcal{R}_t$ a subset of previously queried instances. Combined with initial scores $s_t(x_i)$, we select

$$x_{m+1} = \underset{x_i \in \mathcal{V}_t \setminus \mathcal{B}_m}{\arg\max} \left[s_t(x_i) + \sum_{L \in \mathscr{L}_t} \omega_{t,m}(L) G_{t,m}(x_i; L) + \lambda_{\text{d}} d_{t,m}(x_i) \right], \tag{6}$$

where $\mathscr{L}_t$ contains the acquisition and existing-label scopes, with $\omega_{t,m}(L)$ and λ_{d} weighting gains and distance. After each selection, we update $q_{t,c}^{(m+1)}(L) = \max\{q_{t,c}^{(m)}(L), p_{m+1,c}\}$ and the distances to include x_{m+1} in $\mathcal{B}_{m+1}$, with the initial scores $s_t(x_i)$ remain fixed.

4.3 Retrospective Query Selection

First-time queries provide new training examples but leave the annotation gaps in historical instances unresolved. Rather than assess these instances through model predictions alone, we also use their previously verified labels to guide further annotation. Co-occurrence relationships estimated from cumulative verified annotations after first-time querying connect each instance's observed positives to labels in $U_t(x)$. Together with f_t^{mid}, this evidence guides selection from $\mathcal{X}_t^{\text{hist}}$ in budget b_t^{hist}.

Label Relationships. To assess whether an unobserved label is relevant to a historical instance, we use co-occurrence relationships between that label and the instance's verified positive labels. We estimate these relationships from cumulative verified annotations after first-time querying.

For distinct labels $a, c \in S_t$, with a as the verified-positive source label and c as the target label, let $N^{+\mathrm{v}}_{t,ac}$ count instances where c is also verified and $N^{++}_{t,ac}$ those where c is verified positive. For supported pairs $a \neq c$ and historical instances x_i, we define

$$\pi_{t,ac} = \frac{N^{++}_{t,ac} + \alpha}{N^{+\mathrm{v}}_{t,ac} + 2\alpha}, \tag{7}$$

$$r_{t,ic} = 1 - \prod_{\substack{a \in Y^+_t(x_i) \\ N^{+\mathrm{v}}_{t,ac} > 0}} (1 - \pi_{t,ac}), \qquad c \in U_t(x_i), \tag{8}$$

where α is a smoothing pseudocount. The relationship $\pi_{t,ac}$ estimates how often c is positive among verified positives of a for which c has also been annotated; $r_{t,ic}$ aggregates support from x_i's verified positives for its unobserved label c.

Revisit Selection. To identify missing positive annotations, we combine intermediate predictions with the evidence from each historical instance's verified labels. For a historical instance x_i, let $p^{\mathrm{mid}}_{ic} = f^{\mathrm{mid}}_{t,c}(x_i)$ and let $n^{+,\mathrm{mid}}_{t,c}$ count label c's verified positives after first-time querying. For revisit scoring, we define the prioritized unobserved labels as

$$L_{t,i} = \begin{cases} U_t(x_i) \cap \Delta S_t, & \Delta S_t \neq \varnothing, \\ \{c \in U_t(x_i) : n^{+,\mathrm{mid}}_{t,c} = 0\}, & \Delta S_t = \varnothing. \end{cases}$$

Then, we summarize the strongest joint supports among these labels to refine the revisit score:

$$e_{t,ic} = p^{\mathrm{mid}}_{ic} r_{t,ic}, \qquad E_t(x_i) = \frac{\sum_{c \in \mathcal{T}_{t,i}} e_{t,ic}}{|\mathcal{T}_{t,i}|}, \tag{9}$$

$$\widetilde{s}^{\mathrm{hist}}_t(x_i) = s^{\mathrm{hist}}_t(x_i) + \lambda_{\mathrm{e}} \widehat{E}_t(x_i), \tag{10}$$

where $\mathcal{T}_{t,i}$ contains up to k labels from $L_{t,i}$ with the largest $e_{t,ic}$, $E_t(x_i)$ averages their joint supports, $\widehat{E}_t(x_i)$ is the min–max-normalized $E_t(x_i)$ over historical candidates, $s^{\mathrm{hist}}_t(x_i)$ is the revisit score before adding this evidence, and λ_e controls the weight of this normalized support.

The refined score above initializes historical batch selection: we apply Eq. (6) to the historical candidates under budget b^{hist}_t, using $\tilde{s}^{\mathrm{hist}}_t$ as the fixed initial score. During selection, label gains are computed from f^{mid}_t over $L_{t,i}$, while diversity is measured against the already selected historical instances. Both terms will be updated after each selection. The resulting batch $\mathcal{B}^{\mathrm{hist}}_t$ yields

$$\mathcal{Q}^{\mathrm{hist}}_t = \big\{(x_i, U_t(x_i)) : x_i \in \mathcal{B}^{\mathrm{hist}}_t\big\}. \tag{11}$$

At last, the returned annotations are incorporated into the final model update.

5 Experiments

In this section, We compare ReMuAL with baseline methods, examine the roles of first-time queries and revisits, and evaluate its performance under semi-supervised training. Additional results and analyses are provided in Appendix C.

5.1 Experimental Setup

Datasets. We conduct our main experiments on PASCAL VOC 2007 trainval (20 labels) (Everingham et al., 2010), MS COCO 2014 train (80 labels) (Lin et al., 2014), Visual Genome (1,000 most frequent normalized object names) (Krishna et al., 2017), and NUS-WIDE train (81 concepts) (Chua et al., 2009). LVIS v1 (1,203 categories) (Gupta et al., 2019) serves as a separate long-tail diagnostic.

Table 1: Comparison of Checklist mAP at round r and cumulative AULC through round r under label expansion. "freq." and "name" denote frequency-ordered releases from most to least frequent in the fixed pool and lexicographic releases by label name, respectively. Results are reported as mean ± sample standard deviation over the three available runs, with the best mean in each column in bold. Δ denotes the seed-aligned difference between ReMuAL and the strongest baseline in the same column.

Dataset	Method	$r=4$		$r=8$		$r=12$		$r=16$	
		mAP	AULC	mAP	AULC	mAP	AULC	mAP	AULC
VOC freq.	Random	0.677±.033	0.537±.049	0.697±.010	0.571±.034	0.711±.029	0.592±.026	0.719±.012	0.608±.020
	Entropy	0.596±.109	0.415±.117	0.658±.063	0.479±.084	0.667±.033	0.511±.071	0.715±.012	0.545±.059
	Fixed-label	0.626±.084	0.447±.091	0.670±.062	0.503±.074	0.676±.032	0.528±.066	0.727±.015	0.560±.054
	CRAB-Adapter	0.572±.038	0.432±.034	0.608±.036	0.459±.025	0.629±.013	0.486±.019	0.683±.025	0.516±.019
	ReMuAL	**0.719**±.017	**0.603**±.041	**0.749**±.021	**0.631**±.029	**0.770**±.014	**0.656**±.030	**0.764**±.021	**0.673**±.026
	Δ	+0.043±.049	+0.067±.073	+0.052±.026	+0.060±.046	+0.060±.017	+0.063±.030	+0.037±.020	+0.064±.026
COCO freq.	Random	0.482±.010	0.370±.024	0.579±.007	0.439±.010	0.579±.004	0.464±.005	0.525±.007	0.470±.003
	Entropy	0.463±.021	0.340±.029	0.545±.011	0.401±.019	0.529±.006	0.426±.013	0.493±.006	0.434±.010
	Fixed-label	0.463±.022	0.342±.025	0.540±.019	0.401±.020	0.530±.006	0.426±.013	0.493±.006	0.435±.009
	CRAB-Adapter	0.484±.023	0.361±.033	0.533±.007	0.401±.020	0.508±.010	0.419±.012	0.456±.018	0.422±.005
	ReMuAL	**0.495**±.006	**0.430**±.007	**0.581**±.003	**0.478**±.006	**0.584**±.004	**0.500**±.005	**0.560**±.008	**0.508**±.006
	Δ	+0.010±.019	+0.060±.023	+0.002±.009	+0.039±.011	+0.005±.007	+0.036±.007	+0.035±.002	+0.037±.006
COCO name	Random	0.516±.017	0.351±.032	0.560±.001	0.419±.016	0.558±.001	0.449±.010	0.553±.003	0.465±.007
	Entropy	0.519±.020	0.383±.038	0.529±.014	0.421±.028	0.528±.004	0.442±.022	0.524±.009	0.452±.015
	Fixed-label	0.518±.023	0.383±.038	0.532±.009	0.423±.024	0.529±.003	0.444±.018	0.521±.004	0.454±.013
	CRAB-Adapter	0.528±.018	0.375±.031	0.524±.007	0.408±.019	0.522±.001	0.430±.013	0.488±.010	0.436±.008
	ReMuAL	**0.573**±.009	**0.465**±.010	**0.565**±.014	**0.491**±.007	**0.561**±.018	**0.505**±.011	**0.558**±.018	**0.512**±.012
	Δ	+0.046±.009	+0.082±.046	+0.006±.013	+0.069±.031	+0.003±.017	+0.056±.021	+0.005±.014	+0.048±.020

Dataset	Method	$r=8$		$r=16$		$r=24$		$r=32$		$r=40$	
		mAP	AULC	mAP	AULC	mAP	AULC	mAP	AULC	mAP	AULC
VG name	Random	0.073±.006	0.037±.005	0.092±.004	0.057±.005	0.085±.002	0.066±.004	0.078±.001	0.069±.003	0.070±.002	0.069±.003
	Entropy	0.103±.001	0.059±.004	0.118±.002	0.080±.002	0.115±.002	0.090±.001	**0.112**±.001	0.095±.000	**0.109**±.002	0.097±.000
	Fixed-label	0.098±.009	0.058±.005	0.113±.006	0.078±.006	0.110±.006	0.087±.006	0.108±.005	0.091±.005	0.106±.005	0.094±.005
	CRAB-Adapter	0.077±.002	0.043±.001	0.091±.004	0.061±.001	0.087±.003	0.068±.002	0.089±.002	0.072±.002	0.090±.002	0.075±.002
	ReMuAL	**0.146**±.004	**0.104**±.003	**0.141**±.003	**0.119**±.003	**0.122**±.000	**0.122**±.002	0.108±.003	**0.119**±.002	0.100±.001	**0.115**±.002
	Δ	+0.043±.003	+0.045±.002	+0.024±.005	+0.039±.001	+0.007±.002	+0.032±.002	−0.004±.004	+0.025±.002	−0.010±.003	+0.019±.002

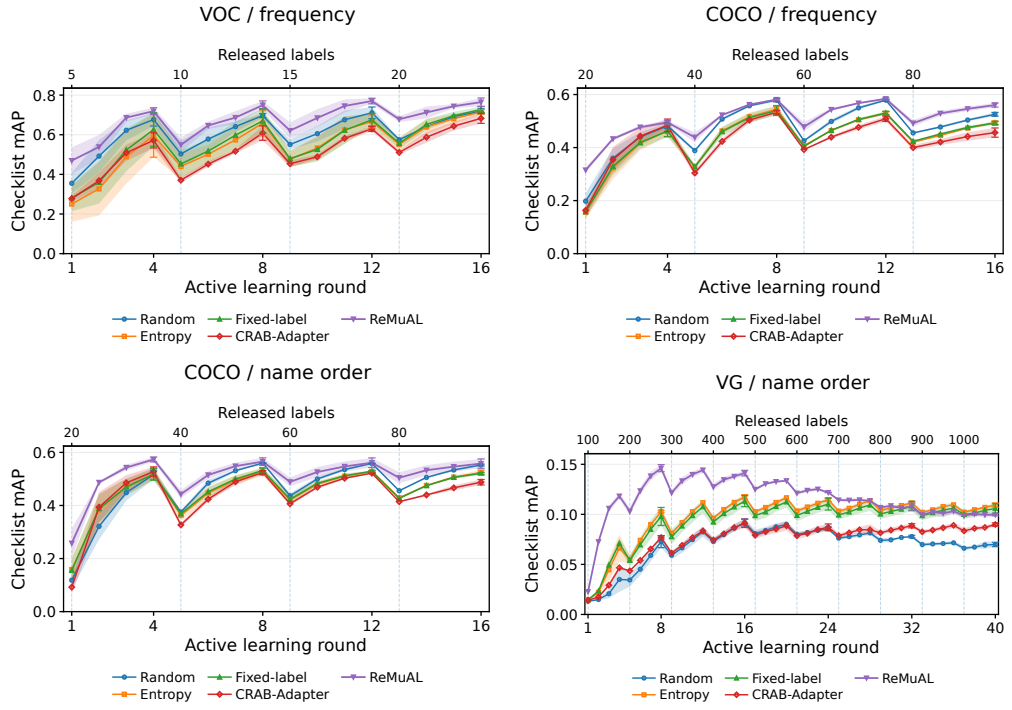


Figure 4: Learning curves of Checklist mAP for the four settings in Table 1. Solid lines denote mean performance, while shaded bands showing ±1 sample STD. Vertical dashed lines mark label releases; the upper axes show the corresponding number of released labels, and panel-specific y-axis ranges preserve visibility of within-setting differences.

Table 2: Performance on NUS-WIDE and LVIS, measured by Checklist mAP at round r and cumulative AULC through round r. Results are mean $\pm$ STD over the three available runs; bold denotes the highest mean in each column, and Δ the seed-aligned difference between ReMuAL and the strongest baseline.

Method	NUS-WIDE				LVIS			
	$r = 8$		$r = 16$		$r = 24$		$r = 48$	
	mAP	AULC	mAP	AULC	mAP	AULC	mAP	AULC
Random	0.450±.003	0.424±.008	0.308±.002	0.370±.004	0.077±.002	0.091±.003	0.028±.001	0.068±.001
Entropy	0.446±.008	0.410±.004	0.314±.016	0.366±.006	0.087±.005	0.099±.007	0.044±.004	0.079±.006
CRAB-Adapter	0.432±.006	0.401±.004	0.280±.027	0.351±.005	0.079±.001	0.081±.001	**0.045**±.001	0.069±.001
ReMuAL	**0.486**±.016	**0.467**±.010	**0.361**±.007	**0.421**±.008	**0.088**±.002	**0.139**±.004	0.033±.002	**0.095**±.003
Δ	+0.036±.016	+0.043±.014	+0.047±.019	+0.050±.012	+0.001±.005	+0.040±.006	-0.011±.003	+0.016±.006

Baselines. We compare ReMuAL with four baselines: Random sampling; Entropy using mean binary predictive entropy; Fixed-label, a binary-feedback adapter aggregating image–label pair scores; and CRAB-Adapter (Qi et al., 2025), which combines predictive entropy, label relationships, and feature diversity. All baselines use first-time and revisit queries under the same within-setting label-release schedule and image-query budget.

Implementation Details. We use ResNet-50 trained with SGD at a learning rate of 0.01 for all comparisons. Labels are released cumulatively every four rounds under frequency- or name-ordered schedules. Per-round budgets allow up to 32 image-query calls for VOC, 256 for Visual Genome, and 512 for COCO and NUS-WIDE. After each round, we compute mean average precision (mAP) on the fixed query pool over currently released labels (Checklist) and, on release rounds, the newly released block (New). We summarize the trajectory using the normalized area under the learning curve (AULC). For the main settings, we report the mean and standard deviation over three runs. Further experimental details are provided in Appendix B.

5.2 Main Results

Comparison with baseline methods. Under frequency-ordered expansion, ReMuAL achieves the highest mean Checklist mAP at every round on VOC and COCO (Table 1 and Figure 4). Gains over the strongest baseline reach 0.114 on VOC, and 0.053 on COCO. Its mean full-trajectory AULC also exceeds Random, the strongest baseline for this metric, by 0.064 and 0.037, respectively. On COCO with name-ordered releases, ReMuAL again leads in mean Checklist mAP and cumulative AULC at every round, with AULC gains up to 0.100 over the strongest baseline. The smaller Checklist gap in later rounds suggests that ReMuAL are particularly helpful when verified annotations are scarce.

Performance with a large label vocabulary. On Visual Genome, a 1,000-label vocabulary is introduced in ten blocks over 40 rounds. ReMuAL leads in mean Checklist mAP mainly during the early and middle stages, with gains up to 0.057. Its mean cumulative AULC exceeds every baseline throughout the trajectory, with gains up to 0.045. This advantage persists even when Entropy leads in late-stage Checklist mAP: on the last released block, ReMuAL reaches 0.062 in New mAP, versus 0.027 for Entropy. As verified annotations accumulate, uncertainty-based selection may become more competitive, while ReMuAL's stronger early-stage gains under sparse supervision remain reflected in its higher cumulative AULC over the full sequence of releases.

Performance on web-image concepts. We further evaluate ReMuAL on NUS-WIDE, whose object and scene concepts provide a complementary test beyond the preceding object-category benchmarks. Under frequency-ordered expansion, ReMuAL achieves the highest mean Checklist mAP and cumulative AULC at both rounds 8 and 16 (Table 2). At round 16, its Checklist mAP reaches 0.361, exceeding Entropy by 4.67 percentage points, while its full-trajectory AULC reaches 0.421, exceeding Random by 0.050. These results extend the observed advantage to web-image concept expansion, with gains in both final predictive performance and cumulative performance.

Long-tail diagnostic. We use LVIS to examine expansion into increasingly infrequent categories. At round 24, ReMuAL has a similar mean Checklist mAP to Entropy (0.088 versus 0.087), but exceeds its cumulative AULC by 0.040 (Table 2). At round 48, ReMuAL retains the highest AULC

Table 3: Ablation study for ReMuAL on COCO and Visual Genome. Checklist mAP at round r and cumulative AULC through round r are reported from a single run; bold marks the best result in each column. † denotes the corresponding full ReMuAL result in Table 1.

Configuration	COCO				VG			
	$r = 8$		$r = 16$		$r = 20$		$r = 40$	
	mAP	AULC	mAP	AULC	mAP	AULC	mAP	AULC
First-time only	0.526	0.403	0.480	0.431	0.114	0.081	0.107	0.092
Random revisits	0.523	0.402	0.474	0.428	0.110	0.078	0.103	0.090
Random first-time	0.575	0.460	0.522	0.478	0.092	0.067	0.071	0.072
w/o New-Coverage	0.513	0.395	0.460	0.414	0.109	0.079	0.104	0.091
w/o Retro-History	0.517	0.401	0.474	0.425	0.110	0.080	0.103	0.091
w/o Retro-Uncertainty	0.510	0.397	0.456	0.419	0.110	0.079	0.104	0.090
ReMuAL†	**0.578**	**0.476**	**0.562**	**0.507**	**0.127**	**0.117**	**0.124**	**0.117**

Table 4: Semi-supervised results on COCO and Visual Genome. Checklist mAP at round r and cumulative AULC through round r are reported from a single run; bold marks the best result in each column.

COCO	$r = 8$		$r = 16$	
Method	mAP	AULC	mAP	AULC
Random	0.452	0.388	0.254	0.345
Entropy	0.295	0.296	0.183	0.255
ReMuAL	**0.591**	**0.495**	**0.567**	**0.526**
VG	$r = 20$		$r = 40$	
Method	mAP	AULC	mAP	AULC
Random	0.042	0.042	0.070	0.050
Entropy	0.042	0.042	0.064	0.049
ReMuAL	**0.107**	**0.099**	**0.092**	**0.098**

of 0.095, exceeding Entropy by 0.016; its Checklist mAP is 0.033, compared with 0.045 for CRAB-Adapter. This pattern is consistent with targeted querying being particularly valuable under sparse early supervision, while later comparisons also reflect accumulated annotations and the introduction of increasingly rare categories under a shared per-round budget for the expanding label set.

5.3 Analysis of Query Strategies

Effect of revisits. Table 3 compares first-time-only querying, random revisits, and the full ReMuAL reference. The full reference leads in Checklist mAP and cumulative AULC at both reported checkpoints on COCO and VG. For example, its final COCO mAP is 0.562 versus 0.480 for first-time-only querying. Random revisits slightly underperform first-time-only querying at both checkpoints on each dataset, suggesting that revisiting alone does not explain these gains.

Effect of query selection. Both randomized configurations remain below the full reference in mAP and AULC. Random revisits yield lower final mAP than random first-time queries on COCO (0.474 versus 0.522), but higher mAP on VG (0.103 versus 0.071). This ordering also holds for AULC and at the earlier checkpoints, suggesting dataset-dependent sensitivity to selection in the two branches. The *w/o New-Coverage*, *w/o Retro-History*, and *w/o Retro-Uncertainty* configurations likewise underperform the full reference in both metrics at both checkpoints. Additional analyses of how queries are allocated and what label feedback they yield are provided in Appendix C.2.

5.4 Results with Semi-Supervised Learning

We next examine ReMuAL under semi-supervised training, where confident predictions on unchecked entries provide auxiliary training targets without changing verified annotations. In the experimental results of Table 4, ReMuAL obtains a final Checklist mAP of 0.567 and an AULC of 0.526 on COCO, compared with 0.254 and 0.345 for Random, the stronger baseline for both metrics. On Visual Genome, the corresponding values are 0.092 and 0.098, versus 0.070 and 0.050 for Random. These comparisons do not isolate the effect of pseudo-labeling: the SSL and primary training profiles also differ in their epoch ceilings.

6 Conclusion

In this work, we studied multi-label active learning under label expansion, where previously complete annotations become partial as new labels are introduced. Since querying new instances alone leaves these historical annotation gaps unresolved, we proposed ReMuAL to coordinate first-time queries and selective revisits under a shared query budget. Experiments on multiple benchmarks show that ReMuAL compares favorably with the evaluated baselines in terms of mAP and AULC under matched image-query budgets. Thus, our work addresses the historical annotation gaps caused by label expansion within a limited query budget.

Reproducibility Statement

Sections 3 and 4 describe the problem setting, query protocol, and learning procedure, with additional query and training details in Appendix A. Section 5.1 and Appendix B specify the datasets, label-release schedules, training configurations, and evaluation metrics. Appendix B.3 specifies run coverage, sample-SD aggregation, and historical provenance limitations. Appendix C provides supplementary results and a current-batch-only comparison. The accompanying seed-level numerical records and aggregation/plotting scripts rebuild the reported statistics, tables, and result figures; this numerical reconstruction does not certify historical training equivalence.

Ethics Statement

Our experiments use existing public image benchmarks, with oracle responses simulated from their annotations. We collect no new images or human annotations and recruit no participants. The reported budgets count image-query calls rather than human annotation effort. Dataset reuse must respect the original terms of use and applicable privacy and consent requirements. Biases in the source annotations may affect the estimated label relationships and query selection. Downstream applications require an assessment of fairness and potential harm in their intended context.

AI Use Statement

Generative AI tools assisted with literature retrieval and the identification of prior works relevant to multi-label active learning and label-space expansion. AI-generated suggestions provided leads for locating relevant publications and examining connections among related studies. Decisions about which sources to cite and how to characterize their contributions rest with the authors. The authors are responsible for verifying references against the original publications and for the accuracy and integrity of the final manuscript.

References

Mohamed Abdelsalam, Mojtaba Faramarzi, Shagun Sodhani, and Sarath Chandar. Iirc: Incremental implicitly-refined classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Klaus Brinker. On active learning in multi-label classification. In *From Data and Information Analysis to Knowledge Engineering*, Studies in Classification, Data Analysis, and Knowledge Organization, pp. 206–213. Springer, 2006. doi: 10.1007/3-540-31314-1_24. URL https://link.springer.com/chapter/10.1007/3-540-31314-1_24.

Tianshui Chen, Tao Pu, Hefeng Wu, Yuan Xie, and Liang Lin. Structured semantic transfer for multi-label recognition with partial labels. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 339–346, 2022. doi: 10.1609/aaai.v36i1.19910. URL https://ojs.aaai.org/index.php/AAAI/article/view/19910.

Tianshui Chen, Tao Pu, Lingbo Liu, Yukai Shi, Zhijing Yang, and Liang Lin. Heterogeneous semantic transfer for multi-label recognition with partial labels. *International Journal of Computer Vision*, 132:6091–6106, 2024. doi: 10.1007/s11263-024-02127-2. URL https://link.springer.com/article/10.1007/s11263-024-02127-2.

Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. NUS-WIDE: A real-world web image database from national university of singapore. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, pp. 1–9, 2009.

Elijah Cole, Oisin Mac Aodha, Titouan Lorieul, Pietro Perona, Dan Morris, and Nebojsa Jojic. Multi-label learning from single positive labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 933–942, 2021.

Zixuan Ding, Ao Wang, Hui Chen, Qiang Zhang, Pengzhang Liu, Yongjun Bao, Weipeng Yan, and Jungong Han. Exploring structured semantic prior for multi label recognition with incomplete labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and*

Pattern Recognition, pp. 3398–3407, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Ding_Exploring_Structured_Semantic_Prior_for_Multi_Label_Recognition_With_Incomplete_CVPR_2023_paper.html.

Thibaut Durand, Nazanin Mehrasa, and Greg Mori. Learning a deep ConvNet for multi-label classification with partial labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 647–657, 2019.

Andrea Esuli and Fabrizio Sebastiani. Active learning strategies for multi-label text classification. In *Advances in Information Retrieval*, volume 5478 of *Lecture Notes in Computer Science*, pp. 102–113. Springer, 2009. doi: 10.1007/978-3-642-00958-7_12. URL https://link.springer.com/chapter/10.1007/978-3-642-00958-7_12.

Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

Hui Wen Goh and Jonas Mueller. Activelab: Active learning with re-labeling by multiple annotators. In *ICLR Workshop on Trustworthy ML*, 2023.

Xiaoqiang Gui, Xudong Lu, and Guoxian Yu. Cost-effective batch-mode multi-label active learning. *Neurocomputing*, 463:355–367, 2021. doi: 10.1016/j.neucom.2021.08.063.

Agrim Gupta, Piotr Dollár, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5356–5364, 2019.

Peiyun Hu, Zachary C. Lipton, Anima Anandkumar, and Deva Ramanan. Active learning with partial feedback. In *International Conference on Learning Representations*, 2019.

Xian-Sheng Hua and Guo-Jun Qi. Online multi-label active learning for large-scale multimedia annotation. Technical Report MSR-TR-2008-103, Microsoft Research, 2008. URL https://www.microsoft.com/en-us/research/wp-content/uploads/2008/06/tr-2008-103.pdf.

Sheng-Jun Huang, Songcan Chen, and Zhi-Hua Zhou. Multi-label active learning: Query type matters. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, pp. 946–952, 2015.

Chen-Wei Hung and Hsuan-Tien Lin. Multi-label active learning with auxiliary learner. In *Proceedings of the Asian Conference on Machine Learning*, volume 20 of *Proceedings of Machine Learning Research*, pp. 315–332. PMLR, 2011. URL https://proceedings.mlr.press/v20/hung11.html.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.

Byung Hyun Lee, Okchul Jung, Jonghyun Choi, and Se Young Chun. Online continual learning on hierarchical label expansion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11761–11770, 2023.

Xin Li and Yuhong Guo. Active learning with multi-label SVM classification. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, pp. 1479–1485, 2013. URL https://www.ijcai.org/Proceedings/13/Papers/221.pdf.

Christopher H. Lin, Mausam, and Daniel S. Weld. Re-active learning: Active learning with relabeling. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pp. 1845–1852, 2016. doi: 10.1609/aaai.v30i1.10315.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pp. 740–755, 2014.

Zhiqiu Lin, Deepak Pathak, Yu-Xiong Wang, Deva Ramanan, and Shu Kong. Continual learning with evolving class ontologies. In *Advances in Neural Information Processing Systems*, volume 35, pp. 7671–7684, 2022.

Lars Möllenbrok, Gencer Sumbul, and Begüm Demir. Deep active learning for multi-label classification of remote sensing images. *IEEE Geoscience and Remote Sensing Letters*, 20:1–5, 2023. doi: 10.1109/LGRS.2023.3305647.

Shishir Muralidhara, Saqib Bukhari, Georg Schneider, Didier Stricker, and René Schuster. Cleo: Continual learning of evolving ontologies. In *European Conference on Computer Vision*, pp. 328–344, 2024. doi: 10.1007/978-3-031-72949-2_19.

Yuanyuan Qi, Jueqing Lu, Xiaohao Yang, Joanne Enticott, and Lan Du. Multi-label bayesian active learning with inter-label relationships. In Silvia Chiappa and Sara Magliacane (eds.), *Proceedings of the Forty-First Conference on Uncertainty in Artificial Intelligence*, volume 286 of *Proceedings of Machine Learning Research*, pp. 3480–3491. PMLR, 2025.

Oscar Reyes, Carlos Morell, and Sebastian Ventura. Effective active learning strategy for multi-label learning. *Neurocomputing*, 273:494–508, 2018. doi: 10.1016/j.neucom.2017.08.001.

Nicholas Roy and Andrew McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pp. 441–448, 2001. URL https://www.lri.fr/~sebag/Examens/Active/roy01toward.pdf.

Weishi Shi and Qi Yu. Fast direct search in an optimally compressed continuous target space for efficient multi-label active learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5769–5778. PMLR, 2019. URL https://proceedings.mlr.press/v97/shi19b.html.

Weishi Shi, Dayou Yu, and Qi Yu. A gaussian process-bayesian bernoulli mixture model for multi-label active learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 27542–27554. Curran Associates, Inc., 2021.

Wei Tan, Lan Du, and Wray Buntine. Diversity enhanced active learning with strictly proper scoring rules. In *Advances in Neural Information Processing Systems*, volume 34, pp. 10906–10918. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/5a7b238ba0f6502e5d6be14424b20ded-Abstract.html.

Wei Tan, Lan Du, and Wray Buntine. Bayesian estimate of mean proper scores for diversity-enhanced active learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46 (5):3463–3479, 2024a. doi: 10.1109/TPAMI.2023.3343359. URL https://doi.org/10.1109/TPAMI.2023.3343359.

Wei Tan, Ngoc Dang Nguyen, Lan Du, and Wray Buntine. Harnessing the power of Beta scoring in deep active learning for multi-label text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 15240–15248, 2024b. doi: 10.1609/aaai.v38i14.29447. URL https://ojs.aaai.org/index.php/AAAI/article/view/29447.

Deepak Vasisht, Andreas C. Damianou, Manik Varma, and Ashish Kapoor. Active learning for sparse bayesian multilabel classification. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 472–481, 2014. doi: 10.1145/2623330.2623759.

Zhen Wang, Liu Liu, and Dacheng Tao. Deep streaming label learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9963–9972. PMLR, 2020. URL https://proceedings.mlr.press/v119/wang20n.html.

Jian Wu, Victor S. Sheng, Jing Zhang, Pengpeng Zhao, and Zhiming Cui. Multi-label active learning for image classification. In *2014 IEEE International Conference on Image Processing*, pp. 5227–5231, 2014. doi: 10.1109/ICIP.2014.7026058.

Yi-Fan Yan and Sheng-Jun Huang. Cost-effective active learning for hierarchical multi-label classification. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pp. 2962–2968, 2018.

Bishan Yang, Jian-Tao Sun, Tengjiao Wang, and Zheng Chen. Effective multi-label active learning for text classification. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 917–926, 2009. doi: 10.1145/1557019.1557119.

Chen Ye, Jian Wu, Victor S. Sheng, Pengpeng Zhao, and Zhiming Cui. Multi-label active learning with label correlation for image classification. In *2015 IEEE International Conference on Image Processing*, pp. 3437–3441, 2015. doi: 10.1109/ICIP.2015.7351442.

Dayou Yu, Minghao Li, Weishi Shi, and Qi Yu. Evidential mixture machines: Deciphering multi-label correlations for active learning sensitivity. In *Advances in Neural Information Processing Systems*, volume 37, pp. 112217–112236, 2024. doi: 10.52202/079017-3563. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/cbb1fa8e7f515e796cda6621a703492f-Abstract-Conference.html.

Guoxian Yu, Xia Chen, Carlotta Domeniconi, Jun Wang, Zhao Li, Zili Zhang, and Xiangliang Zhang. CMAL: Cost-effective multi-label active learning by querying subexamples. *IEEE Transactions on Knowledge and Data Engineering*, 34(5):2091–2105, 2022. doi: 10.1109/TKDE.2020.3003899.

A Additional Method Details

A.1 Query Decisions

Annotation states and notation. The unsuperscripted $B_t(x)$, $Y_t^+(x)$, and $U_t(x)$ denote the state before querying in round t. Superscripts mid and post denote the states after first-time feedback and after revisit feedback, respectively; $U_t^j(x) = S_t \setminus B_t^j(x)$ at each stage j. The image pool and S_t do not change between the two query stages. The historical candidate identities are fixed at the start of the round; instances first queried in that round cannot immediately re-enter the revisit branch. Predictions and label relationships may nevertheless be updated before those historical identities are scored. K_t^j denotes labels with at least one verified positive at stage j, not the released label universe. The symbol ? in the illustrative annotation tables means unqueried, not a negative label or a missing cross-reference.

Allocation and execution. The historical set $\mathcal{H}_t$ includes previously queried images with no remaining gap; only $\mathcal{X}_t^{\mathrm{hist}}$ is eligible for a revisit. Thus ρ_t averages incompleteness over the historical supervision, not just the eligible candidates. Empty history, an empty label universe, or a zero budget gives zero revisit allocation; otherwise half-up rounding gives $\lfloor b_t\rho_t + 1/2 \rfloor$ before the configured bounds are applied. The final allocation also respects the available historical candidates and the per-round call budget. The lower and upper bounds incorporate release, follow-up, and final-block settings; their recorded values are summarized in Appendix B.4. Both branch budgets are limits. Candidate or scope exhaustion can leave unused capacity, so actual calls are read from execution records rather than inferred from a budget formula. A call may contain multiple label judgments and is not a unit of annotation time.

First-time candidate-score definitions. Let $\mathcal{C}_t$ be the pool entering initial candidate scoring, after the configured pool construction and filtering steps. All statistics below use the pre-query model and verified annotations. The pool $\mathcal{C}_t$ is the normalization pool; the reranking pool $\mathcal{V}_t$ in the main text is its retained subset. These definitions expand the ordinary greedy scoring path; configured random mixtures, pair selectors, and batch repair are separate paths. Define

$$n_{t,c}^+ = \sum_{x \in \mathcal{X}} \mathbf{1}[c \in Y_t^+(x)], \qquad K_t = \{c \in S_t : n_{t,c}^+ > 0\}. \tag{12}$$

The unadjusted priority set is

$$F_t^0 = \begin{cases} \Delta S_t, & \Delta S_t \neq \varnothing, \\ S_t \setminus K_t, & \Delta S_t = \varnothing. \end{cases} \tag{13}$$

Let $D_t \subseteq \Delta S_t$ contain the labels temporarily de-emphasized by the release-pacing rule, and let $D_t = \varnothing$ when that rule is inactive. The actual priority scope and supported-label scope are

$$F_t = \begin{cases} F_t^0 \setminus D_t, & F_t^0 \setminus D_t \neq \varnothing, \\ S_t, & \text{otherwise}, \end{cases} \qquad \mathcal{A}_t = \begin{cases} K_t, & K_t \neq \varnothing, \\ S_t, & \text{otherwise}. \end{cases} \tag{14}$$

These sets specify scoring scopes, not the annotation scope of a first-time query, which remains S_t.

For a label set L, define

$$H_i(L) = \frac{1}{|L|} \sum_{c \in L} h_2(p_{ic}), \qquad J_i(L) = 1 - \prod_{c \in L} (1 - p_{ic}), \tag{15}$$

$$C_i(L) = \sum_{c \in L} p_{ic}, \qquad M_i(L) = \frac{1}{|\mathcal{T}_k(i, L)|} \sum_{c \in \mathcal{T}_k(i,L)} p_{ic}, \tag{16}$$

$$W_i(L) = \frac{\sum_{c \in L} a_{t,c} p_{ic}}{\sum_{c \in L} a_{t,c}}, \qquad a_{t,c} = (1 + n_{t,c}^+)^{-1/2}, \tag{17}$$

where h_2 is binary entropy and $\mathcal{T}_k(i, L)$ contains the $\min(k, |L|)$ labels with the largest predicted relevance scores. All five statistics are zero for an empty label set. The feedback vector is

$$\mathbf{v}_i(L) = \big(J_i(L), C_i(L), M_i(L), W_i(L)\big)^\top. \tag{18}$$

Here, J_i is a product-complement proxy for obtaining at least one positive response, C_i is the sum of predicted positive responses, M_i averages the largest relevance scores, and W_i gives larger relative weight to labels with fewer verified positives. The product form of J_i does not establish conditional independence or probability calibration for correlated labels.

For any scalar statistic q_i, write

$$\widehat{q}_i = \begin{cases} \dfrac{q_i - q_{\min}}{q_{\max} - q_{\min}}, & q_{\max} - q_{\min} > \epsilon, \\ 0, & \text{otherwise}, \end{cases} \qquad q_{\min} = \min_{x_j \in \mathcal{C}_t} q_j, \quad q_{\max} = \max_{x_j \in \mathcal{C}_t} q_j, \tag{19}$$

with $\epsilon = 10^{-12}$. Normalization is applied separately to each statistic and each scoring role. The same normalization pool is retained during the ensuing greedy reranking. In particular, $\widehat{\mathbf{v}}$ denotes componentwise normalization, not normalization after combining its entries.

The full initial score for this path is

$$s_t^{\text{new}}(x_i) = \alpha_t \widehat{H}_i(F_t) + \boldsymbol{\beta}_t^\top \widehat{\mathbf{v}}_i(F_t) + g_t(x_i). \tag{20}$$

This is an expansion of the initial score, not an additional query objective. The W and H contributions correspond to those highlighted in the main text; all contributions are added before greedy batch selection. The residual term is

$$\begin{aligned} g_t(x_i) = \gamma_t \big[\widehat{H}_i(\mathcal{A}_t) + \widehat{J}_i(\mathcal{A}_t) + \widehat{W}_i(\mathcal{A}_t) \big] \\ + \delta_t \widehat{H}_i(S_t) + \eta_t \widehat{R}_i(S_t) - \zeta_t \widehat{C}_i(D_t). \end{aligned} \tag{21}$$

The first bracket combines the same uncertainty and positive-response statistics over labels already supported by verified positive examples. The full-label entropy is independent of that scope. The relation score is

$$R_i(L) = \binom{|L|}{2}^{-1} \sum_{\{a,b\} \subseteq L} p_{ia} p_{ib}\, r_{ab}, \qquad r_{ab} = \max\{\pi_{b|a}, \pi_{a|b}\}, \tag{22}$$

with $R_i(L) = 0$ when $|L| < 2$. Here $\pi_{b|a}$ is computed only from instances for which a is a verified positive and b has been verified:

$$\pi_{b|a} = \begin{cases} \dfrac{n_{ab}^{++} + 1}{n_{a,b}^{+\mathrm{v}} + 2}, & n_{a,b}^{+\mathrm{v}} > 0, \\ 0, & n_{a,b}^{+\mathrm{v}} = 0. \end{cases} \tag{23}$$

Distinct labels are used, and unsupported directions contribute zero. This relation table is computed from pre-query annotations here, rather than from the intermediate state used for retrospective selection. The final term of Equation (21) penalizes predicted responses on temporarily deferred labels without removing these labels from the annotation request.

Round-dependent coefficient schedules. Write a for the supported-label weight, s_{r} and s_{l} for its ordinary-release and later-release scales, and b_J, b_C, b_M, b_W for the feedback weights. On ordinary rounds,

$$\alpha_t = 1, \quad \boldsymbol{\beta}_t = (b_J, b_{C,t}, b_M, b_W)^\top, \quad \eta_t = 0, \quad \gamma_t = \begin{cases} a s_{\mathrm{r}}, & \Delta S_t \neq \varnothing, \\ a, & \Delta S_t = \varnothing. \end{cases} \tag{24}$$

Here $b_{C,t} = b_C$, except that a release with at least 16 labels in Γ_t uses $\max\{b_C, 0.90\}$. The later-release branch is selected when $\Delta S_t \neq \varnothing$ and $|K_t| \geq 2|\Delta S_t|$. It uses

$$\alpha_t = a_{\mathrm{l}}, \qquad \boldsymbol{\beta}_t = (0, b_{C,\mathrm{l}}, 0, b_{W,\mathrm{l}})^\top, \qquad \eta_t = b_{R,\mathrm{l}}, \qquad \gamma_t = a s_{\mathrm{l}}. \tag{25}$$

For a later release in the final block, replace γ_t by $\max\{a s_{\mathrm{l}}, a\}$. The full-label coefficient δ_t and pacing coefficient ζ_t are the corresponding classification and pacing weights. These equations retain the configured coefficient values rather than asserting a single fixed profile for all runs.

Scope adjustment and screening. The shared candidate pool is constructed deterministically before method-specific scoring. It is not a feature-centrality prefilter. When release pacing is enabled, the routine first scores candidates with $D_t = \varnothing$ and takes the highest-scoring preliminary batch up to the branch budget. Newly released labels are ordered by their summed predicted relevance over this batch, then by their mean prediction and a stable tie-break key; the configured number of lowest-ranked labels forms D_t. The number of deferred labels is the ceiling of the release size times the configured fraction, bounded below by one and above by the configured cap and release size. The rule is inactive below the configured minimum block size or when the cap is nonpositive. The initial-only restriction disables it once verified-positive labels exist, except for blocks of at least 16 labels. Such blocks also impose a deferred-label fraction of at least 0.25 and a cap of at least five before bounding by the block size. If no preliminary prediction rows are available, the configured stable hash-based label ordering is used instead. A configured filter can then keep candidates with smaller $C_i(D_t)$, subject to the retained-pool bounds. The separate preservation gate retains candidates whose $J_i(S_t \setminus \Delta S_t)$ exceeds its configured threshold, while empty preservation scopes pass automatically. If this gate would remove every candidate, the pre-gate pool is retained. The final retained pool is the normalization pool $\mathcal{C}_t$. After scoring, the highest-scoring candidates up to the configured reranking-pool size are passed to batch construction. Coverage reservations, alternative acquisition paths, and post-selection batch repair are distinct from the initial score defined here.

Retrospective acquisition score. The following definitions specify the score used in the greedy retrospective selection path. All predictions and annotation counts are evaluated after first-time feedback and the intermediate model update. The normalizing pool $\mathcal{C}_t^{\mathrm{hist}}$ is the historical pool entering initial scoring; it is held fixed during greedy selection.

Write $U_i = U_t^{\mathrm{mid}}(x_i)$, $p_{ic} = p_{ic}^{\mathrm{mid}}$, and let K_t^{mid} contain labels with verified positive examples at this stage. The priority set F_t^{hist} consists of the newly released labels when a release occurs, and $S_t \setminus K_t^{\mathrm{mid}}$ otherwise. Let $\mathcal{D}_t^{\mathrm{hist}} = \Gamma_t$ when the current block is nonempty, and $\mathcal{D}_t^{\mathrm{hist}} = S_t$ otherwise. The recovery scope is

$$L_i = \begin{cases} U_i \cap F_t^{\mathrm{hist}}, & U_i \cap F_t^{\mathrm{hist}} \neq \varnothing, \\ U_i \cap \mathcal{D}_t^{\mathrm{hist}}, & \text{otherwise}, \end{cases} \qquad \mathcal{J}_i = \begin{cases} L_i, & L_i \neq \varnothing, \\ U_i, & \text{otherwise}. \end{cases}$$

Here L_i is the recovery scope of the recorded greedy path. It agrees with the main text's priority scope when that scope is nonempty; otherwise the implementation falls back to unresolved labels in the current block. The scarcity scope is constructed from $U_i \cap \mathcal{D}_t^{\mathrm{scar}}$, where $\mathcal{D}_t^{\mathrm{scar}}$ is the first nonempty set among $F_t^{\mathrm{hist}}, \Gamma_t, S_t$. Sort this scope by its verified positive counts, retain the lowest configured fraction, and include all labels tied at the last retained count. Denote the resulting set by $\mathcal{C}_i^{\mathrm{scar}}$.

For a label set L, let $\mathrm{Top}_k(v; L)$ return the labels with the largest up to k values of v_c, retaining all labels if fewer than k are available. Define

$$M_i(L) = \frac{\sum_{c \in \mathrm{Top}_k(p_i; L)} p_{ic}}{|\,\mathrm{Top}_k(p_i; L)|}, \qquad P_i(L) = 1 - \prod_{c \in L} (1 - p_{ic}),$$

$$H_i(L) = \frac{1}{|L|} \sum_{c \in L} h_2(p_{ic}), \qquad W_i(L) = \frac{\sum_{c \in L} p_{ic} (1 + n_{t,c}^{+,\mathrm{mid}})^{-1/2}}{\sum_{c \in L} (1 + n_{t,c}^{+,\mathrm{mid}})^{-1/2}}.$$

Empty-scope statistics are zero. Binary entropy uses the implementation's probability clipping before logarithms. Unsupported directed relationships $\pi_{t,ac}$ have zero contribution. In terms of the historical evidence $r_{t,ic}$ and joint support $e_{t,ic} = p_{ic} r_{t,ic}$ defined in the main text, set

$$R_i(L) = \frac{1}{\binom{|L|}{2}} \sum_{\{a,c\} \subseteq L} p_{ia} p_{ic} \max\{\pi_{t,ac}, \pi_{t,ca}\},$$

$$V_i = \max_{c \in U_i} r_{t,ic}, \qquad E_i = \frac{\sum_{c \in \mathrm{Top}_k(e_{t,i}; L_i)} e_{t,ic}}{|\,\mathrm{Top}_k(e_{t,i}; L_i)|}.$$

The pair score is zero for fewer than two labels, and the evidence aggregates are zero when their indexing sets are empty. Here E_i uses the recovery scope just defined. When the main-text priority scope is nonempty it equals $E_t(x_i)$. The joint-support contribution is added only once.

For each named statistic z_i, normalization is performed separately over $\mathcal{C}_t^{\text{hist}}$:

$$\widehat{z}_i = \begin{cases} (z_i - z_{\min})/(z_{\max} - z_{\min}), & z_{\max} - z_{\min} > \epsilon, \\ 0, & \text{otherwise.} \end{cases}$$

The extrema are over the same named statistic across candidates, even when the candidates have different label scopes. The initial score before adding joint support is

$$\begin{aligned} s_t^{\text{hist}}(x_i) = {} & \widehat{M_i(\mathcal{J}_i)} + \beta_{\text{rec},t}\widehat{M_i(L_i)} + \beta_{\text{hit},t}\widehat{P_i(\mathcal{J}_i)} \\ & + \beta_{\text{all},t}\widehat{P_i(U_i)} + \beta_{\text{unc},t}\widehat{H_i(\mathcal{J}_i)} + \beta_{\text{hist}}\widehat{V}_i \\ & + \beta_{\text{rel}}\widehat{R_i(\mathcal{J}_i)} + \beta_{\text{scarce}}\widehat{W_i(\mathcal{C}_i^{\text{scar}})}. \end{aligned} \tag{26}$$

The score used for selection is

$$\widetilde{s}_t^{\text{hist}}(x_i) = s_t^{\text{hist}}(x_i) + \beta_{\text{joint}}\widehat{E}_i. \tag{27}$$

Thus the base score s_t^{hist} excludes joint support; β_{joint} corresponds to the main-text weight λ_e on this path. The coefficients correspond respectively to the implemented recovery, positive-hit, all-unobserved-hit, uncertainty, historical-evidence, label-relation, joint-support, and scarcity contributions. Their numerical values are supplied by the applicable run configuration, not inferred from constructor defaults. When historical evidence is disabled, both V_i and E_i are zero; disabling uncertainty sets its statistic to zero. The source applies the configured late-round scales to the recovery and hit coefficients. In a final block containing at least sixteen labels, it additionally multiplies the recovery, hit, all-unobserved-hit, and uncertainty coefficients by $1.35, 1.25, 1.15, 0.75$, respectively.

Let $G_{t,m}(x_i; L_i)$ be the positive label-relevance gain used in the main text, with intermediate predictions and initially zero label references. Let $d_{t,m}^{\text{hist}}(x_i)$ be the nearest scaled cosine distance to $\mathcal{B}_m^{\text{hist}}$, initialized to one before any instance is selected. The complete score for a greedy step is

$$A_{t,m}^{\text{hist}}(x_i) = \widetilde{s}_t^{\text{hist}}(x_i) + \beta_{\text{gain}}G_{t,m}(x_i; L_i) + \beta_{\text{dist}}d_{t,m}^{\text{hist}}(x_i). \tag{28}$$

Selection compares this score over the retained remaining candidates. The initial score remains fixed. Selected predictions update the applicable label references, and feature distances are updated when the configured diversity update is enabled. Tie-breaking follows the fixed candidate order. These equations specify the greedy path only: the alternative final-block pair selector and any configured post-selection replacement are separate operations, not algebraically equivalent to this score.

Batch selection and variant names. The main text displays the label-gain and feature-distance parts of greedy selection. Recorded profiles also contain scope-dependent weights, release-specific candidate mixing and optional post-selection repair. These operations are not algebraically interchangeable with a single fixed score. In the historical configuration names, *w/o New-Coverage* refers to the new-image feature-distance switch (`use_coverage=False`), not removal of every coverage-related rule. *w/o Retro-History* disables historical-evidence contributions, including the joint-support statistic in that path; *w/o Retro-Uncertainty* disables its uncertainty contribution. These are configuration descriptions, not claims that all other historical runtime settings have been recovered.

Returned scope versus scoring scope. A first-time query covers S_t. A revisit never overwrites previously verified labels and requests a nonempty subset of the unresolved labels. The full-unresolved rule stated in the main text is the special case $L_q = U_t(x)$. The available current-batch ReMuAL records for E1–E6, not only older variants, contain `query_scope_top_k=5`, `query_scope_final_only=True`, and `query_scope_release_full=False`. These configure a narrowed returned scope in the final block; this cap is distinct from the top-k aggregation of candidate scores. The recorded counts corroborate narrowing: in COCO seed 1 at round 16, 307 revisit calls check 1,535 label entries. Accordingly, the published result records must not be described as uniformly full-unresolved queries. Their performance and feedback remain reported under the recorded scopes. Scope changes do not retroactively verify any label outside a query.

A.2 Training Details

Verified-label objective. Let I_c index examples with a verified judgment for label c at the current training stage, and let $\mathcal{C}_{\text{obs}} = \{c : |I_c| > 0\}$. The recorded ReMuAL loss is

$$\mathcal{L}_{\text{sup}} = \frac{\sum_{c \in \mathcal{C}_{\text{obs}}} w_c |I_c|^{-1} \sum_{i \in I_c} [-a_c y_{ic} \log p_{ic} - (1 - y_{ic}) \log(1 - p_{ic})]}{\sum_{c \in \mathcal{C}_{\text{obs}}} w_c}. \tag{29}$$

Only verified positive and negative entries enter this term. The base current-block weight is 6, the earlier-label weight is 18, and the current-block weight is 12 while a noninitial block still contains labels without verified positives. With n_c^+ and n_c^- denoting the current verified counts, $a_c = \min\{16, \max\{1, (n_c^-/n_c^+)^{0.75}\}\}$ when both counts are nonzero, and $a_c = 1$ otherwise. Labels with no verified entries are omitted. Recorded baseline profiles use unit label weights, no positive rebalancing, and entry-wise normalization. Consequently, sharing a backbone, optimizer and query budget does not establish identical effective losses across methods.

Stage updates and anchoring. Each first-time and final-stage fit starts from the current model parameters but reconstructs its optimizer; a configured scheduler is reconstructed only when enabled. The training seed depends on experiment seed, round and stage. A frozen copy of the incoming model supplies teacher logits for labels in $P_t = S_t \setminus \Gamma_t$. The recorded base anchor weight is 0.75; teacher logits are clipped to $[-8, 8]$ and the Smooth-L_1 parameter is 1. Model parameters are carried between fits, rather than reinitialized from pretraining at every round. Epoch counts are ceilings; stopping uses the configured training-loss threshold and patience, not a held-out validation criterion. Fresh runs begin with an empty verification state. The complete recorded current-batch stage configurations are retained with the numerical supplement; missing historical settings are not inferred from defaults.

Semi-supervised extension. The `torch_ssl` profile samples at most 4,096 images with unchecked released labels. Predictions at least 0.8 or at most 0.2 supply positive or negative pseudo-targets on unchecked entries; their mean binary cross-entropy has weight 0.5. Pseudo-targets affect optimization but do not change B_t^j, $Y_t^{+,j}$, U_t^j or K_t^j for any stage j. Only oracle feedback changes verification status. The recorded SSL experiments use a 40-epoch ceiling rather than the 20-epoch main profile, so cross-profile differences do not isolate the effect of pseudo-labeling.

B Experimental Details

B.1 Data and Release Schedules

Dataset adapters. Experiments operate on a fixed image pool. The main comparisons use VOC 2007 trainval, COCO 2014 train, a Visual Genome object-name vocabulary, and NUS-WIDE train with all 81 concepts. LVIS v1 train supplies a separate long-tail diagnostic. Labels are normalized to lowercase identifiers with spaces replaced by underscores; each adapter constructs image-level positive-label sets as summarized in Table 5. A queried label is assigned its adapter-defined binary target; a label outside the query remains unverified. Evaluation uses this same fixed image pool, not a separate held-out test split.

Table 5: Construction of image-level targets. Counts refer to label identities.

Dataset	Positive-set construction	Labels
VOC 2007	Union normalized XML object names; fixed VOC class list.	20
COCO 2014	Union normalized non-crowd category names; retain category list.	80
Visual Genome	Normalize object names; frequency at least 5; retain top 1,000.	1,000
NUS-WIDE	Align Concepts81 with per-concept Train files; value 1 is positive.	81
LVIS v1	Instance positives; non-exhaustive metadata retained, negative sets omitted.	1,203

COCO uses non-crowd instance annotations; Visual Genome uses normalized object-name entries rather than a guaranteed nonredundant set of semantic concepts. NUS-WIDE uses the per-concept annotation files, not raw user tags. For these adapters, absence from the stored positive set is a negative binary target, not a claim that all real-world semantic absences have been exhaustively

verified. LVIS requires a further distinction: the recorded adapter keeps instance positives and non-exhaustive metadata but omits the category-specific verified-negative sets used by the original benchmark. An absent LVIS positive therefore need not be a verified semantic negative. Its results are image-level adapter diagnostics, not official LVIS instance-detection or segmentation AP.

Label releases and query budgets. Label order is fixed before round 1. Frequency order sorts labels by their positive count over the loaded pool, with normalized names breaking ties. The legacy identifiers `semantic` and `semantic_frontier` mean lexicographic name order here, not semantic-distance ordering. For C labels divided into b blocks, the first $b-1$ blocks contain $\lfloor C/b \rfloor$ labels and the last receives the remainder. Releases occur at rounds 1, 5, 9 and so on; rounds 4, 8 and so on are stage endpoints. All released labels remain in the checklist. Table 6 also records the per-round image-query budget, which must not be confused with the cumulative `budget` field in some exported metric files. E1–E6 records reach their prescribed horizons. Historical configuration and architecture records are interpreted only over their recorded coverage, not automatically assigned a full run.

Table 6: Cumulative release schedules and per-round image-query budgets.

Setting	Order	Block sizes	Releases	Final round	Budget
VOC (E1)	Frequency	5×4	1,5,9,13	16	32
COCO (E2)	Frequency	20×4	1,5,9,13	16	512
COCO (E3)	Name	20×4	1,5,9,13	16	512
VG (E4)	Name	100×10	$1, 5, \ldots, 37$	40	256
NUS-WIDE (E5)	Frequency	20,20,20,21	1,5,9,13	16	512
LVIS (E6)	Frequency	$100 \times 11, 103$	$1, 5, \ldots, 45$	48	200

B.2 Training and Evaluation

Training profiles. Table 7 collects the recorded model, optimizer, batch and epoch settings. Weight names are catalog identifiers; they do not certify the bytes of historical local checkpoints. The label-normalized supervised objective and logit anchor used by ReMuAL are described in Appendix A.2. Baseline loss weighting is not presumed identical simply because model and optimizer names agree. In the current E1–E6 batch, the recorded training-loss stopping threshold is 0.001, patience is 2 and the minimum epoch count is 3; training and prediction worker counts are 8 and automatic mixed precision is enabled. Several historical E1/E2 baseline records do not contain these stopping fields, and they are not backfilled. Additional method-level settings appear in Appendix B.4.

Table 7: Recorded training profiles. Batches are training/prediction; all listed backbones are unfrozen.

Profile	Backbone / weights	Batch	Optimizer	Epochs
E1, E4, E6	ResNet-50 / IN-1K V2	64/256	SGD; lr .01; wd 10^{-4}	20
E2, E3, E5	ResNet-50 / IN-1K V2	128/256	SGD; lr .01; wd 10^{-4}	20
COCO-SSL (E9)	ResNet-50 / IN-1K V2	128/256	SGD; lr .01; wd 10^{-4}	40
VG-SSL (E10)	ResNet-50 / IN-1K V2	64/256	SGD; lr .01; wd 10^{-4}	40
ViT-B/16 check	ViT-B/16 / IN-1K V1	32/128	AdamW; lr 10^{-4}; wd .01	20
ViT-L/16 check	ViT-L/16 / IN-1K V1	16/64	AdamW; lr 10^{-4}; wd .01	20

SGD momentum is 0.9 and the listed profiles have no learning-rate scheduler. For the ViT checks, backbone/head learning rates are $10^{-5}/10^{-4}$; these records remain separate from the ResNet-50 comparisons.

Predictive metrics. Let $\mathcal{C}_{\mathrm{eval}}(L)$ be the labels in scope L with at least one positive evaluation example. At the j-th distinct score threshold, let $P_{c,j}$ and $R_{c,j}$ denote precision and recall for label c. The recorded non-interpolated average precision and macro average are

$$\mathrm{AP}_c = \sum_j (R_{c,j} - R_{c,j-1}) P_{c,j}, \qquad \mathrm{mAP}(L) = \frac{\sum_{c \in \mathcal{C}_{\mathrm{eval}}(L)} \mathrm{AP}_c}{|\mathcal{C}_{\mathrm{eval}}(L)|}. \tag{30}$$

Checklist uses $L = S_t$ and New uses $L = \Delta S_t$ on release rounds. Let $\mathcal{T}_t^{\mathrm{eval}}$ contain the finite Checklist rounds through t, and $\mathcal{T}_t^{\mathrm{rel}}$ the finite release-round New observations. The trajectory summaries are

$$\begin{aligned} \mathrm{AULC}_t &= \frac{1}{|\mathcal{T}_t^{\mathrm{eval}}|} \sum_{r \in \mathcal{T}_t^{\mathrm{eval}}} \mathrm{mAP}_{\mathrm{Checklist},r}, \\ \mathrm{New}_{\max,t} &= \max_{r \in \mathcal{T}_t^{\mathrm{rel}}} \mathrm{mAP}_{\mathrm{New},r}, \qquad \mathrm{New}_{\mathrm{last},t} = \mathrm{mAP}_{\mathrm{New},\max \mathcal{T}_t^{\mathrm{rel}}}. \end{aligned} \tag{31}$$

AULC is the recorded arithmetic mean of Checklist mAP over rounds, not a trapezoidal integral. All E1–E6 trajectories are complete, so the first index set is $\{1, \ldots, t\}$. Empty valid-label sets give missing mAP; labels with no positive examples are omitted, not assigned zero AP. New is missing outside release rounds. New last measures the last block at its release round, not the terminal model re-evaluated on that block. Missing classifier outputs and non-finite scores are mapped to zero before ranking under the recorded evaluator. Neither this rule nor macro averaging fills a missing non-release New observation.

Query and feedback metrics. Let Q_t^{new} and Q_t^{hist} be counts of first-time and revisit calls. Let A_t^{new} count positive first-time label entries for labels without a verified positive before that branch, and let A_t^{front} restrict these entries to Γ_t. Finally, H_t^+ and H_t count positive and total labels checked by revisits. The recorded ratios are

$$\mathrm{NLAE}_t = \frac{A_t^{\mathrm{new}}}{Q_t^{\mathrm{new}}}, \qquad \mathrm{FrontierNLAE}_t = \frac{A_t^{\mathrm{front}}}{Q_t^{\mathrm{new}}}, \qquad \mathrm{RetroHitRate}_t = \frac{H_t^+}{H_t}. \tag{32}$$

The historical field names NAY and FrontierNAY map to NLAE and FrontierNLAE. Activation counts are entries, not numbers of distinct activated categories, so NLAE may exceed one. A zero denominator produces zero by the implementation convention, not an observed negative response. Calls can repeat an image in different rounds. Cumulative calls, checked entries, positive entries and annotation time are different quantities. A cumulative positive-entry fraction $\sum_t H_t^+ / \sum_t H_t$ is also distinct from the final-round RetroHitRate.

B.3 Run Coverage and Statistical Reporting

E1–E5 provide five predictive comparison settings and 24 setting–method combinations. The completed E6 diagnostic adds four combinations. Together these contain 84 recorded runs: seeds 0, 1 and 2 for each of 28 combinations, with 2,088 recorded run-rounds. E1–E5 contribute 72 runs and 1,512 rounds; E6 contributes 12 runs and 576 rounds. Neither NUS-WIDE nor LVIS has a Fixed-label run. E7/E8 query configurations, E9/E10 SSL results and the retained architecture checks are separate seed-0 records. The new E6 runs do not turn these historical experiments into multi-seed comparisons.

Table 8: Run scope of the reported evidence. E6 remains a diagnostic; historical variants are not added to multi-seed totals.

Evidence	Settings	Methods / setting	Seeds	Rounds / run
E1–E3	VOC; COCO, two orders	5	0,1,2	16
E4	VG / name	5	0,1,2	40
E5	NUS-WIDE / frequency	4	0,1,2	16
E6	LVIS / frequency	4	0,1,2	48
E7/E8	Query configurations	6 variants + reference	0	16 / 40
E9/E10	COCO-SSL / VG-SSL	3	0	16 / 40
Backbone checks	COCO, two ViTs	3	0	16

Run provenance and selected versions. The three-run summaries are descriptive. All 28 historical seed-0 runs in E1–E6 lack runtime provenance. In E1–E5, 19 of 24 combinations contain missing historical configuration fields but no conflicting recorded values; the E6 visible configurations agree across seeds apart from the seed. Current-batch seed-1/2 configurations and recorded source hashes agree within each setting and method after accounting for intentional seed fields. For

example, E6 Random's retrospective selector uses seed plus 17; its seed-dependent fingerprint is not itself parameter drift. These observations do not reconstruct missing historical source code, internal overrides or weight bytes. Appendix C.4 reports seeds 1 and 2 separately. Multiple versions with the same seed are not pooled as independent runs; the selected E7–E10 versions are identified in Appendix C.5.

Aggregation and display. For within-run values x_s from n distinct seeds,

$$\bar{x} = \frac{1}{n}\sum_s x_s, \qquad v = \frac{1}{n-1}\sum_s (x_s - \bar{x})^2, \qquad \mathrm{SD} = \sqrt{v}. \tag{33}$$

Where uncertainty is displayed, tables use mean $\pm$ sample SD and bars or bands use ± 1 sample SD, not standard errors or confidence intervals. The compact query-configuration and SSL tables report seed-0 values without SD. Compact query-count tables report descriptive means rounded to integer counts, also without SD. Omission of SD from these displays does not change their source scope; full precision and available uncertainty are retained in the numerical files. No SD is estimated for a single run. All computations precede rounding. Very small positive values must not be interpreted as exact zeros when a three-decimal display rounds to zero.

AULC is computed within each run before averaging across runs; its SD is not the average of the per-round Checklist SDs. Each run also supplies its own New maximum and last-release value before aggregation. In general, the mean of per-run maxima differs from the peak of the mean curve. Cumulative counts are summed within runs and then averaged; ratios are formed within runs before averaging. Changes across rounds partly reflect the expanding evaluation label set and do not isolate forgetting.

Reported improvements. For each setting, round and metric, select a single comparator $b^* = \arg\max_b \bar{x}_b$ from unrounded baseline means, with exact ties resolved by method name. Align observations by recorded seed ID and compute

$$d_s = x_{\mathrm{ReMuAL},s} - x_{b^*,s}, \quad \bar{d} = \frac{1}{n}\sum_s d_s, \quad s_d = \sqrt{\frac{1}{n-1}\sum_s (d_s - \bar{d})^2}. \tag{34}$$

Thus $\bar{d} = \bar{x}_{\mathrm{ReMuAL}} - \max_b \bar{x}_b$; these are differences on the original score scale, not relative percentage changes. Multiplying by 100 converts a difference to percentage points. The displayed improvement SD is computed from the differences, equivalently $s_d^2 = s_x^2 + s_y^2 - 2\widehat{\mathrm{Cov}}(x, y)$, not by adding or subtracting marginal SDs. The comparator is fixed across seeds within a cell. Seed-ID alignment describes observed differences but alone does not certify a matched experimental design. These SDs condition on the selected comparator and do not account for baseline selection uncertainty or establish statistical significance.

B.4 Configuration Coverage

Current-batch settings. The recoverable E1–E6 seed-1/2 provenance includes candidate-pool, selector, allocation and round-training configurations. Table 9 reports selected fields that clarify the implementation rather than substituting constructor defaults for runtime records. The full recorded field snapshots accompany the numerical supplement. Bounds and weights are applied by their configured conditional paths; the table does not assert that every branch executes at every round.

Table 9: Selected recorded ReMuAL settings in the current E1–E6 batch (seeds 1 and 2). Scope caps and score aggregation use different parameters.

Component	Recorded fields	Values
Shared candidate pool	multiplier; size min/max; pool fraction	20; 10,240/32,768; .08
First-time score	positive; count; top-k score; scarcity weights	1; .15; 2.25; 1
Score aggregation	first-time / revisit top-k	5 / 5
Supported-label / distance	active-learning weight; distance weight	4; 1
Retrospective score	recovery; uncertainty; history; joint support	5.5; 1.25; 1.2; 1.75
Retrospective score	hit; all-unobserved hit; relation; scarcity	4.5; 2; 1; .2
Retrospective batch	label-gain; feature-distance weights	1.35; .25
Returned scope	top-k cap; final-only; full at releases	5; true; false
Base budget	maximum old fraction; minimum new fraction	1; 0
Final block budget	maximum old; minimum new	.3; .7
Large final block budget	minimum old; minimum new	.6; .4
Large release budget	maximum old; minimum new	.1; .9
Late large release budget	minimum/maximum old; minimum new	.6/.65; .35

The current first-time selector also records conditional random and CRAB-style batch mixing, plus a final-block pair-selector path. In particular, its base release-random fraction is 1, with initial-large, large, late-large and final-large caps of 0.10, 0.10, 0.25 and 0.35; the large-release CRAB-style fraction is 0.40, with a final-large floor of 0.55. The final-block pair path is enabled with random pair scores and active-label scope. These are recorded conditional settings, not a claim that each executes on every round; the ordinary greedy equations in Appendix A.1 do not replace these branches.

Historical query configurations. The query-configuration results are seed-0 observations: COCO uses checkpoints 8 and 16; VG uses 20 and 40. Midpoint values were supplied from the corresponding metrics records and are retained at six-decimal precision; the displayed table rounds to three decimals. The completed table has no missing midpoint cells. The no-revisit aliases *New-Only diagnostic* and *RALE w/o Retro* have identical recorded endpoint fields and are shown as one *First-time only* row, not two independent repetitions. The full COCO reference comes from E2 seed 0, and the VG reference from the historical E8 record. The variant names specify the intended selection changes; complete historical training and budget-transfer settings have not been recovered for every row. Accordingly, these are recorded configuration comparisons rather than certified one-factor interventions.

Table 10: Display names and historical configuration identities. Variant descriptions do not certify otherwise matched runtime settings.

Displayed configuration	Recorded interpretation
First-time only	No revisit; aliases New-Only diagnostic and RALE w/o Retro.
Random revisits	RALE-New + Random-Old.
Random first-time	Random-New + RALE-Old.
w/o New-Coverage	New-image feature-distance switch disabled.
w/o Retro-History	Historical-evidence contributions disabled in the retrospective scorer.
w/o Retro-Uncertainty	Retrospective uncertainty contribution disabled.
ReMuAL reference	E2 seed 0 on COCO; historical E8 full record on VG.

Semi-supervised settings. E9/COCO and E10/VG use the recorded ResNet-50 SSL profile described in Appendix A.2. Random and Entropy retain their recorded query configurations; ReMuAL combines its query policy with this training profile. Comparisons within an SSL setting must be distinguished from changing a single SSL switch. Pseudo-targets are never counted as oracle-returned labels in the query-feedback tables. E7/E8 runtime fields that were not archived are left unspecified rather than copied from E1–E6.

C Additional Results

C.1 Round-Wise Performance

Figure 5 provides the complete E1–E5 trajectories corresponding to the predictive comparisons in the main text. Panels (a–e) use the three-run descriptive scope; panel (f) uses the separate supplementary VG curve profile, and panels (g,h) use the selected single-run SSL records. Across E1–E5, ReMuAL has the highest terminal mean AULC in all five settings and the highest terminal Checklist mAP in four. On VG, its terminal Checklist mAP is 0.100 compared with Entropy's 0.109, while its AULC is 0.115 compared with 0.097. The pattern is consistent with larger gains while verified supervision is scarce, although this is a stage-wise interpretation rather than a controlled test of supervision size. The E6 diagnostic is reported separately in Appendix C.6; it is not an unshown ninth panel.

The selected VG-SSL record begins below Random in Checklist mAP (0.048 versus 0.051 at round 1) and ends above it (0.092 versus 0.070 at round 40). These single-run observations are not pooled with the multi-run primary comparisons. The supplementary VG curve profile also differs from the historical full reference in the query-configuration table; Appendix C.5 identifies both versions.

Figure 6 reports New mAP only at actual releases. In E1–E5, ReMuAL has the highest mean per-run New maximum and last-release New value. Its corresponding values are 0.367 and 0.040 on NUS-WIDE, and 0.072 and 0.062 on VG. This statement is limited to E1–E5, not extended to the LVIS diagnostic. Each release evaluates a different block, so a decrease across releases is not a measure of forgetting the same categories.

The mean of individual maxima and the maximum of a mean curve are separate summaries. On COCO-name, the former is 0.318 and the mean curve peaks at 0.317 in round 9; mean New last is 0.304 at round 13. For the selected COCO-SSL record, New max is 0.497 at round 5 and New last is 0.169 at round 13. New last is measured at the last release, not by re-evaluating that block with the terminal classifier.

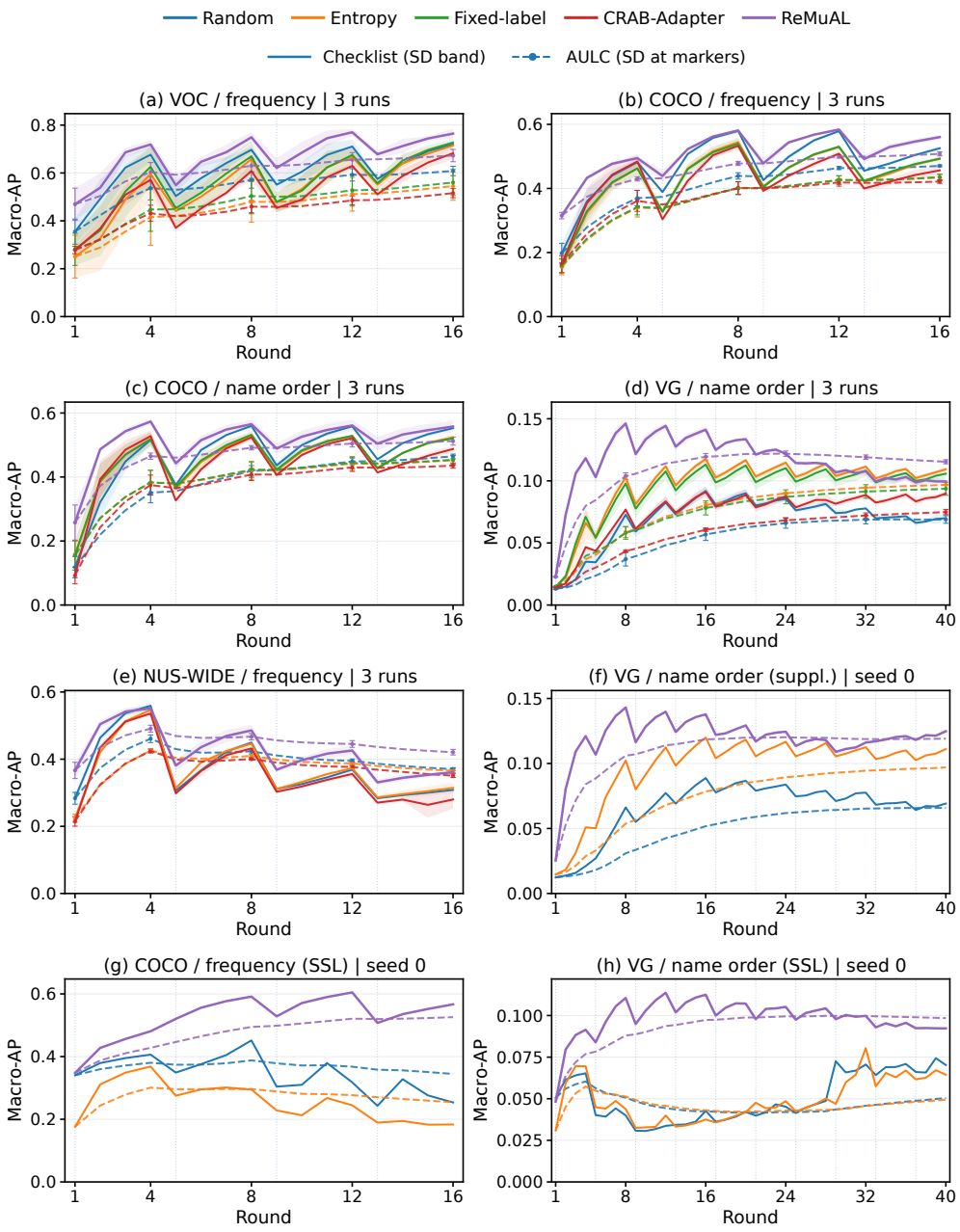


Figure 5: Round-wise Checklist mAP (solid) and cumulative AULC (dashed). Panels (a–e) show descriptive means over available seeds 0–2; Checklist shading and AULC error bars at the marked rounds each show ± 1 sample SD (Appendix B.3). Panels (f–h) retain single-seed observations with no SD estimated. Dotted vertical lines mark releases. Methods use the same colors across result figures; absent methods are not imputed. Releases change the evaluated checklist, so cross-round movement is not a pure measure of learning or forgetting.

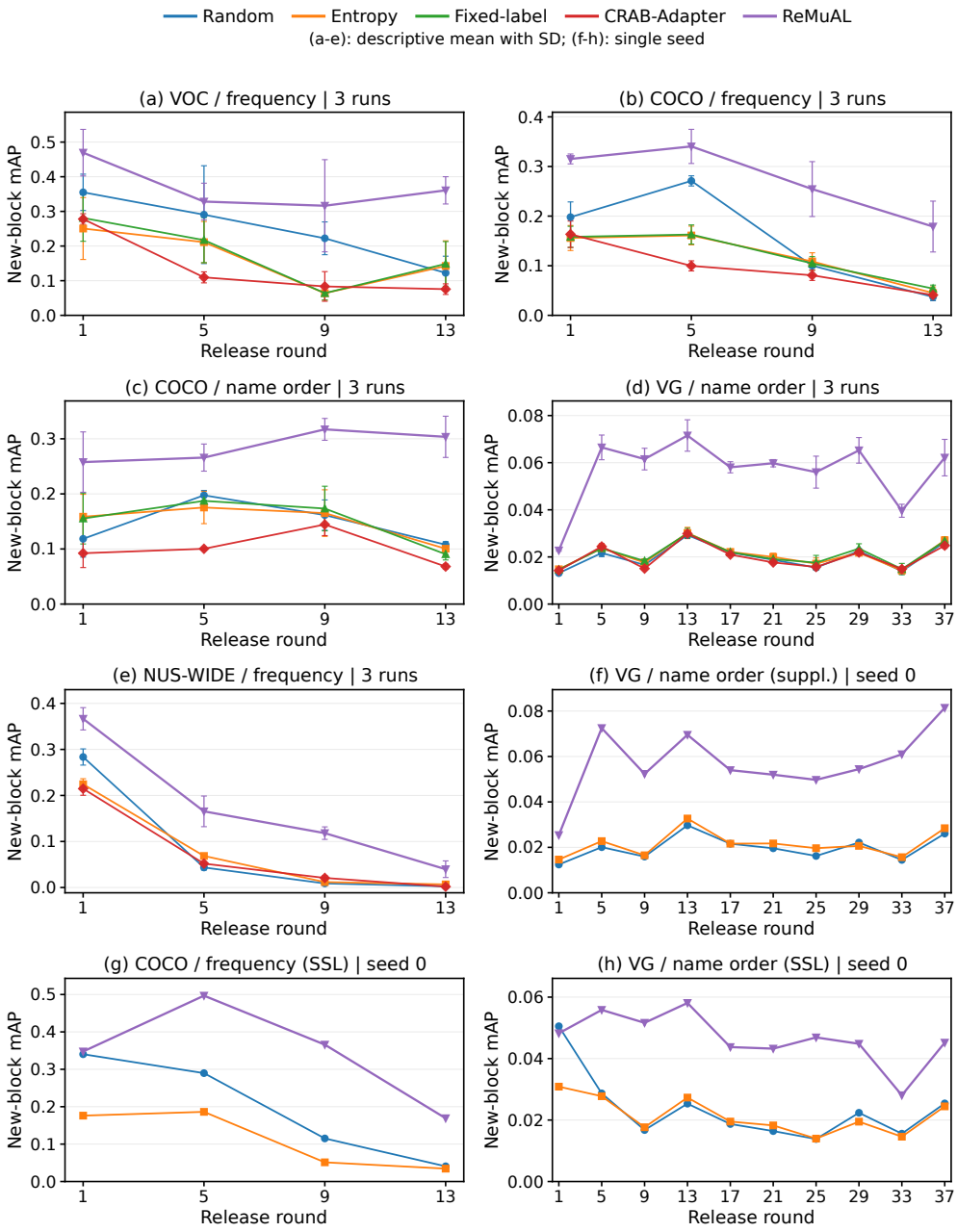


Figure 6: New-block mAP at actual label releases. Panels (a–e) show descriptive means with ± 1 sample-SD error bars over available seeds 0–2; panels (f–h) show single-seed observations without SD estimates. The peak of a mean curve is not necessarily the mean of per-seed New maxima; the rightmost point gives mean New last for the main panels. Unrecorded methods and non-release rounds are not imputed. Each release evaluates a different label block, so cross-release differences reflect both block composition and the contemporaneous model state.

C.2 QUERY OUTCOMES

Query allocation and feedback. Table 11 reports cumulative execution counts at two checkpoints for the main COCO-frequency and VG profiles. It uses means over seeds 0–2, rounded only for display. These are not feedback records for the separate historical E7/E8 variants. On COCO, ReMuAL has made 3,546 first-time calls and 550 revisits by round 8, obtaining 570 positive revisit entries. By round 16, it has made 5,622 first-time calls and 2,570 revisits, checking 39,143 entries and obtaining 2,014 positives. Random and Entropy check approximately 72,893 and 72,947 entries, respectively, with 1,623 and 2,171 positives. Thus ReMuAL's more concentrated feedback in this condition should not be conflated with the largest positive total.

Table 11: Cumulative calls and revisit entries through r: rounded means over seeds 0–2.

COCO	$r = 8$				$r = 16$			
Method	New	Revisit	Checked	Positive	New	Revisit	Checked	Positive
Random	3,484	612	12,233	386	5,726	2,466	72,893	1,623
Entropy	3,484	612	12,233	447	5,694	2,498	72,947	2,171
ReMuAL	3,546	550	11,000	570	5,622	2,570	39,143	2,014
VG	$r = 20$				$r = 40$			
Method	New	Revisit	Checked	Positive	New	Revisit	Checked	Positive
Random	3,596	1,524	249,200	4,328	6,508	3,732	984,167	16,484
Entropy	3,485	1,635	183,000	3,887	6,244	3,996	864,200	14,203
ReMuAL	3,565	1,555	203,433	3,583	6,275	3,965	742,263	11,569

On VG, ReMuAL's rounded cumulative revisit counts are 1,555 and 3,965 at rounds 20 and 40. At round 40, it obtains 11,569 positive entries from 742,263 checked entries, compared with Random's 16,484/984,167 and Entropy's 14,203/864,200. The complete method does not maximize every feedback count or positive fraction. Verified negative entries also contribute supervision. These counts describe executed queries and returned labels, not distinct images, human annotation time or an isolated causal effect of a scoring component.

Totals and final-round ratios. Table 12 preserves the full E1–E5 ReMuAL count summaries, while Table 13 reports final-round ratios, including the completed E6 diagnostic. Counts are summed within runs before averaging; ratios are computed within each final round before averaging. For example, the VOC means of 381 new calls and 131 revisits are identical across the three runs. A zero activation ratio need not mean that a query returns no useful verified labels: the activation metrics count only positive entries for labels without prior verified positives.

Table 12: Cumulative ReMuAL query and label-entry counts in E1–E5, rounded descriptive means over seeds 0–2. Activation counts are entries, not distinct labels.

Condition	New	Revisit	A^{new}	A^{front}	H^{+}	H
VOC	381	131	46	46	53	725
COCO-freq.	5,622	2,570	1,585	1,585	2,014	39,143
COCO-name	5,776	2,416	1,714	1,713	3,003	32,675
VG	6,275	3,965	2,533	2,506	11,569	742,263
NUS-WIDE	5,625	2,567	1,077	1,077	702	40,358

Table 13: Final-round interaction ratios for ReMuAL. Values are descriptive means of per-run ratios over seeds 0–2, not ratios of cumulative mean counts.

Condition	NLAE	FrontierNLAE	RetroHitRate
VOC	0.029	0.029	0.037
COCO-freq.	0.005	0.005	0.096
COCO-name	0.000	0.000	0.192
VG	0.074	0.071	0.076
NUS-WIDE	0.007	0.007	0.025
LVIS diagnostic	0.054	0.000	0.001

Separate supplementary and SSL records. Tables 14 and 15 retain the recorded single-seed counts and terminal ratio fractions. These are not averaged with the primary E1–E6 batch. In the supplementary VG curve profile, ReMuAL obtains 14,876 positive entries using 6,118 revisit calls, compared with Random's 14,992 positives using 3,501 calls. Its positive-entry density is

slightly higher, but positives per revisit call are lower (2.432 versus 4.282). In COCO-SSL, 197 positives from 303 checked revisit entries give a high cumulative positive fraction but a much smaller positive total; the final-round fraction is instead 27/51. Density, total supervision, and call-level yield therefore answer different questions. The available VG-SSL detailed feedback contains only the ReMuAL row; missing baseline feedback is not reconstructed from their mAP curves.

Table 14: Separate supplementary and SSL records (seed 0). N/H are first-time/revisit calls; A/A_f are positive activation entries; H^+/H are positive/checked revisit entries.

Condition	Method	Calls N/H	Activations A/A_f	Entries H^+/H
COCO-SSL corrected	Random	5,632/2,560	1,315/1,315	1,609/76,500
COCO-SSL corrected	Entropy	5,611/2,581	797/797	1,821/65,140
COCO-SSL corrected	ReMuAL	7,889/303	1,553/1,553	197/303
VG-SSL	ReMuAL	6,282/3,958	2,669/2,614	8,577/601,365

Table 15: Final-round ratios for the separate seed-0 records, written as exact numerator/denominator pairs. These are not cumulative fractions.

Condition	Method	NLAE	FrontierNLAE	RetroHitRate
COCO-SSL corrected	Random	0/205	0/205	187/10,960
COCO-SSL corrected	Entropy	0/205	0/205	244/11,420
COCO-SSL corrected	ReMuAL	0/461	0/461	27/51
VG-SSL	ReMuAL	3/103	1/103	74/765

C.3 Results with Different Backbones

We restrict the backbone study to two complete COCO-frequency blocks, each containing ReMuAL and All-Random and All-Entropy controls evaluated within the same backbone block for 16 rounds with seed 0. As summarized in Table 7, the ViT-B/16 block uses the recorded ImageNet-1K V1 catalog identifier, an unfrozen encoder, AdamW, and training/prediction batches of 32/128; the ViT-L/16 block uses the corresponding ImageNet-1K V1 identifier, remains unfrozen, and uses batches of 16/64. Their recorded learning-rate, weight-decay, scheduler, and epoch-ceiling fields are otherwise the same. We therefore compare methods only within each backbone block. Only these two complete single-seed blocks are included; the comparison is not a controlled architecture sweep.

Table 16 reports the two within-backbone comparisons. With ViT-B/16, ReMuAL records 0.609 final checklist mAP, 0.533 AULC, and 0.435 peak release-only mAP, exceeding the stronger control for each metric by 0.029, 0.021, and 0.033, respectively (using unrounded values). With ViT-L/16, the corresponding values are 0.669, 0.607, and 0.556, with margins of 0.026, 0.027, and 0.008. The final and New-maximum margins are smaller than in the ViT-B/16 block, whereas the AULC margin is larger. The terminal retrospective-hit rate is also highest for our system in both blocks (0.151 and 0.356); all ViT-B/16 rows have final-round values of zero on NLAE and FrontierNLAE, whereas only our ViT-L/16 row is nonzero on those rates (both 0.003). These single-seed results support only comparisons among methods evaluated under the same backbone configuration; they do not establish architecture-level robustness or a causal effect of changing the backbone.

Table 16: COCO-frequency backbone comparisons (16 rounds, seed 0). Final and AULC use Checklist mAP; New max includes its release round. NLAE, Frontier, and Retro are final-round ratios. Bold marks the unique maximum within each backbone and metric.

Backbone	Method	Final	AULC	New max (r)	NLAE	Frontier	Retro
ViT-B/16	ReMuAL	**0.609**	**0.533**	**0.435 (5)**	0.000	0.000	**0.151**
ViT-B/16	All-Random	0.580	0.513	0.402 (5)	0.000	0.000	0.017
ViT-B/16	All-Entropy	0.406	0.382	0.253 (1)	0.000	0.000	0.031
ViT-L/16	ReMuAL	**0.669**	**0.607**	**0.556 (5)**	**0.003**	**0.003**	**0.356**
ViT-L/16	All-Random	0.643	0.581	0.549 (5)	0.000	0.000	0.017
ViT-L/16	All-Entropy	0.479	0.453	0.287 (1)	0.000	0.000	0.029

C.4 CURRENT-BATCH-ONLY COMPARISON

Table 17 reports E1–E5 using only seeds 1 and 2. ReMuAL retains the highest mean cumulative AULC at every round in these five settings and the highest mean New max and New last. Terminal Checklist mAP is highest in VOC, COCO-frequency and NUS-WIDE; COCO-name and VG have different terminal rankings. Across all rounds, Checklist leads in 16/16, 15/16, 13/16, 30/40 and 14/16 rounds in the listed setting order. This comparison excludes historical seed 0 rather than asserting that its provenance has been recovered. It is a two-run descriptive comparison, not a significance test.

Table 17: Current-batch-only comparison: mean ± sample SD over seeds 1 and 2 ($n = 2$). Checklist mAP and AULC use the full trajectory endpoint; New max and New last are computed within each seed before aggregation. Bold marks the highest mean within each setting and metric. Historical seed 0 is excluded; no Fixed-label run is available for NUS-WIDE.

Setting	Method	Checklist	AULC	New max	New last
VOC frequency	Random	0.712 ± 0.000	0.606 ± 0.027	0.385 ± 0.010	0.148 ± 0.030
	Entropy	0.720 ± 0.011	0.559 ± 0.077	0.285 ± 0.095	0.176 ± 0.061
	Fixed-label	0.720 ± 0.011	0.559 ± 0.077	0.285 ± 0.095	0.176 ± 0.061
	CRAB-Adapter	0.696 ± 0.015	0.523 ± 0.021	0.279 ± 0.022	0.083 ± 0.010
	ReMuAL	**0.755** ± 0.017	**0.658** ± 0.007	**0.440** ± 0.062	**0.342** ± 0.031
COCO frequency	Random	0.524 ± 0.009	0.472 ± 0.000	0.272 ± 0.014	0.033 ± 0.001
	Entropy	0.491 ± 0.008	0.440 ± 0.003	0.175 ± 0.005	0.051 ± 0.008
	Fixed-label	0.491 ± 0.008	0.440 ± 0.003	0.175 ± 0.005	0.051 ± 0.008
	CRAB-Adapter	0.446 ± 0.000	0.425 ± 0.002	0.179 ± 0.003	0.038 ± 0.008
	ReMuAL	**0.560** ± 0.011	**0.508** ± 0.009	**0.342** ± 0.047	**0.170** ± 0.069
COCO name order	Random	**0.551** ± 0.000	0.469 ± 0.002	0.197 ± 0.010	0.107 ± 0.007
	Entropy	0.519 ± 0.002	0.461 ± 0.002	0.201 ± 0.017	0.095 ± 0.009
	Fixed-label	0.519 ± 0.002	0.461 ± 0.002	0.201 ± 0.017	0.095 ± 0.009
	CRAB-Adapter	0.482 ± 0.005	0.440 ± 0.001	0.137 ± 0.021	0.071 ± 0.000
	ReMuAL	0.548 ± 0.006	**0.505** ± 0.003	**0.306** ± 0.006	**0.284** ± 0.024
VG name order	Random	0.070 ± 0.003	0.070 ± 0.002	0.029 ± 0.001	0.026 ± 0.000
	Entropy	**0.108** ± 0.002	0.097 ± 0.000	0.029 ± 0.001	0.026 ± 0.001
	Fixed-label	**0.108** ± 0.002	0.097 ± 0.000	0.029 ± 0.001	0.026 ± 0.001
	CRAB-Adapter	0.091 ± 0.001	0.076 ± 0.001	0.030 ± 0.001	0.025 ± 0.002
	ReMuAL	0.100 ± 0.001	**0.116** ± 0.001	**0.073** ± 0.009	**0.060** ± 0.009
NUS-WIDE frequency	Random	0.308 ± 0.002	0.373 ± 0.001	0.291 ± 0.017	0.003 ± 0.001
	Entropy	0.309 ± 0.018	0.363 ± 0.005	0.228 ± 0.014	0.003 ± 0.001
	CRAB-Adapter	0.296 ± 0.003	0.353 ± 0.006	0.220 ± 0.016	0.002 ± 0.000
	ReMuAL	**0.359** ± 0.008	**0.417** ± 0.007	**0.357** ± 0.025	**0.040** ± 0.026

Table 18 separately adds the E6 current-batch diagnostic. ReMuAL has the highest full-trajectory AULC and mean per-run New maximum, whereas the highest terminal Checklist mAP is Entropy's and the highest New last is CRAB-Adapter's. Consequently, the E1–E5 claim about New last is not extended to E6. Very small last-release values are shown on a scaled column rather than rounded to indistinguishable zeros.

Table 18: LVIS current-batch diagnostic, mean ± sample SD over seeds 1 and 2. Checklist and AULC use round 48; New last uses release round 45. The last column is scaled; bold marks the highest mean.

Method	Checklist	AULC	New max	New last ($\times 10^{-4}$)
Random	0.028 ± 0.000	0.068 ± 0.002	0.020 ± 0.000	0.58 ± 0.03
Entropy	**0.045** ± 0.004	0.080 ± 0.007	0.030 ± 0.002	0.55 ± 0.21
CRAB-Adapter	0.045 ± 0.001	0.069 ± 0.001	0.020 ± 0.000	**0.99** ± 0.37
ReMuAL	0.032 ± 0.001	**0.094** ± 0.003	**0.064** ± 0.006	0.40 ± 0.13

C.5 Query-Configuration and SSL Record Versions

The seed-0 query-configuration table has both checkpoints for all seven rows: COCO rounds 8 and 16, VG rounds 20 and 40. Midpoints use the six-decimal values supplied from the corresponding metrics records; endpoints retain their original transcription. The full VG reference has AULC 0.117432 and 0.117187 at the two checkpoints, both displayed as 0.117. These are not missing or duplicated cells.

Table 19 distinguishes the query-reference version from the supplementary VG curve profile and identifies the selected COCO-SSL record. Older same-seed versions are neither substituted nor pooled. The selected VG-SSL endpoint is 0.092 Checklist mAP and 0.098 AULC. The SSL two-checkpoint table and curves use matching selected versions; full precision and source identities accompany the numerical records.

Table 19: Selected seed-0 record versions. Different versions with the same seed are not independent repeats. Paired checkpoint entries are in chronological order.

Record	Checkpoint	Checklist	AULC	Use
COCO full reference	8 / 16	0.578 / 0.562	0.476 / 0.507	Query configuration table
VG historical full	20 / 40	0.127 / 0.124	0.117 / 0.117	Query configuration table
VG curve profile	40	0.125	0.119	Supplementary curves
COCO SSL selected	16	0.567	0.526	SSL table and curves
COCO SSL older	16	0.367	0.370	Not pooled
COCO SSL older	16	0.582	0.528	Not pooled

Meaning of the configuration comparisons. The variant definitions in Table 10 identify the changed selection rules. Both random replacement configurations lie below the full reference in Checklist mAP and AULC at the two checkpoints, but their relative order differs between COCO and VG. The historical runtime records do not establish that every other training, query-scope and budget-transfer condition was held fixed. Similarly, the SSL results compare methods within a recorded semi-supervised profile; they are not an SSL-on/off intervention. The recorded 40-epoch ceiling and method-specific training settings preclude attributing all cross-profile differences to pseudo-labeling alone.

C.6 LVIS Long-Tail Diagnostic

E6 contains four methods, seeds 0–2, and 48 recorded rounds per run. Labels are released in decreasing frequency order in twelve blocks, with the final 103-label block arriving at round 45. The main-text checkpoints are rounds 24 and 48. At round 24, ReMuAL has Checklist mAP 0.088 and AULC 0.139; at round 48, the values are 0.033 and 0.095. It has the highest mean Checklist mAP over rounds 1–24 and the highest mean cumulative AULC over all 48 rounds. The stage-wise pattern is consistent with stronger benefits from targeted querying under limited early supervision, while later comparisons also reflect accumulated annotations and increasingly infrequent categories; it does not isolate any one of these factors.

Table 20: LVIS diagnostic, descriptive mean ± sample SD over seeds 0–2. Checklist and AULC use round 48; New last uses release round 45. The last column is scaled; bold marks the highest mean.

Method	Checklist	AULC	New max	New last ($\times 10^{-4}$)
Random	0.028 ± 0.001	0.068 ± 0.001	0.020 ± 0.000	0.58 ± 0.02
Entropy	0.044 ± 0.004	0.079 ± 0.006	0.030 ± 0.002	0.68 ± 0.26
CRAB-Adapter	**0.045** ± 0.001	0.069 ± 0.001	0.020 ± 0.001	**0.95** ± 0.27
ReMuAL	0.033 ± 0.002	**0.095** ± 0.003	**0.062** ± 0.006	0.36 ± 0.11

Table 20 also reports release-based summaries. The mean per-run New maximum is highest for ReMuAL, but the final released block has very low New mAP for every method and CRAB-Adapter has the highest mean on that block. The column scaled by 10^{-4} preserves the distinction between small positive scores and exact zeros. Table 21 reports the available cumulative image-query counts at the same two checkpoints. The E6 export contains these call counts and scalar interaction ratios but not the per-call checked/positive-entry records needed to reconstruct full feedback totals; such totals are not inferred from a ratio.

These results use the image-level adapter in Appendix B.1, not official LVIS instance AP or frequency-group AP. The missing verified-negative sets remain a limitation of that adapter.

Table 21: Cumulative LVIS image-query calls through two checkpoints, rounded means over seeds 0–2. These call counts do not determine the number of checked label entries.

Method	$r = 24$		$r = 48$	
	New	Revisit	New	Revisit
Random	3,252	1,548	5,899	3,701
Entropy	3,239	1,561	5,669	3,931
CRAB-Adapter	3,231	1,569	5,616	3,984
ReMuAL	3,220	1,580	5,806	3,794