Closing the Indexing-Decoding Gap in Multimodal Generative Retrieval via Prefix Retention Optimization

Yufei Chen[1,2] **Zihan Wang**[3,†] **Yubao Tang**[4]
Yukun Zhao[1,2] **Maarten de Rijke**[4] **Zhaochun Ren**[5,†]
[1]Shandong University
[2]Shandong Academy of Artificial Intelligence
[3]CISPA Helmholtz Center for Information Security
[4]University of Amsterdam [5]Leiden University
cyf200409@gmail.com, zhw.cypher@gmail.com
y.tang3@uva.nl, zhaoyukun@sdu.edu.cn, m.derijke@uva.nl
z.ren@liacs.leidenuniv.nl

Abstract

Multimodal generative retrieval formulates multimodal retrieval as discrete identifier generation, eliminating the need for explicit similarity search over external embeddings. Existing approaches construct identifiers via residual quantization and decode them with trie-constrained beam search. This combination introduces an *indexing-decoding gap*: identifier learning objectives, including reconstruction and contrastive losses, do not explicitly enforce prefix discriminability during decoding. As a result, even well-optimized identifiers can be irreversibly pruned early in beam search due to low-ranked prefixes. We theoretically characterize this gap and derive a survival bound that relates prefix retention to three controllable factors in indexing and decoding. Building on this bound, we propose **PRO**, *prefix retention optimization*, a unified framework comprising three mechanisms: (i) prefix ranking distillation aligns quantized prefix rankings with those induced by pre-quantization embeddings using a listwise loss; (ii) vocabulary scheduling increases codebook sizes from shallow to deep residual quantization levels to reduce early competition from non-target prefixes; and (iii) geometric score fusion vectorizes each candidate prefix and incorporates its similarity to the query into beam search scoring, further reducing the indexing–decoding mismatch. Experiments on nine multimodal retrieval tasks show that PRO improves retention of target identifier prefixes and outperforms existing multimodal generative retrieval baselines.[1]

1 Introduction

Multimodal information retrieval requires matching queries and targets across modalities such as text, images, and composed inputs. Dense retrieval dominates this setting by comparing learned continuous embeddings, but it requires maintaining and searching external vector indexes [42]. Generative retrieval (GR) offers an alternative: a sequence-to-sequence model directly generates a discrete identifier for the target item [5, 38, 40]. This paradigm was first studied in text retrieval and recommender systems [31, 40], and has recently been extended to multimodal search [17, 24, 49].

Existing multimodal generative retrieval (MGR) methods, including IRGen [49], GRACE [24], AVG [25], and SemCORE [21], focus on individual cross-modal retrieval tasks. GENIUS [17] takes a step further by introducing a unified generative framework for diverse multimodal retrieval tasks,

[†]Co-corresponding authors.
[1]Our code is available at https://github.com/layingfish/MGR_PRO.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

achieving state-of-the-art performance in this setting. It consists of three stages: (i) an indexing stage that uses a unified multimodal encoder to map queries and target items into a shared continuous embedding space, and quantizes item embeddings into multi-level residual quantization (RQ) codes as identifiers; (ii) a retrieval training stage that trains an autoregressive decoder to generate these identifiers from queries; and (iii) a decoding stage where trie-constrained beam search ensures that the decoder only produces valid candidate identifiers during inference [5]. Despite its versatility, GENIUS exhibits a structural *indexing–decoding gap*, as illustrated in Fig. 1: during indexing, identifier learning objectives, including reconstruction and contrastive losses, are optimized independently of decoding and therefore do not explicitly enforce prefix discriminability for decoding. As a result, prefix survival is difficult to guarantee because trie-constrained beam search expands prefixes sequentially and irrevocably prunes low-ranked candidates at each level. In multimodal retrieval, this gap arises where multimodal representations are converted into discrete identifiers: a target ranked highly by the multimodal encoder can still be lost if its intermediate prefixes do not preserve this ranking.

In this paper, we theoretically characterize this gap and show that existing indexing objectives are insufficient to determine whether a target identifier prefix survives beam search: two tokenizers can achieve the same objective values while one preserves the target prefix and the other prunes it under decoding (see Thm. 1). We further derive a survival bound that relates prefix survival to three controllable factors across indexing and decoding (see Thm. 2). Building on this bound, we propose **PRO**, the *prefix retention optimization* framework, to improve prefix survival across indexing and decoding by targeting the controllable factors. During indexing, prefix ranking distillation enforces consistency between prefix rankings of RQ partial reconstructions and those induced by pre-quantization embeddings via a listwise objective, while vocabulary scheduling progressively increases codebook sizes across RQ levels to mitigate early competition from non-target prefixes. During decoding, geometric score fusion augments beam search scoring with query–prefix similarity derived from reconstructed prefixes, reducing the mismatch between indexing and decoding.

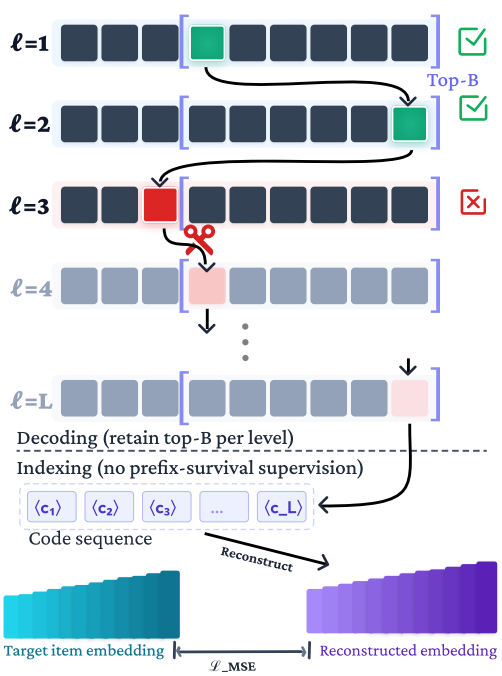


Figure 1: The indexing-decoding gap. **Top** (decoding): Beam search prunes irrevocably at each level; the target item (colored path) survives levels 1–2 but is lost at level 3. **Bottom** (indexing): Existing identifier learning objectives do not explicitly supervise prefix survival during decoding.

We evaluate PRO on nine multimodal retrieval tasks spanning text-to-image, image-to-text, and composed retrieval. PRO outperforms state-of-the-art MGR baselines on all nine tasks, improving the macro-averaged R@1/R@5 of GENIUS by 26.5%/19.1% with only 3.9–4.9% extra latency, and remains stronger than GENIUS and a prefix-weighted training baseline that use a 4× larger beam. Further results show gains on the M-BEIR global pool, across encoders, and on the RQ-based recommender TIGER [31].

Our contributions are: (i) We identify and formalize the indexing–decoding gap in RQ-based MGR as a mismatch between identifier learning objectives and sequential beam search pruning. (ii) We show that existing identifier learning objectives do not determine prefix survival, and derive a bound relating it to three controllable factors across indexing and decoding. (iii) We propose PRO, a framework for improving prefix survival via prefix ranking distillation, vocabulary scheduling, and geometric score fusion. (iv) Experiments on nine multimodal retrieval tasks show that PRO improves prefix survival and consistently surpasses state-of-the-art MGR methods, including under larger beams and on the official M-BEIR global pool, with strong generalizability across multimodal encoders and RQ-based methods.

2 Preliminaries

To facilitate the theoretical analysis of the indexing–decoding gap and the proposed PRO, we briefly introduce the MGR framework proposed by GENIUS [17]. This framework consists of three stages: indexing, retrieval training, and decoding.

Indexing stage. The indexing stage constructs discrete identifiers for items by progressively quantizing their representations, which are extracted by a frozen multimodal encoder. These identifiers serve as supervision for subsequent decoder training.

Multimodal encoder. The encoder maps queries and targets from heterogeneous modalities, including images, texts, and image-text compositions, into a shared d-dimensional embedding space. It is trained with a contrastive loss and kept frozen during both identifier generation and decoder training.

Residual quantization. Let $q \in \mathbb{R}^d$ and $x_j \in \mathbb{R}^d$ denote the pre-quantization fused query and target embeddings, respectively. We use z_j^ℓ to denote the RQ code index of item j at level ℓ. The tokenizer maps each target embedding x_j to a discrete identifier via an L-level residual quantizer. Specifically, it (i) initializes the residual as $r_0 = x_j$; (ii) at each level ℓ, selects the nearest codeword from the level-specific codebook $\{c_{\ell,1}, \ldots, c_{\ell,V_\ell}\} \subset \mathbb{R}^d$ by $z_j^\ell = \arg\min_{v \in \{1,\ldots,V_\ell\}} \|r_{\ell-1} - c_{\ell,v}\|^2$; and (iii) updates the residual as $r_\ell = r_{\ell-1} - c_{\ell,z_j^\ell}$. Here, V_ℓ denotes the vocabulary size at level ℓ. The resulting code sequence $(z_j^1, \ldots, z_j^L)$ forms the discrete identifier of item j, with partial reconstruction $\hat{x}_{j,\ell} = \sum_{t=1}^{\ell} c_{t,z_j^t}$ and full reconstruction $\hat{x}_j = \hat{x}_{j,L}$; applying the same residual quantizer to q gives the full quantized query vector $\hat{q} = \sum_{\ell=1}^{L} c_{\ell,z_q^\ell}$, where z_q^ℓ is selected by the same residual-quantization rule initialized at q. Following GENIUS [17], we adopt a modality-decoupled first code level to distinguish target modalities: the first-level codebook contains three codes (one per modality), while subsequent levels are shared to capture semantic content in the unified embedding space.

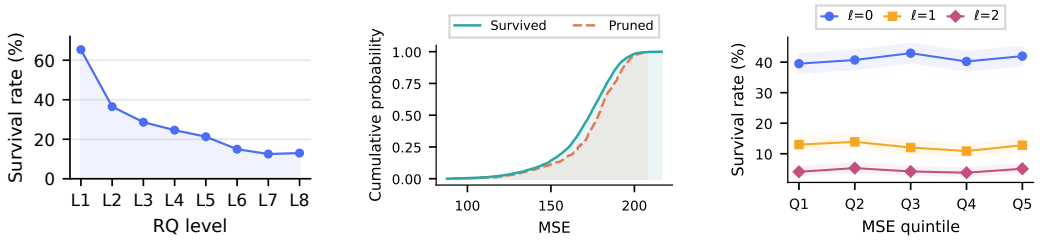
Training objective. The standard RQ tokenizer is trained with three objectives: (i) an RQ loss that stabilizes codebook learning by aligning residuals with selected codewords (updated via exponential moving average); (ii) a mean squared error (MSE) loss aligning quantized query and target vectors, $\mathcal{L}_{\text{mse}} = \|\hat{q} - \hat{x}_j\|_2^2$; and (iii) a contrastive loss [16] on pre-quantization embeddings, with one direction given by $\mathcal{L}_{\text{cl}} = -\log \frac{\exp(\langle q, x_j \rangle / \tau)}{\sum_k \exp(\langle q, x_k \rangle / \tau)}$. During training, the same RQ codebooks are applied to both q and x_j to obtain $\hat{q}$ and $\hat{x}_j$, while only the target code sequence is stored as the item identifier. Here, x_k denotes a positive target or a contrastive negative, and $\tau > 0$ is the temperature. The standard RQ loss is given in App. D. Overall, these objectives constrain local residual-to-codeword fitting, full quantized vectors, and pre-quantization embeddings, but do not explicitly regulate query-conditioned rankings of intermediate prefixes or partial reconstructions $\hat{x}_{j,\ell}$.

Retrieval training and decoding. After tokenizer training, the discrete identifiers supervise an autoregressive decoder. (i) *Training.* The decoder [30] maps query embeddings to code sequences using a cross-entropy loss over the L tokens. (ii) *Decoding.* Candidate identifiers are generated via trie-constrained beam search [5] with beam width B, where decoding is restricted to valid identifiers and beams are ranked by cumulative log-probability.

3 The Indexing-Decoding Gap

In this section, we provide a theoretical analysis of the *indexing-decoding gap* between identifier learning and target item decoding. We first establish a negative result: existing tokenizer objectives used for identifier learning, including contrastive, residual fitting, and reconstruction losses, do not determine whether the target prefix survives beam search. For theoretical analysis, we simulate beam search scoring with a rule that ranks each prefix by the similarity between the query and its RQ partial reconstruction, $\langle q, \hat{x}_{j,\ell} \rangle$. We refer to this scoring rule as the *quantized oracle*. This oracle evaluates prefixes using partial reconstructions induced by RQ codes in the quantized representation space.

Theorem 1. *For any beam width $B \geq 1$ and RQ levels $L \geq 2$, there exist a query q, a set of $B+1$ items, and two L-level residual quantizers $\mathcal{Q}^+$ and $\mathcal{Q}^-$ such that:* (i) *$\mathcal{Q}^+$ and $\mathcal{Q}^-$ share the same pre-quantization query and item embeddings, produce identical greedy code assignments, and achieve identical values for all training objectives, including contrastive, residual-fitting, and reconstruction losses computed from full reconstructions.* (ii) *Under trie-constrained beam search over valid item*



(a) Per-level token survival (b) MSE: survived vs. pruned (c) Survival by MSE quintile

Figure 2: Empirical diagnostics of the indexing–decoding gap on COCO under quantized oracle scoring (B=50, 5,000 queries). (a) The per-level survival rate of the target identifier token drops sharply across early RQ code positions. (b,c) Full reconstruction MSE weakly predicts prefix survival: survived/pruned empirical cumulative distribution functions (ECDFs) nearly overlap, and survival remains flat across MSE quintiles.

identifiers with width B using quantized oracle scores, the target prefix survives at every level in $\mathcal{Q}^+$, whereas it is pruned at level 1 in $\mathcal{Q}^-$.

Proof sketch. We construct two RQ instances $\mathcal{Q}^+$ and $\mathcal{Q}^-$ over the same query and $B+1$ items. First, both quantizers share the same pre-quantization embeddings, select the same RQ code index at every level under the greedy nearest codeword rule, and have identical values for the existing tokenizer objectives. The only difference is in the first two codewords selected by the target item. In $\mathcal{Q}^+$, the first selected codeword has a positive component along the query direction, so the target prefix receives the highest level 1 quantized oracle score. In $\mathcal{Q}^-$, we reverse this query direction component in the first selected codeword, which lowers the target's level 1 prefix score, and add the opposite compensation to the second selected codeword, so that the sum of the first two selected codewords and therefore the full reconstruction remain unchanged. Consequently, the existing tokenizer objectives considered in the theorem assign the same values to the two constructions. As a result, the target prefix survives under $\mathcal{Q}^+$ but is ranked below all B competitors and pruned at level 1 under $\mathcal{Q}^-$. Full details are provided in App. A. If the first code level is reserved for a modality token, the construction applies from levels 2 and 3.

Thm. 1 shows that existing tokenizer objectives cannot guarantee prefix discriminability during beam search. Although later RQ levels can compensate early errors in the final reconstruction, beam search is irreversible and cannot recover a target once its prefix has been pruned. Therefore, prefix discriminability at intermediate levels must be explicitly considered beyond optimizing complete identifiers. This is analogous to the failure of successive refinement in source coding [7]: a code optimized for final distortion need not be useful at intermediate stages.

Empirical analysis. We examine whether the failure mode revealed by Thm. 1 also appears in practice. We measure the per-level survival rate of the target identifier token under quantized oracle scoring on COCO [26, 42] with a RQ-based tokenizer [19] (L=16, V=4096, B=50, 5,000 queries). At each RQ level, this metric conditions on the preceding target prefix and checks whether the target token at that level is ranked among the top-B candidate tokens. As shown in Fig. 2a, the target token becomes much harder to retain after the first RQ level. This illustrates the practical relevance of the failure mode in Thm. 1: due to insufficient prefix discriminability, target identifier tokens are easily lost during beam decoding. We further use complete reconstruction MSE as a diagnostic of reconstruction quality. Fig. 2b shows that the ECDF curves of survived and pruned items nearly overlap. Fig. 2c groups items by MSE quintiles and shows that the prefix survival rate varies little across different MSE ranges. These results suggest that good complete reconstruction alone does not predict whether a target survives beam search; App. B.8 reports the same analysis for other indexing objectives and beam widths.

4 Prefix Retention Optimization

Under what conditions does the target prefix survive and *how can these conditions be formulated as continuous quantities amenable to optimization*? To answer those questions, we first derive a survival

bound that characterizes prefix survival through three quantities: ranking divergence, teacher margin, and decoder mismatch. The bound provides a constructive counterpart to Thm. 1: while existing objectives cannot determine survival, survival is guaranteed when the errors introduced by indexing and decoding are sufficiently small relative to the teacher margin. Guided by this bound, we propose *Prefix Retention Optimization (PRO)*, which reduces ranking divergence, enlarges the teacher margin, and mitigates decoder mismatch via three corresponding components.

Survival bound. We begin with prefix-level notation. For a target item j with RQ code sequence $z_j^{1:L}$, let $\hat{x}_{j,\ell} := \sum_{t=1}^{\ell} c_{t,z_j^t}$ denote its partial reconstruction after ℓ levels, so that $\hat{x}_{j,L} = \hat{x}_j$. Let $\mathcal{P}_\ell$ denote the set of distinct length-ℓ code prefixes, and let $p \in \mathcal{P}_\ell$ denote a generic prefix. Let $p_\ell^+ := z^{1:\ell}$ denote the target prefix at level ℓ. We analyze each beam step through distributions over $\mathcal{P}_\ell$: a teacher distribution π_ℓ^* defined by pre-quantization embedding similarities, a *quantized oracle distribution* $\bar{\pi}_\ell$ induced by partial reconstructions, and a *decoder prefix distribution* $\tilde{\pi}_\ell$ derived from decoder scores. All distributions are conditioned on the query embedding q, and their formal definitions are given in App. A.

The survival of the ground-truth prefix p_ℓ^+ is governed by three quantities:

$$K_\ell = D_{\mathrm{KL}}\big(\pi_\ell^* \,\|\, \bar{\pi}_\ell\big), \qquad \text{(ranking divergence)} \tag{1}$$

$$m_\ell = \pi_\ell^*(p_\ell^+) - \pi_{\ell,B+1}^*, \qquad \text{(teacher margin)} \tag{2}$$

$$\varepsilon_\ell = \mathrm{TV}\big(\bar{\pi}_\ell,\, \tilde{\pi}_\ell\big), \qquad \text{(decoder mismatch)} \tag{3}$$

where $\pi_{\ell,B+1}^*$ denotes the $(B{+}1)$-th largest probability under the teacher distribution (with $V_\ell > B$ in practice). Thus, K_ℓ measures the divergence between teacher and quantized rankings, m_ℓ is the top-B margin under the teacher, and ε_ℓ captures the discrepancy between oracle and decoder distributions.

Theorem 2. *Fix a query q and beam width B. If, for every level $\ell \in \{1, \ldots, L\}$,*

$$\underbrace{\sqrt{\tfrac{1}{2}\,K_\ell(q)}}_{\textit{ranking divergence}} \;+\; \underbrace{\varepsilon_\ell(q)}_{\textit{decoder mismatch}} \;<\; \underbrace{\tfrac{1}{2}\,m_\ell(q)}_{\textit{teacher margin}}\,, \tag{4}$$

then the target prefix remains in the beam through all L levels.

When $m_\ell \le 0$, the bound provides no guarantee, as the target prefix is not ranked within the teacher top-B. All distributions are defined over the full prefix set $\mathcal{P}_\ell$; thus, being top-B globally implies being top-B within any subset containing p_ℓ^+ (see App. A). The proof proceeds in two steps. First, if quantization perturbs the teacher distribution by less than half of the teacher margin, the target prefix remains ahead of all prefixes outside the teacher top-B under the quantized oracle. Pinsker's inequality [4] converts this perturbation into the term $\sqrt{K_\ell/2}$. Second, applying the triangle inequality to incorporate the shift from the oracle to the decoder distribution yields the additional term ε_ℓ. Hence, survival is guaranteed when the combined deviation from quantization and decoding is sufficiently small relative to the teacher margin. The value of Thm. 2 lies in the design principle it exposes: reduce K_ℓ, reduce ε_ℓ, and enlarge m_ℓ at levels where beam pruning is irreversible. Although the bound is conservative, as is typical of Pinsker-based analyses [8, 33, 35], it correctly predicts the monotonic relationship between K_ℓ and prefix survival observed in our experiments (see Sec. 6). This principle directly motivates the three components of PRO.

The PRO Framework. As illustrated in Fig. 3, PRO integrates two tokenizer-side components with one inference-time component. Prefix ranking distillation and vocabulary scheduling operate on identifier learning, while geometric score fusion augments scoring in trie-constrained beam search.

Prefix ranking distillation. Prefix ranking distillation is designed to reduce the ranking divergence K_ℓ. For each query and each RQ level, we compare two distributions over the same candidate prefixes. The teacher distribution scores prefixes using the pre-quantization item embeddings, while the quantized distribution scores the same prefixes using the partial reconstructions produced by the RQ codebooks. If these two distributions disagree, the target prefix can be ranked differently after indexing. We therefore add a listwise KL distillation loss [1, 11] during RQ training:

$$\mathcal{L}_{\mathrm{rank}} \;=\; \mathbb{E}_q\left[\,\frac{1}{L}\sum_{\ell=1}^{L} D_{\mathrm{KL}}\big(\pi_\ell^* \,\|\, \bar{\pi}_\ell\big)\right]. \tag{5}$$

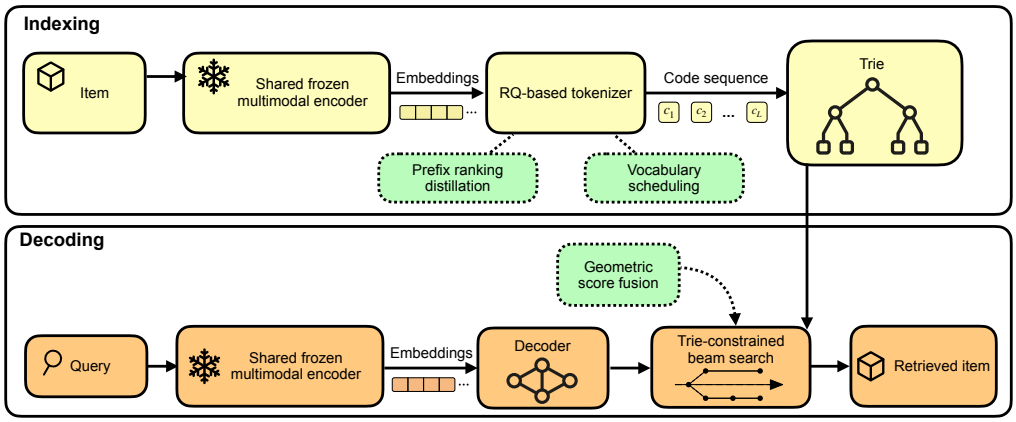


Figure 3: Overview of PRO. During tokenizer training, **prefix ranking distillation** reduces the ranking divergence K_ℓ and **vocabulary scheduling** enlarges the teacher margin m_ℓ. During inference, **geometric score fusion** helps reduce the decoder mismatch ε_ℓ.

Table 1: Performance of three vocabulary schedules on COCO.

	Level 0			Level 1			Level 2			Level 3		
	Desc.	Unif.	**Asc.**	Desc.	Unif.	**Asc.**	Desc.	Unif.	**Asc.**	Desc.	Unif.	**Asc.**
V	2048	1024	**512**	2048	1024	**512**	2048	1024	**512**	2048	1024	**512**
$\bar{m}_\ell \uparrow$	0.035	0.079	**0.109**	0.005	0.013	**0.022**	0.003	0.007	**0.013**	0.002	0.005	**0.010**
$\bar{K}_\ell \downarrow$	1.04	1.02	**0.92**	0.62	0.58	**0.51**	0.46	0.43	**0.35**	0.39	0.35	**0.28**
Surv. (%) $\uparrow$	55.2	66.6	**75.1**	19.5	31.2	**41.1**	16.4	23.9	**31.1**	10.0	18.0	**26.2**

Both distributions are approximated over an in-batch candidate set with shared temperature τ.[2] Minimizing this loss directly reduces the empirical estimate of K_ℓ. It encourages the partial reconstructions at each RQ level to preserve the prefix probability distribution induced by the pre-quantization embeddings for the same query. The ranking distillation loss is incorporated as an auxiliary term in the tokenizer training objective:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cl}} + \beta_{\text{rq}} \mathcal{L}_{\text{rq}} + \gamma_{\text{mse}} \mathcal{L}_{\text{mse}} + \lambda \mathcal{L}_{\text{rank}}, \tag{6}$$

where $\mathcal{L}_{\text{rq}}$ denotes the RQ loss, and β_{rq}, γ_{mse}, and λ control the strengths of the RQ, MSE, and ranking distillation losses.

Vocabulary scheduling. Vocabulary scheduling is designed to enlarge the teacher margin m_ℓ. At shallow RQ levels, beam search makes pruning decisions using only short prefixes. If the early codebook is large, the target prefix must compete with many non-target prefixes before deeper codewords can refine the identifier. A common RQ-VAE design [19] uses the same codebook size across residual levels, and product quantization [15] quantizes separate subspaces with separate subcodebooks, often using the same number of codewords per subspace. We therefore use an ascending schedule $V_1 \leq V_2 \leq \cdots \leq V_L$, allocating less capacity to early levels where pruning is irreversible. A smaller shallow vocabulary reduces the number of early competing prefixes. Under the teacher distribution, probability mass is spread over fewer early prefixes, increasing the probability gap between the target prefix and the strongest prefix just outside the beam. Thus, vocabulary scheduling enlarges the teacher margin at the levels where pruning is most irreversible. The larger deeper vocabularies preserve representation capacity after the early survival bottleneck has been alleviated. Tab. 1 compares three scheduling strategies: descending, uniform, and ascending. The ascending schedule consistently yields the largest $\bar{m}_\ell$ and the highest survival rates, while the descending schedule performs worst. Hence, reducing early-level competition is an effective way to enlarge the teacher margin in practice, despite the smaller early-level vocabulary reducing reconstruction capacity: reconstruction fidelity and beam search survival are distinct objectives.

Geometric score fusion. Geometric score fusion is designed to reduce the decoder mismatch ε_ℓ. During beam search, the decoder assigns a log-probability to each legal next token based on the

[2] In-batch approximation is a common practice in distillation-based training and provides a reasonable estimate of K_ℓ when the batch size is sufficiently large.

query and the current prefix. Each legal token also corresponds to an RQ codeword. Let $\hat{x}_p$ denote the partial reconstruction represented by the current beam prefix p, and let $r_p = q - \hat{x}_p$ denote the residual from this partial reconstruction to the query embedding. This residual is defined only for scoring candidate prefixes during decoding and should not be confused with the RQ residual used during tokenizer assignment. Since the decoder score does not explicitly measure whether a candidate codeword moves the partial reconstruction closer to the query, we use r_p to compute a geometric bias for each candidate token:

$$s_\ell^{\text{fuse}}(v \mid p, q) \;=\; \log P_{\text{decoder}}(v \mid q, p) \;+\; \omega\big(2r_p^\top c_{\ell,v} - c_{\ell,v}^\top c_{\ell,v}\big), \tag{7}$$

Here, $c_{\ell,v}$ is the codeword associated with candidate token v, and ω controls the strength of geometric score fusion. The added term is exactly the reduction in squared distance from the query to the partial reconstruction after appending v, since $\|r_p\|^2 - \|r_p - c_{\ell,v}\|^2 = 2r_p^\top c_{\ell,v} - c_{\ell,v}^\top c_{\ell,v}$. Thus, the bias rewards candidate codewords that reduce the squared query-to-reconstruction distance, with larger values corresponding to larger reductions. After adding this bias to the decoder log-probability, beam search can take the RQ reconstruction geometry into account when scoring candidate prefixes. This better aligns the fused decoder distribution with the quantized distribution and helps reduce decoder mismatch. App. A.4 formalizes this connection by defining the corresponding conditional distributions and showing when the fused distribution moves closer to the quantized oracle distribution. At $\omega = 0$, the score reduces to the original decoder score. Score fusion operates at inference only and requires no changes to tokenizer training.

5 Experimental Settings

We evaluate on 9 dataset–task pairs from the M-BEIR benchmark [42] and Flickr30k [46], covering three query types: single-modality (COCO [26], Flickr30k [46], WebQA [2], NIGHTS [9]), text-to-multimodal, and composed image+text (CIRR [27], OVEN [13], InfoSeek [3]). Candidate pools range from 1K to 612K. Unless otherwise stated, results use task-specific pools and report Recall@1 and Recall@5 without reranking; results on the official M-BEIR global pool are reported separately (Sec. 6). We compare with two dense baselines from UniIR [42], CLIP-SF and BLIP-FF, and five generative methods: IRGen [49], GRACE [24], AVG [25], SemCORE [21], and GENIUS [17]. Among them, GENIUS is the only universal multimodal generative retriever. The remaining four methods support only text-to-image retrieval. For fairness, GENIUS and PRO share the same CLIP-SF encoder and T5-small decoder [30]; all retrieval results are produced by the trained autoregressive decoder with trie-constrained beam search, while the quantized oracle (Sec. 3) is used only for diagnostic analyses. For prior single-task baselines, we report published or reproduced results when available. PRO uses 16 RQ levels, an ascending vocabulary schedule (512, 1024, 2048), prefix ranking distillation (weight 100, temperature 0.05), and geometric score fusion (weight 10) with beam size 50. Unless otherwise stated, retrieval metrics and Fig. 2 use beam size 50; the diagnostics in Figs. 5 and 6 use beam size 20. Full training details are in App. D.

6 Experimental Results

Main experimental comparison. Tab. 2 reports Recall@1 and Recall@5 across all 9 tasks. PRO consistently outperforms GENIUS on every task, improving the macro-averaged R@1 and R@5 by 26.5% and 19.1%. The improvements extend to composed queries that require jointly interpreting visual and textual inputs, where PRO improves the R@5 of GENIUS by 42.5–47.1% on CIRR, OVEN, and InfoSeek. Although dense baselines retain higher absolute performance, PRO consistently narrows the gap, closing over 60% of the R@1 gap on Flickr30k text-to-image, nearly 50% on the two COCO tasks, and 14–35% on the composed-query tasks.

Larger beams and prefix-aware training. Increasing the beam width from 20 to 200 improves both GENIUS and PRO, but GENIUS largely saturates by B=100 (Tab. 3; App. B.3). We also train the GENIUS decoder with PrefixCE, which weights the cross-entropy loss of the k-th semantic code token by $w_k = 2(L - k + 1)/(L + 1)$ to emphasize early identifier tokens. PRO outperforms PrefixCE on all nine tasks at B=50, and on COCO text-to-image, CIRR, OVEN, and InfoSeek, PRO at B=50 also outperforms both GENIUS and PrefixCE at B=200 in R@1 and R@5; on OVEN, for example, quadrupling the beam raises the R@1 of GENIUS by only 0.1 points, whereas PRO adds 8.1 points at the default beam. Hence, the gains of PRO cannot be explained solely by a larger search budget or by emphasizing early identifier tokens.

Table 2: Main results on M-BEIR, retrieved from a task-specific pool. **Bold**: best in each category (dense / generative). A dash (–) indicates that the corresponding baseline does not support the setting.

Method	COCO $t\to i$		Flickr30k $t\to i$		WebQA $t\to t$		WebQA $t\to i,t$		COCO $i\to t$		NIGHTS $i\to i$		CIRR $i,t\to i$		OVEN $i,t\to i,t$		InfoSeek $i,t\to t$	
	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
Dense																		
CLIP-SF	**55.2**	**80.7**	**79.1**	**95.0**	**58.3**	**84.2**	47.6	76.3	65.7	87.5	**8.8**	**31.6**	5.9	43.2	**49.4**	**69.2**	**14.5**	**28.8**
BLIP-FF	53.7	79.6	73.9	92.4	51.9	78.6	**49.5**	**78.1**	**71.1**	**90.9**	8.1	31.0	**24.4**	**50.9**	36.1	56.4	10.5	23.4
Generative																		
IRGen	29.6	50.7	49.0	68.9	–	–	–	–	–	–	–	–	–	–	–	–	–	–
GRACE	16.7	39.5	37.4	59.5	–	–	–	–	–	–	–	–	–	–	–	–	–	–
AVG	31.3	58.0	62.8	85.4	–	–	–	–	–	–	–	–	–	–	–	–	–	–
SemCORE	42.4	57.5	69.0	83.0	–	–	–	–	–	–	–	–	–	–	–	–	–	–
GENIUS	37.1	66.4	56.8	81.3	19.4	29.9	29.1	51.3	47.9	77.3	1.2	11.6	5.5	19.0	26.4	36.0	6.1	10.2
Ours	**45.8**	**72.6**	**70.6**	**87.3**	**25.7**	**40.7**	**35.3**	**57.6**	**59.0**	**84.3**	**3.5**	**19.7**	**8.6**	**27.8**	**34.5**	**51.3**	**7.3**	**15.0**

Table 3: R@1/R@5 of GENIUS with larger beam widths B and of PRO at its default B=50. **Bold**: best result.

Method	COCO $t\to i$		CIRR $i,t\to i$		OVEN $i,t\to i,t$		InfoSeek $i,t\to t$	
	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
GENIUS (B=50)	37.1	66.4	5.5	19.0	26.4	36.0	6.1	10.2
GENIUS (B=100)	37.2	67.1	5.6	19.5	26.6	36.6	6.2	10.6
GENIUS (B=200)	37.2	67.2	5.6	19.6	26.5	36.6	6.2	10.6
PRO (B=50)	**45.8**	**72.6**	**8.6**	**27.8**	**34.5**	**51.3**	**7.3**	**15.0**

Global pool and candidate pool size. On the official M-BEIR global pool, PRO outperforms GENIUS on all eight M-BEIR tasks and raises the macro-averaged R@1/R@5 from 19.4/35.4 to 26.1/42.7 (App. B.4). When the candidate pool grows from 50K to 335K items on OVEN and from 50K to 612K items on InfoSeek, all methods degrade, but PRO outperforms GENIUS at every pool size and consistently narrows the gap to dense retrieval (Fig. 4; App. B.5).

Efficiency and complexity. Following GENIUS, we compare trie-constrained decoding with exact full-corpus dense scanning. As shown in Fig. 7b, PRO maintains nearly constant throughput as the pool grows, whereas exact dense scanning slows steadily, with the curves crossing at around 50K candidates. This claim is limited to exact retrieval: approximate nearest-neighbor indexes such as HNSW [28] accelerate dense retrieval with a recall–latency trade-off, and a matched-quality comparison with them is beyond our scope. The added complexity of PRO is limited: prefix ranking distillation adds about 8% to codebook learning but below 1% to end-to-end training, vocabulary scheduling adds no online computation, and PRO increases latency over GENIUS by only 3.9–4.9% at B=50 (App. B.10), which is small relative to its 26.5%/19.1% macro-averaged R@1/R@5 gains.

Ablation study. To isolate the contribution of each component, we add them cumulatively following the survival bound (see Eq. 4): prefix distillation (reducing ranking divergence, $K_\ell\downarrow$), vocabulary scheduling (increasing the teacher margin, $m_\ell\uparrow$), and score fusion (mitigating decoder mismatch, $\varepsilon_\ell\downarrow$). Tab. 4 reports results on six tasks spanning all three query families; each configuration requires retraining both the tokenizer and the decoder. The full system achieves the best Recall@1 on every task, confirming that the three components are complementary. Two consistent patterns emerge. First, the dominant single-component gain varies by task type, in line with the survival bound. Vocabulary scheduling is most effective when the teacher margin is the primary bottleneck, whereas score fusion yields larger gains when the geometric proxy is more reliable. Second, prefix distillation enhances the effectiveness of vocabulary scheduling. When ranking divergence is large, increasing the margin alone does not reliably preserve the target prefix; once ranking divergence is reduced, vocabulary scheduling becomes consistently beneficial, and the two components exhibit super-additive effects. App. B.6 provides full configurations and pairwise interaction analyses. Additional encoder sensitivity experiments (App. B.11) show that the gap to dense retrieval persists for vanilla GENIUS across three multimodal encoders, while PRO consistently narrows this gap. This suggests that the improvements are not tied to a specific encoder, but arise from mitigating the RQ tokenization and decoding bottleneck.

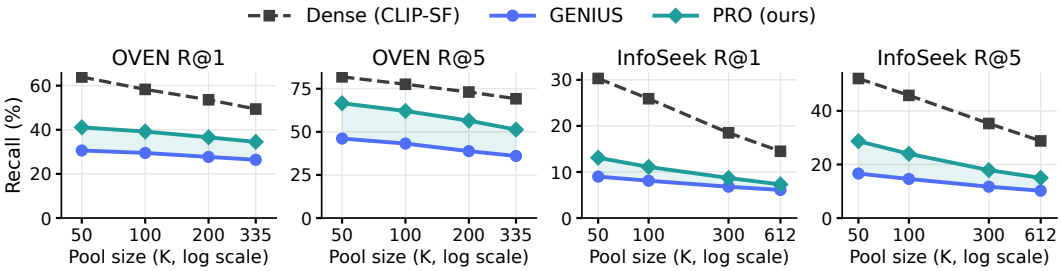
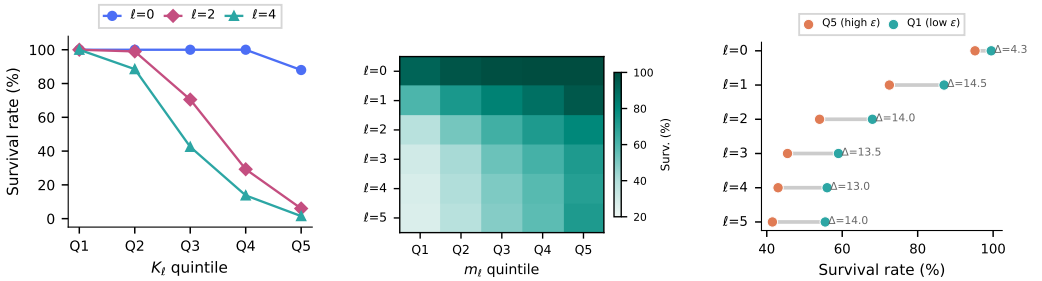


Figure 4: R@1 and R@5 on OVEN and InfoSeek as the candidate pool grows; the largest pool of each task is its full task-specific pool. The shaded area marks the improvement of PRO over GENIUS.



(a) Ranking divergence (b) Teacher margin (c) Decoder mismatch

Figure 5: Survival trends predicted by Thm. 2 on COCO text-to-image (B=20): ranking divergence quintile (Spearman ρ=−0.59), teacher margin quintile, and survival gap between the lowest and highest decoder mismatch quintiles.

Diagnostic analysis. Next, we examine whether the quantities in the survival bound are predictive of prefix survival in practice. Following Thm. 2, we bin 5,000 COCO text-to-image test queries into quintiles of ranking divergence K_ℓ, teacher margin m_ℓ, and decoder mismatch ε_ℓ, and measure cumulative prefix survival at beam width 20. Fig. 5 shows clear monotonic trends for all three quantities. Ranking divergence is the strongest predictor of survival, teacher margin exhibits a consistent positive association, and decoder mismatch shows a consistent negative association across levels. In contrast, the representative reconstruction objective on complete code sequences examined in Sec. 3 shows negligible association with survival, consistent with Thm. 1, which states that reconstruction quality on complete code sequences does not imply prefix survival. Complementarily, Fig. 7a shows that our method improves the per-level survival rate of the target identifier token under the same top-B criterion. In the trained T5 decoder, PRO also raises the rate at which the complete target identifier remains in the beam at B=50 from 56.9% to 66.9% on OVEN and from 13.7% to 23.5% on InfoSeek, while on COCO it is about 90% for both systems (App. B.9). Although the bound derived via Pinsker's inequality is loose, the observed trends align with its predicted dependencies and support the practical relevance of the three quantities. App. B.7 extends this analysis to five datasets across all three query families, confirming that the trends for K_ℓ and m_ℓ generalize, while ε_ℓ remains a broad bottleneck.

Table 4: Ablation study result.

		Base	+PD	+VS	+SF
COCO $t\to i$	R@1	36.8	40.2	42.1	**46.2**
	R@5	66.5	68.2	70.7	**73.0**
COCO $i\to t$	R@1	48.1	51.5	53.6	**59.3**
	R@5	77.2	78.9	81.4	**84.6**
WebQA $t\to t$	R@1	19.5	21.2	21.6	**26.2**
	R@5	30.0	31.7	33.2	**39.7**
WebQA $t\to i,t$	R@1	28.9	31.4	33.5	**35.6**
	R@5	51.4	53.1	55.2	**57.5**
OVEN $i,t\to i$	R@1	5.9	5.6	8.8	**9.1**
	R@5	14.7	14.2	**19.0**	17.6
OVEN $i,t\to i,t$	R@1	26.7	28.4	32.2	**34.4**
	R@5	36.4	39.9	47.5	**51.4**

PD: prefix distill. ($K_\ell\downarrow$), VS: vocab sched. ($m_\ell\uparrow$), SF: score fusion ($\varepsilon_\ell\downarrow$). Bold: best per row.

Generalization Beyond Multimodal Retrieval. Since our analysis and PRO are built on RQ identifiers, we also apply PRO to TIGER [31], a representative RQ-based generative recommendation method, on the Amazon Beauty, Toys, and Sports categories [12]. PRO improves all 12 Recall and NDCG metrics over TIGER (App. B.1), suggesting that our analysis of prefix survival and PRO extend to other RQ-based generative settings.

7 Related Work

Generative retrieval. GR formulates retrieval as sequence generation, where an encoder-decoder model maps queries directly to target identifiers [5, 40]. The paradigm has been studied for document retrieval in DSI and NCI [40, 41], extended to recommendation in TIGER [31], and generalized to multimodal retrieval by IRGen [49], GRACE [24], and GENIUS [17]. ZeroGR [37] extends generative retrieval to zero-shot settings through instruction-driven query generation and semantically meaningful document identifiers. Most related to our work, RIPOR [48] improves scalable generative retrieval through prefix-oriented ranking optimization and relevance-based identifier construction. However, it does not optimize RQ tokenization for target prefix survival under beam search. Our analysis focuses one stage earlier: Thm. 1 shows that the index itself can deterministically prune the target prefix even with a perfect decoder. Accordingly, two of our three mechanisms target tokenizer training. The two approaches address complementary bottlenecks: RIPOR reduces decoder mismatch, while our tokenizer mechanisms reduce ranking divergence and enlarge the teacher margin.

Document identifier design. The identifier is a key factor of GR. Early work studied atomic, semantic, and hierarchical document identifiers in DSI and NCI [40, 41]. Later proposals include multiview identifiers [23], lexical index learning [20], learned tokenization [36], bottleneck-minimal indexing [6], and identifiers with graded relevance [39]. These methods improve identifier semantics, compactness, or relevance alignment. Our work addresses a different axis: even with fixed identifiers, the reconstruction-trained index might fail to preserve the target prefix under beam search (Thm. 1). The mechanisms we propose are therefore complementary to better identifier designs, because any identifier scheme paired with RQ indexing can still face this gap.

Training and inference mismatch. Teacher forcing was introduced for recurrent neural networks [43], and later work studied exposure bias caused by the mismatch between training with teacher forcing and free running decoding [32] and the gap between training objectives and beam search behavior [44]. Inference time score combination, e.g., shallow fusion in neural machine translation, has been used to adapt decoding without retraining [10]. Within GR, reinforcement learning from relevance feedback optimizes retrieval behavior [51], prefix alignment optimization addresses discrepancy between training and inference [47], and RQ refinements improve code utilization and index quality [18]. Autoregressive Ranking [34] introduces SToICaL, a rank-aware training loss that combines item-level reweighting with prefix-tree marginalization to align token probabilities with document relevance. Our setting differs in a key respect: this gap arises from a mismatch between index construction and beam decoding, rather than from train and test discrepancy inside the decoder. The survival bound (Thm. 2) formalizes this coupling by decomposing prefix survival into quantities spanning both stages.

8 Conclusion

This paper identifies and formalizes the indexing–decoding gap in MGR, showing that identifier learning objectives are insufficient to determine prefix survival under beam decoding. We derive a survival bound that relates prefix survival to three controllable factors across indexing and decoding, and propose *PRO*, a unified framework that improves prefix survival via (i) prefix ranking distillation to align prefix rankings, (ii) vocabulary scheduling to reduce early prefix competition, and (iii) geometric score fusion to incorporate query-aware similarity into beam search scoring. Experiments on 9 multimodal retrieval tasks show that PRO consistently outperforms state-of-the-art MGR methods and improves prefix survival, with strong generalizability across multimodal encoders and RQ-based methods.

Limitations. Our analysis relies on the residual, additive structure of RQ and assumes deterministic top-B beam search; similar mismatches may arise for other identifier families and under sampling-based decoding, but our theory does not directly transfer to them (Apps. C.3 and C.4). PRO targets the shared discrete-identifier stage rather than multimodal-specific issues such as the modality gap, strong dense retrievers still achieve higher absolute effectiveness, and our efficiency comparison is limited to exact dense scanning. Finally, our framework assumes a two-stage setup with a fixed encoder and tokenizer, which may limit joint optimization. Future work includes extending the analysis to other identifier families and decoding strategies, and developing end-to-end training with survival-aware objectives.

Acknowledgments and Disclosure of Funding

This research was supported by the Natural Science Foundation of China (662602391), the Natural Science Foundation of Shandong Province (ZR2026QC1067). This research was (partially) supported by the Dutch Research Council (NWO), under project numbers 024.004.022, NWA.1389.20.183, and KICH3.LTP.20.006, and the European Union under grant agreement No. 101201510 (UNITE). Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of their respective employers, funders and/or granting authorities.

References

[1] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: From pairwise approach to listwise approach. In *Proceedings of the International Conference on Machine Learning (ICML)*, ACM International Conference Proceeding Series, pages 129–136. ACM, 2007.

[2] Yingshan Chang, Guihong Cao, Mridu Narang, Jianfeng Gao, Hisami Suzuki, and Yonatan Bisk. WebQA: Multihop and multimodal QA. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16474–16483. IEEE, 2022.

[3] Yang Chen, Hexiang Hu, Yi Luan, Haitian Sun, Soravit Changpinyo, Alan Ritter, and Ming-Wei Chang. Can pre-trained vision and language models answer visual information-seeking questions? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 14948–14968. Association for Computational Linguistics, 2023.

[4] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, 2001.

[5] Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. Autoregressive entity retrieval. In *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2021.

[6] Xin Du, Lixin Xiu, and Kumiko Tanaka-Ishii. Bottleneck-minimal indexing for generative document retrieval. In *Proceedings of the International Conference on Machine Learning (ICML)*, Proceedings of Machine Learning Research, pages 11888–11904. PMLR / OpenReview.net, 2024.

[7] William H. R. Equitz and Thomas M. Cover. Successive refinement of information. *IEEE Transactions on Information Theory*, 37(2):269–275, 1991.

[8] Alexei A. Fedotov, Peter Harremoës, and Flemming Topsøe. Refinements of Pinsker's inequality. *IEEE Transactions on Information Theory*, 49(6):1491–1498, 2003.

[9] Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. DreamSim: Learning new dimensions of human visual similarity using synthetic data. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

[10] Çağlar Gülçehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loïc Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. On using monolingual corpora in neural machine translation. *CoRR*, abs/1503.03535, 2015.

[11] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.

[12] Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian J. McAuley. Bridging language and items for retrieval and recommendation. *CoRR*, abs/2403.03952, 2024.

[13] Hexiang Hu, Yi Luan, Yang Chen, Urvashi Khandelwal, Mandar Joshi, Kenton Lee, Kristina Toutanova, and Ming-Wei Chang. Open-domain visual entity recognition: Towards recognizing millions of Wikipedia entities. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12031–12041. IEEE, 2023.

[14] Weiquan Huang, Aoqi Wu, Yifan Yang, Xufang Luo, Yuqing Yang, Usman Naseem, Chunyu Wang, Qi Dai, Xiyang Dai, Dongdong Chen, Chong Luo, Lili Qiu, and Liang Hu. LLM2CLIP: Powerful language model unlocks richer cross-modality representation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 5131–5139. AAAI Press, 2026.

[15] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128, 2011.

[16] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 18661–18673, 2020.

[17] Sungyeon Kim, Xinliang Zhu, Xiaofan Lin, Muhammet Bastan, Douglas Gray, and Suha Kwak. GENIUS: A generative framework for universal multimodal search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19659–19669. Computer Vision Foundation / IEEE, 2025.

[18] Zhirui Kuai, Zuxu Chen, Huimu Wang, Mingming Li, Dadong Miao, Binbin Wang, Xusong Chen, Li Kuang, Yuxing Han, Jiaxing Wang, Guoyu Tang, Lin Liu, Songlin Wang, and Jingwei Zhuo. Breaking the hourglass phenomenon of residual quantization: Enhancing the upper bound of generative retrieval. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing: Industry Track (EMNLP)*, pages 677–685. Association for Computational Linguistics, 2024.

[19] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11513–11522. IEEE, 2022.

[20] Sunkyung Lee, Minjin Choi, and Jongwuk Lee. GLEN: Generative retrieval via lexical index learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7693–7704. Association for Computational Linguistics, 2023.

[21] Haoxuan Li, Yi Bin, Yunshan Ma, Guoyin Wang, Yang Yang, See-Kiong Ng, and Tat-Seng Chua. SemCORE: A semantic-enhanced generative cross-modal retrieval framework with MLLMs. *CoRR*, abs/2504.13172, 2025.

[22] Xiaoxi Li, Jiajie Jin, Yujia Zhou, Yuyao Zhang, Peitian Zhang, Yutao Zhu, and Zhicheng Dou. From matching to generation: A survey on generative information retrieval. *ACM Transactions on Information Systems*, 43(3):83:1–83:62, 2025.

[23] Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. MINDER: Multiview identifiers enhanced generative retrieval. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6636–6648. Association for Computational Linguistics, 2023.

[24] Yongqi Li, Wenjie Wang, Leigang Qu, Liqiang Nie, Wenjie Li, and Tat-Seng Chua. Generative cross-modal retrieval: Memorizing images in multimodal language models for retrieval and beyond. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 11851–11861. Association for Computational Linguistics, 2024.

[25] Yongqi Li, Hongru Cai, Wenjie Wang, Leigang Qu, Yinwei Wei, Wenjie Li, Liqiang Nie, and Tat-Seng Chua. Revolutionizing text-to-image retrieval as autoregressive token-to-voken generation. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 813–822. ACM, 2025.

[26] 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 *European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science, pages 740–755. Springer, 2014.

[27] Zheyuan Liu, Cristian Rodriguez Opazo, Damien Teney, and Stephen Gould. Image retrieval on real-life images with pre-trained vision-and-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2105–2114. IEEE, 2021.

[28] Yury A. Malkov and Dmitry A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836, 2020.

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, Proceedings of Machine Learning Research, pages 8748–8763. PMLR, 2021.

[30] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:140:1–140:67, 2020.

[31] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Mahesh Sathiamoorthy. Recommender systems with generative retrieval. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

[32] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *International Conference on Learning Representations (ICLR)*, 2016.

[33] Olivier Rioul. A historical perspective on Schützenberger-Pinsker inequalities. In *Geometric Science of Information (GSI)*, Lecture Notes in Computer Science, pages 291–306. Springer, 2023.

[34] Benjamin Rozonoyer, Chong You, Michael Boratko, Himanshu Jain, Nilesh Gupta, Srinadh Bhojanapalli, Andrew McCallum, and Felix Yu. Autoregressive ranking: Bridging the gap between dual and cross encoders. *CoRR*, abs/2601.05588, 2026.

[35] Igal Sason and Sergio Verdú. f-divergence inequalities. *IEEE Transactions on Information Theory*, 62(11):5973–6006, 2016.

[36] Weiwei Sun, Lingyong Yan, Zheng Chen, Shuaiqiang Wang, Haichao Zhu, Pengjie Ren, Zhumin Chen, Dawei Yin, Maarten de Rijke, and Zhaochun Ren. Learning to tokenize for generative retrieval. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

[37] Weiwei Sun, Keyi Kong, Xinyu Ma, Shuaiqiang Wang, Dawei Yin, Maarten de Rijke, Zhaochun Ren, and Yiming Yang. ZeroGR: A generalizable and scalable framework for zero-shot generative retrieval. In *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2026.

[38] Yubao Tang, Ruqing Zhang, Jiafeng Guo, and Maarten de Rijke. Recent advances in generative information retrieval. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region (SIGIR-AP)*, pages 294–297. ACM, 2023.

[39] Yubao Tang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Wei Chen, and Xueqi Cheng. Generative retrieval meets multi-graded relevance. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2024.

[40] Yi Tay, Vinh Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Prakash Gupta, Tal Schuster, William W. Cohen, and Donald Metzler. Transformer memory as a differentiable search index. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022.

[41] Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao, Shibin Wu, Qi Chen, Yuqing Xia, Chengmin Chi, Guoshuai Zhao, Zheng Liu, Xing Xie, Hao Sun, Weiwei Deng, Qi Zhang, and Mao Yang. A neural corpus indexer for document retrieval. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022.

[42] Cong Wei, Yang Chen, Haonan Chen, Hexiang Hu, Ge Zhang, Jie Fu, Alan Ritter, and Wenhu Chen. UniIR: Training and benchmarking universal multimodal information retrievers. In *European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science, pages 387–404. Springer, 2024.

[43] Ronald J. Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280, 1989.

[44] Sam Wiseman and Alexander M. Rush. Sequence-to-sequence learning as beam-search optimization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1296–1306. Association for Computational Linguistics, 2016.

[45] Shiguang Wu, Wenda Wei, Mengqi Zhang, Zhumin Chen, Jun Ma, Zhaochun Ren, Maarten de Rijke, and Pengjie Ren. Generative retrieval as multi-vector dense retrieval. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 1828–1838. ACM, 2024.

[46] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.

[47] Yuanqing Yu, Yifan Wang, Weizhi Ma, Zhiqiang Guo, and Min Zhang. APAO: Adaptive prefix-aware optimization for generative recommendation. *CoRR*, abs/2603.02730, 2026.

[48] Hansi Zeng, Chen Luo, Bowen Jin, Sheikh Muhammad Sarwar, Tianxin Wei, and Hamed Zamani. Scalable and effective generative information retrieval. In *Proceedings of the ACM Web Conference (WWW)*, pages 1441–1452. ACM, 2024.

[49] Yidan Zhang, Ting Zhang, Dong Chen, Yujing Wang, Qi Chen, Xing Xie, Hao Sun, Weiwei Deng, Qi Zhang, Fan Yang, Mao Yang, Qingmin Liao, Jingdong Wang, and Baining Guo. IRGen: Generative modeling for image retrieval. In *European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science, pages 21–41. Springer, 2024.

[50] Yingchen Zhang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. Does generative retrieval overcome the limitations of dense retrieval? *CoRR*, abs/2509.22116, 2025.

[51] Yujia Zhou, Zhicheng Dou, and Ji-Rong Wen. Enhancing generative retrieval with reinforcement learning from relevance feedback. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 12481–12490. Association for Computational Linguistics, 2023.

Supplementary Material

Our supplementary material has the following sections:

- Proof details
- Additional experimental results
- Further discussion
- Implementation details
- NeurIPS paper checklist

A Proof Details

We collect full proofs of the theoretical results stated in the main text. All notation follows Secs. 3–4; we briefly recall the key symbols for convenience.

Notation. We use the following groups of symbols throughout the proofs.

- **Embeddings and codes.** $x_j \in \mathbb{R}^d$ is the dense embedding of target item j. $\hat{x}_{j,\ell} := \sum_{t=1}^{\ell} c_{t,z_j^t}$ is its partial reconstruction after ℓ RQ levels, and $\hat{x}_j := \hat{x}_{j,L}$ is its full reconstruction. $z_j^{1:\ell}$ denotes the length-ℓ code prefix.
- **Beam event.** p_ℓ^+ is the ground-truth prefix at level ℓ, and S_ℓ is the event that p_ℓ^+ survives in the beam.
- **Prefix distributions.** π_ℓ^* scores prefixes using pre-quantization embedding similarities; $\bar{\pi}_\ell$ is the quantized oracle distribution; and $\tilde{\pi}_\ell$ is the decoder prefix distribution. These objects are probability mass functions (PMFs) over the finite prefix set at level ℓ. A lower-case symbol such as p denotes a prefix value; when a random prefix is needed, we use an upper-case symbol and write, for example, $P \sim \pi_\ell^*(\cdot)$. Here, "teacher" refers to π_ℓ^*.
- **Survival factors.** $K_\ell := D_{\mathrm{KL}}(\pi_\ell^* \| \bar{\pi}_\ell)$, $m_\ell := \pi_\ell^*(p_\ell^+) - \pi_{\ell,B+1}^*$, and $\varepsilon_\ell := \mathrm{TV}(\bar{\pi}_\ell, \tilde{\pi}_\ell)$. The KL and TV terms are taken between PMFs over the same prefix set. Here, $\pi_{\ell,B+1}^*$ denotes the probability of the $(B{+}1)$-th ranked prefix under the teacher.

For completeness, for each prefix value p, the 3 prefix PMFs are

$$\pi_\ell^*(p) \propto \exp\big(s_\ell^*(p)/\tau\big), \qquad \bar{\pi}_\ell(p) \propto \exp\big(\bar{s}_\ell(p)/\tau\big), \qquad \tilde{\pi}_\ell(p) \propto \exp\big(\tilde{s}_\ell(p)/\tau\big),$$

where

$$s_\ell^*(p) = \max_{j:\, z_j^{1:\ell}=p} \langle q, x_j \rangle$$

is the frozen-encoder similarity score,

$$\bar{s}_\ell(p) = \langle q, \hat{x}_{j,\ell} \rangle$$

is the partial-reconstruction score shared by all target items with prefix p, and $\tilde{s}_\ell(p)$ denotes the actual decoder score.

A.1 Proof of Theorem 1

Proof idea. The theorem separates existing tokenizer objectives from prefix survival during beam search. We construct two RQ tokenizers over the same query and item embeddings. They select the same code indices and have the same complete query and item reconstructions; their residual-fitting costs also match by construction. Therefore, the contrastive, residual-fitting, and reconstruction objectives considered in the theorem take identical values. The difference lies only in how the first two selected codewords decompose the same complete reconstruction. In $\mathcal{Q}^+$, the target prefix has the highest quantized-oracle score at every level. In $\mathcal{Q}^-$, the first selected target codeword is shifted away from the query direction and the second selected target codeword is shifted back, leaving the complete reconstruction and objective values unchanged. This shift lowers the target first-level prefix score below B competing prefixes, so beam search prunes it. Thus, existing tokenizer objectives do not by themselves determine prefix survival.

Construction. Let $K := B + 1$. We construct the ambient space $\mathbb{R}^d$ with $d = 2K + L + 1$ as a direct sum of mutually orthogonal subspaces:

$$\mathbb{R}^d \;=\; S \,\oplus\, U \,\oplus\, A_1 \,\oplus\, A_2 \,\oplus\, A_3 \,\oplus\, \cdots \,\oplus\, A_L \,\oplus\, R,$$

where $\dim S = \dim U = 1$, $\dim A_1 = \dim A_2 = K$, $\dim A_t = 1$ for $t = 3, \ldots, L$, and $\dim R = 1$.

Let $s \in S$ and $u \in U$ be unit vectors, and fix a large constant $H > 0$. The query embedding is $q := s + Hu$. In A_1, fix K orthonormal vectors $a_1^{(1)}, \ldots, a_K^{(1)}$; in A_2, fix K orthonormal vectors $a_1^{(2)}, \ldots, a_K^{(2)}$. For $t \geq 3$, let $a_\star^{(t)}$ be a unit vector spanning A_t. Let r be a unit vector spanning R. Fix a constant $M > 2$.

Codebooks for $\mathcal{Q}^+$. Level 1 (K+1 codewords): $c_{1,0}^+ = q$; $c_{1,1}^+ = s + Ma_1^{(1)}$; $c_{1,j}^+ = Ma_j^{(1)}$ for $j = 2, \ldots, K$. Level 2 (K+1 codewords): $c_{2,0}^+ = 0$; $c_{2,j}^+ = Ma_j^{(2)}$ for $j = 1, \ldots, K$. Level $t \geq 3$ (two codewords each): $c_{t,0}^+ = 0$; $c_{t,\star}^+ = Ma_\star^{(t)}$. The zero codewords are selected only by the query. Unused dummy codewords can be placed sufficiently far away without changing any assignment, objective value, or beam-search outcome below. Thus the construction only requires finite vocabularies with $V_1, V_2 \geq K + 1$ and $V_t \geq 2$ for $t \geq 3$.

Items. $K = B + 1$ target items:

$$x_j \;=\; c_{1,j}^+ + c_{2,j}^+ + \sum_{t=3}^{L} c_{t,\star}^+ + \alpha\, r, \qquad j = 1, \ldots, K. \tag{8}$$

Target item $j = 1$ is the target.

Lemma A.1 (Greedy assignment and reconstruction optimality under $\mathcal{Q}^+$)**.** *Under $\mathcal{Q}^+$, the greedy nearest-neighbor RQ assigns the query the code sequence $(0, 0, \ldots, 0)$ and target item x_j the code sequence $(j,\, j,\, \star, \ldots, \star)$. The resulting target reconstruction error is α^2 for every item, which is the global minimum over all code assignments for the constructed codebooks.*

Proof. Query. The query q exactly matches $c_{1,0}^+$ at level 1 and has zero residual afterwards, so it selects $(0, 0, \ldots, 0)$.

Level 1. The distance from x_j to the correct codeword $c_{1,j}^+$ is $\|x_j - c_{1,j}^+\|^2 = (L-1)M^2 + \alpha^2$ (only $A_2, \ldots, A_L, R$ components remain). For any incorrect token $k \neq j$, the A_1-component mismatch contributes $\|Ma_j^{(1)} - Ma_k^{(1)}\|^2 = 2M^2$ by orthonormality, so $\|x_j - c_{1,k}^+\|^2 \geq 2M^2 + (L-1)M^2 + \alpha^2$. The query-only token $c_{1,0}^+ = q$ incurs an additional H^2 and is therefore not selected by any target item. The gap is at least $2M^2 > 0$; greedy selects token j.

Level 2. After subtracting $c_{1,j}^+$, the residual has zero A_1-component. By the same orthonormality argument, the correct token j achieves distance $(L-2)M^2 + \alpha^2$. Any incorrect item token incurs an additional $2M^2$, and the query-only zero token leaves an additional M^2 component. Greedy selects token j.

Level $t \geq 3$. The item codeword $c_{t,\star}^+$ removes the remaining A_t component, whereas the query-only zero token does not. Greedy selects $c_{t,\star}^+$.

Full reconstruction and MSE. $\hat{q} = q$ and $\hat{x}_j = c_{1,j}^+ + c_{2,j}^+ + \sum_{t \geq 3} c_{t,\star}^+ = x_j - \alpha\, r$, so the target reconstruction error is α^2.

Global optimality. Every codeword lies in $S \oplus U \oplus A_1 \oplus \cdots \oplus A_L$, so any reconstruction has zero component in R. Since each x_j has a nonzero R-component αr, the Pythagorean theorem gives $\|x_j - \tilde{x}_j\|^2 \geq \alpha^2$ for any code assignment. Equality is achieved above.

Lemma A.2 (Target survival under trie-constrained quantized oracle beam search in $\mathcal{Q}^+$)**.** *Under trie-constrained beam search over the valid item identifiers with quantized oracle scores, the target prefix survives at every level $\ell = 1, \ldots, L$.*

Proof. For the target ($j = 1$): level 1 contributes $c_{1,1}^+ = s + Ma_1^{(1)}$, which has $\langle q, c_{1,1}^+ \rangle = 1$. Every subsequent level adds a codeword orthogonal to q, so $\langle q, \hat{x}_{1,\ell}^+ \rangle = 1$ for all ℓ. For any competitor

($j \geq 2$): $c^+_{1,j} = Ma^{(1)}_j \in A_1$, so $\langle q, \hat{x}^+_{j,\ell}\rangle = 0$ for all ℓ. The target has the unique highest score at every level.

Construction of $\mathcal{Q}^-$. Let $\delta := M - \sqrt{M^2-4}$, so that $4 + (M-\delta)^2 = M^2$. Modify exactly two codewords: $c^-_{1,1} = -s + Ma^{(1)}_1 + \delta a^{(2)}_1$, $c^-_{2,1} = 2s + (M-\delta)a^{(2)}_1$. All other codewords are unchanged.

Proof of Thm. 1. We verify three claims.

Claim 1: Greedy assignment remains valid under $\mathcal{Q}^-$. For the target at level 1, the distance to $c^-_{1,1}$ is $4 + (M-\delta)^2 + (L-2)M^2 + \alpha^2 = (L-1)M^2 + \alpha^2$, while any other token $k \geq 2$ gives $1 + (L+1)M^2 + \alpha^2$. The query-only token $c^-_{1,0} = q$ incurs an additional H^2 and is not selected by any target item. Token 1 remains nearest. At level 2, the residual after $c^-_{1,1}$ is $2s + (M-\delta)a^{(2)}_1 + \sum_{t\geq 3} c^+_{t,\star} + \alpha\, r$; the distance to $c^-_{2,1} = 2s + (M-\delta)a^{(2)}_1$ is $(L-2)M^2 + \alpha^2$. Any incorrect item token incurs an additional $4 + (M-\delta)^2 + M^2$, and the query-only zero token leaves an additional $4 + (M-\delta)^2 = M^2$. For non-target items ($j \geq 2$), the correct distances are unchanged. The modified level-1 token has distance $1 + (L+1)M^2 + \delta^2 + \alpha^2$, and the modified level-2 token has an extra $4 + M^2 + (M-\delta)^2$ over the correct choice. Levels $t \geq 3$ are unchanged. The query still selects $(0, 0, \ldots, 0)$ because the query-only codewords are unchanged. Therefore $\mathcal{Q}^-$ assigns the same code sequences as $\mathcal{Q}^+$.

Claim 2: Identical standard tokenizer objective values. Since code assignments are unchanged, the full reconstruction of the target is $\hat{x}^-_1 = c^-_{1,1} + c^-_{2,1} + \sum_{t\geq 3} c^-_{t,\star} = (-s + Ma^{(1)}_1 + \delta a^{(2)}_1) + (2s + (M-\delta)a^{(2)}_1) + \cdots = (s + Ma^{(1)}_1) + Ma^{(2)}_1 + \cdots = \hat{x}^+_1$. For $j \geq 2$, all codewords are unchanged, so $\hat{x}^-_j = \hat{x}^+_j$. The query reconstruction is also identical because $c^-_{1,0} = c^+_{1,0} = q$ and $c^-_{\ell,0} = c^+_{\ell,0} = 0$ for $\ell \geq 2$. Both quantizers therefore produce identical complete query and item reconstructions. Consequently, objectives evaluated on complete reconstructions take identical values under $\mathcal{Q}^+$ and $\mathcal{Q}^-$, including target reconstruction losses and quantized-space MSE terms such as $\|\hat{q} - \hat{x}_j\|^2$. The target reconstruction error is α^2 under both. The squared residual-fitting term $\sum_\ell \|r_{\ell-1} - c_{\ell,z^\ell}\|^2$ also matches. For the query, the fitting cost is zero under both quantizers. For the target, the level-1 fitting cost under $\mathcal{Q}^-$ is $4 + (M-\delta)^2 + (L-2)M^2 + \alpha^2 = (L-1)M^2 + \alpha^2$, identical to $\mathcal{Q}^+$; the level-2 cost is $(L-2)M^2 + \alpha^2$ under both, and later levels are unchanged. All non-target fitting costs are unchanged because their selected codewords are unchanged. Finally, any contrastive term computed on the pre-quantization embeddings is identical because the query and item embeddings are shared. The Pythagorean lower bound from Lem. A.1 applies to $\mathcal{Q}^-$ as well (all codewords still lie in $S \oplus U \oplus A_1 \oplus \cdots \oplus A_L$), so α^2 remains the global target-reconstruction optimum.

Claim 3: Target is pruned under trie-constrained quantized oracle beam search in $\mathcal{Q}^-$. The level-1 prefix scores are $\langle q, c^-_{1,1}\rangle = -1$ and $\langle q, c^-_{1,j}\rangle = 0$ for $j \geq 2$. The target is strictly last; with beam width $B = K - 1$, it is pruned deterministically.

Impossibility. The target survives under $\mathcal{Q}^+$ (Lem. A.2) and is pruned under $\mathcal{Q}^-$ (Claim 3), yet the two quantizers have identical values for the standard tokenizer-training terms covered in Claim 2. Let $\Phi(\mathcal{Q})$ collect these objective values. Then $\Phi(\mathcal{Q}^+) = \Phi(\mathcal{Q}^-)$, but the target prefix survives under $\mathcal{Q}^+$ and is pruned under $\mathcal{Q}^-$. Thus no criterion depending only on Φ can determine the beam-search outcome, even under trie-constrained quantized oracle beam search.

A.2 Preparatory tools for the survival bound

We recall the definitions and standard results used in the proof of Thm. 2.

Definition A.3 (Total variation distance)**.** *For probability distributions P, Q over a finite set $\mathcal{X}$, the* total variation distance *is*

$$\mathrm{TV}(P, Q) \;:=\; \frac{1}{2}\sum_{x\in\mathcal{X}} |P(x) - Q(x)| \;=\; \max_{A\subseteq\mathcal{X}} |P(A) - Q(A)|. \tag{9}$$

The two expressions are equal by a standard identity.

Lemma A.4 (Pointwise bound)**.** *For any single element $x \in \mathcal{X}$,*

$$|P(x) - Q(x)| \;\le\; \mathrm{TV}(P, Q). \tag{10}$$

Proof. By the second expression in Eq. 9, $\mathrm{TV}(P, Q) = \max_A |P(A) - Q(A)| \ge |P(\{x\}) - Q(\{x\})|$.

Lemma A.5 (Triangle inequality)**.** *For distributions P, Q, R over the same finite set,*

$$\mathrm{TV}(P, R) \;\le\; \mathrm{TV}(P, Q) + \mathrm{TV}(Q, R). \tag{11}$$

Proof. For each x, $|P(x) - R(x)| \le |P(x) - Q(x)| + |Q(x) - R(x)|$. Summing over x and dividing by 2 yields the result.

Lemma A.6 (Pinsker's inequality [4])**.** *For distributions P, Q with P absolutely continuous with respect to Q,*

$$\mathrm{TV}(P, Q) \;\le\; \sqrt{\frac{1}{2} D_{\mathrm{KL}}(P \| Q)}. \tag{12}$$

A.3 Proof of Theorem 2 (Survival bound)

We first establish a per-level result for the quantized oracle alone (Lem. A.7), then extend it to the actual decoder prefix distribution via the triangle inequality to obtain Thm. 2.

Lemma A.7 (Per-level KL controls top-B under quantized oracle)**.** *Fix a query q and a level ℓ. If p_ℓ^+ is in the teacher's top-B under π_ℓ^* (i.e., $m_\ell > 0$) and*

$$K_\ell \;=\; D_{\mathrm{KL}}(\pi_\ell^* \,\|\, \bar{\pi}_\ell) \;<\; \frac{m_\ell^2}{2}, \tag{13}$$

then p_ℓ^+ is also in the top-B under the quantized oracle $\bar{\pi}_\ell$.

Proof. Step 1: assume $\mathrm{TV}(\pi_\ell^*, \bar{\pi}_\ell) < m_\ell/2$ and show ranking preservation.

Let $\delta := \mathrm{TV}(\pi_\ell^*, \bar{\pi}_\ell)$ and suppose $\delta < m_\ell/2$. By Lem. A.4, the probability of each individual prefix can change by at most δ when moving from π_ℓ^* to $\bar{\pi}_\ell$.

The ground-truth prefix can *drop* by at most δ:

$$\bar{\pi}_\ell(p_\ell^+) \;\ge\; \pi_\ell^*(p_\ell^+) - \delta. \tag{14}$$

Any prefix p with teacher rank under π_ℓ^* greater than B (i.e., $\pi_\ell^*(p) \le \pi_{\ell,B+1}^*$) can *rise* by at most δ:

$$\bar{\pi}_\ell(p) \;\le\; \pi_\ell^*(p) + \delta \;\le\; \pi_{\ell,B+1}^* + \delta. \tag{15}$$

Subtracting Eq. 15 from Eq. 14:

$$\begin{aligned} \bar{\pi}_\ell(p_\ell^+) - \bar{\pi}_\ell(p) \;&\ge\; \big[\pi_\ell^*(p_\ell^+) - \delta\big] - \big[\pi_{\ell,B+1}^* + \delta\big] \\ &=\; \underbrace{\pi_\ell^*(p_\ell^+) - \pi_{\ell,B+1}^*}_{=\, m_\ell} - 2\delta \\ &=\; m_\ell - 2\delta \;>\; 0, \end{aligned} \tag{16}$$

where the strict inequality uses $\delta < m_\ell/2$. Since $\bar{\pi}_\ell(p_\ell^+)$ is strictly greater than the quantized oracle probability of *every* prefix whose teacher rank exceeds B, and at most $B - 1$ other prefixes can have teacher rank $\le B$, we conclude $p_\ell^+ \in \text{Top-B}(\bar{\pi}_\ell)$.

Step 2: use Pinsker's inequality to convert the KL condition to a TV condition.

By Pinsker's inequality (Lem. A.6):

$$\delta = \mathrm{TV}(\pi_\ell^*, \bar{\pi}_\ell) \le \sqrt{\frac{K_\ell}{2}}.$$

For the condition in Step 1 ($\delta < m_\ell/2$) to hold, it suffices that

$$\sqrt{\frac{K_\ell}{2}} < \frac{m_\ell}{2}.$$

Squaring both sides (both are non-negative):

$$\frac{K_\ell}{2} < \frac{m_\ell^2}{4}, \qquad \text{i.e.,} \qquad K_\ell < \frac{m_\ell^2}{2}.$$

This is precisely the hypothesis in Eq. 13.

We now extend Lem. A.7 from the quantized oracle to the actual decoder prefix distribution, arriving at Thm. 2.

Proof of Thm. 2. Fix a query q for which the condition $\sqrt{K_\ell/2} + \varepsilon_\ell < m_\ell/2$ holds at every level ℓ. We prove by induction that p_ℓ^+ remains in the beam at every level.

Per-level argument. Fix any level ℓ and assume $S_{\ell-1}$ holds (the inductive hypothesis; for $\ell = 1$, there is no previous pruning step). We need to show $p_\ell^+ \in \text{Top-B}(\tilde{\pi}_\ell)$ over the candidate set at level ℓ.

By the triangle inequality for TV (Lem. A.5):

$$\text{TV}(\pi_\ell^*, \tilde{\pi}_\ell) \;\le\; \underbrace{\text{TV}(\pi_\ell^*, \bar{\pi}_\ell)}_{\text{quantization shift}} + \underbrace{\text{TV}(\bar{\pi}_\ell, \tilde{\pi}_\ell)}_{\text{decoder mismatch}}. \tag{17}$$

Applying Pinsker (Lem. A.6) to the first term and using the definition $\varepsilon_\ell := \text{TV}(\bar{\pi}_\ell, \tilde{\pi}_\ell)$ for the second:

$$\text{TV}(\pi_\ell^*, \tilde{\pi}_\ell) \;\le\; \sqrt{\frac{K_\ell}{2}} \;+\; \varepsilon_\ell. \tag{18}$$

By the hypothesis of the theorem, the right-hand side is strictly less than $m_\ell/2$. Denoting the left-hand side by $\delta' := \text{TV}(\pi_\ell^*, \tilde{\pi}_\ell)$, we have $\delta' < m_\ell/2$.

The remainder of the argument is identical to Step 1 of Lem. A.7, but with $\tilde{\pi}_\ell$ in place of $\bar{\pi}_\ell$: by Lem. A.4, the ground-truth prefix can drop by at most δ', and any competitor ranked below B by the teacher can rise by at most δ'. Subtracting (as in Eq. 16):

$$\tilde{\pi}_\ell(p_\ell^+) - \tilde{\pi}_\ell(p) \;\ge\; m_\ell - 2\delta' \;>\; 0.$$

Therefore $p_\ell^+ \in \text{Top-B}(\tilde{\pi}_\ell)$ over the full prefix set $\mathcal{P}_\ell$.

Subset argument. Actual beam search at level ℓ considers only children of the B surviving prefixes from level $\ell-1$, i.e., a subset $\mathcal{P}'_\ell \subseteq \mathcal{P}_\ell$. Since $S_{\ell-1}$ holds by the inductive hypothesis, $p_{\ell-1}^+$ is in the beam, so p_ℓ^+ is indeed a candidate in $\mathcal{P}'_\ell$. If p_ℓ^+ is in the top-B of the full set $\mathcal{P}_\ell$, it is *a fortiori* in the top-B of any subset containing it, because removing competitors can only improve its rank.

Applying the per-level argument at $\ell = 1, 2, \ldots, L$ completes the induction, establishing S_L.

A.4 Score fusion and decoder mismatch

We formalize when geometric score fusion can reduce the decoder mismatch ε_ℓ. Fix a query q, a level ℓ, and a surviving parent prefix $p = z^{1:\ell-1}$. Let $\hat{x}_p$ denote the partial reconstruction represented by this parent prefix, and define the decoding residual $r_p := q - \hat{x}_p$. This residual is the query-to-prefix residual used at decoding time, not the RQ assignment residual in Sec. 2. Let $\mathcal{A}_\ell(p)$ denote the set of legal child tokens. Each legal child token $v \in \mathcal{A}_\ell(p)$ corresponds to a codeword $c_{\ell,v}$ and produces the candidate partial reconstruction $\hat{x}_p + c_{\ell,v}$. All conditional distributions below are PMFs over the finite token set $\mathcal{A}_\ell(p)$; lower-case symbols such as u and v denote token values. When an expectation or covariance is used, the associated upper-case symbol denotes a random token drawn from the stated PMF. The conditional quantized oracle assigns higher probability to legal tokens whose candidate reconstructions have larger similarity to the query, while the conditional decoder distribution is produced by the autoregressive decoder over the same legal tokens. Decoder mismatch measures the distance between these two conditional distributions. Score fusion adds a

codebook-derived bias to the decoder log-probability so that candidate tokens reducing the query residual receive higher beam scores.

The conditional quantized oracle over $\mathcal{A}_\ell(p)$ is

$$\bar{\pi}_\ell(v \mid p, q) = \frac{\exp\big(q^\top c_{\ell,v}/\tau\big)}{\sum_{u \in \mathcal{A}_\ell(p)} \exp\big(q^\top c_{\ell,u}/\tau\big)}.$$

The parent term $q^\top \hat{x}_p$ is omitted because it is constant across all children of the same parent prefix. The conditional decoder distribution is $\tilde{\pi}_\ell(v \mid p, q)$, and the geometric score used by score fusion is

$$g_\ell(v) = 2r_p^\top c_{\ell,v} - c_{\ell,v}^\top c_{\ell,v}.$$

Equivalently,

$$g_\ell(v) = \|r_p\|^2 - \|r_p - c_{\ell,v}\|^2,$$

so $g_\ell(v)$ is the reduction in squared distance from the query to the partial reconstruction after appending token v.

We now rewrite $g_\ell(v)$ in terms of the conditional quantized oracle. Let

$$Z_\ell(p, q) = \sum_{u \in \mathcal{A}_\ell(p)} \exp\big(q^\top c_{\ell,u}/\tau\big).$$

Then

$$q^\top c_{\ell,v} = \tau \log \bar{\pi}_\ell(v \mid p, q) + \tau \log Z_\ell(p, q).$$

Substituting $r_p = q - \hat{x}_p$ into $g_\ell(v)$ gives

$$\begin{aligned} g_\ell(v) &= 2(q - \hat{x}_p)^\top c_{\ell,v} - c_{\ell,v}^\top c_{\ell,v} \\ &= 2q^\top c_{\ell,v} - \big(2\hat{x}_p^\top c_{\ell,v} + c_{\ell,v}^\top c_{\ell,v}\big) \\ &= 2\tau \log \bar{\pi}_\ell(v \mid p, q) + \kappa_\ell(p, q) - \eta_\ell(v), \end{aligned} \tag{19}$$

where

$$\kappa_\ell(p, q) := 2\tau \log Z_\ell(p, q), \qquad \eta_\ell(v) := 2\hat{x}_p^\top c_{\ell,v} + c_{\ell,v}^\top c_{\ell,v}.$$

The term $\kappa_\ell(p, q)$ is constant over child tokens of the same parent prefix, whereas $\eta_\ell(v)$ is token-dependent. Consequently, the fused distribution takes the form

$$\pi_\ell^\omega(v \mid p, q) \;\propto\; \tilde{\pi}_\ell(v \mid p, q) \;\cdot\; \bar{\pi}_\ell(v \mid p, q)^{2\tau\omega} \;\cdot\; e^{-\omega\, \eta_\ell(v)}. \tag{20}$$

Proposition A.8 (Geometric expert approximates quantized oracle)**.** *Define the geometric expert distribution*

$$\rho_\ell(v \mid p, q) \;:=\; \frac{\exp\big(g_\ell(v)/(2\tau)\big)}{\sum_{u \in \mathcal{A}_\ell(p)} \exp\big(g_\ell(u)/(2\tau)\big)}.$$

Then

$$\rho_\ell(v \mid p, q) \;=\; \frac{\bar{\pi}_\ell(v \mid p, q)\, \exp\big(-\eta_\ell(v)/(2\tau)\big)}{\sum_{u \in \mathcal{A}_\ell(p)} \bar{\pi}_\ell(u \mid p, q)\, \exp\big(-\eta_\ell(u)/(2\tau)\big)},$$

where $\eta_\ell(v) := 2\hat{x}_p^\top c_{\ell,v} + c_{\ell,v}^\top c_{\ell,v}$. If the distortion oscillation satisfies $\mathrm{osc}(\eta_\ell) := \max_v \eta_\ell(v) - \min_v \eta_\ell(v) \le \delta_\ell$, where this oscillation measures the token-dependent deviation between the geometric score and the conditional-oracle log-probability after removing the parent-wise constant, then

$$D_{\mathrm{KL}}\big(\bar{\pi}_\ell(\cdot \mid p, q) \,\big\|\, \rho_\ell(\cdot \mid p, q)\big) \;\le\; \frac{\delta_\ell}{2\tau}, \qquad \mathrm{TV}\big(\bar{\pi}_\ell(\cdot \mid p, q),\, \rho_\ell(\cdot \mid p, q)\big) \;\le\; \sqrt{\frac{\delta_\ell}{4\tau}}. \tag{21}$$

Proof. The identity for ρ_ℓ follows directly from $g_\ell(v) = 2\tau \log \bar{\pi}_\ell(v \mid p, q) + \kappa_\ell(p, q) - \eta_\ell(v)$ (Eq. 19): dividing by 2τ and exponentiating cancels $\kappa_\ell/(2\tau)$ in the normalization.

For the KL bound, write $\bar{\eta} := \mathbb{E}_{V \sim \bar{\pi}_\ell(\cdot \mid p, q)}[\eta_\ell(V)]$ and $\Delta_v := \eta_\ell(v) - \bar{\eta}$ for each $v \in \mathcal{A}_\ell(p)$. Then

$$D_{\mathrm{KL}}\big(\bar{\pi}_\ell(\cdot \mid p, q) \,\big\|\, \rho_\ell(\cdot \mid p, q)\big) \;=\; \log \mathbb{E}_{V \sim \bar{\pi}_\ell(\cdot \mid p, q)}\Big[e^{-\Delta_V/(2\tau)}\Big].$$

Since $\max_v(-\Delta_v) \le \mathrm{osc}(\eta_\ell) \le \delta_\ell$, every term in the expectation is at most $e^{\delta_\ell/(2\tau)}$, giving $D_{\mathrm{KL}}(\bar{\pi}_\ell(\cdot \mid p, q) \| \rho_\ell(\cdot \mid p, q)) \le \delta_\ell/(2\tau)$. The TV bound follows from Pinsker's inequality (Lem. A.6).

Prop. A.8 explains why the geometric score is aligned with the conditional quantized oracle when the token-dependent distortion is small. The next result states when adding this score to the decoder reduces the local decoder mismatch.

Proposition A.9 (Local reduction of decoder mismatch)**.** *Define the fused family*

$$\pi_\ell^\omega(v \mid p,q) \;:=\; \frac{\tilde\pi_\ell(v \mid p,q)\,\exp\big(\omega\, g_\ell(v)\big)}{\sum_{u\in\mathcal{A}_\ell(p)} \tilde\pi_\ell(u \mid p,q)\,\exp\big(\omega\, g_\ell(u)\big)}, \qquad \omega \ge 0.$$

Assume there is no tie between the conditional oracle and decoder probabilities, i.e.,

$$\bar\pi_\ell(v \mid p,q) \neq \tilde\pi_\ell(v \mid p,q) \qquad \textit{for every } v \in \mathcal{A}_\ell(p),$$

and let

$$S_\ell^+ \;:=\; \big\{v \in \mathcal{A}_\ell(p) :\; \bar\pi_\ell(v \mid p,q) > \tilde\pi_\ell(v \mid p,q)\big\}.$$

Let $V \sim \tilde\pi_\ell(\cdot \mid p,q)$ be a random legal token drawn from the decoder distribution. Then

$$\frac{d}{d\omega}\,\mathrm{TV}\big(\bar\pi_\ell(\cdot \mid p,q),\, \pi_\ell^\omega(\cdot \mid p,q)\big)\Big|_{\omega=0} \;=\; -\,\mathrm{Cov}\Big(g_\ell(V),\; \mathbf{1}\{V \in S_\ell^+\}\Big). \tag{22}$$

In particular, if $\mathrm{Cov}\big(g_\ell(V),\, \mathbf{1}\{V \in S_\ell^+\}\big) > 0$, then there exists $\omega_0 > 0$ such that for all $0 < \omega \le \omega_0$,

$$\mathrm{TV}\big(\bar\pi_\ell(\cdot \mid p,q),\, \pi_\ell^\omega(\cdot \mid p,q)\big) \;<\; \mathrm{TV}\big(\bar\pi_\ell(\cdot \mid p,q),\, \tilde\pi_\ell(\cdot \mid p,q)\big).$$

Proof. For each legal token $v \in \mathcal{A}_\ell(p)$, define the probability masses

$$\bar p_v := \bar\pi_\ell(v \mid p,q), \qquad \tilde p_v := \tilde\pi_\ell(v \mid p,q), \qquad p_v^\omega := \pi_\ell^\omega(v \mid p,q).$$

We use $V_\omega \sim \pi_\ell^\omega(\cdot \mid p,q)$ for a random token drawn from the fused distribution at parameter ω.

Step 1: derivative of π^ω. Since $\pi_\ell^\omega(\cdot \mid p,q)$ is an exponential-family tilt of $\tilde\pi_\ell(\cdot \mid p,q)$,

$$\frac{d}{d\omega}\, p_v^\omega \;=\; p_v^\omega \,\big(g_\ell(v) - \mathbb{E}_{V_\omega \sim \pi_\ell^\omega(\cdot|p,q)}[g_\ell(V_\omega)]\big).$$

At $\omega = 0$, $p_v^0 = \tilde p_v$ for every token v, so

$$\frac{d}{d\omega}\, p_v^\omega\Big|_{\omega=0} = \tilde p_v\big(g_\ell(v) - \mathbb{E}_{V\sim\tilde\pi_\ell(\cdot|p,q)}[g_\ell(V)]\big).$$

Step 2: sign preservation. By the no-tie assumption and the finite support, $\bar p_v - \tilde p_v \neq 0$ for every v. By continuity of p_v^ω in ω, there exists $\epsilon > 0$ such that for all $0 < \omega < \epsilon$,

$$\mathrm{sign}\big(\bar p_v - p_v^\omega\big) \;=\; \mathrm{sign}\big(\bar p_v - \tilde p_v\big) \qquad \text{for every } v.$$

Step 3: TV derivative. In this regime,

$$\mathrm{TV}\big(\bar\pi_\ell(\cdot \mid p,q), \pi_\ell^\omega(\cdot \mid p,q)\big) = \sum_{v\in S_\ell^+} \big(\bar p_v - p_v^\omega\big).$$

Differentiating at $\omega = 0$:

$$\begin{aligned}
&\frac{d}{d\omega}\,\mathrm{TV}\big(\bar\pi_\ell(\cdot \mid p,q), \pi_\ell^\omega(\cdot \mid p,q)\big)\Big|_{\omega=0} \\
&= -\sum_{v\in S_\ell^+} \tilde p_v\,\big(g_\ell(v) - \mathbb{E}_{V\sim\tilde\pi_\ell(\cdot|p,q)}[g_\ell(V)]\big) \\
&= -\,\mathrm{Cov}_{V\sim\tilde\pi_\ell(\cdot|p,q)}\big(g_\ell(V),\; \mathbf{1}\{V \in S_\ell^+\}\big).
\end{aligned}$$

If this covariance is positive, the derivative is strictly negative, so by continuity there exists $\omega_0 > 0$ with

$$\mathrm{TV}\big(\bar\pi_\ell(\cdot \mid p,q), \pi_\ell^\omega(\cdot \mid p,q)\big) < \mathrm{TV}\big(\bar\pi_\ell(\cdot \mid p,q), \tilde\pi_\ell(\cdot \mid p,q)\big)$$

for all $0 < \omega \le \omega_0$.

Table 5: Comparison with TIGER on RQ-based generative recommendation.

	Beauty				Toys				Sports			
Method	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10
TIGER	0.039038	0.055270	0.026157	0.031408	0.035906	0.052596	0.024123	0.029479	0.021771	0.033092	0.014115	0.017762
PRO	**0.042794**	**0.062961**	**0.028821**	**0.035295**	**0.040645**	**0.060066**	**0.026946**	**0.033198**	**0.024046**	**0.037474**	**0.015183**	**0.019501**
Δ (abs.)	+0.003756	+0.007691	+0.002664	+0.003887	+0.004739	+0.007470	+0.002823	+0.003719	+0.002275	+0.004382	+0.001068	+0.001739
Δ (rel.)	+9.62%	+13.92%	+10.18%	+12.38%	+13.20%	+14.20%	+11.70%	+12.62%	+10.45%	+13.24%	+7.57%	+9.79%

Corollary A.10 (Oracle-alignment decomposition)**.** *Under the decomposition* $g_\ell(v) = 2\tau \log \bar{\pi}_\ell(v \mid p, q) + \kappa_\ell - \eta_\ell(v)$*, let* $V \sim \tilde{\pi}_\ell(\cdot \mid p, q)$*. Then*

$$\mathrm{Cov}\big(g_\ell(V), \mathbf{1}\{V \in S_\ell^+\}\big) = \underbrace{2\tau\, \mathrm{Cov}\big(\log \bar{\pi}_\ell(V \mid p, q),\, \mathbf{1}\{V \in S_\ell^+\}\big)}_{\textit{oracle alignment}} - \underbrace{\mathrm{Cov}\big(\eta_\ell(V),\, \mathbf{1}\{V \in S_\ell^+\}\big)}_{\textit{distortion interference}}. \tag{23}$$

This decomposition makes the sufficient condition in Prop. A.9 easier to interpret. The first covariance is positive when tokens underweighted by the decoder tend to have high conditional-oracle probability. The second covariance is the token-dependent distortion introduced by $\eta_\ell(v)$. Therefore, score fusion locally reduces ε_ℓ when the oracle-alignment term dominates the distortion-interference term. When ε_ℓ is large (as observed empirically in App. B.7, mean $\mathrm{TV} > 0.95$), many oracle-preferred tokens are underweighted by the decoder. In that case, S_ℓ^+ concentrates on tokens with high $\log \bar{\pi}_\ell(v \mid p, q)$, which makes the oracle-alignment term large and positive.

Remark A.11. Props. A.8 and A.9 operate on the per-parent conditional distribution. The global decoder mismatch $\varepsilon_\ell = \mathrm{TV}(\bar{\pi}_\ell, \tilde{\pi}_\ell)$ in Thm. 2 is defined over all length-ℓ prefixes. The local result therefore does not by itself prove a global reduction for arbitrary beams. It does explain the mechanism used in our decoder: if the alignment condition in Prop. A.9 holds across the surviving parents, then reducing mismatch inside these conditional distributions also reduces the global decoder mismatch over length-ℓ prefixes. The empirical decoder mismatch is consistently large across datasets (App. B.7: mean $\mathrm{TV} > 0.95$, std < 0.03), which supports treating this mismatch as a systematic decoding issue rather than an isolated parent-prefix case.

B Additional Experimental Results

B.1 Generality to generative recommendation

Tab. 5 reports the additional experiment on TIGER [31]. We evaluate on the Amazon Beauty, Toys, and Sports categories [12]. PRO improves all Recall and NDCG metrics over TIGER, which supports the generality of our analysis of prefix survival and PRO in RQ-based generative recommendation.

B.2 Full retrieval results

Tabs. 6–8 report the complete retrieval results on M-BEIR, extending Tab. 2 with R@10, grouped by query type.

Table 6: Full retrieval results on M-BEIR: text↔image tasks. **Bold**: best generative method per column.

Method	COCO $t\to i$			Flickr30k $t\to i$			COCO $i\to t$		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Dense									
CLIP-SF	**55.2**	**80.7**	**88.4**	**79.1**	**95.0**	**97.5**	65.7	87.5	92.6
BLIP-FF	53.7	79.6	87.3	73.9	92.4	95.3	**71.1**	**90.9**	**95.4**
Generative									
IRGen	29.6	50.7	56.3	49.0	68.9	72.5	–	–	–
GRACE	16.7	39.5	50.3	37.4	59.5	66.2	–	–	–
AVG	31.3	58.0	–	62.8	85.4	–	–	–	–
SemCORE	42.4	57.5	–	69.0	83.0	–	–	–	–
GENIUS	37.1±.33	66.4±.28	76.7±.22	56.8±.42	81.3±.25	85.5±.18	47.9±.35	77.3±.24	86.0±.17
Ours	**45.8**±.38	**72.6**±.30	**81.3**±.22	**70.6**±.45	**87.3**±.22	**89.2**±.15	**59.0**±.40	**84.3**±.25	**90.9**±.14

Table 7: Full retrieval results on M-BEIR (cont.): text-to-multimodal and image-to-image tasks. "–": method does not support the task.

Method	WebQA $t\to t$			WebQA $t\to i,t$			NIGHTS $i\to i$		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Dense									
CLIP-SF	**58.3**	**84.2**	**90.0**	47.6	76.3	83.9	**8.8**	**31.6**	52.8
BLIP-FF	51.9	78.6	84.9	**49.5**	**78.1**	**86.7**	8.1	31.0	**54.8**
Generative									
IRGen	–	–	–	–	–	–	–	–	–
GRACE	–	–	–	–	–	–	–	–	–
GENIUS	19.4±.32	29.9±.28	32.2±.25	29.1±.36	51.3±.28	58.2±.24	1.2±.14	11.6±.28	30.7±.35
Ours	**25.7**±.40	**40.7**±.35	**45.2**±.30	**35.3**±.42	**57.6**±.32	**64.6**±.26	**3.5**±.20	**19.7**±.35	**40.7**±.38

Table 8: Full retrieval results on M-BEIR (cont.): composed image+text queries. "–": method does not support the task.

Method	CIRR $i,t\to i$			OVEN $i,t\to i,t$			InfoSeek $i,t\to t$		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Dense									
CLIP-SF	5.9	43.2	56.5	**49.4**	**69.2**	**75.3**	**14.5**	**28.8**	**37.0**
BLIP-FF	**24.4**	**50.9**	**61.3**	36.1	56.4	63.2	10.5	23.4	30.2
Generative									
IRGen	–	–	–	–	–	–	–	–	–
GRACE	–	–	–	–	–	–	–	–	–
GENIUS	5.5±.24	19.0±.30	28.1±.28	26.4±.38	36.0±.30	43.9±.26	6.1±.22	10.2±.25	12.9±.22
Ours	**8.6**±.30	**27.8**±.36	**38.2**±.32	**34.5**±.44	**51.3**±.35	**61.3**±.28	**7.3**±.25	**15.0**±.30	**19.3**±.28

B.3 Larger beams and prefix-aware decoder training

PrefixCE. PrefixCE is a simple prefix-aware decoder training baseline. It keeps the identifiers, decoder architecture, and decoding procedure of GENIUS, and only reweights the decoder cross-entropy loss of the k-th semantic code token by

$$w_k = \frac{2(L-k+1)}{L+1}, \qquad k = 1, \ldots, L,$$

where L=16 is the number of semantic code tokens. The weights decrease linearly with token position and average to one, so PrefixCE emphasizes early identifier tokens without changing the overall loss scale. All other training and decoding settings follow GENIUS. At the default beam width

B=50, PrefixCE improves GENIUS on all nine tasks (Tab. 9), confirming the benefit of emphasizing early identifier tokens; however, its gains remain smaller than those of PRO on every task.

Table 9: Comparison with the prefix-aware decoder training baseline PrefixCE at B=50 (R@1/R@5). **Bold**: best result.

Task	GENIUS	PrefixCE	PRO
COCO $t\to i$	37.1/66.4	38.2/67.1	**45.8/72.6**
Flickr30k $t\to i$	56.8/81.3	57.3/81.9	**70.6/87.3**
WebQA $t\to t$	19.4/29.9	20.7/32.0	**25.7/40.7**
WebQA $t\to i,t$	29.1/51.3	30.3/53.1	**35.3/57.6**
COCO $i\to t$	47.9/77.3	49.3/78.5	**59.0/84.3**
NIGHTS $i\to i$	1.2/11.6	1.9/13.1	**3.5/19.7**
CIRR $i,t\to i$	5.5/19.0	6.8/20.7	**8.6/27.8**
OVEN $i,t\to i,t$	26.4/36.0	27.4/38.2	**34.5/51.3**
InfoSeek $i,t\to t$	6.1/10.2	6.4/11.1	**7.3/15.0**

Beam width. Tab. 10 varies the beam width $B \in \{20, 50, 100, 200\}$ on four tasks covering text-to-image and composed retrieval. Larger beams provide modest improvements for both GENIUS and PRO, and GENIUS largely saturates by B=100. PRO at the default B=50 remains stronger than GENIUS at B=200 in every R@1 and R@5 comparison. PrefixCE at B=200 also remains below PRO at B=50 on all four tasks and both metrics. Thus, the gains of PRO cannot be explained solely by increased search capacity or by emphasizing early identifier tokens.

Table 10: Retrieval performance (R@1/R@5) under different beam widths B. PrefixCE is evaluated at B=50 and B=200 (–: not evaluated). The default beam width is B=50.

Task	Method	B=20	B=50	B=100	B=200
COCO $t\to i$	GENIUS	36.2/63.6	37.1/66.4	37.2/67.1	37.2/67.2
	PrefixCE	–	38.2/67.1	–	38.4/68.2
	PRO	44.0/67.9	45.8/72.6	46.1/73.7	46.0/73.9
CIRR $i,t\to i$	GENIUS	5.2/17.8	5.5/19.0	5.6/19.5	5.6/19.6
	PrefixCE	–	6.8/20.7	–	6.7/21.1
	PRO	8.2/25.6	8.6/27.8	8.8/28.4	8.8/28.4
OVEN $i,t\to i,t$	GENIUS	25.6/34.2	26.4/36.0	26.6/36.6	26.5/36.6
	PrefixCE	–	27.4/38.2	–	27.7/38.9
	PRO	33.3/48.7	34.5/51.3	34.8/52.1	34.7/52.1
InfoSeek $i,t\to t$	GENIUS	5.8/9.4	6.1/10.2	6.2/10.6	6.2/10.6
	PrefixCE	–	6.4/11.1	–	6.5/11.4
	PRO	6.9/13.8	7.3/15.0	7.4/15.5	7.4/15.6

B.4 Official M-BEIR global pool

The main results use task-specific candidate pools. Tab. 11 additionally evaluates GENIUS and PRO on the official M-BEIR global pool [42], which merges the candidates of all M-BEIR tasks. PRO outperforms GENIUS on all eight evaluated tasks in both R@1 and R@5, raising the macro-averaged R@1/R@5 from 19.4/35.4 to 26.1/42.7. Because the two settings use different candidate pools, these results should be compared with each other rather than with the task-specific results in Tab. 2.

Table 11: Results (R@1/R@5) on the official M-BEIR global pool. Flickr30k is not part of M-BEIR and is therefore excluded. **Bold**: better result.

Task	GENIUS	PRO
COCO $t \to i$	25.7/53.0	**39.2/62.5**
COCO $i \to t$	43.4/76.7	**58.7/84.2**
NIGHTS $i \to i$	1.2/11.6	**3.5/19.6**
WebQA $t \to t$	18.9/28.7	**24.1/37.9**
WebQA $t \to i,t$	28.7/50.2	**34.2/56.2**
CIRR $i,t \to i$	5.3/18.0	**8.9/26.2**
OVEN $i,t \to i,t$	25.8/35.8	**33.2/41.7**
InfoSeek $i,t \to t$	5.9/ 9.5	**6.8/13.1**
Macro average	19.4/35.4	**26.1/42.7**

B.5 Effect of candidate pool size

To examine how retrieval effectiveness changes with corpus size, Tab. 12 evaluates dense retrieval (CLIP-SF), GENIUS, and PRO on OVEN and InfoSeek with candidate pools of increasing size, up to the full task-specific pools (335K and 612K items, respectively). All methods degrade as the pool grows. At every evaluated pool size, PRO outperforms GENIUS and reduces the gap between the RQ-based generative retriever and dense retrieval, indicating that its improvements remain robust as the candidate pool grows.

Table 12: Retrieval performance (R@1/R@5) as the candidate pool grows on OVEN and InfoSeek. The largest pool of each task is its full task-specific pool. Dense: CLIP-SF.

OVEN $i,t \to i,t$ **Method**	50K	100K	200K	335K
Dense	63.9/81.9	58.3/77.6	53.6/73.1	49.4/69.2
GENIUS	30.6/46.1	29.5/43.2	27.7/38.8	26.4/36.0
PRO	41.1/66.6	39.2/62.1	36.6/56.5	34.5/51.3
InfoSeek $i,t \to t$ **Method**	50K	100K	300K	612K
Dense	30.3/52.1	25.9/45.8	18.5/35.3	14.5/28.8
GENIUS	9.0/16.6	8.1/14.6	6.8/11.7	6.1/10.2
PRO	13.1/28.7	11.1/24.0	8.7/17.9	7.3/15.0

B.6 Complete ablation

Tab. 13 reports all seven configurations tested in the ablation study, complementing the additive summary in Tab. 4. We organize the results by individual components, pairwise combinations, and the full system.

Per-component analysis. Among individual components, the strongest contributor varies by task: vocabulary scheduling gives the largest standalone gains on the OVEN settings, score fusion is strongest on WebQA $t \to t$, and prefix distillation is strongest on COCO. These patterns are consistent with the view from the survival bound that different tasks are dominated by different quantities.

Pairwise interactions. Prefix distillation and vocabulary scheduling show robust positive interaction across the ablation settings. This supports the hypothesis that reducing ranking divergence makes subsequent margin enlargement more effective. In contrast, prefix distillation + score fusion shows mixed interaction (positive on 3 tasks, near-zero or slightly negative on 3), confirming that the synergy between the two codebook-learning components is more robust than the codebook–decoder interaction. Notably, vocabulary scheduling alone hurts WebQA $t \to t$ by -0.4, but after prefix distillation the marginal vocabulary scheduling gain turns positive ($+0.4$), consistent with the hypothesis that reducing K_ℓ stabilizes subsequent margin enlargement.

Table 13: Complete ablation results. Individual components (top), pairwise combinations (middle), and full approach (bottom). Abbreviations: PD: Prefix distillation; VS: Vocabulary scheduling; SF: Score fusion. **Bold**: best per column. All entries: mean±std over 5 seeds.

Configuration	COCO $t\to i$		COCO $i\to t$		WebQA $t\to t$		WebQA $t\to i,t$		OVEN $i,t\to i$		OVEN $i,t\to i,t$	
	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
Baseline												
Base (vanilla RQ)	36.8±.30	66.5±.24	48.1±.32	77.2±.22	19.5±.34	30.0±.28	28.9±.35	51.4±.26	5.9±.24	14.7±.27	26.7±.32	36.4±.26
Individual components												
+ PD ($K_\ell\downarrow$)	40.2±.33	68.2±.26	51.5±.35	78.9±.23	21.2±.36	31.7±.30	31.4±.38	53.1±.28	5.6±.23	14.2±.26	28.4±.34	39.9±.27
+ VS ($m_\ell\uparrow$)	38.5±.32	68.6±.25	49.4±.33	79.5±.22	19.1±.35	29.6±.29	30.1±.36	52.7±.27	8.5±.28	18.5±.30	30.4±.35	44.1±.28
+ SF ($\varepsilon_\ell\downarrow$)	39.7±.34	69.8±.26	51.7±.36	81.1±.24	23.4±.40	36.6±.33	31.2±.38	54.3±.29	5.3±.22	13.3±.25	28.6±.33	40.7±.28
Pairwise combinations												
+ PD + SF	42.9±.36	71.4±.27	55.3±.38	82.5±.24	24.9±.42	**40.1**±.34	33.7±.40	55.9±.30	6.2±.25	14.5±.27	30.2±.35	44.8±.29
+ PD + VS	42.1±.35	70.7±.26	53.6±.37	81.4±.23	21.6±.38	33.2±.31	33.5±.39	55.2±.29	8.8±.28	**19.0**±.30	32.2±.36	47.5±.29
Full approach												
+ PD + VS + SF	**46.2**±.38	**73.0**±.28	**59.3**±.42	**84.6**±.25	**26.2**±.44	39.7±.34	**35.6**±.42	**57.5**±.32	**9.1**±.30	17.6±.30	**34.4**±.38	**51.4**±.31

R@5 exceptions. The full system achieves the best R@5 on 4 of 6 tasks. Two exceptions: prefix distillation + score fusion outperforms Full on WebQA $t\to t$ R@5 by $+0.4$, and prefix distillation + vocabulary scheduling outperforms Full on OVEN $i,t\to i$ R@5 by $+1.4$. Both exceptions involve tasks where score fusion or vocabulary scheduling provide weaker marginal gains at R@5, suggesting that inference-time score fusion is less effective on small text-only pools and on composed queries where the geometric proxy is noisy.

B.7 Cross-dataset diagnostic analysis

Sec. 6 validates the survival bound's predictions on COCO. Fig. 6 extends this analysis to four additional datasets spanning all three query families, using $\tau{=}1.0$ and beam width 20.

The top row shows K_ℓ quintile versus survival at levels 0, 3, and 5. All five datasets exhibit a consistent monotone decline: higher ranking divergence predicts lower prefix survival at every level, confirming that the relationship identified on COCO generalizes across query modalities and dataset characteristics. The effect is strongest on composed-query tasks (CIRR, OVEN), where K_ℓ variance is largest, and mildest on Flickr30k, where the overall survival rate is high and leaves less room for differentiation.

The bottom row shows m_ℓ quintile versus survival. All datasets display a monotone increase: larger teacher margin correlates with higher survival at every level. The effect is milder than for K_ℓ, consistent with the main-text finding that ranking divergence is the dominant bottleneck in the survival bound.

Decoder mismatch is a systemic bottleneck. Unlike K_ℓ and m_ℓ, which exhibit substantial per-query variance, the decoder mismatch ε_ℓ is uniformly high across all queries and all five datasets (mean TV > 0.95, std < 0.03). Consequently, quintile-binning ε_ℓ reveals no monotone trend with survival—not because decoder mismatch is unimportant, but because it affects *every* query equally. This finding reframes the role of score fusion (Sec. 4): rather than selectively rescuing queries with unusually high decoder mismatch, it compensates for a systemic gap between the decoder scoring distribution and the quantized oracle, explaining why score fusion yields broad-based improvements across all tasks and query types (Tab. 4).

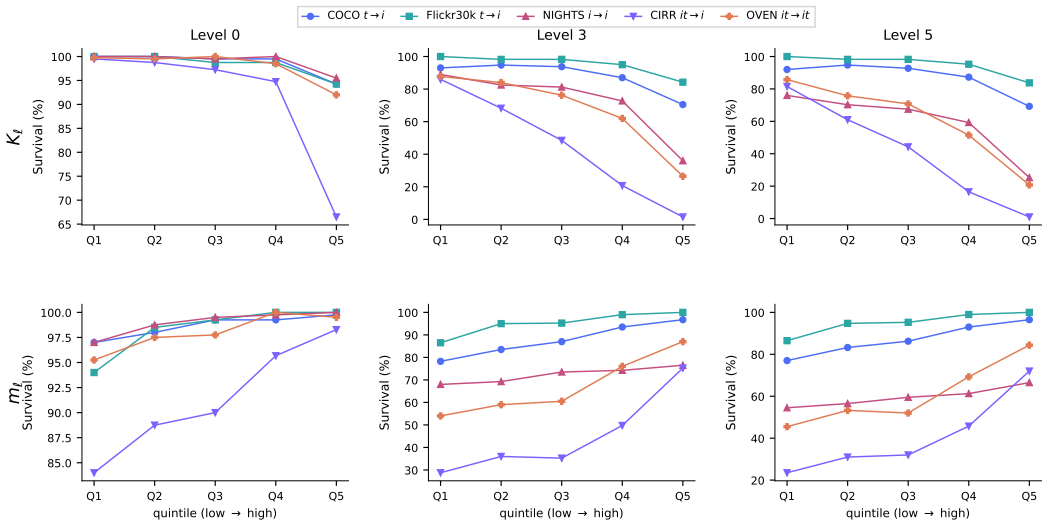


Figure 6: Cross-dataset diagnostic analysis. Top row: K_ℓ quintile vs. survival. Bottom row: m_ℓ quintile vs. survival. All five datasets show monotone trends consistent with the survival bound (Thm. 2). Beam width B=20.

B.8 Indexing objectives and prefix survival across beam widths

Thm. 1 proves that standard RQ objectives, including reconstruction, contrastive, and residual-fitting objectives, cannot guarantee the discriminability and survival of an intermediate target prefix. Fig. 2 examines one concrete consequence of this result under quantized oracle scoring with B=50: full reconstruction MSE has little association with prefix survival. It does not reproduce the paired construction used in the proof, but provides empirical grounding for the practical relevance of the theorem.

Tab. 14 extends this analysis to $B \in \{20, 50, 100, 200\}$ and to the three objective classes considered in Thm. 1: full reconstruction MSE, RQ fitting quality, and the pre-quantization contrastive objective. Since the contrastive loss is defined over batch-level pre-quantization similarities, we use its query-level positive-to-negative margin as the corresponding diagnostic signal. The same pattern appears across beam widths. Full reconstruction MSE has almost no predictive value for survival, RQ fitting quality has weak predictive value, and the contrastive margin carries a moderate query-difficulty signal. Existing objectives may therefore contain signals correlated with survival, but they do not guarantee prefix discriminability under decoding, which is consistent with Thm. 1.

Table 14: Association between indexing-objective signals and target prefix survival under quantized oracle scoring, in the setting of Fig. 2 with varying beam width B. Each signal cell reports AUROC/Spearman ρ with survival; Survival is the target prefix survival rate.

B	Survival	MSE	RQ fitting	Contrastive margin
20	73.0%	0.53/0.03	0.56/0.09	0.63/0.19
50	88.6%	0.52/0.02	0.57/0.08	0.64/0.16
100	95.0%	0.54/0.02	0.55/0.06	0.62/0.12
200	97.8%	0.51/0.01	0.57/0.04	0.61/0.08

B.9 Target survival under the trained decoder

All end-to-end results in Sec. 6 are produced by an actual autoregressive generative retriever: both GENIUS and PRO train a T5-small decoder to generate item identifiers and retrieve through trie-constrained beam search. The quantized oracle is used only for diagnostic analyses of indexing behavior. To complement these diagnostics with decoder-level evidence, Tab. 15 reports the final target survival rate under the trained decoder, i.e., whether the complete target identifier remains in

the beam after all decoding steps. At the default B=50, PRO improves final survival from 56.91% to 66.91% on OVEN and from 13.74% to 23.51% on InfoSeek, and the improvement on InfoSeek grows with the beam width. On COCO text-to-image, survival is already about 90% for both systems at B=50, leaving limited room for improvement on this diagnostic; there, the R@1/R@5 gains of PRO (Tab. 2) mainly reflect better ranking of the retained candidates.

Table 15: Final target survival under the trained T5 decoder: the fraction of queries whose complete target identifier remains in the beam after all decoding steps.

Task	Method	B=20	B=50	B=100	B=200
COCO $t\to i$	GENIUS	77.80%	90.32%	96.14%	98.93%
	PRO	76.65%	90.28%	96.42%	99.14%
OVEN $i,t\to i,t$	GENIUS	48.40%	56.91%	63.58%	70.24%
	PRO	58.39%	66.91%	73.24%	79.22%
InfoSeek $i,t\to t$	GENIUS	8.15%	13.74%	20.12%	28.91%
	PRO	13.09%	23.51%	35.42%	51.01%

B.10 Efficiency comparison

Prefix survival and throughput scaling. Fig. 7a reports the per-level survival rate of the target identifier token under quantized oracle scoring. At each RQ level, this metric conditions on the preceding target prefix and checks whether the target token at that level is ranked among the top-B candidate tokens. Fig. 7b compares query throughput as the candidate pool grows from 5K to 300K on a single GPU. Exact dense retrieval stores all N embeddings and scans them at query time, with cost $O(N{\cdot}d)$; generative decoding stores only discrete RQ codes and traverses a trie at $O(L{\cdot}B{\cdot}V)$, independent of N. Dense throughput declines steadily as the pool grows, while generative throughput remains nearly constant. The two curves cross near N=50K; beyond this point generative retrieval is faster than exact scanning. Adding score fusion (Eq. 7) has little effect on throughput, as it only computes lightweight residual inner products per decoding step.

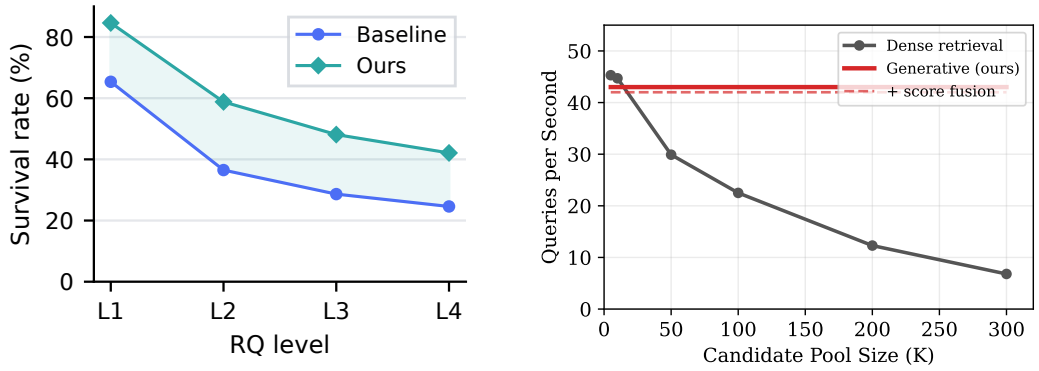
Scope of the efficiency comparison. Following GENIUS, we compare trie-constrained generative decoding with exact scanning over the full corpus, which isolates the core retrieval operations and their behavior as the corpus grows. Approximate nearest-neighbor indexes such as HNSW [28] provide an important system-level acceleration for dense retrieval, but they introduce additional index configurations and a trade-off between recall and latency. A rigorous comparison with such indexes would require matched retrieval quality and careful tuning of both systems, which is beyond the scope of this method-focused study; our efficiency claims therefore apply to the exact retrieval setting.

Latency and training overhead. Tab. 16 reports per-query inference latency at B=50. PRO increases latency over GENIUS by 3.9–4.9%, mainly due to geometric score fusion, since vocabulary scheduling introduces no additional online computation. On the training side, prefix ranking distillation increases the cost of codebook learning by about 8%, while the end-to-end training overhead remains below 1% because decoder training is unchanged and dominates the total training cost (App. D). Relative to GENIUS, PRO improves the macro-averaged R@1 and R@5 over the nine tasks by 26.5% and 19.1%, so these effectiveness gains come with limited training and inference overhead.

Table 16: Per-query inference latency (ms/query) at B=50.

Task	GENIUS	PRO	Overhead
COCO $t\to i$	171.94	180.14	4.77%
CIRR $i,t\to i$	170.67	177.37	3.93%
OVEN $i,t\to i,t$	193.37	202.84	4.90%
InfoSeek $i,t\to t$	187.35	195.72	4.47%

Index size. Tab. 17 compares the index footprint at several pool sizes. The generative index stores L=16 integer codes per target item (two bytes each), yielding a constant compression ratio of over ninety times relative to storing full 768-dimensional floating-point embeddings.



(a) Per-level token survival

(b) Throughput scaling

Figure 7: End-to-end diagnostics and efficiency. (a) Per-level survival rate of the target identifier token on COCO text-to-image under quantized oracle scoring; our method improves target token retention over the baseline indexing. (b) Query throughput vs. candidate pool size on a single GPU. Exact dense scanning scales as $O(N \cdot d)$; generative decoding remains nearly constant. Score fusion adds little overhead (the two generative curves nearly overlap).

Table 17: Index size comparison. Dense retrieval stores float32 embeddings ($N \times 768 \times 4$ B); generative retrieval stores int16 RQ codes ($N \times 16 \times 2$ B).

Pool size N	Dense (MB)	Generative (MB)	Ratio
5 K	14.6	0.15	96×
50 K	146.5	1.53	96×
335 K	979.0	10.2	96×
1 M	2 930	30.5	96×

B.11 Encoder sensitivity

To verify that the performance gap between RQ-based MGR and dense retrieval is not primarily driven by encoder choice, we compare three multimodal encoders under vanilla GENIUS (RQ $L = 16$, $V = 4096$) with a T5-small decoder (beam width 20, no re-ranking), across four M-BEIR tasks (MSCOCO $t \to i$, Flickr30k $t \to i$, WebQA $t \to t$, NIGHTS $i \to i$). The encoders span different capacities and training regimes:

- **CLIP-OpenAI** (ViT-L/14, 768-d): zero-shot CLIP without M-BEIR fine-tuning, serving as the non-instruct baseline.
- **LLM2CLIP** [14] (LLM2Vec-8B + ViT-L/14-336, 1280-d): higher-capacity encoder via an LLM backbone, also evaluated in the non-instruct regime.
- **CLIP-SF** [42] (ViT-L/14, 768-d): UniIR's instruction-tuned CLIP, used in its native instruct regime.

Tab. 18 reports dense-retrieval (query-target similarity), vanilla GENIUS, and our method Recall@{1,5} across the four tasks.

Across encoders of different capacities and training regimes, the vanilla GENIUS column exhibits a similar proportional gap between generative and dense retrieval: switching to a higher-capacity encoder or a retrieval-specialized one shifts both scores upward but does not close their relative gap. This suggests that the gap is not caused by insufficient encoder expressiveness; rather, the RQ-based tokenization-and-decoding bottleneck prevents the decoder from fully preserving the discrimination in the encoder space. The Ours column consistently narrows this decay across encoders, indicating that our method mitigates the RQ bottleneck rather than simply benefiting from a particular encoder choice.

Table 18: Recall@{1,5} with three multimodal encoders on four M-BEIR tasks. Dense: query-target inner-product similarity; GENIUS: vanilla GENIUS pipeline with the corresponding encoder; Ours: our pipeline with the corresponding encoder.

Task	Encoder	Dense		GENIUS		Ours	
		R@1	R@5	R@1	R@5	R@1	R@5
MSCOCO $t \to i$	CLIP-OpenAI	38.2	62.5	27.8	53.7	38.6	63.7
	LLM2CLIP	54.4	78.5	34.0	58.6	45.5	68.5
	CLIP-SF	55.2	80.7	37.1	66.4	45.8	72.6
Flickr30k $t \to i$	CLIP-OpenAI	66.9	88.9	55.4	80.7	68.3	87.8
	LLM2CLIP	82.5	95.8	57.5	80.2	75.6	89.9
	CLIP-SF	79.1	95.0	56.8	81.3	70.6	87.3
WebQA $t \to t$	CLIP-OpenAI	22.2	40.9	9.9	21.6	17.4	25.1
	LLM2CLIP	50.6	78.8	17.6	27.6	29.5	44.2
	CLIP-SF	58.3	84.2	19.4	29.9	25.7	40.7
NIGHTS $i \to i$	CLIP-OpenAI	6.8	27.8	0.5	6.9	3.3	19.3
	LLM2CLIP	8.4	31.2	2.2	14.1	3.4	20.0
	CLIP-SF	8.8	31.6	1.2	11.6	3.5	19.7

C Further Discussion

C.1 Generative and dense retrieval

We view GR and dense retrieval (DR) as two distinct retrieval paradigms. Tab. 19 summarizes their differences: beyond the retrieval operation itself, they differ in how they represent and organize the corpus, learn the index, model relevance, and connect retrieval with generative models. These differences give GR research value along three dimensions.

Table 19: Comparison between standard dense retrieval and generative retrieval.

Aspect	Standard dense retrieval	Generative retrieval
Retrieval formulation	Encodes queries and items as continuous vectors and retrieves through similarity search	Maps a query directly to a discrete target identifier through sequence generation
Corpus representation	Stores continuous item embeddings in an external vector index	Represents each item by a discrete identifier in a learned identifier space
Indexing and retrieval learning	Representation learning and index construction are typically separated from retrieval	Identifier construction and query-to-identifier prediction can be coordinated within a unified framework
Online retrieval interface	Exact or approximate nearest-neighbor search	Constrained sequence decoding
Relevance modeling	Commonly fixed-dimensional vector similarity	Conditional autoregressive likelihood over identifier sequences
Integration with generative tasks	Retrieval and downstream generation are generally separate modules	Can share model components and a sequence-generation interface with recommendation or generation tasks

Learnable identifier space. GR represents the corpus and the retrieval outputs within the same discrete identifier space, providing a direct path toward coordinating identifier construction with query-to-identifier retrieval. Identifier construction determines the output space of the retrieval model and can therefore incorporate query-conditioned retrieval signals, and semantic identifiers can encode hierarchical or task-oriented corpus structures. PRO reflects this opportunity by introducing query-conditioned prefix ranking into identifier construction and reusing identifier geometry during decoding, bringing the learned identifier structure closer to the behavior of the actual retrieval procedure.

Beyond fixed-dimensional matching. Typical DR compresses each query and item into fixed-dimensional vectors and models relevance mainly through an inner product or another bilinear interaction. GR instead maps a query into the corpus identifier space through parameterized, query-conditioned identifier prediction, which unfolds as a sequence of conditional decisions. Decoder-level GR relevance can be interpreted within a multi-vector matching framework [45], clarifying how its relevance computation differs from single-vector DR. This formulation may allow representational capacity to scale with model capacity and sequential decision making, potentially alleviating the global low-rank bottleneck of fixed-dimensional dense representations; recent theoretical and controlled empirical work examines these possibilities [50].

Explicit prefix decisions. RQ-based GR represents retrieval as a sequence of prefix-level decisions. Hierarchical retrieval is not unique to GR, but here each prefix is both part of the item identifier and an intermediate retrieval output generated by the model, so it can be directly supervised, analyzed, and optimized. The learned identifier structure functions as intended only when target prefixes remain available throughout decoding, which makes the indexing–decoding gap a key bottleneck in realizing the benefits of structured identifiers.

Generating identifiers also avoids explicit nearest-neighbor search over an external embedding index and provides a common sequence-generation interface with generative recommendation and other generative tasks, allowing model architectures and training mechanisms to be shared [22, 38]. Together, these properties give GR a distinct design space for jointly studying corpus representation, identifier design, and retrieval prediction. Our contribution lies within this paradigm: PRO improves the effectiveness of RQ-based GR by addressing the indexing–decoding gap, while strong dense retrievers still achieve higher absolute effectiveness on the evaluated benchmarks.

C.2 Relevance to multimodal retrieval

PRO builds on the RQ-based GR framework of GENIUS and targets a retrieval bottleneck that arises when multimodal representations are converted into discrete identifiers: rankings established by the multimodal encoder in the original representation space can be lost at intermediate RQ levels. Since this gap occurs in the shared discrete-identifier retrieval stage, our analysis and method apply across modality processing architectures, and also to other RQ-based settings such as generative recommendation (App. B.1). This generality is reflected in consistent improvements across query modalities and task formulations. Relative to GENIUS, PRO improves R@1/R@5 by 23.5%/9.3% on COCO text-to-image, 24.3%/7.4% on Flickr30k text-to-image, and 23.2%/9.1% on COCO image-to-text. On the composed-query tasks CIRR, OVEN, and InfoSeek, the relative improvements are 56.4%/46.3%, 30.7%/42.5%, and 19.7%/47.1%, and PRO closes 14.3–35.2% of the R@1 gap and 25.8–46.1% of the R@5 gap between GENIUS and the strongest dense baseline (Tab. 2). More multimodal-specific issues, including the modality gap, modality-aware identifier construction, and cross-modal alignment, remain meaningful directions for future work.

C.3 Beyond RQ identifiers

Our theoretical analysis, proofs, and method design rely on the residual and additive structure of RQ, so the current results do not directly transfer to other identifier structures. Similar failure modes may nonetheless arise whenever the objective used to design or optimize an identifier differs from the way the identifier is processed by the actual decoding procedure. For example, a hierarchical identifier may produce semantically meaningful leaf codes while its intermediate nodes remain insufficiently discriminative during tree traversal. A learned multi-token identifier may be optimized for complete-sequence prediction while decoding decisions depend on partial sequences. For lexical or multiview identifiers [20, 23], the ordering and grouping of identifier tokens may interact with decoding in ways that are not reflected in the identifier learning objective. Depending on the decoding method, such a mismatch may appear through deterministic pruning, stochastic candidate coverage, or subsequent reranking. These settings motivate a broader theoretical analysis of the relationship between identifier design and decoding, which we leave to future work.

C.4 Sampling-based decoding

Our analysis is based on deterministic top-B beam search, whereas sampling turns target retention into a stochastic coverage problem. Under K independent full-sequence samples, a target identifier

z^* with probability $p_\theta(z^* \mid q)$ is sampled at least once with probability $1 - \big(1 - p_\theta(z^* \mid q)\big)^K$. For sampling followed by reranking, successful retrieval decomposes into whether the target identifier is covered by the sampled candidate set and whether it is subsequently ranked sufficiently high. A mismatch between identifier design and decoding may also arise in this setting: if the learned identifier structure causes the decoder to assign insufficient probability mass to target prefixes, the target may have low sampling coverage even when its complete representation is well optimized. Prefix-aware indexing or geometry-guided sampling may therefore remain relevant, but we leave a formal treatment of sampling-based decoding to future work.

D Implementation Details

All experiments use the CLIP-SF encoder (ViT-L/14, 768-d embeddings) [42] and T5-small decoder [30]. CLIP weights are frozen throughout; only the RQ codebooks and T5 decoder parameters are trained. All training uses a single node with 4 NVIDIA RTX 4090 GPUs (24GB each) and PyTorch DDP. Unless otherwise specified, results on the nine multimodal retrieval tasks are averaged over 5 random seeds.

Datasets. Tab. 20 summarizes the 9 dataset–task pairs from the M-BEIR benchmark [42] used in our evaluation. Each task defines a query modality, a target modality, and a task-specific candidate pool. Pool sizes range from 1K (Flickr30k) to 612K (InfoSeek), spanning three orders of magnitude.

Table 20: Dataset statistics. All splits follow the M-BEIR benchmark [42]. Pool sizes refer to the task-specific test candidate pool.

Dataset	Task	Query	Target	Train	Test	Pool
COCO	$t \to i$	text	image	100K	24.8K	5.0K
Flickr30k	$t \to i$	text	image	145K	5.0K	1.0K
COCO	$i \to t$	image	text	113K	5.0K	24.8K
NIGHTS	$i \to i$	image	image	16K	2.1K	40K
WebQA	$t \to t$	text	text	16K	2.5K	544K
WebQA	$t \to it$	text	image+text	17K	2.5K	403K
CIRR	$it \to i$	image+text	image	26K	4.2K	22K
OVEN	$it \to it$	image+text	image+text	305K	14.7K	335K
InfoSeek	$it \to t$	image+text	text	284K	11.3K	612K

D.1 Existing assets and licenses

Tab. 21 and Tab. 22 summarize the existing datasets, pretrained models, and code assets used in our experiments. We use these assets only for research evaluation, cite their original papers or release pages, and do not redistribute raw datasets or pretrained checkpoints in our released code package.

Item codes. Each target item is represented by a sequence of $L{+}1{=}17$ discrete tokens: one modality token ($V_{\text{mod}}{=}3$) followed by $L{=}16$ RQ code tokens with an ascending vocabulary schedule:

$$V_\ell = \begin{cases} 512 & \ell = 0, \dots, 3 \quad \text{(4 levels)}, \\ 1024 & \ell = 4, \dots, 11 \quad \text{(8 levels)}, \\ 2048 & \ell \geq 12 \quad \text{(4 levels)}. \end{cases}$$

The baseline (GENIUS) uses a uniform vocabulary $V{=}4096$ at all 16 levels with the same modality token.

Codebook-learning objective. For an input embedding y quantized by the RQ codebooks, let $r_0 = y$, $z^\ell = \arg\min_v \|r_{\ell-1} - c_{\ell,v}\|^2$, and $r_\ell = r_{\ell-1} - c_{\ell,z^\ell}$. Following the standard RQ objective used in GENIUS [17], the residual-quantization loss is

$$\mathcal{L}_{\text{rq}}(y) \;=\; \sum_{\ell=1}^{L} \big\| r_{\ell-1} - \text{sg}\big(c_{\ell,z^\ell}\big) \big\|_2^2 , \tag{24}$$

Table 21: Dataset assets used in this paper.

Asset	Use	License / terms	Source
M-BEIR	Main multimodal retrieval benchmark	MIT for the benchmark release; underlying media and data follow the original source terms specified by M-BEIR	Wei et al. [42]; Hugging Face
Flickr30k	Additional text-to-image retrieval benchmark	Images are obtained from Flickr and must follow the Flickr Terms of Use; the dataset maintainers state that the data are provided only for non-commercial research and educational purposes	Young et al. [46]; dataset page
Amazon Reviews'23	Beauty, Toys, and Sports categories for the TIGER generalization experiment	No explicit dataset license assigned by the release; made available primarily for research purposes; raw data are not redistributed in our package	Hou et al. [12]; dataset page; license discussion

Table 22: Pretrained model and code assets used in this paper.

Asset	Use	License / terms	Source
UniIR CLIP-SF and BLIP-FF checkpoints	Dense baselines and frozen CLIP-SF encoder for GENIUS and PRO	MIT	Wei et al. [42]; Hugging Face
UniIR codebase	Reference code for M-BEIR data loading and evaluation conventions	MIT	Wei et al. [42]; GitHub
OpenAI CLIP ViT-L/14	Encoder-sensitivity experiment	MIT	Radford et al. [29]; GitHub
T5-small	Autoregressive decoder backbone	Apache-2.0	Raffel et al. [30]; Hugging Face
LLM2CLIP	Encoder-sensitivity experiment	Apache-2.0	Huang et al. [14]; Hugging Face
GENIUS official implementation	Reference implementation for the reproduced GENIUS baseline	MIT	Kim et al. [17]; GitHub

where $\mathrm{sg}(\cdot)$ denotes the stop-gradient operator. In codebook learning, $\mathcal{L}_{\mathrm{rq}}$ is averaged over the query and target embeddings in the batch. The full objective for codebook learning is

$$\mathcal{L}_{\mathrm{cb}} \;=\; \mathcal{L}_{\mathrm{cl}} \;+\; \beta_{\mathrm{rq}}\,\mathcal{L}_{\mathrm{rq}} \;+\; \gamma_{\mathrm{mse}}\,\mathcal{L}_{\mathrm{mse}} \;+\; \lambda\,\mathcal{L}_{\mathrm{rank}}\,, \tag{25}$$

where $\mathcal{L}_{\mathrm{cl}}$ is a bidirectional contrastive loss on pre-quantization embeddings, $\mathcal{L}_{\mathrm{rq}}$ is the residual-quantization loss between residuals and assigned codewords, $\mathcal{L}_{\mathrm{mse}} = \|\hat{q} - \hat{x}_j\|_2^2$ is the MSE between quantized query and target vectors, and $\mathcal{L}_{\mathrm{rank}}$ is the prefix ranking distillation loss (Sec. 4). We set β_{rq}=100, γ_{mse}=100, λ=100.

Codebook-learning schedule. Training runs for 20 epochs with learning rate 10^{-4} (Adam), batch size 512 per GPU (effective batch 2048 with 4 GPUs), and no learning rate warmup. Prefix ranking distillation uses λ_{rank}=100, shared temperature τ=0.05, top-k=128 in-batch candidates, with 10% linear warmup.

Decoder training. The T5-small decoder (60M parameters) is trained for 30 epochs with learning rate 10^{-4} (Adam) and batch size 512. Input query embeddings are projected to a single soft token via a linear layer, then fed to the T5 encoder. The decoder autoregressively generates the 17-token code sequence (modality + 16 RQ tokens), supervised with per-token cross-entropy loss.

Inference. We use trie-constrained beam search and geometric score fusion (Eq. 7) adds a bias ω=10 at each decoding step, computed from the query residual and the current-level codebook. No reranking is applied after beam search.

Hyperparameters. Tab. 23 summarizes all key hyperparameters.

Training overhead. Prefix ranking distillation adds a KL term over in-batch candidates at each of the L levels; no additional forward pass is required. On 4×RTX 4090 GPUs, the codebook-learning stage (Stage 1) takes $\sim$4 min/epoch with distillation versus $\sim$3.8 min/epoch without, an overhead of $\sim$8%. Since decoder training (Stage 2, $\sim$23 min/epoch) is unchanged and dominates wall-clock time,

Table 23: Hyperparameter summary. All results averaged over 5 seeds.

Component	Hyperparameter	Value
Codebook (RQ)	RQ levels L	16
	Vocab schedule V_ℓ	$512/1024/2048$
	Learning rate	10^{-4}
	Training epochs	20
Prefix distillation	λ_{rank}	100
	τ (shared)	0.05
	Top-k candidates	128
	Warmup fraction	10%
Decoder (T5)	Learning rate	10^{-4}
	Training epochs	30
	Batch size (per GPU)	512
Inference	Beam width B	50
	Score fusion ω	10

the end-to-end training overhead is under 1%. Score fusion adds only lightweight inner products per decoding step at inference; end-to-end, PRO increases per-query latency over GENIUS by 3.9–4.9% at B=50 (Tab. 16).

Vocabulary schedule design. The ascending schedule $V_1 \leq V_2 \leq \cdots \leq V_L$ follows directly from the survival bound (Eq. 4): early levels face irreversible pruning, so fewer codewords enlarge the teacher margin m_ℓ (Sec. 4); later levels contribute less to survival risk and benefit from larger vocabularies for reconstruction fidelity. We adopt a three-tier split ($512/1024/2048$) that allocates the smallest vocabulary to the first 4 levels—where survival decay is steepest (Sec. 3)—and the largest to the final 4 levels. The 8 intermediate levels use a moderate size as a transition. This schedule was selected based on the principle above and not tuned exhaustively; Tab. 1 confirms that even coarser ascending schedules consistently outperform uniform and descending alternatives.

Baselines. For IRGen [49] and GRACE [24], we report the results reproduced by GENIUS [17]. For AVG [25] and SemCORE [21], we report the results from SemCORE [21]. We have verified that the test splits and candidate pools used in these works are consistent with those of the M-BEIR benchmark.

D.2 Statistical variability

GENIUS (reproduced) and our method are each trained with 5 independent random seeds. Tabs. 6–8 and Tab. 13 report per-cell mean±std. Across all tasks, the maximum standard deviation is 0.45 (R@1) and 0.36 (R@5); the corresponding 95% confidence interval half-widths ($1.96\,\sigma/\sqrt{5}$) are below 0.40 in all cases. All pairwise comparisons between our full system and GENIUS are statistically significant ($p < 0.01$, two-sample t-test), including the smallest gap (InfoSeek R@1, Δ=1.2, $p < 0.01$). Dense baselines (CLIP-SF, BLIP-FF) use deterministic pretrained weights.

Table 24: Hyperparameter sensitivity (COCO $t\to i$). Default values in **bold**. Each panel varies one parameter with the others fixed.

λ_{rank}	R@1 / R@5	τ	R@1 / R@5	ω	R@1 / R@5
0	41.3 / 70.2	0.01	40.3 / 68.7	0	42.1 / 70.7
1	41.8 / 70.6	0.02	44.6 / 71.9	1	42.4 / 70.9
10	44.7 / 72.4	**0.05**	**46.2 / 73.0**	3	43.9 / 72.3
50	46.5 / 72.1	0.1	45.8 / 72.8	5	45.6 / 72.1
100	**46.2 / 73.0**	0.2	43.1 / 71.0	**10**	**46.2 / 73.0**
200	44.8 / 72.5	0.5	39.7 / 67.4	20	45.5 / 73.4
500	41.5 / 69.3	1.0	38.2 / 66.1	50	42.0 / 70.1

D.3 Hyperparameter sensitivity

Tab. 24 reports Recall@$\{1, 5\}$ on COCO $t \to i$ (single seed) as each key hyperparameter is varied while the others are held at their default values (λ_{rank}=100, τ=0.05, ω=10).

All three parameters exhibit a broad plateau around the chosen operating point. For the distillation strength λ_{rank}, values in $[50, 200]$ yield R@1 within 1.7 points of the best; the dominant effect is whether distillation is enabled at all (λ=0 vs. λ>0: $+5$ R@1). The temperature τ is stable in $[0.05, 0.1]$; extreme values (≤ 0.01 or ≥ 0.5) hurt because the softmax becomes either too peaked or too flat for effective distillation. The fusion weight ω is stable in $[5, 20]$; at ω=0 (pure T5 decoding) R@1 drops by 4 points, while ω=50 over-weights the geometric term and degrades performance.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the indexing-decoding gap, the theoretical results, the PRO framework, and the empirical scope of the paper. The claims are supported by the theoretical analysis in Secs. 3–4 and the experiments in Sec. 6.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a limitations paragraph in Sec. 8, discussing the RQ-based scope, the beam-search decoding assumption, the remaining gap between generative and dense retrieval, the exact-scanning scope of the efficiency comparison, and the frozen-encoder assumption. App. C further discusses non-RQ identifiers and sampling-based decoding.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The main theoretical statements are given in Thms. 1 and 2. Complete proofs and auxiliary results are provided in App. A.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Sec. 5 and App. D describe the datasets, baselines, model components, hyperparameters, training schedules, decoding settings, and random seeds. Our code, including scripts and configuration files, is publicly available via the repository linked in the abstract.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The experiments use public benchmarks and pretrained assets cited in the paper. Our code repository, linked in the abstract, provides scripts for reproducing the main experiments; raw datasets and pretrained checkpoints are not redistributed and should be obtained from their original sources.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Sec. 5 gives the main experimental setup, and App. D provides dataset statistics, item-code construction, training objectives, optimization details, inference settings, and hyperparameters.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: App. D.2 reports results over 5 independent random seeds, per-cell mean$\pm$std, confidence-interval half-widths, and two-sample t-tests for comparisons with GENIUS.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: App. D reports the hardware setup, training schedule, and overhead measurements, including the use of 4 NVIDIA RTX 4090 GPUs and the per-epoch cost of codebook learning and decoder training.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work studies a technical retrieval method on public benchmarks and does not involve human subjects, private data collection, or deployment of a user-facing system. The authors have reviewed and followed the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This work studies a technical indexing and decoding problem in generative retrieval and does not introduce a deployed system, a new dataset, or a user-facing application. Potential societal impacts would mainly arise from downstream retrieval or recommendation deployments rather than from the technical contribution studied here.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk models, scraped datasets, image generators, or other assets that require special safeguards beyond standard research-use licenses and dataset terms.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Existing datasets, pretrained models, and code assets are cited in the paper, and App. D.1 summarizes their licenses and terms of use. Our released code includes reproduction scripts but does not redistribute raw datasets or pretrained checkpoints. Users will be instructed to obtain M-BEIR, Flickr30k, Amazon Reviews'23 resources, pretrained checkpoints, and referenced code assets from their original sources under the corresponding licenses and terms of use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We release our code for reproducing the proposed method and main experiments via the repository linked in the abstract. The released materials contain documentation and scripts, while relying on users to obtain external datasets and checkpoints from their original sources.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: This work does not involve crowdsourcing or human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method and experiments do not use LLMs as an important, original, or non-standard component.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.