Beyond Codebook Uniformity: Triplet-Level Metrics and Triplet-Driven Semantic IDs for Generative Retrieval

Anonymous ACL submission

Abstract

Generative retrieval replaces item IDs with *Semantic IDs* (SIDs) decoded autoregressively by a sequence model. Playing the role of tokenizer, the codebook largely determines the quality of generative retrieval. Existing codebook metrics, however, only characterize the global uniformity of code usage and do not reveal whether an individual SID correctly represents its item or separates it from others. To address this problem, we propose two metrics and a matching training objective. Firstly we build a query-centered triplet dataset (q, a, p, n) from large-scale user behavior. On top of this dataset we define **Overlap**, a prefix-based distance between two SIDs, and **Pairwise Overlap Ranking (POR)**, the fraction of triplets where a codebook places the positive closer to the anchor than the hard negative. Both metrics correlate strongly with downstream Hit Rate in GR. We then introduce the **Differentiable Soft Overlap Loss (DSOL)** to incorporate these two metrics into training and optimize the codebook directly. On a public e-commerce benchmark (ESCI-us) and a large-scale industrial video search dataset, DSOL improves HR and POR over standard SID baselines.

1 Introduction

Generative retrieval (GR) (Wang et al., 2022; Bevilacqua et al., 2022a; Geng et al., 2022) frames retrieval as a sequence-to-sequence problem in which a decoder autoregressively emits the identifier of the target item, and it is becoming a practical alternative to the ANN-based two-tower paradigm (Tay et al., 2022; Bevilacqua et al., 2022b). Nowadays, mainstream GR methods replaces arbitrary integer IDs with *Semantic IDs* (SIDs), tuples of discrete codes produced by a residual-quantization variational autoencoder (RQ-VAE) (Bevilacqua et al., 2022b; Zeghidour et al., 2021), after which a generative decoder uses beam search to produce candidate document IDs for a given query q. Because the codebook acts as tokenizer that grounds every step of this decoder, it largely determines the quality of GR (Rajput et al., 2023). A growing body of follow-up work targets the codebook directly: some efforts redesign the quantizer itself (Wang et al., 2024; Si et al., 2024; Wan et al., 2026), while others reshape the prefix structure with contrastive signals (Pang et al., 2025; Liu et al., 2026; Hu et al., 2026). Since there are many methods dedicated to improving codebooks, there should also be assessments for codebooks. This raises a consequent question: *how do we measure codebook quality?*

Figure 1: An example of SIDs in video search: clips of two football players (Ronaldo, Messi) share a long SID prefix, while a related but semantically off video about a coach (Guardiola) diverges earlier on its SID.

A widely accepted intuition is that a good codebook should produce SIDs that fully represent the semantics of an item and remain discriminative against other items (Liu et al., 2026). Figure 1 illustrates this intuition with a concrete example. In video search, among the videos relevant to a query

about football players' highlights, match clips of Cristiano Ronaldo and Messi should share a long SID prefix, while a topically related but semantically off video about Guardiola, who is a coach rather than a player, should diverge from them on SID. A useful codebook-quality metric should be able to measure exactly this kind of discrimination.

However, Today's codebook metrics which are frequently used, such as Independent Coding Rate (ICR) (Chen et al., 2025; Kuai et al., 2024) and Gini coefficient (Fu et al., 2025), lack the ability to capture semantic relevance at all. They only measure how evenly the codewords are used and say nothing about whether an individual SID correctly represents the semantics of its item. As a result, the offline numbers reported for a codebook are largely disconnected from the downstream retrieval quality we actually care about.

Building on this observation, we construct a query-centered triplet dataset in which (a, p) are drawn from clicked, high-relevance items under the same query. The hard negative n is drawn from items that appeared under the query but were ignored by users and have low query-item relevance, which supplies hard negatives at no extra cost. On top of this triplet structure we introduce **Overlap**, a prefix-based distance between two SIDs, and **Pairwise Overlap Ranking (POR)**, the fraction of triplets (a, p, n) on which a codebook correctly places p closer to a than n. POR is per-triplet by design and probes the property that matters most for generative retrieval, namely how representative a SID is relative to other candidate items. In the section of experiments,we prove that both POR and the average $\mathrm{Overlap}(a, p)$ track HR@100 strongly, while ICR and Gini show only weak or no correlation with HR. The ranking of codebook methods on POR is also highly correlated with their ranking on HR, so POR can serve as a relative reference signal for retrieval quality and partially obviates expensive end-to-end evaluation.

Finally, we turn Overlap and POR into a training signal. Overlap and POR both rely on hard prefix matching and are not differentiable as written. Building on prior work on differentiable indices (Wang et al., 2026; Fu et al., 2026), we introduce the **Differentiable Soft Overlap Loss (DSOL)**, which relaxes Overlap into a differentiable form and uses a margin loss to align training with POR. On a public e-commerce retrieval benchmark (ESCI-us) and a large-scale industrial video search dataset, DSOL improves both HR and POR over standard codebook-learning baselines. Overall, we make the following contributions:

- We identify that current codebook metrics are global uniformity statistics and do not reflect downstream retrieval quality.
- We propose **Overlap** and **POR**, two triplet-based metrics that probe per-SID representativeness, and show that they correlate strongly with HR@100 across a large tokenizer sweep.
- We propose **DSOL**, a differentiable relaxation of Overlap with a margin loss aligned to POR, which improves both HR and POR over standard SID baselines on ESCI-us and a large-scale industrial video search dataset.

2 Related Work

2.1 Generative Recommendation with Semantic Identifiers

The seminal work in generative recommendation, **TIGER** (Rajput et al., 2023), replaces arbitrary IDs with *Semantic IDs* (SIDs) that encode item semantics hierarchically. A seq2seq model is then trained to autoregressively generate the next item's SID via beam search over the code vocabulary. Subsequent work improves the codebook along different axes. A kind of work refines the *quantizer architecture and regularization* (Si et al., 2024; Liu et al., 2026; Wan et al., 2026; Chen et al., 2025), by injecting collaborative signals, tree structure, category priors, or reference-vector guidance into the RQ-VAE objective to improve codebook utilization and semantic cohesion. Another kind of work brings *contrastive learning into codebook training* (Hu et al., 2026; Pang et al., 2025; Wang et al., 2024), using either Hamming distance (Norouzi et al., 2012) or InfoNCE (Rusak et al., 2025) to align semantic IDs. Both adopt a *pairwise* formulation that pulls a positive together and pushes other items apart. There are also some methods that attempt to further separate the SID to avoid collisions, using methods such as OPQ (Ge et al., 2013; Zhao et al., 2026) and FSQ (Mentzer et al., 2024; Xia et al., 2026) to open up the codebook space.

These work concentrating on the codebook itself reflects a common premise: **the quality of the SID codebook largely determines the quality of generative retrieval**. When the codebook fails to reflect semantic structure, or when too many items collide on the same identifier, downstream retrieval degrades accordingly.

2.2 Metrics for Codebook Quality

Measuring codebook quality is an active line of research. A widely used metric (Chen et al., 2025) is the **Independent Coding Rate (ICR)**, the fraction of items that receive a globally unique SID. Fu et al. (2025) propose the **Gini coefficient** of per-codeword assignment frequencies as a finer-grained indicator of usage imbalance, arguing that a codebook can be evenly utilized yet highly skewed. Hu et al. (2026) report *SID entropy*. There are also some similar metrics, such as collision rate (Li et al., 2025) and codebook utilization rate. All of these metrics, however, characterize only *global* properties of the codebook assignment, i.e. how evenly the codebook is used, or how cleanly clusters separate in the embedding space. None of them directly ask whether an individual SID *represents* its item and *distinguishes* it from other semantically similar items, which is the property that ultimately matters for beam-search retrieval.

3 Method

As discussed in Section 2, existing codebook metrics characterize only global properties of the codebook and offer little guidance toward the downstream retrieval metric. We fill this gap in two steps: we first define a principled way to evaluate codebook quality, and then convert that evaluation signal into a training objective. An overview of the full pipeline is shown in Figure 2.

First of all, we construct a query-centered triplet dataset that supplies ground-truth notions of "similar" and "dissimilar" items for any query. On top of these triplets we define two lightweight metrics — **Overlap**, a prefix-based distance between two SIDs, and **POR**, which aggregates Overlap values into a pairwise ranking indicator. Finally, taking direct optimization of Overlap and POR as our design principle, we derive the **Differentiable Soft Overlap Loss (DSOL)**, a fully differentiable objective whose optimization improves both metrics above and Hit Rate.

3.1 Triplet Dataset Construction

To evaluate whether a SID correctly represents the semantics of its item, we need pairs of items whose relative similarity is known. We obtain such supervision from the large-scale query log of an *industrial video search* platform. For each query q we sample an *anchor* item a that is clicked by user under q with high query–item relevance. We then sample a *positive* item p that is also clicked under the same q and is also highly relevant. By construction, p is semantically close to a. We sample a *negative* item n that appears under q but is not clicked and has low query–item relevance. Because n shares the same query context as a and p, it is topically related in a broad sense, yet it differs from a in the fine-grained semantics that led the user to ignore it. This yields a triplet dataset

$$\mathcal{D} = \{(q_i,\, a_i,\, p_i,\, n_i)\}_{i=1}^{N},$$

in which the pair (a, p) should be closer than (a, n) in any reasonable semantic space.

For the public ESCI-us benchmark, where user clicks are unavailable, we take the role of "clicks" from the `Exact` relevance label and the role of "low-relevance negatives" from `Substitute` or `Complement` items under the same query; this preserves the same structure — positives genuinely answer the query, hard negatives are topically relevant substitutes that do not.

3.2 Overlap: Measuring SID Similarity

Given a SID of length L, written as $\mathbf{c}(x) = (c_0(x), c_1(x), \ldots, c_{L-1}(x))$, we measure the similarity between two SIDs by the length of their shared prefix. Since a SID is generated in a coarse-to-fine manner, agreement on early tokens indicates agreement on broad semantic category, while agreement on later tokens indicates agreement on finer distinctions. Two items that share no prefix should be considered far apart in the SID space, regardless of how many later tokens happen to match.

Formally, we define the *prefix length* of a pair of items A and B as the largest ℓ such that the first ℓ tokens agree:

$$\ell(A, B) = \max\big\{\, \ell \in \{0, 1, \ldots, L\} \,\big|\, c_d(A) = c_d(B),\ \forall\, 0 \le d < \ell \,\big\}. \tag{1}$$

The **Overlap** between two SIDs is the prefix length normalized by the SID length:

$$\mathrm{Overlap}(A, B) = \frac{\ell(A, B)}{L}, \tag{2}$$

which takes values in $\{0,\ 1/L,\ 2/L,\ \ldots,\ 1\}$.

This design matches the hierarchical nature of residual quantization: a broken prefix implies that two items were routed into different regions at the coarsest stage of quantization, after which any later-stage coincidence is semantically meaningless. It also matches beam search: once a beam commits to the wrong first token, no amount of matching at later steps can recover the correct item.

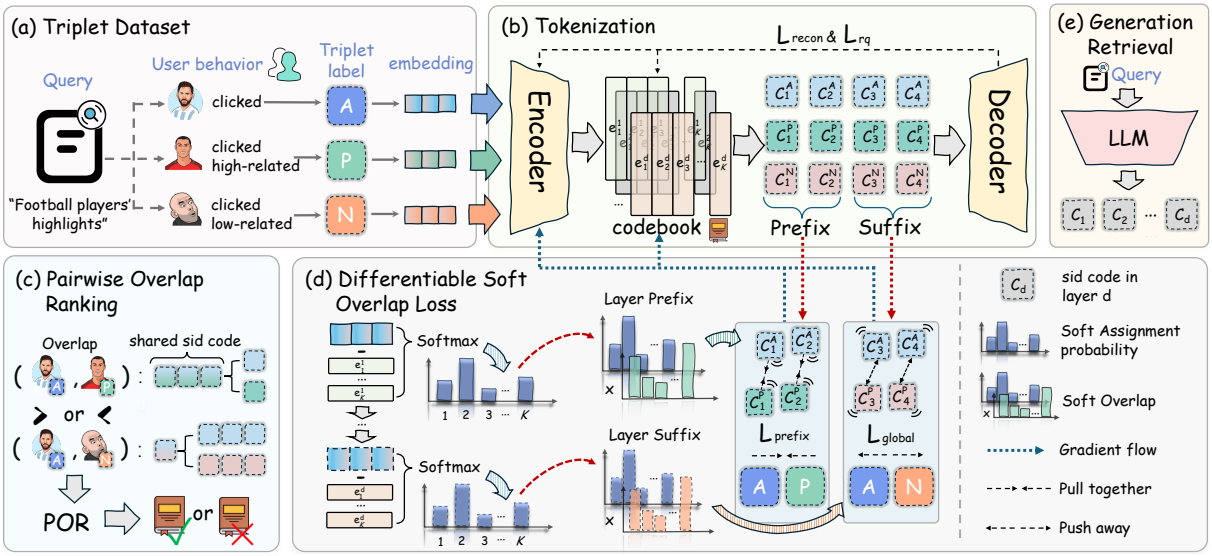


Figure 2: Overview of our pipeline. From a query-centered triplet dataset, we define **Overlap** as the prefix length between two SIDs and aggregate it into **POR**, a pairwise ranking indicator over (a, p, n) triplets. The **Differentiable Soft Overlap Loss (DSOL)** relaxes Overlap into a differentiable objective and trains the SID tokenizer toward higher POR, which directly transfers to downstream Hit Rate.

3.3 POR: Pairwise Overlap Ranking

Given the raw distance between two SIDs, we can now reason about the geometric relations *among* the three members of a triplet. For a single triplet (a, p, n), a good codebook should satisfy $\text{Overlap}(a, p) > \text{Overlap}(a, n)$: this comparative relation is precisely the way we quantify how much genuine semantic information the reconstructed SIDs carry about their items. We now turn this intuition into a formal indicator.

Concretely, for a triplet $(a, p, n) \in \mathcal{D}$ we define the indicator

$$\mathbb{I}_{>}(a, p, n) = \begin{cases} 1, & \text{if } \text{Overlap}(a, p) \\ & \quad > \text{Overlap}(a, n), \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The **Pairwise Overlap Ranking (POR)** of a codebook is the expectation of this indicator over the dataset:

$$\text{POR}(\mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{t \in \mathcal{D}} \mathbb{I}_{>}(t) \in [0, 1], \tag{4}$$

where $t = (a, p, n)$ denotes a triplet in $\mathcal{D}$. POR answers a precise question: *on what fraction of triplets does the codebook correctly place the positive closer to the anchor than the negative?* Unlike ICR or Gini, which measure only global codebook statistics, POR probes whether an individual SID represents its item and separates it from similar but less relevant competitors. Alongside POR, we also track the Overlap averages $\mathbb{E}[\text{Overlap}(a, p)]$, which quantify how tightly positive pairs are pulled together. Subsequently we show empirically that both POR and $\mathbb{E}[\text{Overlap}(a, p)]$ correlate strongly with downstream Hit Rate, confirming that they capture the codebook property most relevant to generative retrieval.

3.4 Preliminaries: Reconstruction Losses in Generative Retrieval

Standard generative retrieval tokenizers (Rajput et al., 2023) encode an item into a latent $\mathbf{z} = \text{Encoder}(\mathbf{x}) \in \mathbb{R}^D$ and apply an L-layer residual quantizer to produce a discrete code tuple $(c_0, \ldots, c_{L-1})$. At layer d, given residual $\mathbf{r}^{(d)}$ (with $\mathbf{r}^{(0)} = \mathbf{z}$) and codebook $\{\mathbf{e}_1^{(d)}, \ldots, \mathbf{e}_K^{(d)}\} \subset \mathbb{R}^D$, the quantizer picks the nearest codeword and passes the remainder to the next layer:

$$\hat{k}^{(d)} = \arg\min_k \|\mathbf{r}^{(d)} - \mathbf{e}_k^{(d)}\|^2, \tag{5}$$

$$\mathbf{r}^{(d+1)} = \mathbf{r}^{(d)} - \mathbf{e}_{\hat{k}^{(d)}}^{(d)}. \tag{6}$$

The quantized representation $\hat{\mathbf{z}} = \sum_{d=0}^{L-1} \mathbf{e}_{\hat{k}^{(d)}}^{(d)}$ approximates $\mathbf{z}$, and a decoder $D(\cdot)$ reconstructs the original input from $\hat{\mathbf{z}}$. The tokenizer is trained end-to-end with a reconstruction loss

$$\mathcal{L}_{\text{recon}} = \|\mathbf{x} - D(\hat{\mathbf{z}})\|_2^2, \tag{7}$$

together with a residual-quantization loss

$$\mathcal{L}_{\text{rq}} = \sum_{d=0}^{L-1} \Big(\|\text{sg}[\mathbf{r}^{(d)}] - \mathbf{e}^{(d)}_{\hat{k}^{(d)}}\|_2^2 + \eta \, \|\mathbf{r}^{(d)} - \text{sg}[\mathbf{e}^{(d)}_{\hat{k}^{(d)}}]\|_2^2 \Big), \quad (8)$$

where $\text{sg}[\cdot]$ is the stop-gradient operator. The two terms inside $\mathcal{L}_{\text{rq}}$ are the standard codebook loss and commitment loss inherited from the VQ-VAE family, balanced by η.

3.5 Differentiable Soft Overlap Loss

Having defined POR and Overlap as evaluation indicators, we consequently try to make them participate in training and directly optimize these two. The obstacle is that both quantities are discrete: the prefix-length function in Eq. 1 and the pairwise comparison in Eq. 3 are non-differentiable, so neither can serve as a gradient signal. Inspired by previous works on Gumbel softmax (Jang et al., 2017), we therefore introduce an end-to-end differentiable continuous relaxation of Overlap, on top of which we build a training objective the **Differentiable Soft Overlap Loss (DSOL)**, casting the Pairwise Overlap Ranking relation itself into a margin-based loss so that the triplet-level ordering drives every gradient step.

Soft Assignment. Instead of the hard assignment in Eq. 5, we define a soft assignment probability over codebook entries at layer d:

$$p_k^{(d)}(\mathbf{r}) = \frac{\exp\Big(-\|\mathbf{r}^{(d)} - \mathbf{e}_k^{(d)}\|^2/\tau\Big)}{\sum_{j=1}^{K} \exp\Big(-\|\mathbf{r}^{(d)} - \mathbf{e}_j^{(d)}\|^2/\tau\Big)}, \quad (9)$$

where $\tau > 0$ is a temperature parameter.

Layer-wise Soft Overlap. For two items A and B with residual vectors $\mathbf{r}_A^{(d)}$ and $\mathbf{r}_B^{(d)}$ at layer d, their *soft overlap* is defined as the inner product of their soft assignment distributions:

$$\text{Overlap}_d(A, B) = \sum_{k=1}^{K} p_k^{(d)}(\mathbf{r}_A) \cdot p_k^{(d)}(\mathbf{r}_B). \quad (10)$$

This quantity lies in $(0, 1]$: it approaches 1 when both items are assigned to the same codebook entry, and approaches 0 when they are assigned to entirely different entries. Crucially, this construction makes the Overlap computation end-to-end differentiable with respect to the codebook.

Global Margin Ranking Loss. We define a weighted global overlap that emphasizes suffix layers:

$$\widetilde{\text{Overlap}}(A, B) = \frac{\sum_{d=0}^{L-1} w_d \cdot \text{Overlap}_d(A, B)}{\sum_{d=0}^{L-1} w_d}, \quad (11)$$

with layer weights $w_0 = w_1 = 0$, $w_2 = w_3 = 1$. The zero weights on prefix layers ensure that this loss does not push negatives away in early layers, allowing negatives to naturally maintain prefix consistency with the anchor. The global margin ranking loss is:

$$\mathcal{L}_{\text{global}} = \mathbb{E}\big[\max\big(0, m + \widetilde{\text{Overlap}}(a, n) - \widetilde{\text{Overlap}}(a, p)\big)\big], \quad (12)$$

where m is the margin hyperparameter.

Prefix Alignment Loss. We add a complementary *pull-only* prefix loss whose sole job is to pull both (a, p) and (a, n) toward the same coarse buckets at the first two prefix layers. Treating each layer overlap as a Bernoulli success probability and reusing the layer weighting of Eq. 11:

$$\mathcal{L}_{\text{prefix}} = -\textstyle\sum_d \tilde{w}_d \, \mathbb{E}\big[\log \text{Overlap}_d(a, p) + \log \text{Overlap}_d(a, n)\big], \quad (13)$$

with normalized layer weights $\tilde{w}_d = w_d / \sum_{d'} w_{d'}$ and $w_0 = w_1 = 1$, $w_2 = w_3 = 0$. The zero weights restrict $\mathcal{L}_{\text{prefix}}$ to suffix layers 0–1. The $-\log$ form gives large gradients at small overlaps and saturates as overlap approaches one. With $\mathcal{L}_{\text{prefix}}$ on layers 0–1 and $\mathcal{L}_{\text{global}}$ on layers 2–3, prefix alignment and suffix separation act on disjoint depth ranges.

Combined SOL Objective. The full Soft Overlap Loss combines the reconstruction terms with the two complementary triplet terms:

$$\mathcal{L}_{\text{SOL}} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{rq}} + \alpha \cdot \mathcal{L}_{\text{global}} + \beta \cdot \mathcal{L}_{\text{prefix}}, \quad (14)$$

where $\mathcal{L}_{\text{global}}$ pushes negatives away at the suffix layers and $\mathcal{L}_{\text{prefix}}$ aligns positives at the prefix layers, with α and β scaling their contributions on top of the standard reconstruction objective. The downstream generative retrieval model is then trained in the usual way on the frozen SIDs.

4 Experiments

We design our experiments around three questions. (**Q1**) Does DSOL improve downstream generative

Table 1: Performance comparison of different retrieval methods on two datasets. HR@k denotes Hit Rate at k and MRR@k denotes Mean Reciprocal Rank at k. ICR and POR are codebook-quality metrics and only apply to SID-based methods. Bold indicates the best result and underline indicates the second-best result. Dashes indicate metrics could not computed in the current run. The results of the table are obtained from multiple runs.

Model	Industrial						ESCI-us					
	HR@10	HR@100	MRR@10	MRR@100	ICR(%)	POR(%)	HR@10	HR@100	MRR@10	MRR@100	ICR(%)	POR(%)
BM25	0.153	0.191	0.085	0.097	–	–	0.255	0.405	0.148	0.157	–	–
DPR	0.126	0.162	0.069	0.084	–	–	<u>0.318</u>	<u>0.482</u>	<u>0.158</u>	<u>0.169</u>	–	–
RQ-VAE	0.113	0.149	0.059	0.081	<u>71.32</u>	30.58	0.166	0.237	0.087	0.093	69.68	30.41
RQ-kmeans	0.128	0.157	0.071	0.084	70.81	32.26	0.231	0.291	0.136	0.140	71.23	31.52
K-means	0.146	0.187	0.083	0.098	**72.73**	<u>34.87</u>	0.245	0.306	0.142	0.146	<u>74.72</u>	32.43
RQ-OPQ	0.145	0.181	<u>0.086</u>	0.101	68.02	33.92	0.183	0.267	0.098	0.102	**76.36**	19.82
CRQ-VAE	<u>0.151</u>	<u>0.196</u>	0.079	<u>0.103</u>	61.32	34.51	0.314	0.451	0.157	0.162	62.83	<u>33.02</u>
DSOL (Ours)	**0.166**	**0.210**	**0.093**	**0.108**	66.70	**35.31**	**0.337**	**0.495**	**0.172**	**0.179**	73.39	**33.96**

retrieval relative to standard SID baselines on both public and industrial datasets? (**Q2**) Is POR a better metric for codebook quality than the global uniformity metrics (ICR, Gini) used today, i.e., does it track HR more reliably? (**Q3**) Do the individual components of DSOL (the prefix alignment loss and the global margin loss) contribute to its final performance?

4.1 Experimental Setup

Datasets. We evaluate on two retrieval corpora chosen to cover the public e-commerce and industrial search settings. **ESCI-us** (Reddy et al., 2022) is the US slice of the Amazon ESCI benchmark with ∼482k query–item pairs and four relevance grades (*Exact*, *Substitute*, *Complement*, *Irrelevant*). For triplet construction we take Exact items as the anchor/positive pool and Substitute/Complement items under the same query as hard negatives. **Industrial** is an industrial-scale search log from a commercial search platform; triplets are built from user clicks and a calibrated relevance score. Both datasets are tokenized with SIDs of length $L = 4$ and codebook size $K = 256$.

Baselines. We compare against a ladder of retrievers, organized by paradigm: (1) **Sparse** — BM25 (Robertson and Zaragoza, 2009) over the item text fields; (2) **Dense** — DPR (Karpukhin et al., 2020) with FAISS search (Johnson et al., 2019); (3) **Quantization tokenizers** — **RQ-VAE** (Rajput et al., 2023) (the TIGER tokenizer), **K-means** (Bradley et al., 2000; Luo et al., 2025) with residual clustering, and **RQ-OPQ** (Chen et al., 2025); (4) **SID improvements** — **CRQ-VAE** (Pang et al., 2025), a cluster-regularized residual quantizer trained with contrastive supervision. For all quantization-based methods we fix the downstream generative model and only replace the SID tokenizer, which isolates the contribution of codebook quality.

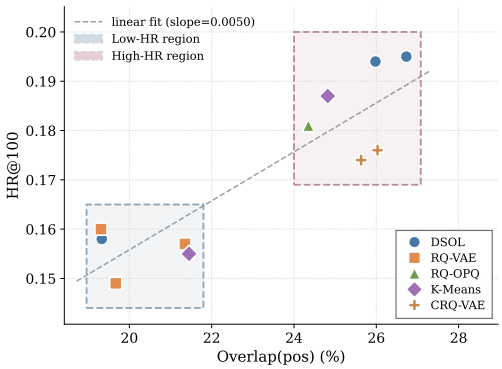
Generative backbone and training. We use a decoder-only Qwen3-0.6B (Yang et al., 2025) as the GR backbone on Industrial and an MT5-base (Xue et al., 2021) on ESCI-us. Training uses AdamW with learning rate 10^{-4}, batch size 32–128 (dataset dependent) and 30 epochs.

Metrics. We report Hit Rate at $\{10, 100\}$ (HR@k) and Mean Reciprocal Rank at $\{10, 100\}$ (MRR@k) on the SID generation task, together with the codebook-quality metrics $\mathrm{Overlap}(a, p)$ and POR. For the correlation study (Section 4.3) we additionally report ICR and the Gini coefficient to demonstrate the contrast.

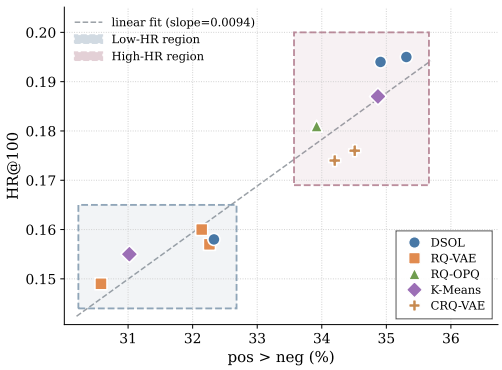
4.2 Main Results

Table 1 reports HR@k, MRR@k, ICR, and POR on Industrial and ESCI-us. Three patterns stand out. First, against every generative retrieval baseline as well as BM25 and DPR, DSOL attains the best HR on both datasets: HR@100 of 0.210 on Industrial and 0.495 on ESCI-us, which shows that our codebook optimization is effective. DSOL also leads on MRR@10 and MRR@100 on both datasets. Second, DSOL achieves the highest POR among SID tokenizers (35.31% on Industrial, 33.96% on ESCI-us), confirming that the loss does optimize the pairwise overlap ranking it was designed for. We further quantify the correlation between POR and HR in Section 4.3.

The margin between DSOL and CRQ-VAE deserves a closer look, since both methods apply some form of contrastive supervision to the codebook. CRQ-VAE lacks an explicit suffix-separation mechanism: its ICR is markedly lower (61.32 on Industrial), suffix codes are reused more often, dis-



(a) Overlap(a, p) vs. HR@100



(b) POR vs. HR@100

Figure 3: POR-family metrics track HR@100 with a single near-linear trend: $r = 0.89$ for Overlap(a, p) and $r = 0.95$ for POR.

tinct items collide on the same SID, and retrieval performance drops accordingly. DSOL avoids this by combining prefix alignment with an explicit suffix margin loss, which helps to yield both higher ICR and higher HR.

4.3 Analysis of Codebook Metrics

To address Q2, we sweep several SID tokenizers (RQ-VAE, K-means, CRQ-VAE, RQ-OPQ and DSOL) on the same corpus with identical downstream training and record the codebook metrics together with HR@100. Figure 3 shows that the two POR-family metrics track HR@100 along a single monotone, near-linear trend, with Pearson $r = 0.89$ for Overlap(a, p) and $r = 0.95$ for POR. In contrast, the global uniformity metrics in Table 2 are uninformative ($|r| < 0.35$ for both ICR and Gini): they describe how evenly the codebook is used, but not whether a given SID is representative of its item. We will display the sample points of Gini and ICR in the Appendix to show their trends with HR@100.

Table 2: Pearson correlation between codebook metrics and downstream HR@100. POR-family metrics are far better aligned with retrieval quality than global uniformity metrics.

Metric	Pearson r vs. HR@100
ICR	-0.33
Gini	0.24
Overlap(a, p)	**0.89**
POR	**0.95**

Table 3: Ablation of DSOL components on Industrial. HR@100 is the downstream retrieval metric; POR is the triplet-ordering metric DSOL is designed to optimize.

Variant	HR@100	POR
RQ-VAE (baseline)	0.149	30.6%
+ $\mathcal{L}_{\text{global}}$	0.145	30.2%
+ $\mathcal{L}_{\text{prefix}}$	0.193	34.1%
DSOL (full)	**0.210**	**35.3%**

4.4 Ablation Study

Table 3 adds DSOL's two terms one at a time on top of an RQ-VAE baseline. Starting from baseline, adding $\mathcal{L}_{\text{global}}$ slightly lowers both HR and POR (0.145, 30.2%): on its own, the suffix-only push pulls codewords apart and gives no compensating prefix alignment. Adding $\mathcal{L}_{\text{prefix}}$, in contrast, raises both metrics (0.193, 34.1%), since constraining the prefix keeps semantically similar items in the same coarse bucket. Full DSOL combines the two and reaches the highest POR and HR@100. The two terms thus play complementary roles, with $\mathcal{L}_{\text{global}}$ providing the suffix-level separation and $\mathcal{L}_{\text{prefix}}$ providing the prefix-level alignment.

4.5 Qualitative Case Study

For one high-volume query from the industrial-search log—*"Peppa Pig Complete Collection (Chinese version)"*—we compare the SIDs that DSOL and RQ-VAE assign to seven returned videos (Table 4). Under DSOL all SIDs share the prefix `A_77, B_71`. In contrast, under RQ-VAE the videos scatter across four first-layer codes, and even the four sharing `A_2` diverge at the second layer. The same gap shows up at the corpus level (Figure 4): DSOL's `A_77` pre-image consists of children-animation tokens (*Peppa*, *cartoon*, *kids*), whereas RQ-VAE's `A_2` pre-image mixes the cartoon head with off-topic words highlighted in red (e.g., game- and horror-related), which we manually marked after the run as semantically inconsistent with the query intent. The numerical gap

Table 4: SIDs assigned by DSOL and RQ-VAE to distinct videos retrieved under the same query *Peppa Pig Complete Collection (Chinese version)* on Industrial. Tokens in red (DSOL) and blue (RQ-VAE) mark prefix layers on which all listed items share the same code.

Q: Peppa Pig Complete Collection (Chinese version)	
DSOL (Ours)	**RQ-VAE**
<A_77,B_71,C_137,D_208>	<A_2,B_68,C_246,D_218>
<A_77,B_71,C_191,D_248>	<A_2,B_223,C_250,D_117>
<A_77,B_71,C_239,D_115>	<A_114,B_196,C_23,D_102>
<A_77,B_71,C_171,D_84>	<A_247,B_21,C_195,D_218>
<A_77,B_71,C_239,D_166>	<A_151,B_89,C_170,D_218>
<A_77,B_71,C_129,D_119>	<A_2,B_68,C_50,D_218>
<A_77,B_71,C_137,D_8>	<A_2,B_63,C_103,D_218>

(a) DSOL — A_77 (b) RQ-VAE — A_2

Figure 4: Word clouds of the dominant first-layer prefix under the same query. Red words in (b) are annotations for tokens semantically off the query intent.

measured by $\mathrm{Overlap}(a, p)$ and HR@100 thus has a clean linguistic correlate: DSOL's prefix codes carry query intent and RQ-VAE's do not.

4.6 Hyperparameter Sensitivity

We sweep the two SOL coefficients α and β over a 5×5 grid in $\{0.5, 0.2, 0.1, 10^{-2}, 10^{-3}\}$ on Industrial and report HR@100 in Figure 5. Across the broad central region (any α paired with $\beta \in \{10^{-2}, 10^{-1}\}$), HR@100 stays within 19.1–21.0, indicating that DSOL is largely insensitive to either coefficient. Performance only drops sharply at $\beta = 0.5$, where HR@100 falls to about 14.5 because an overweighted prefix loss collapses the codebook into a single coarse bucket. $\beta = 10^{-3}$ is also slightly worse since the prefix signal becomes too weak. The plateau around $(\alpha, \beta) = (0.2, 10^{-2})$ confirms that DSOL is robust within a sensible range.

4.7 Cost of POR Evaluation vs. Downstream SFT

POR is designed as a cheap proxy that tells us whether a codebook is worth pushing through SFT. Table 5 compares the cost of computing POR with that of a full SFT round on Qwen3-0.6B (8×H20 96G). Compared with SFT, POR adds only about

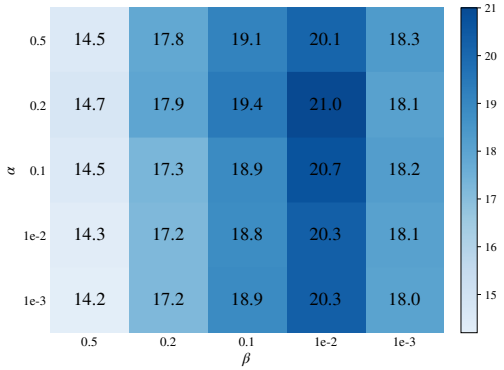


Figure 5: Sensitivity analysis of the $\mathcal{L}_{\text{global}}$ weight α and the $\mathcal{L}_{\text{prefix}}$ weight β.

1.8 GB of disk for the triplet embedding cache and 4.8 MB for the docid triplet table. Because POR only compares the longest matching prefix among the three triplet members, the computation is extremely lightweight. On the 629,736-triplet Industrial set it finishes in roughly 10 seconds with no GPU usage, while SFT takes 118 minutes to converge. This asymmetry matters when sweeping codebook sizes, layer counts, loss variants, and seeds. Using POR as a pre-filter, where SFT runs only on candidates above a POR threshold, lets compute concentrate on the candidates with the highest expected downstream gain.

Table 5: Resource cost of POR vs. SFT (8×H20 96G).

Stage	Extra Disk	Wall Time
POR	1.8 GB emb. + 4.8 MB triplets	$\sim$ 10 s (CPU)
SFT	none	118 min

5 Conclusion

We argued that codebook quality for generative retrieval should be measured by a per-triplet ranking indicator rather than a global uniformity statistic. On top of a prefix-based Overlap distance we proposed POR, which correlates strongly with downstream HR (Pearson $r \approx 0.9$) where ICR and Gini do not. Turning POR into a training objective gave DSOL, a differentiable loss with asymmetric prefix-pull and suffix-push constraints. DSOL improves HR and POR over the strongest SID baseline on three datasets, showing that the metric and the loss optimize the same property. The metric-as-loss recipe of defining what matters per item and then deriving its continuous relaxation should generalize to other generative-retrieval problems where discrete structure meets ranking objectives.

Limitations

Firstly, we do not yet report online A/B test results: the online serving pipeline is not fully ready in time for this submission, and we plan to include a live A/B comparison in the camera-ready version. Additionally, our triplet construction relies on both an interaction signal (e.g., clicks) and a query–item relevance signal. It therefore generalizes naturally to any industrial setting that exposes the two signals together, but it does not directly cover datasets that provide neither. In such cold-supervision scenarios, building reliable triplets is itself an open problem that we leave to future work. Moreover, the layer weights inside $\mathcal{L}_{\text{prefix}}$ and $\mathcal{L}_{\text{global}}$ in Eqs. 11 and 13 are set by hand to 0 or 1 to enforce the prefix/suffix split. This is a deliberately simple choice; the broader weighting space (graded or learned per-layer weights, asymmetric prefix/suffix budgets, etc.) is left to future work.

References

Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Scott Yih, Sebastian Riedel, and Fabio Petroni. 2022a. Autoregressive search engines: Generating substrings as document identifiers. *Advances in Neural Information Processing Systems*, 35:31668–31683.

Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Wen-tau Yih, Sebastian Riedel, and Fabio Petroni. 2022b. Autoregressive entity retrieval. In *International Conference on Learning Representations*.

Paul S Bradley, Kristin P Bennett, and Ayhan Demiriz. 2000. Constrained k-means clustering. *Microsoft Research, Redmond*, 20(0):0.

Zheng Chen and 1 others. 2025. OneSearch: A preliminary exploration of the unified end-to-end generative framework for E-commerce search. In *Companion Proceedings of the ACM Web Conference*.

Junchen Fu, Xuri Ge, Alexandros Karatzoglou, Ioannis Arapakis, Suzan Verberne, Joemon M. Jose, and Zhaochun Ren. 2026. Differentiable semantic id for generative recommendation. *Preprint*, arXiv:2601.19711.

Kairui Fu, Tao Zhang, Shuwen Xiao, Ziyang Wang, Xinming Zhang, Chenchi Zhang, Yuliang Yan, Junjun Zheng, Yu Li, Zhihong Chen, Jian Wu, Xiangheng Kong, Shengyu Zhang, Kun Kuang, Yuning Jiang, and Bo Zheng. 2025. Forge: Forming semantic identifiers for generative retrieval in industrial datasets. *Preprint*, arXiv:2509.20904.

Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. 2013. Optimized product quantization for approximate nearest neighbor search. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2946–2953.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM conference on recommender systems*, pages 299–315.

Zheng Hu, Yuxin Chen, Yongsen Pan, Xu Yuan, Yuting Yin, Daoyuan Wang, Boyang Xia, Zefei Luo, Hongyang Wang, Songhao Ni, Dongxu Liang, Jun Wang, Shimin Cai, Tao Zhou, Fuji Ren, and Wenwu Ou. 2026. Stop treating collisions equally: Qualification-aware semantic id learning for recommendation at industrial scale. *Preprint*, arXiv:2603.00632.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations (ICLR)*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE transactions on big data*, 7(3):535–547.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 6769–6781.

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. 2024. Breaking the hourglass phenomenon of residual quantization: Enhancing the upper bound of generative retrieval. *Preprint*, arXiv:2407.21488.

Guanghan Li, Xun Zhang, Yufei Zhang, Yifan Yin, Guojun Yin, and Wei Lin. 2025. Semantic convergence: Harmonizing recommender systems via two-stage alignment and behavioral semantic tokenization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 12040–12048.

Xiaoyu Liu, Fuwei Zhang, Yiqing Wu, Xinyu Jia, Zenghua Xia, Fuzhen Zhuang, Zhao Zhang, Fei Jiang, and Wei Lin. 2026. Cat-id2: Category-tree integrated document identifier learning for generative retrieval in e-commerce. In *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining*, WSDM '26, page 426–435, New York, NY, USA. Association for Computing Machinery.

Xinchen Luo, Jiangxia Cao, Tianyu Sun, Jinkai Yu, Rui Huang, Wei Yuan, Hezheng Lin, Yichen Zheng, Shiyao Wang, Qigen Hu, and 1 others. 2025. Qarm: Quantitative alignment multi-modal recommendation at kuaishou. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 5915–5922.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. 2024. Finite scalar quantization: Vq-vae made simple. In *International Conference on Learning Representations*, volume 2024, pages 51772–51783.

Mohammad Norouzi, David J Fleet, and Russ R Salakhutdinov. 2012. Hamming distance metric learning. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Yunsheng Pang, Zijian Liu, Yudong Li, Shaojie Zhu, Zijian Luo, Chenyun Yu, Sikai Wu, Shichen Shen, Cong Xu, Bin Wang, and 1 others. 2025. HiGR: Efficient generative slate recommendation via hierarchical planning and multi-objective preference alignment. *arXiv preprint arXiv:2512.24787*.

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

Chandan K. Reddy, Lluís Màrquez, Fran Valero, Nikhil Rao, Hugo Zaragoza, Sambaran Bandyopadhyay, Arnab Biswas, Anlu Xing, and Karthik Subbian. 2022. Shopping queries dataset: A large-scale ESCI benchmark for improving product search. *Preprint*, arXiv:2206.06588.

Stephen Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond*, volume 4. Now Publishers Inc.

Evgenia Rusak, Patrik Reizinger, Attila Juhos, Oliver Bringmann, Roland S. Zimmermann, and Wieland Brendel. 2025. Infonce: Identifying the gap between theory and practice. *Preprint*, arXiv:2407.00143.

Zihua Si, Zhongxiang Sun, Jiale Chen, Guozhang Chen, Xiaoxue Zang, Kai Zheng, Yang Song, Xiao Zhang, Jun Xu, and Kun Gai. 2024. Generative retrieval with semantic tree-structured item identifiers via contrastive learning. *Preprint*, arXiv:2309.13375.

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

Qiang Wan, Ze Yang, Dawei Yang, Ying Fan, Xin Yan, Siyang Liu, Yicong Liu, Chenwei Zhang, Wei Xu, Jiahao Qin, and Ke Wang. 2026. R3-vae: Reference vector-guided rating residual quantization vae for generative recommendation. *Preprint*, arXiv:2604.11440.

Wenjie Wang, Honghui Bao, Xinyu Lin, Jizhi Zhang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. 2024. Learnable item tokenization for generative recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM)*, pages 2400–2409.

Xujing Wang, Yufeng Chen, Boxuan Zhang, Jie Zhao, Chao Wei, Cai Xu, Ziyu Guan, Wei Zhao, Weiru Zhang, and Xiaoyi Zeng. 2026. Differentiable geometric indexing for end-to-end generative retrieval. *Preprint*, arXiv:2603.10409.

Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao, Shibin Wu, Qi Chen, Yuqing Xia, Chengmin Chi, Guoshuai Zhao, Zheng Liu, and 1 others. 2022. A neural corpus indexer for document retrieval. *Advances in Neural Information Processing Systems*, 35:25600–25614.

Tian Xia, Jiaqi Zhang, Yueyang Liu, Hongjian Dou, Tingya Yin, Jiangxia Cao, Xulei Liang, Tianlu Xie, Lihao Liu, Xiang Chen, Shen Wang, Changxin Lao, Haixiang Gan, Jinkai Yu, Keting Cen, Lu Hao, Xu Zhang, Qiqiang Zhong, Zhongbo Sun, and 9 others. 2026. Qarm v2: Quantitative alignment multimodal recommendation for reasoning user sequence modeling. *Preprint*, arXiv:2602.08559.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. *Preprint*, arXiv:2010.11934.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. 2021. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507.

Qihang Zhao, Zhongbo Sun, Xiaoyang Zheng, Xian Guo, Siyuan Wang, Zihan Liang, Mingcan Peng, Ben Chen, and Chenyi Lei. 2026. Coins: Semantic ids enhanced cold item representation for click-through rate prediction in e-commerce search. In *Proceedings of the ACM Web Conference 2026*, pages 8529–8532.

A Triplet Construction Details per Dataset

On Industrial, positives are items with click > 0 and a relevance model score above 0.6 (out of 1) under the anchor's query; negatives are items with zero clicks and a relevance score below 0.35 under the same query; the anchor itself is drawn from the clicked high-relevance pool. On ESCI-us, where click logs are absent, the positive pair uses two items with `Exact` label under the same query, and the negative is drawn from `Irrelevant` or `Complement` items under the same query. We require anchor, positive and negative to be distinct items and filter out duplicates. Because the number of triplets that can be constructed on ESCI-us is large, we randomly sub-sample $50{,}000$ triplets to keep training cost comparable across datasets. The resulting corpus sizes are summarized in Table 6.

Table 6: Statistics of the processed datasets used in this paper. **Triplet** denotes the number of (anchor, positive, negative) triplets used for SID training; on ESCI we randomly sub-sample $50{,}000$ from a much larger pool.

Dataset	Docs	Queries	Triplet
ESCI-us	482,105	277,348	50,000
Industrial	5,406,830	408,031	629,736

B Hyperparameters

We use temperature $\tau = 0.5$, margin $m = 0.2$, and prefix weight $\beta = 1.0$ throughout. These were selected on a held-out validation slice of Industrial and re-used on the other two datasets without re-tuning.

C Effect of Generative Backbone Scale

The main results in Section 4 fix the generative retriever to Qwen3-0.6B so that any HR/MRR change can be attributed to the SID tokenizer alone. To check that DSOL's gain over TIGER is not specific to one backbone, we replace it with Qwen3-0.6B/1.7B/4B and rerun SFT on Industrial; all other settings match the main experiments.

Table 7 indicates that backbone scale has only a limited impact on retrieval quality in this setup, while the tokenizer choice is consistently decisive: at every scale, DSOL outperforms TIGER on both HR@100 and MRR@100.

Table 7: Effect of generative backbone scale on Industrial.

Backbone	**Tokenizer**	**HR@100**	**MRR@100**
Qwen3-0.6B	TIGER	0.149	0.081
	DSOL	**0.210**	**0.106**
Qwen3-1.7B	TIGER	0.155	0.083
	DSOL	**0.214**	**0.109**
Qwen3-4B	TIGER	0.163	0.088
	DSOL	**0.221**	**0.118**

D Codebook Geometry Ablations

We further ablate the two axes of the residual quantization codebook: the per-layer codebook size K and the SID length L. We report HR@$\{10, 50, 100\}$ together with POR for both TIGER and DSOL on Industrial.

D.1 Effect of Codebook Size

We fix $L = 4$ and sweep the per-layer codebook size $K \in \{128, 256, 512, 1024\}$. Results are shown in Table 8. Both tokenizers peak at $K = 256$ and then regress as K keeps increasing. When the codebook is too small, the SID space becomes overcrowded and semantically distinct items collide on neighbouring codes; when it is too large, each codeword covers only a thin slice of items, so the per-bucket signal is no longer well concentrated. POR is consistent with HR@100 and also peaks at $K = 256$.

Table 8: Effect of codebook size K on Industrial.

K	**Tokenizer**	HR@10	HR@50	HR@100	POR
128	TIGER	0.093	0.120	0.139	29.2
	DSOL	**0.146**	**0.184**	**0.197**	**33.4**
256	TIGER	0.113	0.142	0.149	30.6
	DSOL	**0.166**	**0.198**	**0.210**	**35.3**
512	TIGER	0.110	0.137	0.140	29.0
	DSOL	**0.163**	**0.190**	**0.201**	**33.4**
1024	TIGER	0.075	0.109	0.123	28.1
	DSOL	**0.141**	**0.174**	**0.182**	**32.3**

D.2 Effect of Number of Layers

We fix $K = 256$ and sweep the SID length $L \in \{3, 4, 5\}$ on Industrial (Table 9). When $L = 3$ we set the prefix region to the first layer, and when $L = 5$ we extend it to the first three layers. HR@100 again peaks at $L = 4$ for both tokenizers: a shorter SID under-tokenizes the catalog, while a longer SID stretches the residual cascade

into deeper layers where the per-layer signal is dominated by noise. For DSOL, POR decreases monotonically as L grows ($37.7 \rightarrow 35.3 \rightarrow 31.7$), which does not exactly track HR; this confirms that POR is a relative, comparison-based indicator rather than a strict replacement of HR. Across all three depths, DSOL outperforms TIGER, indicating that the loss remains stable as L varies.

Table 9: Effect of number of layers L on Industrial.

L	**Tokenizer**	HR@10	HR@50	HR@100	POR
3	TIGER	0.105	0.131	0.140	32.8
	DSOL	**0.148**	**0.187**	**0.199**	**37.7**
4	TIGER	0.113	0.142	0.149	30.6
	DSOL	**0.166**	**0.198**	**0.210**	**35.3**
5	TIGER	0.098	0.130	0.138	28.0
	DSOL	**0.161**	**0.189**	**0.199**	**31.7**

E ICR / Gini vs. Downstream HR

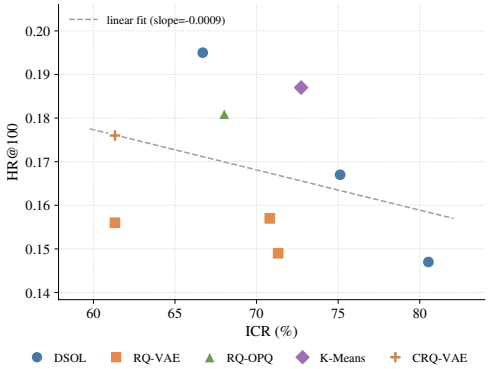
Section 4.3 reports that ICR and Gini are weakly correlated with downstream retrieval quality (Table 2). Figure 6 visualizes the same sweep of eight tokenizer variants used in Figure 3 but plots ICR and Gini against HR@100. Neither variable yields a clean trend: the points form a flat cloud, and within-method variants spread along the metric axis without consistently moving HR@100. This contrasts with the near-linear trends in Figure 3, confirming that uniformity-based metrics describe how evenly the codebook is used but say little about whether a particular SID is representative of its item.

F Visualization of Triplet Data

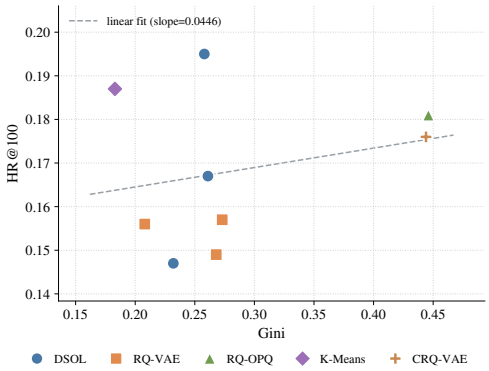
To illustrate the structure of our triplet dataset and how it guides codebook training, we visualize a few cases of different genres in Figure 7. For each triplet, we also show the SIDs produced by DSOL and by the RQ-VAE baseline so that the codebook-level difference is directly observable.

F.1 Headphone Review

In Figure 7(a), both the anchor a and the positive p are reviews of the Sennheiser HD800S headphone, while the negative n is a side-by-side comparison of several headphone models that does not include the HD800S. All three videos belong to the headphone category, but a and p share a clearly stronger semantic link, so the codebook should give (a, p) a longer shared prefix than (a, n). Under DSOL the prefix structure follows this expectation, whereas RQ-VAE fails to separate n from (a, p) at the early layers.



(a) ICR vs. HR@100



(b) Gini vs. HR@100

Figure 6: Global uniformity metrics show no clear trend with HR@100, in contrast to the POR-family metrics in Figure 3.

F.2 Concert Performance

In Figure 7(b), the anchor a and positive p are two clips of the same Spring Festival Gala duet performed by Qishan Huang and Curley G, while the negative n is a solo performance by Curley G alone. Although n shares a performer with (a, p), the song and the staging are different. DSOL's SIDs reflect this by giving (a, p) a longer common prefix while letting n diverge earlier, while RQ-VAE assigns more uniform prefixes that fail to capture this distinction.

F.3 Highway Drowsiness

In Figure 7(c), a and p are short clips about *how to stay awake while driving on the highway*, while n is nominally about "highway drowsiness" but is actually a review of advanced driver-assistance systems. DSOL pulls a and p together at the prefix layers, whereas under RQ-VAE a and p already

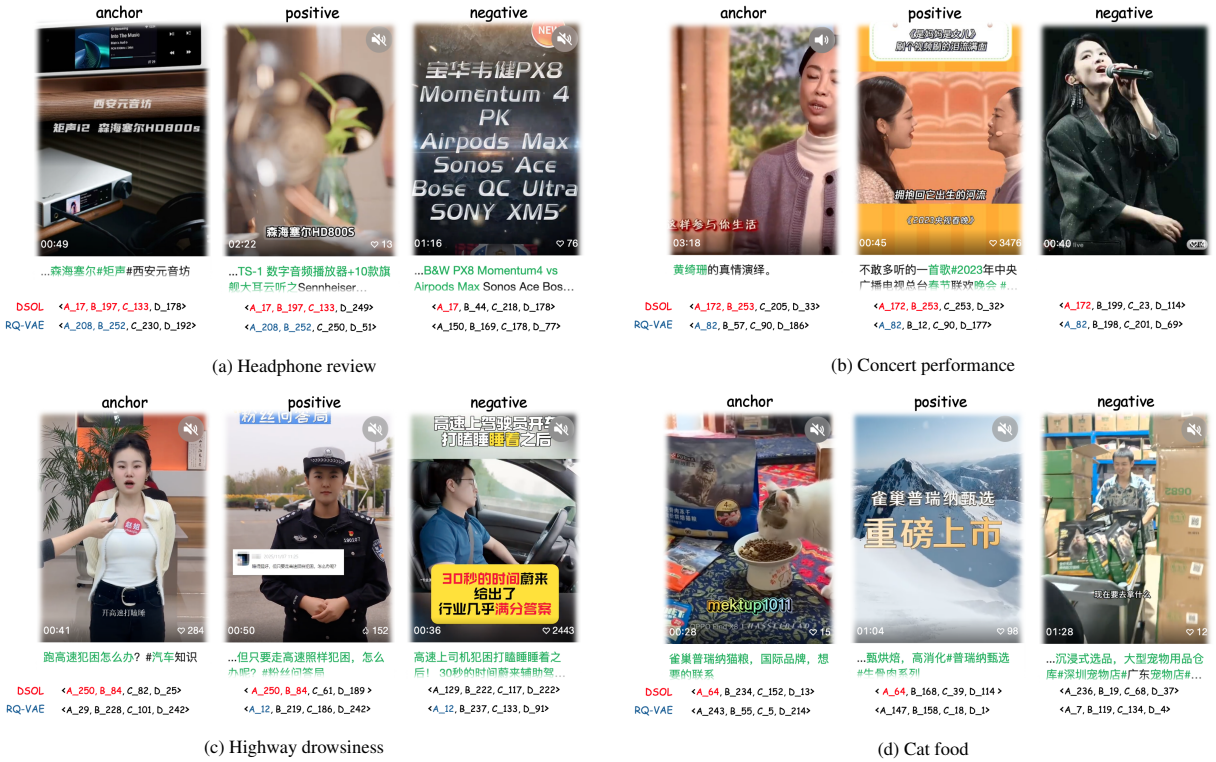


Figure 7: Triplet examples from the Industrial dataset, together with the SIDs assigned by DSOL and by the RQ-VAE baseline. (a) headphone reviews; (b) Spring Festival Gala concert clips; (c) highway drowsiness; (d) cat food.

disagree at the first codebook layer, which illustrates the baseline's weakness at grouping items that share fine-grained semantics.

F.4 Cat Food

In Figure 7(d), a and p are both promotional clips for the same brand of cat food, while n is a behind-the-scenes look at a cat-food factory. DSOL again gives (a, p) a long shared prefix. The fact that n shares no codes with (a, p) under either tokenizer most likely reflects a genuinely large semantic gap between consumer-facing promotion and industrial-line content.