ON-DEMAND HASH EXPANSION FOR GROWING CROSS-MODAL DATABASES

Anonymous authors
Paper under double-blind review

ABSTRACT

In growing cross-modal databases, adapting representations to continuous data arrivals while maintaining historical search indexes poses a persistent challenge. Although new arrivals provide supervision to improve representations, materializing newly learned hash bits requires an encoding and writing pass over historical entries. We introduce On-Demand Elastic Hashing (ODEH), which expands ranking capacity without widening the stored search index. ODEH retrieves candidates via a compact base index and ranks them by complete Hamming distance using residual bits generated on demand from frozen backing features. This decoupled design preserves lightweight index scans while trading historical code writes for retained feature storage and candidate computation. Residual blocks are trained against candidate ranking boundaries with earlier bits frozen. A two-tier rule selects the shortest extension meeting a prescribed Top-K quality target, falling back to statistically tested improvement when no extension meets the target. Experiments on COCO, NUS-WIDE, and MIRFlickr compare ODEH with six external baselines using shared frozen features. ODEH achieves the highest mean in five of six dataset–direction pairs for each of Precision@10, nDCG@10, and query success, while maintaining a 32-bit base index.

1 INTRODUCTION

Cross-modal hashing maps images and text to compact binary codes for semantic retrieval (Jiang & Li, 2017; Chen et al., 2024; Fan & Cao, 2025). As a database grows, arriving data provide new supervision, while historical items retain codes produced by earlier models. Online and lifelong hashing update representations as data arrive (Zhan et al., 2022; Xu et al., 2024; Kang et al., 2025; 2026), and bit-extendable methods increase code capacity (Wang et al., 2025; Liu et al., 2026).

Making newly learned bits available for historical retrieval creates a deployment problem. When an encoder generates these bits from item features, storing them for the historical gallery requires a new encoding and writing pass. Searching only the original codes leaves the added capacity unused. The challenge is to use the new bits for historical items while keeping their stored search codes unchanged.

We propose *On-Demand Elastic Hashing* (ODEH), which separates the codes used to find candidates from those used to rank them. ODEH keeps short base codes in the search index and retains fixed continuous feature vectors in a separate store. For each query, the base codes select a shortlist; mappings trained on arriving relevance labels generate extra bits from the query and candidate features. Ranking uses the base and additional bits together, allowing historical items to benefit from new capacity without storing longer codes (Figure 1). This design trades historical code writes for feature storage and candidate computation.

ODEH preserves existing bits, so removing an extension recovers the prior code exactly. New bits are trained to place a required number of relevant items in Top-K. With the feature extractor and earlier encoders frozen, training focuses on the relevant and irrelevant candidates whose ordering determines success. The fraction of unsuccessful queries defines the failure rate for length selection. ODEH retains the fewest added bits meeting a target failure rate in both directions; if none qualifies, it accepts a tested improvement or keeps the existing code.

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

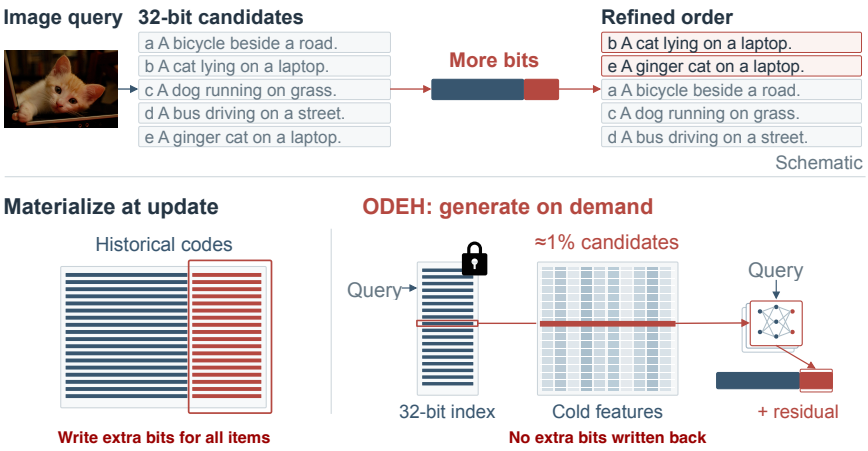


Figure 1: New bits refine candidate rankings. Materialized expansion writes them for all historical items; ODEH generates them for the query and shortlisted candidates. Texts and rankings are schematic.

Under a shared-feature protocol with three training seeds, ODEH achieves the highest mean against six external methods in five of six dataset–direction pairs for each of Top-10 precision, normalized discounted cumulative gain, and query success.

Our contributions are:

- An architecture that learns and serves additional hash bits without rewriting historical search codes or changing earlier coordinates.
- A candidate-boundary objective and quality-driven rule for selecting the retained code length.
- Bounds on failures that added bits cannot repair, alongside matched evaluations of retrieval quality, historical writes, feature storage, and query computation.

2 Related Work

Cross-modal and continual hashing. Cross-modal hashing learns common binary codes through joint feature and code learning (Jiang & Li, 2017), visual–textual prompts (Chen et al., 2024), discriminative boundaries (Qin et al., 2025), or globally informed hard negatives (Cheng et al., 2026). As data arrive, DLCH preserves database codes (Xu et al., 2024), while LCDH transfers offline knowledge online (Li et al., 2025b). SILCH and MCDLCH address task-scale variation and multi-scale consistency, respectively (Yang et al., 2026a; Xu et al., 2025a). POLISH and OH-ELS handle expanding classes (Zhan et al., 2024; Fan et al., 2026), while FPO-CMH addresses partial-modal streams (Li et al., 2025a). For code expansion, BITS duplicates historical codes (Wang et al., 2025), PBEH compares extended queries with shorter stored codes by truncation (Liu et al., 2026), and RCMH regenerates codes at a specified length (Jiang et al., 2026). MHR's residual adaptors over a frozen encoder's hash logits can change earlier signs (Hua & Xiao, 2026). ODEH freezes earlier coordinates and computes historical-item extensions only after candidate retrieval.

Adaptive and cascaded retrieval. Nested representations expose multiple retrieval budgets: MRL and AdANNS use continuous prefixes (Kusupati et al., 2022; Rege et al., 2023), while NHL learns binary prefixes (He et al., 2025). AdaWidth selects embedding width per query (Yang & Zhao, 2026). DiskANN places compressed vectors in RAM and full vectors on SSD (Jayaram Subramanya et al., 2019). Extended RaBitQ refines distance estimates with more quantization bits (Gao et al., 2025); MRQ progressively evaluates projected and full vectors (Yang et al., 2026b). ODEH retains features to generate new binary coordinates from arriving supervision and selects one shared length across queries. These coordinates refine candidate rankings rather than approximate a fixed vector metric at higher precision.

Risk-controlled selection. Retrieval risk can be controlled through set calibration for probabilistic coverage (Cai et al., 2025) or joint treatment of retrieval and ranking (Xu et al., 2025b). Conformal Risk Control calibrates a parameter governing a bounded monotone loss (Angelopoulos et al., 2024); Learn then Test supports more general configuration families through hypothesis testing (Angelopoulos et al., 2025). Because prefix failure need not decrease with length, ODEH calibrates a finite family of frozen prefixes using binomial bounds for the quality target (Clopper & Pearson, 1934) and paired tests for improvement.

3 METHOD

ODEH uses a base index to retrieve candidates and learns additional bits to refine their ranking. A quality rule determines how many new bits to retain (Figure 2).

3.1 SETTING AND NOTATION

We consider a database that grows through paired image–text arrivals. Training uses labels and all previously arrived training data, excluding evaluation queries and future arrivals. Queries retrieve items in the other modality: image-to-text (I2T) or text-to-image (T2I). Query sources are disjoint from the evaluation gallery.

The initial hasher produces a *base hash code* h_0 of width b_0, stored in the index for candidate search. Expansions add *residual bits* computed from each item's features. The base code and retained residual bits form the existing *logical code* h_{old}. Their encoding functions and the feature extractor ϕ remain frozen while a new block is learned.

A query succeeds when its returned list contains enough relevant items. In our Top-10 setting, at least five relevant results are required whenever ten or more are available in the gallery. Relevance is defined by semantic-label overlap, with captions inheriting their source image's labels.

For one snapshot and retrieval direction, let $\mathcal{D}$ be the target gallery and $\mathcal{P}(q)$ its items relevant to query q. The gallery contains at least K items. Eligible queries have a relevant gallery item and follow distribution Q. For an increment of d bits, $T_{K,d}(q)$ denotes the returned Top-K list. The required relevant count is $k(q) = \lceil \rho \min\{K, |\mathcal{P}(q)|\} \rceil$, where $\rho \in (0, 1]$; we use $K = 10$ and $\rho = 0.5$. This count decreases when relevant items are scarce.

Query failure and its population risk are

$$Z_d(q) = \mathbf{1}\{|T_{K,d}(q) \cap \mathcal{P}(q)| < k(q)\}, \qquad R(d) = \mathbb{E}_{q\sim Q}[Z_d(q)]. \tag{1}$$

Using a fixed candidate fraction and risk target ε, ODEH selects an increment from a finite set $\mathcal{B}$ containing zero. The selected increment applies to every query in the snapshot.

3.2 FROZEN COORDINATES AND ON-DEMAND RETRIEVAL

At each expansion, we train a new residual head for each modality while freezing the feature extractor, base hasher, and retained heads. For an image or text item z, the new head maps its fixed feature vector $\phi(z)$ to a binary block $r(z)$. A prefix of length d consists of the first d output bits, which extend the old code as

$$h_d(z) = [h_{\text{old}}(z); r_{1:d}(z)]. \tag{2}$$

A zero-length prefix keeps the existing code. Appendix B gives the head architecture and recurrence across growth stages.

For a query q, ODEH first scans the base codes to retrieve a shortlist $\mathcal{C}(q)$ of the nearest items in Hamming distance. It then reads their features and evaluates the residual heads for the query and candidates. Ranking by the complete distance $d_H(h_d(q), h_d(z))$ gives the returned Top-K list. Both steps break distance ties by item identifier.

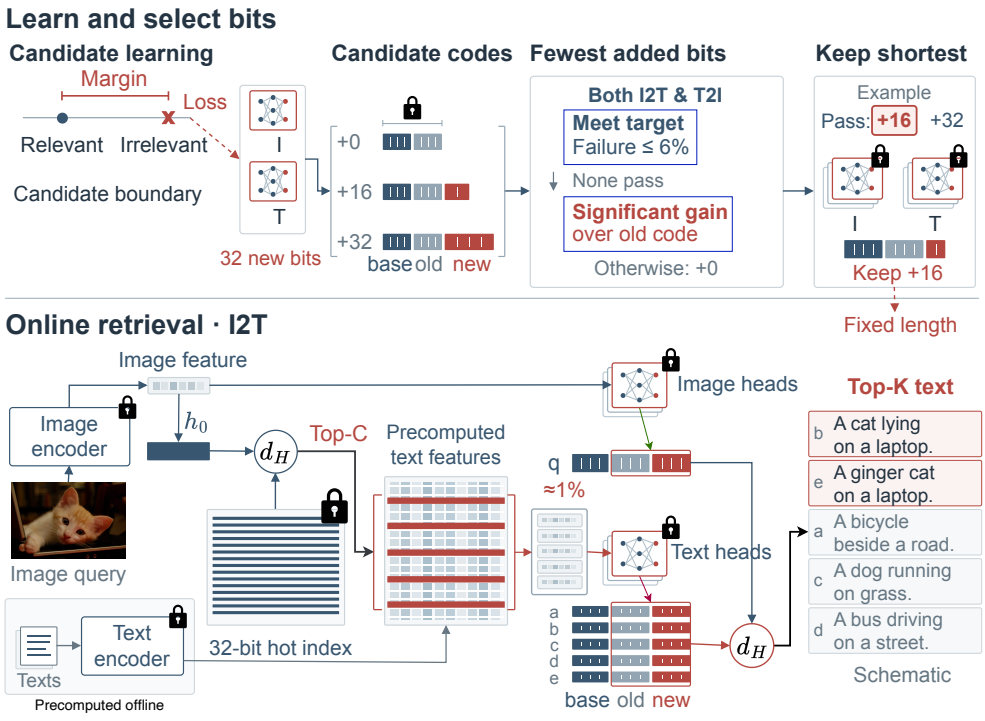


Figure 2: Top: residual learning and length selection on Validation; gain denotes passing the paired-improvement criterion. Bottom: frozen heads generate old and new residual bits for the query and candidates. Locks denote frozen encoding functions; d_H denotes Hamming distance. Texts and rankings are schematic.

Every proposed length uses this same shortlist. A fixed candidate fraction f determines its size, bounded by K and the gallery size: $C = \min\{|\mathcal{D}|, \max\{K, \lceil f|\mathcal{D}|\rceil\}\}$. Complete distance counts disagreements across the base and all retained residual bits: $d_H(u, v) = (b - u^\top v)/2$ for sign vectors $u, v \in \{-1, +1\}^b$.

New arrivals add base-code and feature rows; accepting a residual block leaves historical codes untouched. Storage comprises $|\mathcal{D}|\lceil b_0/8\rceil$ base-code bytes, identifiers, features, and retained heads. Features held in RAM count toward resident memory.

The candidate fraction f controls feature reads and residual encoding, while base-code search scans the gallery. All positive prefixes share feature reads, parameters, and full head computation because the output is evaluated before slicing. Prefix length controls logical capacity and distance operations.

3.3 Learning residual bits for Top-K retrieval

The new residual block is trained to place at least $k(q)$ relevant candidates among the Top-K results, as required by Equation 1. For example, when $K = 10$ and $k = 5$, making the fifth nearest relevant candidate strictly closer than the sixth nearest irrelevant candidate ensures at least five relevant results in Top-10. This comparison defines the ranking boundary used for learning.

For one query and length, write $k = k(q)$. We compute the boundary loss when $\mathcal{C}(q)$ contains at least k relevant and $K - k + 1$ irrelevant items. Let D_j^+ and D_j^- be the j-th smallest complete Hamming distances among these relevant and irrelevant candidates, respectively. The boundary margin and loss are

$$M_d(q) = D^-_{K-k+1} - D^+_k, \qquad \ell_d(q) = \log\Big(1 + e^{-M_d(q)}\Big). \tag{3}$$

Minimizing this loss encourages a positive margin, which guarantees $Z_d(q) = 0$ under any tie ordering.

To optimize this loss, we retrieve fixed base-code candidates from the arrived training gallery and compute complete-code distances. A straight-through estimator (Bengio et al., 2013) retains exact binary forward distances while passing gradients through continuous head outputs. The boundary loss is averaged over queries satisfying the candidate counts above, with equal weight for each direction and positive prefix. Regularization uses one sampled image–text pair per source and epoch to penalize cross-modal disagreement and quantization error, and to encourage balanced, decorrelated bits (Weiss et al., 2008). Appendix B specifies the full objective and hyperparameters.

Candidate coverage and code length limit which errors can be corrected. A shortlist with fewer than k relevant items fails at every length. For queries meeting the boundary-loss conditions, appending d bits changes the margin by at most d, so an old margin below $-d$ cannot be repaired by that increment. All queries remain in evaluation. Appendix C gives the gradient rule and proofs.

3.4 Selecting a shared length for each snapshot

Selection during growth. A shared increment gives all queries one code width; separate directional checks prevent acceptance based only on a favorable average. At each stage, ODEH evaluates increments $\mathcal{B} = \{0, 16, 32\}$ from a trained 32-bit residual block on held-out development queries (Validation) with the same candidate fraction. It keeps the shortest increment with empirical failure at most $\varepsilon = 0.06$ in both directions. Otherwise, it selects the shortest positive increment passing the paired-improvement test in both directions, or retains the old code if none passes. Choosing zero discards the new block.

The improvement test compares each positive increment with the current retained code using queries whose success status changes. For direction g, let $G_g(d)$ count queries changing from failure to success and $H_g(d)$ count the reverse. The exact binomial criterion (Fagerland et al., 2013) is

$$G_g(d) > H_g(d), \qquad \Pr\big\{\mathrm{Binom}\big(G_g(d) + H_g(d), \tfrac{1}{2}\big) \geq G_g(d)\big\} \leq \frac{0.05}{36}. \tag{4}$$

Passing this test is recorded separately from meeting the 6% target. Selection targets query-level success, which need not maximize mean precision or discounted gain.

The divisor 36 counts three seeds, three stages, two positive increments, and two directions. Reusing Validation makes this cutoff a development criterion, also applied to longer growth trajectories.

Independent terminal selection. After growth, independent Calibration queries reselect only the last-stage increment from $\{0, 16, 32\}$, using the frozen code entering that stage and its trained residual block. Earlier increments stay fixed; Test evaluates the resulting state.

Calibration follows the same two-tier order, replacing empirical failure rates with one-sided Clopper–Pearson risk bounds (Clopper & Pearson, 1934; Angelopoulos et al., 2025) and the improvement-test cutoff with $\beta = \alpha/J$. Here $\alpha = 0.05$ is the joint error budget and J counts all prespecified bounds and tests. Certification assumes i.i.d. queries from Q, conditional on models, galleries, and search settings frozen before Calibration access. Allocating β to each bound and test gives a joint $1 - \alpha$ guarantee by a union bound. Appendix D gives the allocation and proof.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We used MS-COCO, NUS-WIDE-21, and MIRFlickr-24 (Lin et al., 2014; Chua et al., 2009; Huiskes & Lew, 2008) with frozen CLIP features (Radford et al., 2021): ViT-B/16 for COCO and ViT-B/32 for the other datasets (Dosovitskiy et al., 2021). A source is one image with about five captions (COCO) or one tag set (NUS-WIDE/MIRFlickr). I2T searches texts; T2I searches images. Shared semantic labels define relevance; query sources are excluded from galleries. Growth uses 2,500 and 7,500 sources before reaching 22,500 on COCO/NUS-WIDE or 14,983 on MIRFlickr. Appendix A gives splits and gallery sizes.

Seeds 0/1/2 repeat complete training with shared arrival order and evaluation text. Test has 4,952/5,000/2,500 sources per direction on COCO/NUS-WIDE/MIRFlickr. We report precision (P@10), normalized discounted cumulative gain (nDCG@10) (Järvelin & Kekäläinen, 2002), and query failure (Equation 1) as three-seed means and sample SDs. ODEH uses 32-bit base codes and 1% candidates. Growth and learning controls use Validation; paired intervals resample query sources (Efron, 1979), conditional on the fitted models.

Table 1: Final Test quality under the shared-feature protocol (%, mean ± sample SD; I2T / T2I). Bits gives logical width, with ranges across seeds. ODEH stores 32-bit hot codes plus features and heads.

Method	Bits	P@10↑	nDCG@10↑	Failure↓
COCO				
PBEH †	64–65	93.77±1.85 / 94.98±0.13	93.86±1.88 / 95.08±0.14	3.93±1.55 / 3.38±0.16
BITS-CM †	128	95.11±0.05 / 93.73±0.08	95.16±0.03 / 93.80±0.07	**2.86**±0.10 / 4.99±0.02
OH-ELS	64	71.99±2.23 / 68.46±1.79	72.00±2.26 / 68.57±1.74	25.98±2.78 / 31.99±1.87
MCDLCH	64	53.98±0.78 / 61.61±1.51	53.97±0.34 / 61.47±1.45	47.44±0.43 / 38.64±1.45
DGHDGH S	64	90.31±1.66 / 90.03±1.73	90.41±1.73 / 90.30±1.69	6.56±1.12 / 6.75±1.45
DDBH S	64	82.04±1.88 / 76.61±3.82	81.58±2.58 / 76.71±4.25	12.90±0.56 / 17.70±0.79
ODEH †	80	**95.76**±0.12 / **95.29**±0.29	**95.85**±0.12 / **95.51**±0.23	2.90±0.10 / **3.15**±0.24
NUS-WIDE				
PBEH	64	81.90±1.23 / 75.95±1.96	82.16±1.19 / 76.12±1.91	12.12±1.00 / 19.47±2.02
BITS-CM	32	86.64±0.13 / 80.10±0.18	86.67±0.15 / 80.17±0.18	11.85±0.13 / 18.53±0.22
OH-ELS	64	86.51±1.45 / 75.62±0.97	86.49±1.47 / 75.62±1.01	13.15±1.57 / 24.21±0.97
MCDLCH	64	68.94±1.01 / 66.15±3.93	68.13±0.50 / 65.96±4.07	26.71±4.97 / 28.86±2.33
DGHDGH S	64	85.32±0.44 / 81.15±0.30	85.41±0.39 / 81.20±0.30	8.95±0.32 / 13.88±0.85
DDBH S	64	78.56±1.56 / 73.09±0.98	76.94±2.00 / 73.08±0.51	12.33±0.13 / 18.43±1.36
ODEH	80	**86.80**±1.76 / **83.72**±1.12	**87.30**±2.04 / **84.42**±1.11	**7.00**±0.90 / **9.06**±0.73
MIRFlickr				
PBEH	64–65	92.60±0.75 / 87.19±0.22	92.78±0.74 / 87.32±0.16	3.39±0.54 / 8.56±0.21
BITS-CM	32	95.92±0.11 / 87.56±0.17	95.91±0.08 / 87.59±0.14	3.65±0.18 / 11.12±0.28
OH-ELS	64	**96.95**±0.13 / 92.39±0.04	**96.96**±0.11 / 92.41±0.04	3.05±0.27 / 7.57±0.08
MCDLCH	64	90.13±1.12 / 83.55±0.55	90.01±1.05 / 83.19±0.68	6.41±0.27 / 12.72±0.98
DGHDGH S †	64	95.59±0.09 / 92.74±0.07	95.66±0.13 / 92.80±0.04	1.87±0.32 / 5.09±0.12
DDBH S	64	85.18±1.19 / 80.18±5.07	85.29±1.87 / 78.88±7.61	6.72±1.89 / 12.31±3.23
ODEH †	48–64	95.54±0.86 / **93.58**±0.63	95.73±0.89 / **93.64**±0.72	**1.67**±0.23 / **3.33**±0.29

†: empirical failure is at most 6% in both directions in every seed. S: full retraining; other methods preserve historical prefixes. External methods: full scan. ODEH: 1% candidates, Calibration-selected. Bold: best mean per dataset, metric, and direction.

4.2 OVERALL RETRIEVAL COMPARISON

Table 1 compares the final Test quality of Calibration-selected ODEH with six full-gallery baselines reproduced or adapted to shared features (Appendix Table S1). DGHDGH and DDBH retrain on all arrived data. Figure 3 follows these methods during growth on Validation (Section 4.3).

On final COCO Test, ODEH's 80-bit ranking code exceeds 128-bit BITS-CM by 0.65 / 1.56 points in P@10 and 0.69 / 1.71 in nDCG@10. The exceptions to ODEH's lead are BITS-CM's 0.04-point lower COCO I2T failure (smaller than either method's seed SD) and OH-ELS's higher MIRFlickr I2T P@10/nDCG@10.

Expansion and coordinate preservation. We use two internal controls to examine added capacity and coordinate preservation. Against the same 32-bit base, learned extensions improve Test P@10 by 4.29–13.12 points across all six dataset–direction pairs (Appendix Table S5).

Residual refitting replaces old heads, allowing earlier coordinates to change alongside the total width. On NUS-WIDE Test, it reaches 5.89% / 8.17% failure at 48 bits versus ODEH's 7.00% / 9.06% at 80 bits (Appendix Table S5), with lower latency (Appendix Table S6). On COCO Validation, a 16-bit append and 48-bit refit yield near-equal failure (Appendix Table S9). Both share the base index and candidates and avoid historical writes through on-demand execution; ODEH also preserves the code entering each expansion as a prefix.

4.3 Growth and quality-driven length

During growth, ODEH leads NUS-WIDE T2I at all three stages, with its P@10 advantage over the strongest baseline, DGHDGH, reaching 3.61 points at the last stage (Figure 3). On COCO, ODEH exceeds both full-retraining baselines throughout growth, while BITS-CM and PBEH achieve higher terminal means. On MIRFlickr, its early T2I lead shifts to OH-ELS and DGHDGH. ODEH's growth-time Validation selections retain 48→64→64 bits on COCO, 48→64→80 on NUS-WIDE, and terminal widths of 32/48/48 across MIRFlickr seeds.

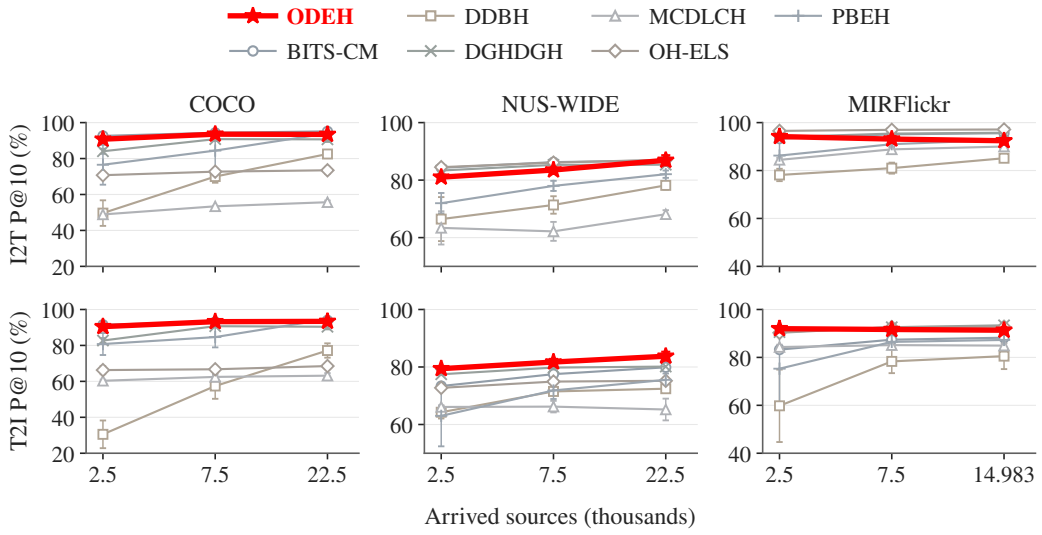


Figure 3: Growth-time Validation P@10 before terminal Calibration (three seeds, mean ± sample SD). ODEH reranks 1% candidates; external methods scan the full gallery. Stages are equally spaced and labelled by arrived source count. Table 1 reports final Test results after independent terminal selection.

Improvement beyond the absolute gate. Target-only selection keeps NUS-WIDE at 32 bits, whereas the paired-improvement tier enables successive additions that reduce terminal Validation failure from 17.61% / 20.76% to 6.76% / 8.92% (Figure 4).

Candidate coverage. On NUS-WIDE, shortlists contain too few relevant items for 0.69% / 1.95% of queries, setting a failure floor that extra bits cannot reduce. The remaining 6.07 / 6.97 points of failure occur despite sufficient candidates, so ranking is the larger bottleneck. To assess the effect of candidate coverage across datasets, we vary the candidate budget with development models frozen. Using 1% candidates instead of the full gallery changes mean failure by -0.60 to $+0.16$ points across the six dataset–direction pairs (Appendix Figure S3).

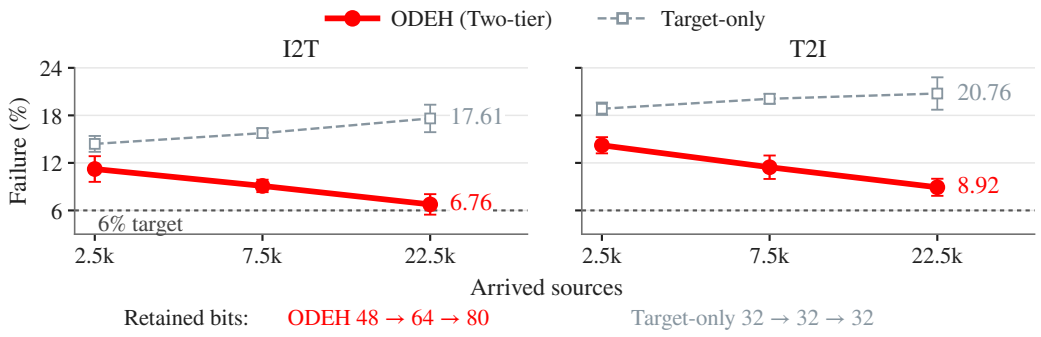


Figure 4: Two-tier selection reduces NUS-WIDE Validation failure as the database grows (three seeds, mean ± sample SD). The shared annotation gives retained ranking widths; both rules keep a 32-bit index and use 1% candidates.

Independent terminal selection. For final Test evaluation, Calibration reselects the last increment while keeping the earlier growth history fixed. This retains 80/80/80 bits on COCO and NUS-WIDE and 48/64/64 on MIRFlickr. Of the 12 dataset–scale–seed configurations, including COCO at 100,000 sources, nine certify failure risk at most 6% in both directions; the three NUS-WIDE configurations certify improvement only (Appendix Table S2). At 100,000 sources, selected lengths are 80/64/64, with Test P@10 of 94.66% / 94.72% and failure of 3.63% / 3.47%.

4.4 Quality and cost of on-demand ranking

Figure 5 compares learned residuals, precomputed residual codes, and FP32 cosine using the final selected models, shared features, and 1% base-code candidates. RAM timings use fixed Validation queries and a shared optimized scan, starting with query features and base codes available.

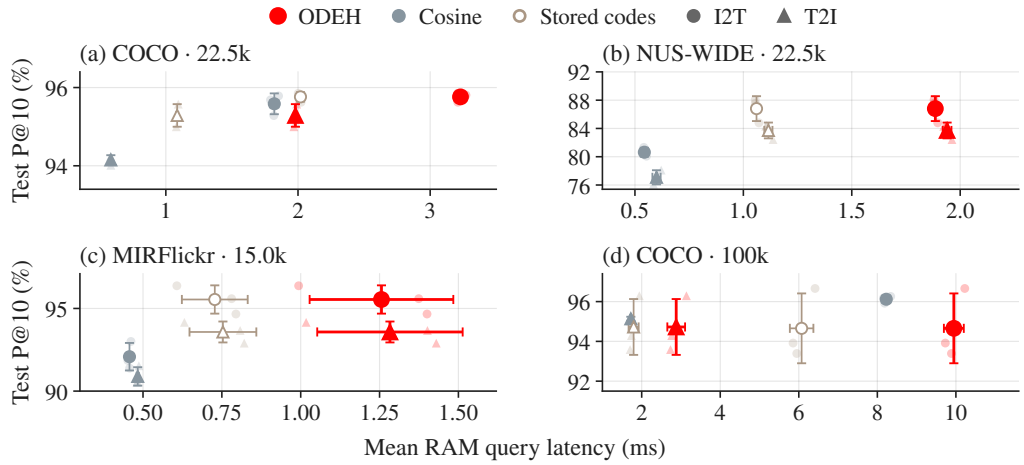


Figure 5: Quality and latency of the final selected models: Test P@10 versus RAM query latency (three seeds, mean ± sample SD). Faint points show individual seeds.

Ranking quality and query computation. On the galleries in Table 1, ODEH improves Test P@10 over cosine by 6.17 / 6.63 points on NUS-WIDE, 3.46 / 2.69 on MIRFlickr, and 0.18 / 1.14 on COCO (I2T / T2I). These gains come at a query-time cost: ODEH is slower in all eight dataset–scale–direction groups. At 100,000 COCO sources, cosine also leads P@10 by 1.46 / 0.41 points.

A separate comparison with PBEH times each method's serving implementation from frozen query features to Top-10, including query encoding and transfer. ODEH is 46.5% faster on COCO I2T at

22,500 sources and 7.9% slower on T2I. At 100,000 sources, latency reductions are 62.4% / 47.0% (Appendix Table S3); PBEH has higher quality there (Appendix Table S5).

Fixed index and historical writes. ODEH keeps four code bytes per item in the index, alongside an identifier, while retaining a 2,048-byte FP32 feature and shared residual heads for on-demand ranking. Expansion can reuse features already retained for training. Materializing each accepted d_t-bit extension for H_t historical items accumulates $\sum_t H_t d_t$ logical bits of writes; ODEH adds the same ranking capacity without these historical writes.

Stored residuals give identical rankings with lower query latency. Encoding the full retained chain from a base-only index and writing its codes to the OS cache is amortized after 33–154 queries, excluding common training (Appendix Figure S2). Incremental expansion reuses stored residuals; its update cost remains unmeasured.

4.5 Learning controls

Figure 6 compares 16- and 32-bit additions to COCO's 48/64/48-bit development histories, with shared initialization, 80 epochs, and 1% evaluation candidates. A1 replaces the candidate pool with global in-batch samples; A2 uses a pairwise loss. A3 replaces the new head's continuous feature input with an embedding of the old binary code, and A4 omits old residual distances during training.

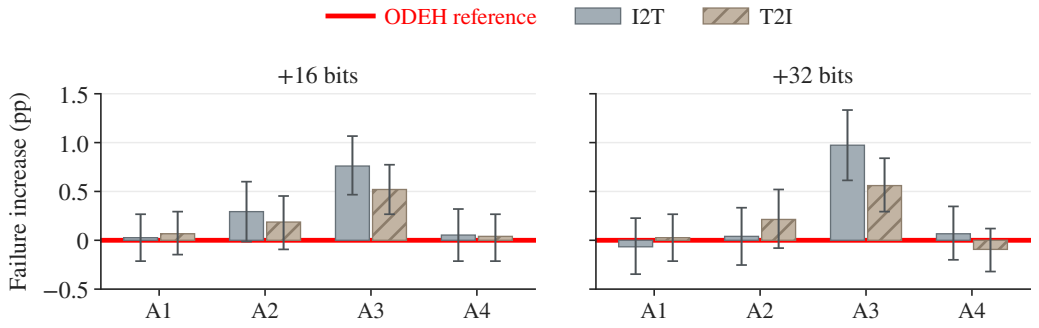


Figure 6: Matched COCO Validation controls with 16 or 32 added bits. Bars show control-minus-ODEH failure; positive values favor ODEH. Whiskers are 95% paired-source intervals conditional on three fitted models.

Continuous features benefit both expansion budgets: replacing them with old-code embeddings (A3) raises failure by 0.760 / 0.520 points with 16 added bits and 0.973 / 0.560 points with 32. All four A3 intervals lie above zero, while A1, A2, and A4 intervals cross zero at both widths. Appendix Table S10 reports seed variation for the 32-bit additions.

Limitations. Selection acts locally: COCO's two-tier history ends 0.557 / 0.429 P@10 points below target-only selection. Dataset comparisons also combine captions versus tags with different CLIP backbones, so the cause of the cosine gaps remains unresolved. The timing measurements use serial execution; concurrent serving may change the balance between candidate computation and shared feature access.

5 Conclusion

ODEH expands ranking capacity through candidate-level residual generation while preserving the stored base index and every retained coordinate. Across three datasets, learned extensions improve retrieval over the base codes. Quality-driven selection connects expansion to a query-level requirement, and coverage and distance bounds identify limits on what new bits can repair. The resulting trade-off replaces historical code writes with retained feature storage and candidate computation.

Reproducibility statement

The anonymized supplement provides ODEH's training and retrieval implementation, two-tier selection and calibration routines, and configurations for the three datasets. Section 3 defines retrieval and learning. Appendix A specifies data and comparison protocols, Appendix B gives training details, and Appendices C–E cover proofs, calibration, and timing.

Ethics statement

This study uses existing COCO, NUS-WIDE, and MIRFlickr benchmarks and collects no new participant data. The experiments use source identifiers, semantic labels, and NUS-WIDE tag text. Images, feature arrays, and pretrained weights are obtained separately under their providers' terms. Dataset labels and pretrained features can carry social and annotation biases, which may affect retrieved content. Applications should assess these biases in their intended use context.

AI use statement

Generative AI tools were used solely for language polishing and manuscript structural organization. All scientific content, research methodology, source code, experimental design and verification, and final conclusions were independently developed and verified by the authors. The authors take full responsibility for the final content of this work.

References

Anastasios N Angelopoulos, Stephen Bates, Emmanuel J Candès, Michael I Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *The Annals of Applied Statistics*, 19(2):1641–1662, 2025.

Anastasios Nikolas Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=33XGfHLtZg.

Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432, 2013. URL http://arxiv.org/abs/1308.3432.

Kaiwen Cai, Chris Xiaoxuan Lu, Xingyu Zhao, Wei Huang, and Xiaowei Huang. Risk controlled image retrieval. In Toby Walsh, Julie Shah, and Zico Kolter (eds.), *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pp. 27224–27232. AAAI Press, 2025. doi: 10.1609/AAAI.V39I26.34931. URL https://doi.org/10.1609/aaai.v39i26.34931.

Bingzhi Chen, Zhongqi Wu, Yishu Liu, Biqing Zeng, Guangming Lu, and Zheng Zhang. Enhancing cross-modal retrieval via visual-textual prompt hashing. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, pp. 623–631. ijcai.org, 2024. URL https://www.ijcai.org/proceedings/2024/69.

Kun Cheng, Qibing Qin, Wenfeng Zhang, Lei Huang, and Jie Nie. Deep global-sense hard-negative discriminative generation hashing for cross-modal retrieval. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=GAQEsnnQtG.

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 Stéphane Marchand-Maillet and Yiannis Kompatsiaris (eds.), *Proceedings of the 8th ACM International Conference on Image and Video Retrieval, CIVR 2009, Santorini Island, Greece, July 8-10, 2009*. ACM, 2009. doi: 10.1145/1646396.1646452. URL https://doi.org/10.1145/1646396.1646452.

C. J. Clopper and E. S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934. doi: 10.1093/biomet/26.4.404. URL https://doi.org/10.1093/biomet/26.4.404.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

B. Efron. Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, 7(1):1 – 26, 1979. doi: 10.1214/aos/1176344552. URL https://doi.org/10.1214/aos/1176344552.

Morten W. Fagerland, Stian Lydersen, and Petter Laake. The McNemar test for binary matched-pairs data: mid-p and asymptotic are better than exact conditional. *BMC Medical Research Methodology*, 13(1):91, July 2013. ISSN 1471-2288. doi: 10.1186/1471-2288-13-91. URL https://doi.org/10.1186/1471-2288-13-91.

Haozhi Fan and Yuan Cao. Vision-guided text mining for unsupervised cross-modal hashing with community similarity quantization. In Toby Walsh, Julie Shah, and Zico Kolter (eds.), *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pp. 2843–2851. AAAI Press, 2025. doi: 10.1609/AAAI.V39I3.32290. URL https://doi.org/10.1609/aaai.v39i3.32290.

Wentao Fan, Chao Zhang, Chunlin Chen, and Huaxiong Li. Online cross-modal hashing with expanding label space. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor (eds.), *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pp. 21002–21010. AAAI Press, 2026. doi: 10.1609/AAAI.V40I25.39242. URL https://doi.org/10.1609/aaai.v40i25.39242.

Jianyang Gao, Yutong Gou, Yuexuan Xu, Yongyi Yang, Cheng Long, and Raymond Chi-Wing Wong. Practical and asymptotically optimal quantization of high-dimensional vectors in euclidean space for approximate nearest neighbor search. *Proc. ACM Manag. Data*, 3(3):202:1–202:26, 2025. doi: 10.1145/3725413. URL https://doi.org/10.1145/3725413.

Liyang He, Yuren Zhang, Rui Li, Zhenya Huang, Runze Wu, and Enhong Chen. Nested hash layer: A plug-and-play module for multiple-length hash code learning. arXiv:2412.08922v2, 2025. URL https://arxiv.org/abs/2412.08922v2.

Peichun Hua and Yunming Xiao. Matryoshka hash representations for model-aware compact semantic retrieval, 2026. URL https://arxiv.org/abs/2609.07276.

Mark J. Huiskes and Michael S. Lew. The MIR flickr retrieval evaluation. In Michael S. Lew, Alberto Del Bimbo, and Erwin M. Bakker (eds.), *Proceedings of the 1st ACM SIGMM International Conference on Multimedia Information Retrieval, MIR 2008, Vancouver, British Columbia, Canada, October 30-31, 2008*, pp. 39–43. ACM, 2008. doi: 10.1145/1460096.1460104. URL https://doi.org/10.1145/1460096.1460104.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002. doi: 10.1145/582415.582418. URL http://doi.acm.org/10.1145/582415.582418.

Suhas Jayaram Subramanya, Fnu Devvrit, Harsha Vardhan Simhadri, Ravishankar Krishnawamy, and Rohan Kadekodi. DiskANN: Fast Accurate Billion-point Nearest Neighbor Search on a Single Node. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/hash/09853c7fb1d3f8ee67a61b6bf4a7f8e6-Abstract.html.

Kaihang Jiang, Wai Keung Wong, Xiaozhao Fang, Weijun Sun, Guoxu Zhou, Shengli Xie, and Xiaochun Cao. An efficient regenerated cross-modal hashing: Improving existing hash codes with the arbitrary length. *IEEE Trans. Pattern Anal. Mach. Intell.*, 48(9):10382–10400, 2026. doi: 10.1109/TPAMI.2026.3688816. URL https://doi.org/10.1109/TPAMI.2026.3688816.

Qing-Yuan Jiang and Wu-Jun Li. Deep cross-modal hashing. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 3270–3278. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.348. URL https://doi.org/10.1109/CVPR.2017.348.

Xiao Kang, Xingbo Liu, Xuening Zhang, Wen Xue, Xiushan Nie, and Yilong Yin. Semi-supervised online cross-modal hashing. In Toby Walsh, Julie Shah, and Zico Kolter (eds.), *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pp. 17770–17778. AAAI Press, 2025. doi: 10.1609/AAAI.V39I17.33954. URL https://doi.org/10.1609/aaai.v39i17.33954.

Xiao Kang, Xingbo Liu, Shuo Pan, Xuening Zhang, Xiushan Nie, and Yilong Yin. PEOCH: online cross-modal hashing with semi-supervised streaming data driving prototype evolution. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor (eds.), *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pp. 15009–15017. AAAI Press, 2026. doi: 10.1609/AAAI.V40I17.38523. URL https://doi.org/10.1609/aaai.v40i17.38523.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham M. Kakade, Prateek Jain, and Ali Farhadi. Matryoshka representation learning. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/c32319f4868da7613d78af9993100e42-Abstract-Conference.html.

Fengling Li, Yang Sun, Tianshi Wang, Lei Zhu, and Xiaojun Chang. Fast partial-modal online cross-modal hashing. *IEEE Trans. Image Process.*, 34:4440–4455, 2025a. doi: 10.1109/TIP.2025.3586504. URL https://doi.org/10.1109/TIP.2025.3586504.

Jiaxing Li, Lin Jiang, Zeqi Ma, Kaihang Jiang, Xiaozhao Fang, and Jie Wen. Lightweight contrastive distilled hashing for online cross-modal retrieval. In Toby Walsh, Julie Shah, and Zico Kolter (eds.), *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pp. 4779–4787. AAAI Press, 2025b. doi: 10.1609/AAAI.V39I5.32505. URL https://doi.org/10.1609/aaai.v39i5.32505.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pp. 740–755. Springer, 2014. doi: 10.1007/978-3-319-10602-1_48. URL https://doi.org/10.1007/978-3-319-10602-1_48.

Wenhao Liu, Tianshi Wang, Fengling Li, Jingjing Li, and Lei Zhu. Prompt-driven bit extension hashing for continual cross-modal retrieval. *IEEE Trans. Circuits Syst. Video Technol.*, 36(5):7125–7139, 2026. doi: 10.1109/TCSVT.2025.3642760. URL https://doi.org/10.1109/TCSVT.2025.3642760.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Qibing Qin, Yadong Huo, Wenfeng Zhang, Lei Huang, and Jie Nie. Deep discriminative boundary hashing for cross-modal retrieval. *IEEE Trans. Circuits Syst. Video Technol.*, 35(10):10557–10570, 2025. doi: 10.1109/TCSVT.2025.3570128. URL https://doi.org/10.1109/TCSVT.2025.3570128.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021. URL http://proceedings.mlr.press/v139/radford21a.html.

Aniket Rege, Aditya Kusupati, Sharan Ranjit S, Alan Fan, Qingqing Cao, Sham M. Kakade, Prateek Jain, and Ali Farhadi. Adanns: A framework for adaptive semantic search. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/f062da1973ac9ac61fc6d44dd7fa309f-Abstract-Conference.html.

Yongxin Wang, Zhen-Duo Chen, Xin Luo, and Xin-Shun Xu. BITS: bit-extendable incremental hashing in open environments. *IEEE Trans. Image Process.*, 34:6550–6563, 2025. doi: 10.1109/TIP.2025.3613924. URL https://doi.org/10.1109/TIP.2025.3613924.

Yair Weiss, Antonio Torralba, and Robert Fergus. Spectral hashing. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou (eds.), *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pp. 1753–1760. Curran Associates, Inc., 2008. URL https://proceedings.neurips.cc/paper/2008/hash/d58072be2820e8682c0a27c0518e805e-Abstract.html.

Liming Xu, Hanqi Li, Bochuan Zheng, Weisheng Li, and Jiancheng Lv. Deep lifelong cross-modal hashing. *IEEE Trans. Circuits Syst. Video Technol.*, 34(12):13478–13493, 2024. doi: 10.1109/TCSVT.2024.3450490. URL https://doi.org/10.1109/TCSVT.2024.3450490.

Liming Xu, Hanqi Li, Jie Shao, Xianhua Zeng, and Weisheng Li. Multi-scale consistency deep lifelong cross-modal hashing. *ACM Trans. Multim. Comput. Commun. Appl.*, 21(10):283:1–283:23, 2025a. doi: 10.1145/3704636. URL https://doi.org/10.1145/3704636.

Yunpeng Xu, Mufang Ying, Wenge Guo, and Zhi Wei. Two-stage risk control with application to ranked retrieval. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025*, pp. 9104–9111. ijcai.org, 2025b. doi: 10.24963/IJCAI.2025/1012. URL https://doi.org/10.24963/ijcai.2025/1012.

Haoran Yang, Ying An, Zhan Yang, and Jun Long. Scale-invariant lifelong cross-modal hashing via gradient equilibrium. In Won Kim, Jae-Gil Lee, Kyuseok Shim, James Caverlee, David F. Gleich, Xiangliang Zhang, Qi He, Faisal Farooq, Rómer Rosales, Sayan Ranu, Haixun Wang, Yan Liu, Carl Yang, Leman Akoglu, Nitesh V. Chawla, and Vagelis Papalexakis (eds.), *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2, KDD 2026, Jeju Island, Korea, August 9-13, 2026*, pp. 5986–5997. ACM, 2026a. doi: 10.1145/3770855.3817901. URL https://doi.org/10.1145/3770855.3817901.

Mingyu Yang, Liuchang Jing, Wentao Li, and Wei Wang. Quantization meets projection: A happy marriage for approximate k-nearest neighbor search. *Proc. VLDB Endow.*, 19(6):1240–1249, 2026b. URL https://www.vldb.org/pvldb/vol19/p1240-yang.pdf.

Shubing Yang and Dongfang Zhao. Adawidth: Query-adaptive embedding width for dense retrieval. *CoRR*, abs/2608.23862, 2026. doi: 10.48550/ARXIV.2608.23862. URL https://doi.org/10.48550/arXiv.2608.23862.

Yu-Wei Zhan, Yongxin Wang, Yu Sun, Xiao-Ming Wu, Xin Luo, and Xin-Shun Xu. Discrete online cross-modal hashing. *Pattern Recognit.*, 122:108262, 2022. doi: 10.1016/J.PATCOG.2021.108262. URL https://doi.org/10.1016/j.patcog.2021.108262.

Yu-Wei Zhan, Xin Luo, Zhen-Duo Chen, Yongxin Wang, Yinwei Wei, and Xin-Shun Xu. POLISH: adaptive online cross-modal hashing for class incremental data. In Tat-Seng Chua, Chong-Wah Ngo, Ravi Kumar, Hady W. Lauw, and Roy Ka-Wei Lee (eds.), *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*, pp. 4470–4478. ACM, 2024. doi: 10.1145/3589334.3645716. URL https://doi.org/10.1145/3589334.3645716.

A DATASET PROTOCOL AND COMPARISON SCOPE

COCO sources and splits. Filtering sources without labels leaves 122,218 images and 611,420 captions over 80 categories, with Karpathy train/validation/test counts 112,318/4,948/4,952. From the 117,266 non-Test sources, we reserve 5,000 for Validation, 2,500 for development calibration, and 2,500 for regression, leaving a pool of 107,266 sources with a fixed arrival order. Growth uses its first 100,000 sources; positions 100,001–102,500 are held out for independent terminal Calibration. Final Test contains the 4,952 held-out sources. At the three main growth stages, I2T galleries contain 12,506/37,517/112,565 captions and T2I galleries 2,500/7,500/22,500 images. T2I uses one fixed caption per query source. Split, arrival, and evaluation-caption seeds are 20260904, 0, and 0.

Figures 1–2 use an illustrative photograph with authored candidate texts. Their codes, candidate order, and selection outcomes are schematic.

Tag datasets. NUS-WIDE represents each source by an image and its active official 1k tags; 21 selected concepts define relevance. From 190,420 available sources, the official training portion supplies 106,617 Train, 5,000 Validation, and 2,500 Calibration sources; final Test uses 5,000 official test sources. Our preprocessing of MIRFlickr (Huiskes & Lew, 2008) removes 2,098 empty-tag sources from 24,581 positive-label sources, retaining 22,483 pairs: 14,983 Train and 2,500 each for Validation, Calibration, and Test. Exact-image duplicates stay together in Train, with no image-content overlap across roles. Both use frozen, independently normalized 512-dimensional ViT-B/32 features.

Baseline adaptations. All methods share dataset-specific frozen features, source identities, and arrival orders. Table S1 identifies the implementations, adaptations, and access to historical data. Continual updates, cumulative retraining, and internal controls form distinct comparison regimes.

Table S1: Implementation scope of the six external comparisons. "Fixed codes" refers to previously published gallery codes; training may still use the historical information listed below.

Method	Implementation	Adaptation and historical access
PBEH	Official-code adaptation	Author training and bit trigger; features scaled by $\sqrt{512}$, independent-query inference, and item-width-truncated Hamming scoring. Published codes and widths remain fixed.
BITS-CM	Cross-modal adaptation	Shared semantic centers and separate modality projections retain BITS discrete updates and expansion trigger. Kernel parameters are fixed from the initial chunk; published codes remain fixed.
OH-ELS	Formula-based	Discrete updates and sufficient statistics adapted to stages without new classes. Historical code/label anchors are replayed; published codes remain fixed.
MCDLCH	Paper-based, corrected	Three-network learning with CLIP image projection, text-coordinate pooling, and posterior-mean inference. Objective/indexing corrections are documented in the implementation. Historical features are sampled; published codes remain fixed.
DGHDGH	Official-code adaptation	Author losses, graph learning, and optimization on frozen features. Heads are retrained on all arrived pairs; historical gallery codes are regenerated.
DDBH	Official-code adaptation	Author boundary/quantization losses and optimization; equivalent vectorized execution on NUS-WIDE and MIRFlickr. Cumulative retraining regenerates historical codes.

B ARCHITECTURE AND OPTIMIZATION

Growth state and binary outputs. At stage t, each modality $m \in \{I, T\}$ has a new head with relaxed output $v_t^m(z) = \tanh A_t^m(\phi^m(z))$ and binary output $r_t^m(z) = \operatorname{sign} v_t^m(z)$, with $\operatorname{sign}(0) = +1$. Let $d_s^\star$ be the increment retained at stage s. Before stage t, the retained representation is

$$h_{t-1}^m(z) = \left[h_0^m(z); r_{1,1:d_1^\star}^m(z); \ldots; r_{t-1,1:d_{t-1}^\star}^m(z)\right], \qquad b_{t-1} = b_0 + \sum_{s<t} d_s^\star.$$

Thus the main text's h_{old} is h_{t-1}, and a proposal gives $h_{t,d}^m = [h_{t-1}^m; r_{t,1:d}^m]$. Zero-length blocks contribute no coordinates. The complete Hamming distance decomposes as

$$D_{t,d}(q,z) = d_H(h_0(q), h_0(z)) + \sum_{s<t} d_H(r_{s,1:d_s^\star}(q), r_{s,1:d_s^\star}(z)) + d_H(r_{t,1:d}(q), r_{t,1:d}(z)).$$

Candidate pools depend only on h_0.

Each modality uses a separate residual head with architecture

$$A_t^m(u) = W_3^m\ \mathrm{GELU}(W_2^m\,\mathrm{GELU}(W_1^m u + u))\,.$$

We omit stage subscripts on the weights and use bias-free linear maps. For the current 512-dimensional features, W_1 has width 512, W_2 maps to 256 dimensions, and W_3 outputs a 32-bit proposal. The feature skip adds no parameters. When a prefix is retained, the whole head remains available and evaluates all outputs before slicing the retained coordinates.

For a batch of B source pairs and proposal width D, stack the relaxed image and text outputs $v_i^I, v_i^T \in \mathbb{R}^D$ in $V \in \mathbb{R}^{2B\times D}$. Four regularizers penalize cross-modal disagreement, quantization error, bit imbalance, and correlation, respectively:

$$\mathcal{L}_{\text{cm}} = \frac{1}{BD}\sum_{i=1}^{B}\|v_i^I - v_i^T\|_2^2, \qquad \mathcal{L}_q = \frac{1}{2BD}\sum_{i,j}(|V_{ij}|-1)^2,$$

$$\mathcal{L}_b = \left\|\frac{1}{2B}\sum_i V_{i,:}\right\|_2^2, \qquad \mathcal{L}_d = \left\|\frac{V^\top V}{2B} - I_D\right\|_F^2.$$

Here I_D is the D-dimensional identity matrix.

For each direction and positive code prefix, the boundary objective is averaged over trainable queries, with an empty set contributing zero. These terms are then averaged equally. Queries are batched by source, with one training caption sampled per source per epoch, and the regularizers operate on these source-balanced query pairs. Candidate features shared by several queries are encoded once within a batch. The total loss is

$$\mathcal{L} = \mathcal{L}_{\text{boundary}} + \lambda_{\text{cm}}\mathcal{L}_{\text{cm}} + \lambda_q\mathcal{L}_q + \lambda_b\mathcal{L}_b + \lambda_d\mathcal{L}_d.$$

Training uses prefixes 16 and 32, zero margin, unit temperature, $\lambda_{\text{cm}} = 0.1$, and $\lambda_q = \lambda_b = \lambda_d = 0.01$. It runs for 80 epochs with AdamW (Loshchilov & Hutter, 2019), batch size 256, learning rate 10^{-3}, and weight decay 10^{-4}. Checkpoint selection uses Validation failure at the shorter proposed prefix, ordered by worst direction, bidirectional mean, and the Validation objective. The paired global-training control replaces the base-code candidate pool with the in-batch pool, keeping the architecture, loss, initialization, and update count fixed.

Base hasher. The base code is the first 32 coordinates of a separately trained 64-bit hasher. This base uses a shared, bias-free $512 \to 256 \to 64$ projection for image and text, with GELU between the linear layers and a final tanh. It trains on the first 2,500 arrived sources, averaging its objective over 32/64-bit prefixes and both retrieval directions. The objective combines equally weighted pairwise and boundary losses, with cross-modal consistency weight 0.1 and quantization, balance, and decorrelation weights 0.01. The pairwise term averages positive and negative logistic losses

separately with logit $(b - 2D_{ij})/\sqrt{b}$, where b is the prefix width and D_{ij} the Hamming distance between examples i and j. The boundary term uses $K = 10$, $\rho = 0.5$, and in-batch relevance counts. Base consistency averages over semantic-positive pairs. AdamW uses batch size 256, learning rate 10^{-3}, and weight decay 10^{-4}. Training runs for at most 50 epochs with patience 20, selecting the lowest Validation loss. On COCO and NUS-WIDE, this loss uses all 5,000 Validation sources, whereas quality comparisons use the fixed 2,500-source subset. Each full training seed fits its own base, which is then frozen.

C Distance bounds and exact retrieval

Binary distances and boundary objective. We fix one snapshot and direction, using the main-text notation. For candidate $z \in \mathcal{C}(q)$, define the frozen distance $D_{\mathrm{old}}(q, z) = d_H(h_{\mathrm{old}}(q), h_{\mathrm{old}}(z))$. Let $v_{1:d}$ denote the new head's relaxed outputs at prefix $d > 0$. The relaxed complete distance is

$$\widetilde{D}_d(q, z) = D_{\mathrm{old}}(q, z) + \frac{1}{2}\left[d - v_{1:d}(q)^\top v_{1:d}(z)\right].$$

Let D_d^{bin} be the corresponding exact binary Hamming distance. Training uses the straight-through distance

$$D_d^{\mathrm{ST}} = \mathrm{stopgrad}(D_d^{\mathrm{bin}} - \widetilde{D}_d) + \widetilde{D}_d,$$

where $\mathrm{stopgrad}$ preserves its argument's value and sets its gradient to zero. Forward values therefore match binary retrieval distances, while gradients follow the relaxed inner product. The frozen term includes old residual blocks, even though candidate selection uses only the base 32 bits.

For a trainable query, write $k = k(q)$. If fewer than k relevant items enter the first K positions, the $(K - k + 1)$-th irrelevant item precedes the k-th relevant item and has distance no larger. Hence failure implies $M_d(q) \le 0$, so the loss $\ell_d(q) = \log(1 + \exp(-M_d(q)))$ is at least $\log 2$. Thus $Z_d(q) \le \ell_d(q)/\log 2$. The implementation averages gradients uniformly over items tied at the selected order statistic, so permuting equally distant candidates leaves the update unchanged.

Candidate coverage and attainable risk. Let $A(q) = \mathbf{1}\{|\mathcal{C}(q) \cap \mathcal{P}(q)| < k(q)\}$, and let $B(q)$ indicate a trainable query, whose candidates contain at least $k(q)$ relevant and $K - k(q) + 1$ irrelevant items. If $A = 1$, every ranking of the candidates fails. If $A = B = 0$, every Top-K ranking succeeds. Combining these cases with the boundary bound gives

$$A(q) \le Z_d(q) \le A(q) + B(q)\frac{\ell_d(q)}{\log 2}.$$

The second term is defined as zero outside the trainable set. Consequently, $R(d) \ge \Pr_{q\sim Q}\{A(q) = 1\}$ for every prefix: extra bits can reorder candidates but cannot recover excluded items.

What a frozen-code extension can repair. Fix the current snapshot, candidate identifiers, relevance, and required count. For a trainable query, write $a(z) = d_H(h_{\mathrm{old}}(q), h_{\mathrm{old}}(z))$. Appending d unweighted binary coordinates gives distance $a(z) + \delta(z)$, where $0 \le \delta(z) \le d$. By monotonicity of order statistics, each relevant or irrelevant boundary distance satisfies $D_j^{\pm,\mathrm{old}} \le D_j^{\pm,\mathrm{new}} \le D_j^{\pm,\mathrm{old}} + d$. Subtracting the relevant boundary from the irrelevant boundary gives

$$M_{\mathrm{old}}(q) - d \le M_d(q) \le M_{\mathrm{old}}(q) + d.$$

If $M_{\mathrm{old}}(q) < -d$, at least $K - k + 1$ irrelevant items remain strictly closer than the k-th relevant item after any such extension, forcing failure. Combining this event with the disjoint candidate-shortfall event gives

$$R(d) \ge \Pr\{A(q) = 1\} + \Pr\{B(q) = 1,\ M_{\mathrm{old}}(q) < -d\}.$$

This is an impossibility bound: satisfying $M_{\mathrm{old}} \ge -d$ does not guarantee that a successful extension exists or can be learned. At equality, success can depend on the fixed identifier tie rule.

C.1 Optional exact-result check

For evaluated candidates $\mathcal{C}(q)$, let D_{in} be their K-th smallest complete distance and let $D_{\text{out}} = \min_{z \notin \mathcal{C}(q)} d_H(h_0(q), h_0(z))$ be the smallest omitted base-code distance. Every omitted residual distance is nonnegative, so $D_{\text{in}} < D_{\text{out}}$ certifies agreement with full-gallery complete-code Top-K. The exact check expands by complete base-distance layers until this condition holds or the gallery is exhausted. ODEH's fixed-fraction service instead obtains its semantic-risk bound from the lists it actually returns.

A COCO development model, trained with seed 0 at 22,500 arrived sources, was used for a single-model correctness check. Exact expansion matched full-gallery complete-code Top-10 for all 200 fixed Validation queries per direction at each checked width (48, 64, and 80 bits). Mean examined fractions were 3.17% / 5.18% at 48 bits and 21.05% / 26.41% at 80 bits (I2T / T2I).

D Selection and independent calibration

D.1 Validation selection and state transition

For direction $g \in \{\text{I2T}, \text{T2I}\}$, pair the same n_g Validation queries under the current retained code (increment zero) and each positive increment d. The empirical failure is $\widehat{R}_g(d) = n_g^{-1} \sum_i Z_d^g(q_i)$. Define the discordant-query counts

$$G_g(d) = \sum_{i=1}^{n_g} Z_0^g(q_i)[1 - Z_d^g(q_i)], \qquad H_g(d) = \sum_{i=1}^{n_g} [1 - Z_0^g(q_i)] Z_d^g(q_i).$$

Thus $\widehat{R}_g(d) - \widehat{R}_g(0) = [H_g(d) - G_g(d)]/n_g$. The one-sided paired score is $p_g(d) = \Pr\{X \geq G_g(d)\}$, where $X \sim \text{Binom}(G_g(d) + H_g(d), 1/2)$; set $p_g(d) = 1$ when no queries are discordant. The rule requires both $G_g(d) > H_g(d)$ and $p_g(d) \leq 0.05/36$ in each direction. The divisor comes from the base grid of three seeds, three stages, two positive increments, and two directions within a dataset. The same cutoff is retained for five-stage growth to 100,000 sources as a fixed development threshold. Validation reuse and adaptive histories provide no 5% familywise certification of those trajectories.

Section 3.4 gives the two-tier decision rule for increments $\{0, 16, 32\}$. A positive retained increment freezes a new block for subsequent stages. The target-only control in Table S8 omits the improvement tier. Each rule trains proposals from its own retained history.

D.2 Finite-family calibration

The joint release family covers 12 terminal seed cells: COCO at 22,500 and 100,000 sources, NUS-WIDE at 22,500, and MIRFlickr at 14,983, each with three seeds. Before Calibration access, each cell freezes the code entering the last stage, its trained proposal, the data, and the search settings. Its increments are $0, 16, 32$. Each dataset contributes $n = 2{,}500$ Calibration query sources per direction. Conditional on the frozen models and galleries, the guarantee assumes i.i.d. queries from the corresponding eligible-query distribution. The family allocates 72 absolute bounds and 48 paired tests:

$$12 \times 2 \times (3 + 2) = 120, \qquad \beta = 0.05/120.$$

Every allocation is retained even when a gate is skipped or a cell fails.

For direction g and frozen configuration j, let $e_{g,j}$ be the Calibration failure count and $R_{g,j}$ its population failure risk. The one-sided Clopper–Pearson upper bound is (Clopper & Pearson, 1934)

$$U_{g,j} = \begin{cases} 1, & e_{g,j} = n, \\ \text{Beta}^{-1}(1 - \beta; e_{g,j} + 1, n - e_{g,j}), & e_{g,j} < n. \end{cases}$$

Here $\text{Beta}^{-1}(p; a, b)$ is the p-quantile of the beta distribution with shape parameters a, b.

Within each cell, release first selects the shortest increment whose two bounds are at most 0.06. If none qualifies, the paired counts $G_g(d), H_g(d)$ and exact binomial tail above are recomputed on

Calibration against the incoming chain. Release then selects the shortest positive increment with $G_g(d) > H_g(d)$ and $p_g(d) \le \beta$ in both directions. This certifies relative improvement only, not the 6% target. If both gates fail, the incoming chain is kept without a certificate.

Each absolute bound has $\Pr_{\text{Cal}}\{R_{g,j} > U_{g,j}\} \le \beta$. For each paired test, the null is $\Pr(Z_0 = 1, Z_d = 0) \le \Pr(Z_0 = 0, Z_d = 1)$. Conditional on the discordant count, its gain probability is at most $1/2$, so the exact upper-tail test has false-rejection probability at most β. A union bound gives

$$\Pr_{\text{Cal}}\{\text{any invalid absolute bound or false improvement claim}\} \le (72 + 48)\beta = 0.05.$$

Thus the joint 95% guarantee covers both gates and directions across all 12 frozen terminal cells, without requiring independence between tests. Updating these models after Calibration requires a new independent set or an adaptive procedure with its own justification.

Table S2: Independent release and final evaluation. Bounds and Test failure are percentages; paired entries follow I2T / T2I. Each row identifies a terminal gallery and full training seed.

Seed	Bits	Cal upper (%)	Paired p	Test failure (%)	Release
COCO, 22,500 sources					
0	80	3.55/4.69	1.2×10^{-9} / 6.5×10^{-11}	2.99/3.43	Absolute
1	80	3.55/4.83	2.8×10^{-6} / 4.2×10^{-6}	2.93/3.01	Absolute
2	80	3.50/4.69	2.1×10^{-9} / 2.6×10^{-6}	2.79/3.01	Absolute
NUS-WIDE, 22,500 sources					
0	80	9.88/12.45	1.8×10^{-20} / 1.4×10^{-22}	8.04/9.90	Relative
1	80	7.79/10.45	3.3×10^{-20} / 9.6×10^{-19}	6.52/8.66	Relative
2	80	8.51/10.76	4.7×10^{-18} / 1.3×10^{-13}	6.44/8.62	Relative
MIRFlickr, 14,983 sources					
0	48	2.11/4.74	2.2×10^{-8} / 5.2×10^{-14}	1.84/3.00	Absolute
1	64	2.01/5.39	5.5×10^{-6} / 1.8×10^{-4}	1.76/3.52	Absolute
2	64	2.32/5.29	9.5×10^{-2} / 9.8×10^{-5}	1.40/3.48	Absolute
COCO, 100,000 sources					
0	80	3.41/4.17	2.5×10^{-9} / 1.6×10^{-10}	2.28/2.56	Absolute
1	64	5.62/5.57	*0-bit increment*	4.20/3.63	Absolute
2	64	5.52/5.52	*0-bit increment*	4.42/4.20	Absolute

Pairs: I2T/T2I. Calibration uses 2,500 queries per direction. The frozen terminal family has 120 statements (72 absolute bounds and 48 paired tests), with joint 95% coverage and $\beta = 0.05/120$ per statement. Absolute: both Clopper–Pearson upper bounds are at most 6%; Relative: both paired one-sided tests pass. 0-bit increment: the unchanged code is its own reference, so no paired test applies. Test does not determine release or code length.

E System measurement procedures

Execution environment. Final-model RAM, feature-input, and materialization timings use an NVIDIA RTX 4090 and a host with 96 available logical CPUs and 251 GiB RAM, running Python 3.13.13 and PyTorch 2.6.0 with CUDA 12.4. The timing workers use eight PyTorch intra-op threads and one inter-op thread; OpenMP, OpenBLAS, and MKL are also limited to eight threads. FP32 computation uses the highest precision setting with TF32 disabled.

Matched RAM execution. The three execution modes use the same Calibration-selected models, frozen query features, 32-bit base codes, and 1% candidates. ODEH generates candidate residuals on demand; precomputed execution reads complete candidate codes and generates query residuals; cosine scores the FP32 features directly. All share the optimized packed-code scan; on-demand and cosine execution read features from host RAM. Each mode receives 32 warm-up queries and 512 fixed Validation queries per direction over three repetitions, with execution order permuted. We average latency within each training seed, then report the mean and sample SD over three seeds. The quality–latency figure pairs these timings with final Test quality from the same models.

Before Calibration, the optimized and reference implementations produced identical bits and Top-10 results for 60,000 query pairs across 12 development models; optimization reduced latency by 42–64%. The matched RAM comparisons use the optimized implementation throughout.

Common feature-input timing. Table S3 compares ODEH with PBEH from frozen query features through Top-10 output, including query encoding, transfer, packing, scanning, and selection. ODEH uses optimized serving and PBEH its native evaluator. Timings use fixed Validation queries and the frozen models evaluated on Test; each method retains its own encoder and scan implementation.

Table S3: COCO latency from frozen query features to Top-10 (ms; three-seed mean ± sample SD). Backbone feature extraction is excluded.

Sources	Direction	ODEH	PBEH	Change
22.5k	I2T	3.633 ± 0.015	6.787 ± 0.315	-46.5%
22.5k	T2I	2.308 ± 0.018	2.139 ± 0.070	$+7.9\%$
100k	I2T	10.304 ± 0.316	27.409 ± 1.182	-62.4%
100k	T2I	3.233 ± 0.227	6.104 ± 0.234	-47.0%

Change is relative to PBEH; negative values indicate lower latency. Each method retains its own query encoder and scan implementation.

Storage components. Table S4 reports gallery and residual-head payloads in MiB (2^{20} bytes), summing both retrieval directions and counting shared heads once. The base index uses four code bytes and an eight-byte identifier per item; each 512-dimensional FP32 feature adds 2,048 bytes. Precomputed execution stores full codes and query-side heads, while cosine requires features but no residual heads. Common base encoders and transient workspaces are outside this component accounting.

Table S4: Serving payloads of the final selected models (MiB). Entries are three-seed means; nonzero sample SDs are shown.

Gallery	Execution	Codes	IDs	Features	Heads
COCO 22.5k	**ODEH**	0.515	1.030	263.799	9.188
	Cosine	0.515	1.030	263.799	0
	Stored codes	1.288	1.030	0	9.188
NUS-WIDE 22.5k	**ODEH**	0.172	0.343	87.891	9.188
	Cosine	0.172	0.343	87.891	0
	Stored codes	0.429	0.343	0	9.188
MIRFlickr 15.0k	**ODEH**	0.114	0.229	58.527	5.104 ± 1.768
	Cosine	0.114	0.229	58.527	0
	Stored codes	0.210 ± 0.033	0.229	0	5.104 ± 1.768
COCO 100k	**ODEH**	2.290	4.580	1172.424	7.146 ± 1.768
	Cosine	2.290	4.580	1172.424	0
	Stored codes	4.961 ± 0.661	4.580	0	7.146 ± 1.768

Gallery codes, IDs, and features are summed over both directions; shared heads are counted once. Common base encoders and transient buffers are excluded.

Physical feature reads. Figure S1 compares warm and cold retrieval on COCO with packed codes, using the development-selected 64-bit models and reference implementation. This separate measurement batch uses an RTX 4090, eight PyTorch threads, PyTorch 2.6/CUDA 12.4, and ext4 storage on a WDC WUH722016CLE6L4 HDD and a KIOXIA EXCERIA G2 NVMe SSD. The underlying HDD/NVMe study also covers unpacked codes. Dedicated feature files are written and synchronized before measurement. Before each cold query, `MADV_DONTNEED` followed by `POSIX_FADV_DONTNEED` evicts their pages; `mincore` must report less than 1% residency. Eviction and residency checks are outside the timer. Per-query changes in `/proc/self/io` read bytes verify that all 73,728 cold measurements access physical storage; warm queries read zero physical bytes. Each plotted group summarizes three training seeds.

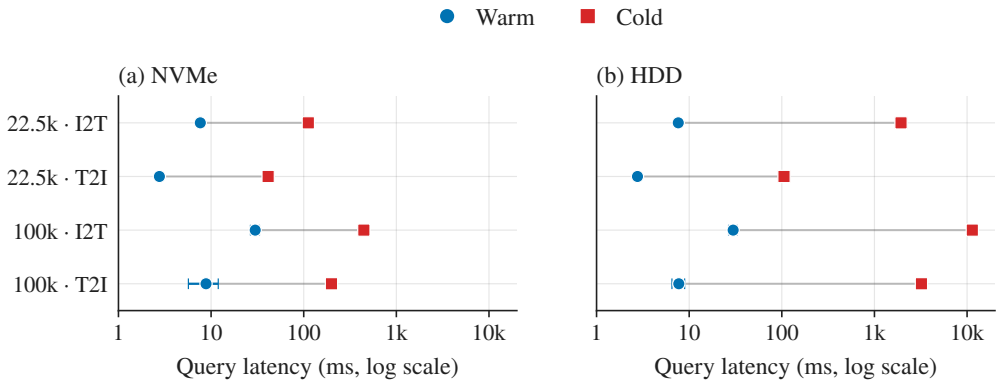


Figure S1: Warm versus cold COCO retrieval with 64-bit development models. Paired points show mean latency and bars show sample SD over three seeds; both panels use the same logarithmic axis.

The base hasher, gallery, and candidate-byte budget match the final models, allowing the feature-read component to be compared separately from model computation. At 100,000 sources, this component is 417.15 / 189.64 ms on NVMe and 11.36 / 3.21 s on HDD (I2T / T2I). The complete cold-query timings in the figure remain specific to the 64-bit models.

E.1 One-time materialization and query volume

On-demand and materialized execution produce identical codes and rankings for a fixed model, gallery, and candidate rule. Full materialization evaluates all retained heads from the base-code index and writes packed full-length codes. Let B_{full} be this one-time cost and $L_{\text{demand}}, L_{\text{stored}}$ the paired mean query times, in seconds. Over n serial queries,

$$T_{\text{demand}}(n) - T_{\text{stored}}(n) = n(L_{\text{demand}} - L_{\text{stored}}) - B_{\text{full}}. \tag{S1}$$

Figure S2 gives each seed's first query favoring stored codes, computed from measured costs. Across eight dataset–scale–direction groups, the thresholds range from 33 to 154 queries. B_{full} covers encoding and OS-cache writes without `fsync`; common training is excluded.

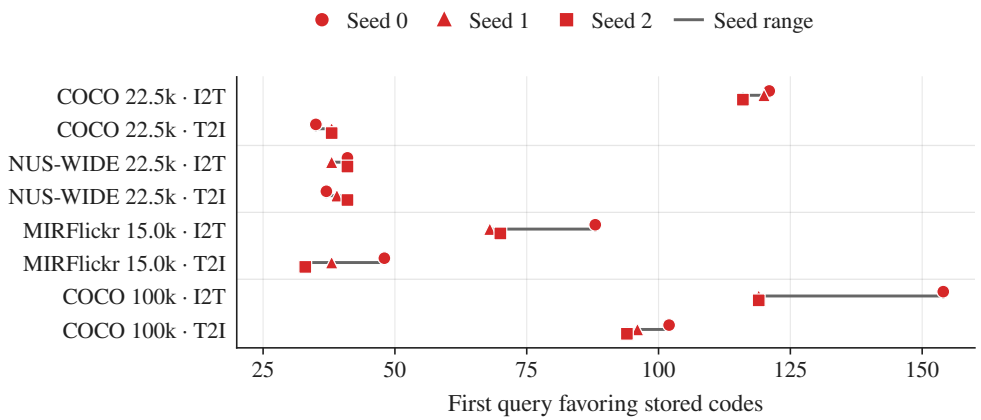


Figure S2: Queries needed to amortize one-time full-code materialization, computed using Equation S1. Points represent three training seeds; segments span their minimum and maximum.

Across T accepted expansions, let H_t historical items receive d_t new bits at update t. Reusing stored residuals gives cumulative logical writes

$$W_{\text{stored}} = \sum_{t=1}^{T} H_t d_t, \qquad W_{\text{demand}} = 0. \tag{S2}$$

Both executions store new arrivals and may access historical data for training. Equation S2 counts bits before padding; incremental encoding and write costs are unmeasured and differ from B_{full}, which covers the full retained chain.

F Supplementary results and matched controls

F.1 Final Test controls and query cost

Table S5 reports final Test controls using the same selected ODEH states as Table 1, together with PBEH at 100,000 COCO sources. Base-32 and cosine share ODEH's base index and candidates. The corresponding COCO feature-input timings are in Table S3.

Table S5: Final Test controls and the 100,000-source external comparison (%, mean ± sample SD over three complete training seeds; I2T / T2I). Base-32, cosine, ODEH, and residual refitting share 1% base-code candidates; PBEH scans the full gallery. ODEH freezes all old coordinates; residual refitting permits replacing old heads. Differences in the text are computed before rounding.

Ranking	Bits	P@10↑	nDCG@10↑	Failure↓
COCO, 22,500 sources				
Base code	32	86.24±0.44 / 86.10±0.47	86.33±0.31 / 86.16±0.41	10.37±0.30 / 9.01±0.45
Cosine	FP32	95.58±0.27 / 94.15±0.12	95.73±0.28 / 94.40±0.11	**2.61**±0.38 / 3.67±0.22
ODEH	80	**95.76**±0.12 / **95.29**±0.29	**95.85**±0.12 / **95.51**±0.23	2.90±0.10 / **3.15**±0.24
NUS-WIDE, 22,500 sources				
Base code	32	73.69±2.59 / 73.93±2.06	72.15±1.55 / 73.70±2.58	17.07±1.41 / 20.29±2.00
Cosine	FP32	80.64±0.63 / 77.09±1.00	80.87±0.61 / 77.38±0.90	12.45±0.33 / 17.23±1.15
Residual refit^R	48	**90.12**±0.41 / **86.07**±0.61	**90.46**±0.48 / **86.51**±0.61	**5.89**±0.40 / **8.17**±0.17
ODEH	80	86.80±1.76 / 83.72±1.12	87.30±2.04 / 84.42±1.11	7.00±0.90 / 9.06±0.73
MIRFlickr, 14,983 sources				
Base code	32	90.97±1.08 / 89.29±1.30	91.02±0.93 / 89.11±1.37	3.17±0.10 / 6.77±0.74
Cosine	FP32	92.08±0.83 / 90.89±0.56	92.28±0.84 / 90.94±0.56	2.89±0.31 / 6.13±0.12
ODEH	48–64	**95.54**±0.86 / **93.58**±0.63	**95.73**±0.89 / **93.64**±0.72	**1.67**±0.23 / **3.33**±0.29
COCO, 100,000 sources				
Base code	32	86.03±0.19 / 86.17±0.43	85.99±0.31 / 86.15±0.43	10.54±0.20 / 9.49±0.47
Cosine	FP32	96.12±0.18 / 95.13±0.10	**96.24**±0.19 / 95.29±0.07	**2.17**±0.17 / 3.05±0.16
PBEH (full scan)	64–65	**96.23**±0.23 / **96.25**±0.09	96.22±0.22 / **96.28**±0.10	2.23±0.27 / **2.89**±0.05
ODEH	64–80	94.66±1.76 / 94.72±1.40	94.72±1.75 / 94.85±1.42	3.63±1.18 / 3.47±0.83

Values: I2T/T2I, percent; mean ± sample SD across three seeds. PBEH uses full scan; all other rows share a 32-bit base index and its 1% candidates. Cosine uses 512-dimensional FP32 features. R: old residual heads may be retrained. Bold: best mean within each comparison.

Residual-refit query cost. Table S6 measures the same NUS-WIDE models: 80-bit ODEH and 48-bit residual refit, using the shared base index and 1% candidates. Both can generate residuals on demand or read precomputed codes. Refitting permits changes to old residual coordinates and changes total width; the comparison therefore varies both update constraints and capacity. Timings use fixed Validation queries.

F.2 Validation metrics and uncertainty

Table S7 provides a separate Validation analysis of full-gallery mAP@all and actual-service R@10. ODEH uses the development-selected 64/64/64, 80/80/80, and 32/48/48-bit states on COCO, NUS-WIDE, and MIRFlickr, respectively, before terminal Calibration. We generate complete logical codes for full-gallery mAP and use 1% candidates for R@10.

Table S6: NUS-WIDE query latency at 22,500 sources (ms, mean ± sample SD over three training seeds; I2T / T2I). The base-code-ready timer starts with query features and base codes available; the feature-input timer also includes query encoding and transfer.

Model	Bits	Execution	I2T (ms)↓	T2I (ms)↓
Base code and query features ready				
ODEH	80	On demand	1.885±0.010	1.940±0.020
ODEH	80	Precomputed	1.061±0.010	1.116±0.019
Residual refit	48	On demand	1.081±0.014	1.132±0.026
Residual refit	48	Precomputed	0.670±0.011	0.728±0.021
Frozen query feature ready				
ODEH	80	On demand	2.254±0.017	2.319±0.030
ODEH	80	Precomputed	1.412±0.013	1.466±0.020
Residual refit	48	On demand	1.473±0.016	1.528±0.028
Residual refit	48	Precomputed	1.036±0.013	1.087±0.024

Mean ± sample SD over three training seeds, with 512 fixed Validation queries per direction and three timing repeats per seed. NUS-WIDE has 22,500 sources; both models use the same 32-bit index and 1% candidates. Features reside in RAM. The second starting point includes base-code query encoding and transfers; both exclude backbone feature extraction. Precomputed stores the same full codes before querying.

Metric definitions. Let y_i indicate relevance at rank i, and let $N_q = |\mathcal{P}(q)|$. Precision and recall are $P@K = K^{-1}\sum_{i=1}^{K} y_i$ and $R@K = N_q^{-1}\sum_{i=1}^{K} y_i$. Binary nDCG divides $\sum_{i=1}^{K} y_i/\log_2(i+1)$ by $\sum_{i=1}^{\min(K,N_q)} 1/\log_2(i+1)$. Full-gallery average precision is $N_q^{-1}\sum_{i=1}^{|\mathcal{D}|} y_i P@i$. Metrics are averaged over queries within each direction, then summarized across training seeds. Because a gallery can contain thousands of relevant items under label-overlap relevance, R@10 measures a different quantity from Top-10 precision or query failure.

Paired uncertainty estimates. Source bootstrap keeps paired outputs for each query source together. In Figure 6, each source's control-minus-ODEH difference is averaged across three fitted models separately for each retrieval direction. We resample 2,500 sources with replacement 2,000 times using random seed 20260920 and use the 2.5th/97.5th percentiles as pointwise 95% intervals conditional on those models. Sample SD separately describes variation across complete training seeds.

The capacity contrasts in Table S9 first average each source's paired difference over directions and three models, then use 2,000 source-bootstrap draws with seed 0. The COCO two-tier versus target-only comparison uses 2,000 draws with seed 20260915, sharing each source draw across seeds and directions and reporting directions separately.

F.3 Capacity and update constraints

Table S8 compares target-only and two-tier selection under actual candidate retrieval, together with the NUS-WIDE residual-refit control.

Table S9 compares target-only retained states, prespecified expansion candidates, and residual-refit controls, all using 1% candidates. Residual-refit controls keep the 32-bit base but replace residual heads at 48 or 64 total bits, relaxing ODEH's constraint on previously retained coordinates. The comparison pairs new-head initialization, arrived training data, objective, and 80 epochs. Total chain capacity and parameters differ, so it does not isolate freezing under equal total capacity.

Under COCO target-only histories, additional 16-bit proposals at total lengths 64/80/64 reduce bidirectional failure by 1.20 points (95% paired-source CI $[0.95, 1.47]$). These capacity probes are distinct from the deployed selection decisions.

In this matched COCO Validation comparison, the 16-bit extension and the refitted 48-bit control have nearly identical mean failure: their bidirectional difference is -0.007 percentage points, with paired 95% interval $[-0.20, 0.18]$. Appending 32 rather than 16 bits changes mean failure by $+0.013$ points, with interval $[-0.12, 0.15]$. These source-paired intervals are conditional on the three trained models. The COCO estimate concerns matched Validation proposals; the NUS-WIDE comparison in Table S5 uses final Test queries.

Table S7: Development-selected Validation metrics before terminal Calibration (%, mean ± sample SD over three complete training seeds). mAP uses full-gallery ranking; R@10 uses each method's actual service path and divides relevant hits by all relevant gallery items. COCO has about five captions per image, so I2T has a larger recall denominator.

Method	Bits	mAP@all (%)↑		R@10 (%)↑	
		I2T	T2I	I2T	T2I
COCO					
PBEH	64–65	64.88±2.46	68.29±1.16	0.1051±0.0045	0.5346±0.0008
BITS-CM	128	63.17±0.07	62.00±0.06	0.1102±0.0008	0.5439±0.0034
OH-ELS	64	58.93±0.37	58.61±0.48	0.0422±0.0055	0.1602±0.0190
MCDLCH	64	54.25±0.54	61.30±0.36	0.0143±0.0006	0.0950±0.0074
DGHDGH [S]	64	71.02±1.21	72.00±1.51	0.0945±0.0042	0.4625±0.0176
DDBH [S]	64	69.57±0.77	63.83±2.69	0.0774±0.0027	0.3959±0.0061
ODEH	64	69.53±0.43	70.22±0.37	0.1029±0.0012	0.5070±0.0045
NUS-WIDE					
PBEH	64	55.84±1.55	54.18±1.40	0.1606±0.0049	0.1459±0.0053
BITS-CM	32	69.78±0.04	65.40±0.06	0.1800±0.0006	0.1622±0.0001
OH-ELS	64	78.79±0.26	70.27±0.27	0.1573±0.0069	0.1317±0.0067
MCDLCH	64	66.70±2.48	70.58±1.03	0.0959±0.0084	0.1023±0.0053
DGHDGH [S]	64	70.12±0.89	70.17±0.85	0.1645±0.0010	0.1476±0.0003
DDBH [S]	64	68.93±0.64	69.64±1.18	0.1429±0.0029	0.1281±0.0026
ODEH	80	60.45±0.93	60.18±2.26	0.1716±0.0057	0.1603±0.0058
MIRFlickr					
PBEH	64–65	71.02±1.19	69.27±0.46	0.1278±0.0008	0.1183±0.0014
BITS-CM	32	78.26±0.03	74.67±0.01	0.1327±0.0010	0.1182±0.0005
OH-ELS	64	92.24±0.22	87.30±0.12	0.1318±0.0007	0.1227±0.0013
MCDLCH	64	85.65±0.42	81.30±0.40	0.1173±0.0024	0.1069±0.0014
DGHDGH [S]	64	81.26±0.26	83.21±0.37	0.1333±0.0006	0.1263±0.0007
DDBH [S]	64	77.75±0.50	77.29±2.50	0.1126±0.0026	0.1015±0.0072
ODEH	32–48	78.10±0.47	80.26±0.67	0.1258±0.0025	0.1214±0.0025

S: full retraining on all arrived data.

Table S8: Validation selection controls with actual 1% service (%, mean ± sample SD; I2T / T2I). Both selection rules freeze old bits; the NUS-WIDE refit control replaces old residuals.

Selection / control	Bits	P@10↑	nDCG@10↑	Failure↓
COCO				
Target only	48–64	94.00±0.13 / 93.83±0.06	94.05±0.11 / 93.96±0.11	4.25±0.12 / 3.93±0.12
Two-tier selection	64	93.44±0.58 / 93.40±0.37	93.48±0.58 / 93.48±0.37	4.49±0.33 / 4.17±0.33
NUS-WIDE				
Target only	32	73.79±2.52 / 73.55±1.85	72.30±1.53 / 73.40±2.38	17.61±1.73 / 20.76±2.04
Two-tier selection	80	86.90±1.76 / 83.75±1.55	87.38±2.10 / 84.48±1.48	6.76±1.29 / 8.92±1.08
Residual refit	48	90.53±0.24 / 86.11±1.01	90.91±0.28 / 86.62±0.99	5.31±0.15 / 7.85±0.52
MIRFlickr				
Target only	32–48	92.43±1.02 / 91.34±1.11	92.51±1.04 / 91.37±1.17	2.75±0.79 / 4.92±0.63
Two-tier selection	32–48	92.43±1.02 / 91.34±1.11	92.51±1.04 / 91.37±1.17	2.75±0.79 / 4.92±0.63

Table S9: Capacity controls on terminal COCO Validation. Failure rates are percentages, mean ± sample standard deviation across three complete training seeds. Slash-separated lengths follow seeds 0/1/2.

Configuration	Logical bits	I2T failure (%)	T2I failure (%)
Retained before the update	48/64/48	4.25±0.12	3.93±0.12
Append 16, old bits frozen	64/80/64	2.91±0.16	2.88±0.17
Append 32, old bits frozen	80/96/80	2.84±0.04	2.97±0.13
Refit residuals, 48 bits	48	2.93±0.12	2.87±0.09
Refit residuals, 64 bits	64	2.84±0.32	2.88±0.07

F.4 MATCHED TERMINAL-EXPANSION CONTROLS

All controls freeze the same 48/64/48-bit old histories for seeds 0/1/2 and train only the new terminal block. A1 uses global source-shuffled batches of 256 paired examples, sampling one caption per source per epoch. Its boundary candidates are the opposite-modality examples within the batch. A2 keeps retrieved candidate pools and the boundary-valid query mask, replacing the boundary term with a pairwise objective. For each positive, softplus compares its logit with the log-mean-exp of the negative logits. For complete Hamming distance D, the logit is $-D/s$, with $s = \sqrt{b_{t-1} + d}/2$ and temperature one. Positive terms are averaged within each query, then queries are weighted equally. Both controls retain the source-pair regularizers, historical distances, new-head initialization, and update count.

A3 replaces only the new head's input with a fixed, normalized, orthogonal 512-dimensional embedding of the complete old code. Frozen older heads still use CLIP features. A4 trains with base-code and new-bit distances, omitting old residual distances. All controls use complete logical codes for Validation and service, with the same 1% evaluation candidates. Checkpoint selection follows ODEH's shorter-prefix ordering: worst directional failure, bidirectional mean, and the original boundary-based Validation objective.

Figure 6 reports control-minus-ODEH failure at both increments. With 16 added bits, A3's I2T/T2I failure differences have paired 95% intervals $[0.467, 1.067]$ / $[0.267, 0.773]$ points. For A2, the failure intervals are $[-0.013, 0.600]$ / $[-0.093, 0.453]$. Table S10 gives seed variation for the 32-bit additions.

The corresponding precision comparison for the 16-bit A2 control gives P@10 differences of $+0.079$ / $+0.508$ points relative to ODEH, with paired 95% intervals $[-0.143, 0.300]$ / $[0.311, 0.720]$. Here, higher mean precision coexists with higher mean failure, distinguishing average relevance from query-level success.

Table S10: Matched COCO Validation controls with 32 added bits. Entries are control-minus-ODEH failure in percentage points, mean ± sample SD of three paired seed differences. Positive values favor ODEH; all controls use the same 1% service candidates.

Control	I2T	T2I
A1: global in-batch candidates	-0.067 ± 0.061	$+0.027 \pm 0.061$
A2: pairwise objective	$+0.040 \pm 0.317$	$+0.213 \pm 0.167$
A3: old-code input	$+0.973 \pm 0.220$	$+0.560 \pm 0.212$
A4: omit old residual distance	$+0.067 \pm 0.083$	-0.093 ± 0.023

F.5 GROWTH HISTORIES AND FROZEN-SNAPSHOT SENSITIVITY

Local selection and subsequent growth. On COCO, two-tier selection retains 48 bits at the first stage in seeds 0 and 2, where target-only keeps 32. The two rules end at 64/64/64 and 48/64/48 bits, respectively. Terminal P@10 decreases by 0.557 / 0.429 points (95% paired-source CIs $[-0.803, -0.328]$ / $[-0.625, -0.229]$), while failure rises by 0.24 points in each direction, with intervals crossing zero. Stage-local acceptance therefore need not improve final quality. MIRFlickr follows identical histories under both rules.

Frozen-snapshot sensitivity. Figure S3 varies candidate fraction or absolute target on frozen terminal development models. The budget sweep compares 0.5%, 1%, 2%, and 5% candidates with full-gallery ranking at the same retained widths: COCO 64, NUS-WIDE 80, and MIRFlickr 32/48/48 bits. Slashes separate seeds 0/1/2.

The target sweep fixes 1% candidates and applies only the absolute tier at 4%, 6%, 8%, or 10% in both directions. Incoming widths are COCO 64, NUS-WIDE 64, and MIRFlickr 32/48/48; proposals and fitted histories stay fixed. "None" means no feasible prefix; a selected total width equal to the incoming width means no expansion.

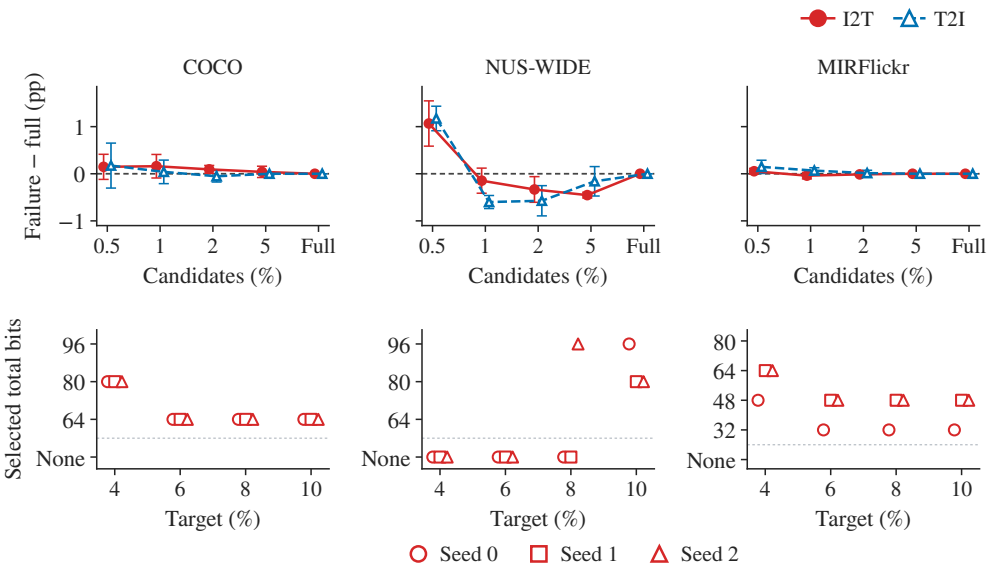


Figure S3: Frozen-snapshot Validation sensitivity. Top: candidate-minus-full-gallery failure (three seeds, mean ± sample SD). Bottom: shortest total width meeting the target in both directions under the absolute tier alone, with fixed proposals.

Growth with no additional training. In the fixed-base growth study, 32-bit failure changes from 10.60% / 9.04% to 10.00% / 8.72% as the gallery grows from 2,500 to 22,500 sources, with paired change intervals including zero. Table S11 extends COCO galleries without further training. The target-only chain retains its 22,500-source widths of 48/64/48 and meets 6% in every seed at both larger scales. The nested 64-bit base was trained at initialization without residual heads. Gallery growth need not exhaust a fixed representation: arrivals also add relevant items under label-overlap relevance.

Table S11: Frozen-encoder COCO growth: failure (%) with 1% candidates and no training after 22,500 sources (I2T / T2I; three seeds, mean ± sample SD).

Frozen representation	67,500 sources	100,000 sources
Base 32	10.56 ± 0.28 / 8.99 ± 0.36	10.33 ± 0.10 / 9.00 ± 0.49
Nested base 64	9.08 ± 0.53 / 7.49 ± 0.33	9.37 ± 0.56 / 7.59 ± 0.28
Retained chain, 48/64/48	4.35 ± 0.10 / 4.03 ± 0.13	4.31 ± 0.14 / 3.92 ± 0.04