

PRQ-KMeans: Projection Residual Quantization for Semantic ID Tokenization

Yunxiao Luo*, Siyuan Wang[†], Ben Chen[‡], Chenyi Lei

¹Kuaishou Technology

Abstract

Semantic identifiers (SIDs) represent entities as hierarchical token sequences for generative retrieval and recommendation. Residual-quantization tokenizers construct these sequences by selecting a codeword at each level and passing a residual to the next. We view this process as **progressive commonality removal**: each token captures a component shared within its group, while later tokens should model the remaining differences. This view reveals three limitations: a corpus-wide shared component can consume first-level capacity, hard assignment ignores graded similarities to nearby codewords, and full-codeword subtraction can leave variation along the selected-codeword direction in the next residual. We therefore develop our solution in the post-hoc setting, where residual construction is not constrained by input reconstruction. Specifically, we propose **PRQ-KMeans**, which removes the global-mean component, refines centroids with Top- k similarity-weighted updates, and replaces full-codeword subtraction with a projection residual that removes each representation’s selected-centroid component. Experiments on a large-scale industrial search dataset and four public recommendation benchmarks show that PRQ-KMeans achieves the strongest overall performance among the evaluated tokenizers, including gains of up to 7.4% in HitRate and 11.8% in MRR on the industrial dataset.

1 Introduction

Semantic identifiers (SIDs) replace atomic entity identifiers with short token sequences for generative retrieval (Tay et al. 2022; Wang et al. 2022; Sun et al. 2023) and recommendation (Rajput et al. 2023; Singh et al. 2024). Hierarchical SIDs organize these tokens from coarse to fine: related entities can share early tokens, while later tokens distinguish entities that remain under the same prefix. Constructing such an identifier involves two coupled decisions at each level: which token represents the entity at the current position, and what representation is passed forward to construct the next token.

Residual quantization provides a common framework for making these decisions. At each level, it selects a codeword for the current token, subtracts that codeword from the current representation, and passes the resulting residual to the

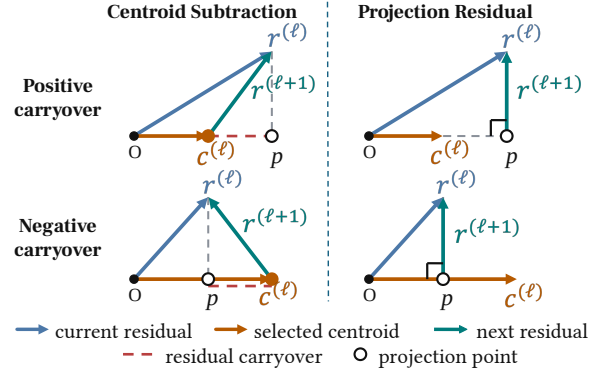


Figure 1: Full-centroid subtraction and projection residuals. Full-centroid subtraction may leave positive or negative residual carryover along the selected-centroid direction, whereas projection removes this component and yields a residual orthogonal to that centroid.

next level. Two major tokenizer paradigms use this recursion. VAE-based methods (Lee et al. 2022; Rajput et al. 2023; Wan et al. 2026) learn hierarchical SIDs through a residual-quantization autoencoder. Post-hoc methods (e.g., RQ-KMeans and RQ-GMM) (Luo et al. 2025; Deng et al. 2025; Tong et al. 2026) instead fit hierarchical codebooks directly over existing embeddings. In both paradigms, the residual becomes the representation processed at the next level. We refer to the component that remains along the selected-codeword direction after subtraction as residual carryover.

Since each later token uses the preceding residual, residual construction determines what information passes across levels. We therefore view residual quantization as **progressive commonality removal**: the selected codeword represents a component shared within the current group, while the residual should retain the differences that remain for later tokens. This view reveals three challenges. **First**, commonality can exist before any cluster-level token is formed. A component shared across the embedding collection may dominate the initial representations. Because it is broadly shared, this component contributes little to distinguishing entities, yet the first codebook may spend part of its capacity modeling this global background rather than coarse differences among entities.

*Equal contribution.

[†]Equal contribution.

[‡]Corresponding author.

Second, hard assignment limits how accurately a codeword can summarize its group. Each representation updates only its selected codeword, even when it is also similar to nearby candidates, discarding graded similarity around assignment boundaries. **Third**, full-codeword subtraction does not generally remove the selected-codeword component accurately from every representation. As Figure 1 illustrates, representations assigned to the same codeword may contain different amounts of this component, although the same codeword is subtracted from all of them. This mismatch can leave a positive or negative component along the selected direction, causing the next codebook to revisit variation already represented by the current token. Our preliminary study finds this component measurable in industrial RQ-KMeans residuals and shows that retaining more of it monotonically reduces next-level SID-prefix utilization (Section 3).

VAE-based tokenizers can reshape their latent representations and codebooks through reconstruction and auxiliary SID objectives (Wang et al. 2024; Zhu et al. 2024; Wan et al. 2026). Their residuals, however, are constrained by additive reconstruction: everything left after subtracting the current codeword is passed to later codewords, even a component along the same direction. Discarding that component independently would alter the codeword sum used for reconstruction. Post-hoc tokenizers instead operate on existing embeddings without requiring the codewords to reconstruct the input, allowing the residual to preserve only the differences needed by later SID levels. We develop our solution in this post-hoc setting using RQ-KMeans.

To address these challenges, we propose **PRQ-KMeans**, a post-hoc hierarchical SID tokenizer built around projection residuals. To keep the corpus-wide component from consuming first-level capacity, PRQ-KMeans removes each input representation’s projection onto the global mean before fitting the first codebook. To better estimate the component shared within each cluster, Top- k soft refinement lets every representation contribute to several nearby centroids in proportion to their cosine similarities, rather than updating only its most similar centroid. To prevent the selected-centroid component from being passed to the next level, PRQ-KMeans removes each representation’s own projection onto its selected centroid after hard assignment, rather than subtracting the full centroid vector. The resulting residual is orthogonal to that centroid and is passed forward to fit the next codebook. Together, these operations implement progressive commonality removal from global to local.

Our main contributions are threefold. (1) We introduce the view of hierarchical SID construction as progressive commonality removal and identify three challenges in determining what is passed between successive token levels. (2) We propose PRQ-KMeans, which combines global component removal and Top- k soft centroid refinement with a projection residual that removes this component separately for each representation. (3) We demonstrate the effectiveness of PRQ-KMeans through extensive evaluation on industrial search and four public recommendation benchmarks.

2 Related Work

2.1 Semantic IDs in Generative Retrieval and Recommendation

Semantic IDs represent entities as short sequences of discrete tokens. For generative retrieval, DSI (Tay et al. 2022) establishes the paradigm of directly generating document identifiers, and NCI (Wang et al. 2022) extends this paradigm with hierarchical semantic identifiers. GenRet (Sun et al. 2023), GLEN (Lee, Choi, and Lee 2023), and MERGE (Zhang et al. 2025a) incorporate semantic, lexical, or relevance signals into identifier construction. OneSearch (Chen et al. 2025) applies hierarchical SIDs to large-scale generative e-commerce search. For generative recommendation and ranking, TIGER (Rajput et al. 2023), Semantic IDs for Ranking (Singh et al. 2024), and OneRec (Deng et al. 2025) use hierarchical SIDs to organize the output space. LETTER (Wang et al. 2024), CoST (Zhu et al. 2024), ETEGRec (Liu et al. 2024), DIGER (Fu et al. 2026), and UniSID (Jiang et al. 2026) further incorporate collaborative, contrastive, differentiable, or end-to-end supervision into SID learning. For embedding-derived SIDs, the tokenizer determines how continuous entity representations are mapped to discrete token sequences.

2.2 Quantization Methods for Hierarchical SIDs

Residual quantization constructs a hierarchical SID by successively quantizing the residual passed from one level to the next. RQ-VAE (Lee et al. 2022; Rajput et al. 2023) and R3-VAE (Wan et al. 2026) use autoencoders and recursive residual quantization to learn hierarchical SIDs. Post-hoc methods fit the hierarchy over existing embeddings: QARM (Luo et al. 2025) and OneRec (Deng et al. 2025) use RQ-KMeans, while RQ-GMM (Tong et al. 2026) uses probabilistic mixture components. Beyond residual quantization, parallel methods generate multiple codes from separate vector subspaces or token branches, including PQ (Jegou, Douze, and Schmid 2011), OPQ (Ge et al. 2013), LLDA-Rec (Shi et al. 2025), and ACERec (Xia et al. 2026a). Moreover, hybrid tokenizers, including OneSearch (Chen et al. 2025), QARM V2 (Xia et al. 2026b), and Quantizing Intent (Choi et al. 2026), combine residual and non-residual quantizers to capture hierarchical commonality and finer distinctions. Recently, methods such as DOS (Yin et al. 2026), ReSID (Liang et al. 2026), and HHQ (Wang et al. 2026) reshape the quantization space through orthogonal or hyperspherical structures. However, these methods leave the cross-level residual unexplored. Under full-centroid subtraction, the residual can still carry a selected-centroid component into the next codebook, making later levels revisit variation represented earlier. PRQ-KMeans addresses this gap by modeling this carryover and removing it through per-representation projection.

3 Preliminaries and Motivation

3.1 RQ-KMeans for Hierarchical SIDs

In RQ-KMeans, each codeword is a K-Means centroid. Let $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N \subset \mathbb{R}^d$ be the embeddings used to fit the tokenizer. A depth- L RQ-KMeans tokenizer learns one codebook $\mathcal{C}^{(\ell)} = \{\mathbf{c}_j^{(\ell)}\}_{j=1}^{K_\ell}$ at each level and maps entity i to the

SID $(z_i^{(1)}, \dots, z_i^{(L)})$. Starting from $\mathbf{r}_i^{(1)} = \mathbf{x}_i$, Euclidean K-Means alternates between nearest-centroid assignment and centroid update:

$$\begin{aligned} z_i^{(\ell)} &= \arg \min_j \left\| \mathbf{r}_i^{(\ell)} - \mathbf{c}_j^{(\ell)} \right\|_2, \\ \mathbf{c}_j^{(\ell)} &\leftarrow \frac{1}{|\mathcal{I}_j^{(\ell)}|} \sum_{i \in \mathcal{I}_j^{(\ell)}} \mathbf{r}_i^{(\ell)}. \end{aligned} \quad (1)$$

where $\mathcal{I}_j^{(\ell)} = \{i : z_i^{(\ell)} = j\}$. After fitting, $z_i^{(\ell)}$ becomes the token at level ℓ , and RQ-KMeans constructs the next-level representation by subtracting the selected centroid:

$$\mathbf{r}_i^{(\ell+1)} = \mathbf{r}_i^{(\ell)} - \mathbf{c}_{z_i^{(\ell)}}^{(\ell)}. \quad (2)$$

Some implementations additionally apply the in-place update $\mathbf{r}_i^{(\ell+1)} \leftarrow \text{norm}(\mathbf{r}_i^{(\ell+1)})$ before the next level. Following previous work (Chen et al. 2025, 2026), the industrial analysis below uses this normalized variant. Here, $\text{norm}(\mathbf{v}) = \mathbf{v} / \|\mathbf{v}\|_2$ for nonzero $\mathbf{v}$. In reconstruction-oriented quantization, the residual is the approximation error left after selecting a centroid; in hierarchical SID tokenization, it also determines the representation available to the next level. Under *progressive commonality removal*, the selected centroid summarizes information shared within its cluster, while the residual should preserve the remaining differences. We next examine whether full-centroid subtraction achieves this goal for each representation.

3.2 Residual Carryover after Centroid Subtraction

A centroid is shared by all representations assigned to its cluster, although these representations may contain different amounts of its directional component. Consider one representation–centroid pair and abbreviate it as $\mathbf{r}$ and $\mathbf{c} \neq \mathbf{0}$. Write $\mathbf{c} = \rho \hat{\mathbf{c}}$, where $\hat{\mathbf{c}} = \mathbf{c} / \|\mathbf{c}\|_2$ and $\rho = \|\mathbf{c}\|_2$. Let $\alpha = \mathbf{r}^\top \hat{\mathbf{c}}$ and $\mathbf{u} = \mathbf{r} - \alpha \hat{\mathbf{c}}$, so that $\mathbf{u} \perp \hat{\mathbf{c}}$. Then

$$\begin{aligned} \mathbf{r} &= \alpha \hat{\mathbf{c}} + \mathbf{u}, \\ \mathbf{r} - \mathbf{c} &= (\alpha - \rho) \hat{\mathbf{c}} + \mathbf{u}. \end{aligned} \quad (3)$$

Full-centroid subtraction applies the shared coefficient ρ , whereas representation $\mathbf{r}$ contains the instance-specific coefficient α . When $\alpha \neq \rho$, the residual retains $(\alpha - \rho)\hat{\mathbf{c}}$ along the selected centroid direction; we call this component *residual carryover*. Figure 1 illustrates the positive- and negative-carryover cases. An arithmetic-mean centroid matches the average coefficient within its assigned cluster, but not necessarily the coefficient of every representation. Appendix A provides the corresponding identity and derivation.

3.3 Empirical Motivation

We first measure residual carryover using a three-level RQ-KMeans tokenizer fitted to representations from the industrial dataset, with codebook sizes 1024, 512, and 128. Section 5.1 and Appendix C provide the setup. For representation

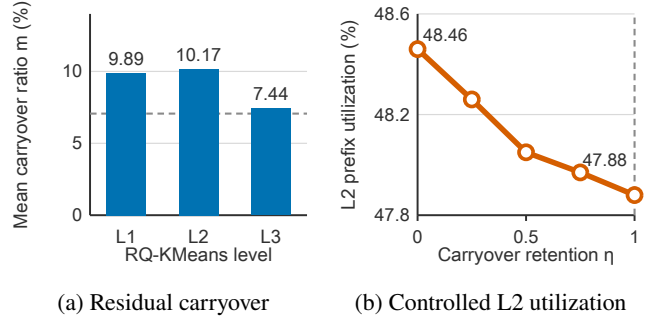


Figure 2: Empirical motivation on the industrial dataset. (a) Mean residual carryover ratio at each RQ-KMeans level. The dashed line marks the expected absolute cosine between two independent isotropic directions in 128 dimensions. (b) L2 prefix utilization after retaining a fraction η of the L1 carryover. L1 assignments stay fixed; L2 is refit for every η .

i , define the relative carryover magnitude at level ℓ as

$$m_i^{(\ell)} = \frac{|\alpha_i^{(\ell)} - \rho_i^{(\ell)}|}{\left\| \mathbf{r}_i^{(\ell)} - \mathbf{c}_{z_i^{(\ell)}}^{(\ell)} \right\|_2}. \quad (4)$$

Figure 2(a) reports mean ratios of 9.89%, 10.17%, and 7.44% at L1, L2, and L3, respectively. For scale, two independent isotropic directions in 128 dimensions have an expected absolute cosine of 7.07%; Appendix D.1 provides the derivation. The L1 and L2 means, corresponding to residuals passed to subsequent codebooks, exceed this reference by 2.82 and 3.10 percentage points, respectively, whereas the terminal L3 value is close to it.

We next test how this component affects the immediately following SID level. Keeping the L1 assignments fixed, we retain a fraction $\eta \in [0, 1]$ of the L1 carryover:

$$\mathbf{r}_i^{(2)}(\eta) = \text{norm} \left(\eta \left(\alpha_i^{(1)} - \rho_i^{(1)} \right) \hat{\mathbf{c}}_i^{(1)} + \mathbf{u}_i^{(1)} \right). \quad (5)$$

Here, $\hat{\mathbf{c}}_i^{(1)}$ is the direction of the selected L1 centroid. The endpoints $\eta = 1$ and $\eta = 0$ retain the full RQ-KMeans carryover and remove it completely, respectively. For each η , we refit L2 with the same Euclidean K-Means configuration and measure the fraction of the $K_1 K_2$ possible two-token item prefixes assigned to at least one item. Appendix D.1 provides the equivalent operational update and implementation details.

Figure 2(b) shows that L2 prefix utilization increases monotonically from 47.88% at $\eta = 1$ to 48.46% at $\eta = 0$. Under this controlled protocol, reducing selected-centroid carryover allows the immediately following level to use a broader set of prefixes. This observation motivates constructing the next-level representation by removing the selected-centroid component through projection.

4 Methodology

4.1 Overview

Figure 3 gives an overview of PRQ-KMeans. It first removes the global component from the input embeddings. It then

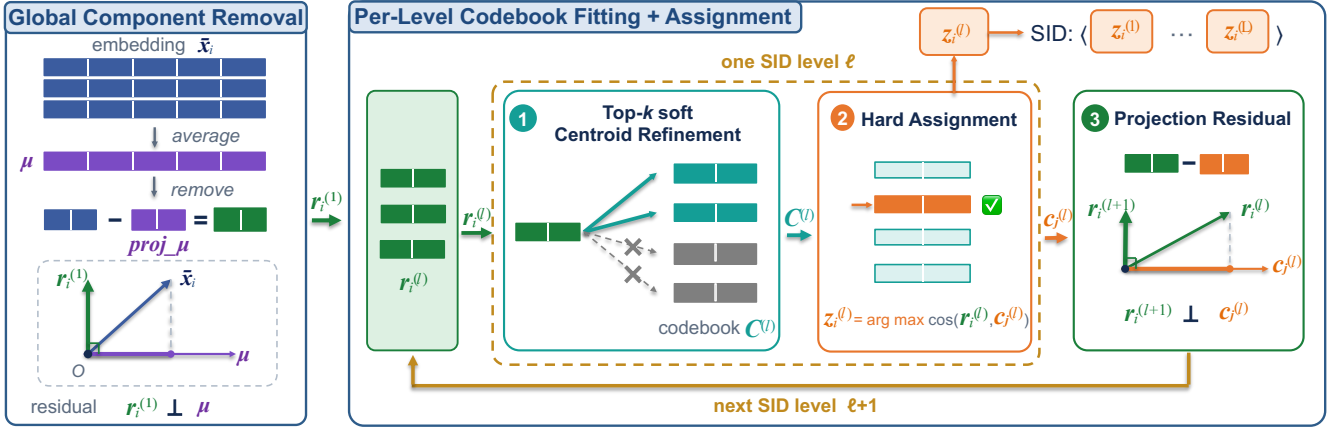


Figure 3: Overview of PRQ-KMeans. PRQ-KMeans initializes the first-level residual by removing the global-mean component. At level ℓ , Top- k soft centroid refinement fits the codebook from the current residuals. Hard cosine assignment then selects a centroid and emits token $z_i^{(\ell)}$; projection removes the selected-centroid component before the residual proceeds to level $\ell + 1$. SID encoding repeats the same hard-assignment and projection recursion using the learned codebooks.

learns the codebooks sequentially: at each level, Top- k soft weights refine the current centroids, hard cosine assignment emits a token, and the projection residual becomes the input used to fit the next codebook. Once the hierarchy is trained, SID encoding repeats the hard assignment and residual update using the learned global mean and codebooks during inference. Projection defines the cross-level update, while global removal and soft refinement prepare its input representations and centroids. Appendix B gives the complete algorithm.

4.2 Projection Residual Construction

PRQ-KMeans removes the selected-centroid component by requiring the next residual $\mathbf{v}$ to be orthogonal to the selected centroid $\mathbf{c}$. Orthogonality alone, however, does not determine a useful residual; even the zero vector satisfies this constraint. We therefore choose the orthogonal residual that changes the current representation $\mathbf{r}$ as little as possible:

$$\mathbf{v}^* = \arg \min_{\mathbf{v}} \|\mathbf{v} - \mathbf{r}\|_2^2 \quad \text{s.t.} \quad \mathbf{v}^\top \mathbf{c} = 0. \quad (6)$$

Solving this problem gives

$$\mathbf{v}^* = \mathbf{r} - \frac{\mathbf{r}^\top \mathbf{c}}{\|\mathbf{c}\|_2^2} \mathbf{c}. \quad (7)$$

This solution directly removes the carryover identified in Eq. (3). Recall that $\mathbf{c} = \rho \hat{\mathbf{c}}$, $\alpha = \mathbf{r}^\top \hat{\mathbf{c}}$, and $\mathbf{r} = \alpha \hat{\mathbf{c}} + \mathbf{u}$. The component subtracted in Eq. (7) is

$$\frac{\mathbf{r}^\top \mathbf{c}}{\|\mathbf{c}\|_2^2} \mathbf{c} = \alpha \hat{\mathbf{c}}.$$

Consequently, $\mathbf{v}^* = \mathbf{r} - \alpha \hat{\mathbf{c}} = \mathbf{u}$. Unlike the RQ-KMeans residual $(\alpha - \rho) \hat{\mathbf{c}} + \mathbf{u}$, the projection residual contains no remaining component along the selected centroid.

Proposition 1 formalizes these guarantees.

Proposition 1 (Minimum-change orthogonal residual)

For any $\mathbf{r} \in \mathbb{R}^d$ and nonzero $\mathbf{c}$, the projection residual $\mathbf{v}^*$ in Eq. (7) is the unique solution to Eq. (6). It satisfies $\mathbf{c}^\top \mathbf{v}^* = 0$ and has the smallest Euclidean distance to $\mathbf{r}$ among all vectors orthogonal to $\mathbf{c}$.

Centroid magnitude does not affect the component removed by Eq. (7). PRQ-KMeans therefore compares representations and centroids using cosine similarity, which is likewise unaffected by centroid magnitude. The same cosine scores are used for Top- k soft refinement and hard assignment. The complete proofs are provided in Appendix A.

4.3 Global Component Removal

A component shared across the entire embedding collection can dominate the first partition. Standard K-Means uses the mean of each cluster as its centroid, summarizing information shared by the representations in that cluster. PRQ-KMeans applies the same idea before the first SID level: it uses the mean of all input representations to summarize information shared across the collection. We write $\bar{\mathbf{x}}_i = \text{norm}(\mathbf{x}_i)$ for the L2-normalized input embedding. Their global mean is

$$\boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^N \bar{\mathbf{x}}_i. \quad (8)$$

Before emitting any token, PRQ-KMeans removes the component along this mean and obtains the first-level representation:

$$\mathbf{r}_i^{(1)} = \text{norm} \left(\bar{\mathbf{x}}_i - \frac{\bar{\mathbf{x}}_i^\top \boldsymbol{\mu}}{\|\boldsymbol{\mu}\|_2^2} \boldsymbol{\mu} \right). \quad (9)$$

The global mean is learned once during tokenizer fitting and reused during encoding. Equation (9) produces the representations used to fit the first codebook; subsequent levels apply the same residual construction to the selected centroid.

4.4 Top- k Soft Centroid Refinement

Because the residual update depends on the selected centroid, reliable centroid estimation matters. Hard K-Means uses each representation to update only its assigned centroid, ignoring its similarity to nearby centroids around a cluster boundary. During each fitting iteration, PRQ-KMeans computes the cosine scores and selects the Top- k candidate centroids:

$$s_{ij}^{(\ell)} = \cos(\mathbf{r}_i^{(\ell)}, \mathbf{c}_j^{(\ell)}), \quad (10)$$

$$\mathcal{N}_k(\mathbf{r}_i^{(\ell)}) = \text{TopK}_{j \in \{1, \dots, K_\ell\}} s_{ij}^{(\ell)}.$$

Within this candidate set, the soft weight is

$$w_{ij}^{(\ell)} = \frac{\exp(\beta s_{ij}^{(\ell)})}{\sum_{t \in \mathcal{N}_k(\mathbf{r}_i^{(\ell)})} \exp(\beta s_{it}^{(\ell)})} \quad (11)$$

for $j \in \mathcal{N}_k(\mathbf{r}_i^{(\ell)})$, and zero otherwise, where $\beta > 0$ is the soft-max concentration (inverse-temperature) parameter; larger β produces sharper weights.

Each centroid is updated using the representations for which it is a candidate:

$$\mathbf{c}_j^{(\ell)} \leftarrow \frac{\sum_i w_{ij}^{(\ell)} \mathbf{r}_i^{(\ell)}}{\sum_i w_{ij}^{(\ell)}}. \quad (12)$$

Restricting the weights to the k nearest centroids keeps each update local while allowing a boundary representation to contribute to nearby candidates. The score, weight, and centroid-update steps are repeated for a fixed number of fitting iterations. Soft weights are used only while refining the current codebook. After refinement, hard assignment emits the token and constructs the residual passed to the next level.

4.5 Sequential Codebook Fitting and SID Encoding

In the recursion, Eq. (7) is applied with $\mathbf{r} = \mathbf{r}_i^{(\ell)}$ and $\mathbf{c} = \mathbf{c}_{z_i^{(\ell)}}^{(\ell)}$. After fitting $\mathcal{C}^{(\ell)}$, PRQ-KMeans assigns each fitting representation to one centroid using hard cosine assignment:

$$z_i^{(\ell)} = \arg \max_j \cos(\mathbf{r}_i^{(\ell)}, \mathbf{c}_j^{(\ell)}). \quad (13)$$

The selected index $z_i^{(\ell)}$ gives the current token. Writing the selected centroid as $\mathbf{c}_i^{(\ell)} = \mathbf{c}_{z_i^{(\ell)}}^{(\ell)}$, PRQ-KMeans removes its component and normalizes the result in one update:

$$\mathbf{r}_i^{(\ell+1)} = \text{norm} \left(\mathbf{r}_i^{(\ell)} - \frac{\mathbf{r}_i^{(\ell)\top} \mathbf{c}_i^{(\ell)}}{\|\mathbf{c}_i^{(\ell)}\|_2^2} \mathbf{c}_i^{(\ell)} \right). \quad (14)$$

For a nonzero projection residual, the normalization in Eq. (14) is well defined; vector $\mathbf{r}_i^{(\ell+1)}$ is used to fit $\mathcal{C}^{(\ell+1)}$. Repeating this process learns the hierarchy level by level.

Once fitting is complete, SID encoding reuses the learned global mean and codebooks, repeating the same hard assignment, token emission, projection, and normalization steps without further updates.

5 Experiments

5.1 Experimental Setup

Datasets. We primarily evaluate PRQ-KMeans on a publicly released industrial e-commerce search dataset from prior work (Li et al. 2026). It contains approximately 7.8 million items and 9.0 million training-query records for tokenizer fitting. Following OneSearch (Chen et al. 2025), these data form three training stages of approximately 36 million, 54 million, and 46 million examples, with separate Order and Click test sets from logged purchases and clicks. We also evaluate on the Amazon Sports, Toys, and Clothing benchmarks (He and McAuley 2016) and LastFM (Cantador, Brusilovsky, and Kuflik 2011), following prior work (Rajput et al. 2023; Wan et al. 2026). Appendix C provides the dataset details.

Baselines. We compare PRQ-KMeans with learned residual tokenizers RQ-VAE (Lee et al. 2022) and R3-VAE (Wan et al. 2026), post-hoc tokenizers RQ-KMeans (Luo et al. 2025) and RQ-GMM (Tong et al. 2026), and parallel quantizers PQ-KMeans (Jegou, Douze, and Schmid 2011) and OPQ-KMeans (Ge et al. 2013). For the five-level industrial setting, we also evaluate hybrid tokenizers that combine a hierarchical residual prefix with suffix tokens (Chen et al. 2025; Xia et al. 2026b). RQ-OPQ (Chen et al. 2025) uses two RQ-KMeans levels, a balanced K-Means third level, and two OPQ suffix tokens. ResKmeansFSQ (Xia et al. 2026b) uses three RQ-KMeans levels followed by two FSQ suffix tokens. PRQ-OPQ replaces the first two RQ-KMeans levels of RQ-OPQ with PRQ-KMeans, while retaining its balanced third-level K-Means and two OPQ suffix tokens. This construction evaluates the compatibility of PRQ-KMeans with the balanced third level and OPQ suffix used in RQ-OPQ. Appendix C provides the reproduction and alignment details.

Implementation and Evaluation. For the industrial dataset, we follow OneSearch (Chen et al. 2025), using its 128-dimensional distilled-BGE query and item representations and BART-Base retriever (Lewis et al. 2020). All tokenizers share the three-stage training, decoding, and evaluation pipeline. The three-level setting uses a 1024-512-128 codebook; the five-level setting uses the same three-level prefix followed by two 64-way suffix tokens. For the public benchmarks, we use 768-dimensional Sentence-T5-Base item embeddings (Ni et al. 2021), a 256-256-256 codebook, and the TIGER pipeline (Rajput et al. 2023; Wan et al. 2026). PRQ-KMeans uses $k = 2$ on the public benchmarks and $k = 5$ on the industrial dataset due to their different codebook sizes, with $\beta = 15$ in both settings. Following OneSearch, industrial evaluation reports HitRate@50 and MRR@50 for Order and Click, together with the independent-code ratio (ICR), prefix utilization, and used-prefix Gini for codebook quality. Public evaluation reports item-level Recall@20 and NDCG@20; Appendix C provides evaluation details. The code will be released upon acceptance.

5.2 Overall Performance

Industrial benchmark. We compare three-level codebook quality (Table 1). Among all evaluated methods, PRQ-

Method	Downstream Evaluation				Codebook Quality						
	Order		Click		Full SID	Utilization (%)			Gini		
	HitRate	MRR	HitRate	MRR	ICR (%)	L1	L2	L3	L1	L2	L3
Codebook: 1024–512–128											
RQ-VAE	0.2474	0.0501	0.2921	0.0446	54.24	100.00	47.55	3.22	0.407	0.795	0.601
R3-VAE	0.2289	0.0449	0.2788	0.0402	54.96	99.41	<u>51.04</u>	3.34	0.427	0.783	0.592
RQ-KMeans	<u>0.2913</u>	<u>0.0644</u>	<u>0.3279</u>	<u>0.0540</u>	<u>55.52</u>	100.00	47.88	<u>3.61</u>	0.413	<u>0.774</u>	<u>0.569</u>
RQ-GMM	0.2687	0.0594	0.3071	0.0483	50.66	99.90	38.57	3.06	0.412	0.783	0.598
PQ-KMeans	0.0400	0.0061	0.0521	0.0054	50.45	100.00	22.52	0.67	0.351	0.934	0.891
OPQ-KMeans	0.1242	0.0225	0.1393	0.0164	47.35	100.00	21.85	1.15	0.276	0.916	0.804
PRQ-KMeans	0.3128	0.0720	0.3488	0.0588	58.45	100.00	57.78	3.99	<u>0.298</u>	0.758	0.546
Codebook: 1024–512–128–64–64											
ResKmeansFSQ	0.4038	0.1194	0.4364	0.0969	89.19	100.00	<u>47.88</u>	3.61	<u>0.413</u>	<u>0.774</u>	0.569
RQ-OPQ	<u>0.4276</u>	<u>0.1340</u>	<u>0.4650</u>	<u>0.1083</u>	98.99	100.00	<u>47.88</u>	<u>8.00</u>	<u>0.413</u>	<u>0.774</u>	<u>0.266</u>
PRQ-OPQ	0.4359	0.1364	0.4713	0.1089	<u>98.74</u>	100.00	57.78	8.42	0.298	0.758	0.241

Table 1: Codebook quality and downstream generative retrieval performance on the industrial dataset. ICR denotes the independent-code ratio, the percentage of occupied full SIDs assigned to exactly one item. For each codebook block, best and second-best values are bold and underlined, respectively; lower Gini is better.

Method	Sports		Toys		Clothing		LastFM	
	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG
RQ-VAE	0.0522	0.0219	0.0751	0.0321	0.0357	0.0141	0.0082	0.0033
R3-VAE	0.0514	0.0211	0.0812	<u>0.0354</u>	<u>0.0361</u>	<u>0.0147</u>	0.0074	0.0029
RQ-KMeans	<u>0.0528</u>	<u>0.0215</u>	<u>0.0824</u>	0.0346	0.0330	0.0132	<u>0.0115</u>	<u>0.0042</u>
RQ-GMM	0.0506	0.0209	0.0711	0.0287	0.0341	0.0133	0.0082	0.0024
PQ-KMeans	0.0087	0.0033	0.0177	0.0080	0.0082	0.0034	0.0016	0.0007
OPQ-KMeans	0.0379	0.0151	0.0403	0.0171	0.0233	0.0087	0.0033	0.0014
PRQ-KMeans	0.0531	0.0219	0.0851	0.0361	0.0382	0.0151	0.0179	0.0076

Table 2: Performance evaluation on the public benchmarks.

KMeans obtains the highest ICR and L2–L3 prefix utilization together with the lowest L2–L3 Gini. Relative to RQ-KMeans, ICR increases from 55.52% to 58.45%, L2–L3 utilization rises from 47.88%/3.61% to 57.78%/3.99%, and L2–L3 Gini decreases from 0.774/0.569 to 0.758/0.546.

PRQ-KMeans also improves downstream generative retrieval. RQ-KMeans is the strongest three-level baseline on all four metrics. Relative to RQ-KMeans, PRQ-KMeans improves Order HitRate and MRR by 7.4% and 11.8%, respectively, and Click HitRate and MRR by 6.4% and 8.9%. In the five-level hybrid setting, PRQ-OPQ achieves the best Order and Click results on all four downstream metrics among the three aligned tokenizers and improves over RQ-OPQ by 0.6%–1.9%. This result supports the compatibility of the PRQ-KMeans prefix with an OPQ suffix under the same industrial tokenizer interface.

Public benchmarks. Across four benchmarks, PRQ-KMeans obtains the best or tied-best value on all eight metrics and exceeds RQ-KMeans in every comparison (Table 2). The gains are modest on Sports and Toys, larger on Clothing,

Variant	Order		Click	
	HitRate	MRR	HitRate	MRR
PRQ-KMeans	0.3128	0.0720	0.3488	0.0588
– Global	0.3017	0.0697	0.3379	0.0560
– Soft	<u>0.3067</u>	<u>0.0705</u>	<u>0.3460</u>	<u>0.0584</u>
– Projection	0.2985	0.0687	0.3328	0.0548

Table 3: Ablation study on the industrial dataset.

and most pronounced on LastFM, where Recall increases from 0.0115 to 0.0179 and NDCG from 0.0042 to 0.0076 relative to RQ-KMeans. These results extend the evidence beyond the industrial setting to multiple public domains.

5.3 Ablation Study

Tables 3 and A1 (in Appendix) report the industrial and public ablations, respectively. We compare the complete model with three variants: – *Global* omits global component removal; – *Soft* replaces Top-*k* soft refinement with hard centroid updates; and – *Projection* retains Global and Top-*k* refinement but replaces cosine fitting and assignment with Euclidean counterparts and projection residualization with full-centroid subtraction. The complete PRQ-KMeans model achieves the best result on every industrial metric and the best or tied-best result on all eight public metrics, tying – *Global* on Clothing NDCG. On the industrial dataset, removing Projection produces the largest decrease on all four metrics, while removing Global or Soft also yields consistent drops. On the public benchmarks, removing Projection produces the largest decrease on all six Amazon metrics, while LastFM is most sensitive to Soft. Removing Global lowers seven of eight metrics and ties the complete model on Clothing NDCG.

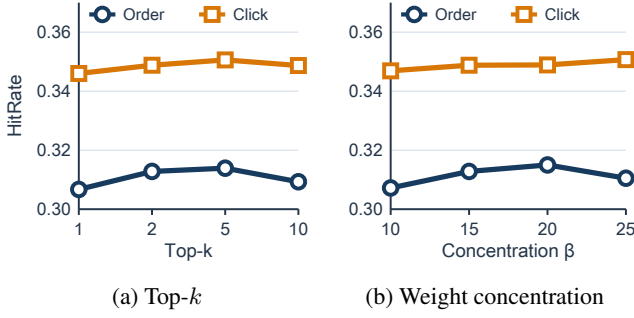


Figure 4: Hyperparameter sensitivity on the industrial dataset.

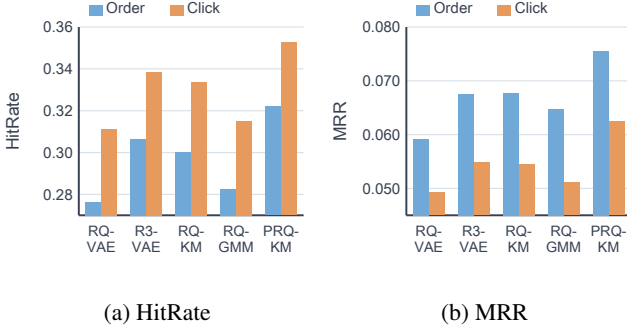


Figure 5: Embedding robustness on the industrial dataset. The aligned BGE representations used in the main experiment are replaced with aligned Qwen3 representations, while the codebook sizes and downstream protocol remain unchanged. KM denotes K-Means.

5.4 Sensitivity and Robustness

Hyperparameter sensitivity. We study sensitivity to k , the number of nearby centroids receiving weighted updates, and β , which controls weight concentration. Figure 4 shows that $k = 2$ and $k = 5$ perform better than $k = 1$ and $k = 10$. This suggests a trade-off: a small neighborhood limits sharing near cluster boundaries, while larger neighborhoods may weaken locality by updating less similar centroids. For β , $\beta = 10$ performs worse, while results remain comparable across $\beta = 15$ – 25 . This suggests that diffuse weights may obscure similarity differences among the centroids, whereas sharper weights better prioritize the closest candidates. Appendix D.3 provides results on the public benchmarks.

Embedding robustness. To evaluate robustness across input embeddings, we replace the aligned BGE representations used in the main experiment with aligned Qwen3-Embedding-0.6B representations (Zhang et al. 2025b). We rerun five tokenizers using these 128-dimensional representations, with the alignment procedure, codebook sizes, and downstream protocol unchanged. Figure 5 shows that PRQ-KMeans achieves the highest HitRate and MRR on both Order and Click under the alternative embedding, supporting its robustness to different input embeddings.

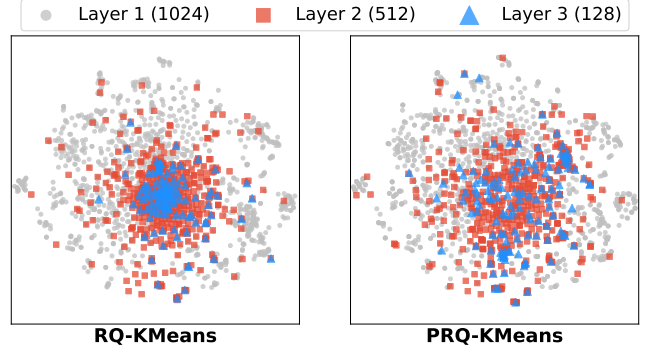


Figure 6: Centroid visualization on the industrial dataset. L1–L3 centroids from RQ-KMeans and PRQ-KMeans are jointly embedded using one t-SNE fit; both panels share coordinates and axis limits, and colors and markers denote SID levels.

5.5 Qualitative Analysis

Centroid visualization. To visualize how residual construction shapes the learned codebooks, we use the L1–L3 centroids of RQ-KMeans and PRQ-KMeans and jointly embed them in two dimensions with t-SNE (Van der Maaten and Hinton 2008). In RQ-KMeans, L1 centroids span a broad region, whereas L2 and especially L3 centroids form a dense core. This pattern is consistent with residual carryover: when the selected-centroid component remains, later codebooks receive representations that still vary along earlier centroid directions and may revisit previously represented variation. In contrast, PRQ-KMeans removes this component before fitting the next codebook, and its L2 and L3 centroids cover a broader region. Together with the higher later-level utilization and lower Gini in Table 1, this visualization provides qualitative support for progressive commonality removal.

Case study. We trace 12 Kitty-themed slippers assigned to one RQ-KMeans full SID. Under PRQ-KMeans, these items retain one shared level-1 prefix, separate into two level-2 prefixes, and form three full SIDs containing four items each. In this group, the later tokens distinguish observed indoor soft-soled, outdoor/dormitory, and mixed-use variants while retaining the shared product theme. Appendix D.4 reports the exact collision-group statistics and selection procedure; it also presents a case study of the weak-intent query “hotel.”

6 Conclusion

This paper presents PRQ-KMeans, a post-hoc hierarchical SID tokenizer based on progressive commonality removal. It removes the global component before the first level and refines each codebook with Top- k soft weights. After assignment, it removes the selected-centroid component through projection before fitting the next codebook. PRQ-KMeans achieves the best overall downstream retrieval performance on the industrial dataset and four public benchmarks. Component ablations support the complete design. Together, these results support explicitly controlling what is passed between successive SID levels.

References

- Cantador, I.; Brusilovsky, P.; and Kuflik, T. 2011. Second workshop on information heterogeneity and fusion in recommender systems (HetRec2011). In *Proceedings of the fifth ACM conference on Recommender systems*, 387–388.
- Chen, B.; Guo, X.; Wang, S.; Liang, Z.; Lv, Y.; Ma, Y.; Xiao, X.; Xue, B.; Zhang, X.; Yang, Y.; et al. 2025. One-search: A preliminary exploration of the unified end-to-end generative framework for e-commerce search. *arXiv preprint arXiv:2509.03236*.
- Chen, B.; Wang, S.; Ma, Y.; Liang, Z.; Zhang, X.; Lv, Y.; Yang, Y.; Dai, H.; Mao, L.; Zhao, T.; et al. 2026. OneSearch-V2: The Latent Reasoning Enhanced Self-distillation Generative Search Framework. *arXiv preprint arXiv:2603.24422*.
- Choi, J.; Ye, H.; Ding, Z.; Long, B.; Zelditch, B.; and Vats, A. 2026. Quantizing Intent: Cross-Domain Semantic IDs from Organic Activity for Industrial Ranking. *arXiv preprint arXiv:2606.01396*.
- Deng, J.; Wang, S.; Cai, K.; Ren, L.; Hu, Q.; Ding, W.; Luo, Q.; and Zhou, G. 2025. Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment. *arXiv preprint arXiv:2502.18965*.
- Fu, J.; Ge, X.; Karatzoglou, A.; Arapakis, I.; Verberne, S.; Jose, J. M.; and Ren, Z. 2026. Differentiable Semantic ID for Generative Recommendation. *arXiv preprint arXiv:2601.19711*.
- Ge, T.; He, K.; Ke, Q.; and Sun, J. 2013. Optimized product quantization for approximate nearest neighbor search. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2946–2953.
- He, R.; and McAuley, J. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, 507–517.
- Jegou, H.; Douze, M.; and Schmid, C. 2011. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 33(01): 117–128.
- Jiang, J.; Zhang, X.; Zhang, E.; Xiong, Y.; Zhang, J.; Wang, J.; Yu, H.; Wang, Y.; Wang, H.; Yan, X.; et al. 2026. End-to-End Semantic ID Generation for Generative Advertisement Recommendation. *arXiv preprint arXiv:2602.10445*.
- Lee, D.; Kim, C.; Kim, S.; Cho, M.; and Han, W.-S. 2022. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11523–11532.
- Lee, S.; Choi, M.; and Lee, J. 2023. GLEN: Generative retrieval via lexical index learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 7693–7704.
- Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, 7871–7880.
- Li, Y.; Chen, B.; Cheng, M.; Liu, Z.; Zhang, X.; Lei, C.; and Ou, W. 2026. KuaiSearch: A Large-Scale E-Commerce Search Dataset for Recall, Ranking, and Relevance. *arXiv preprint arXiv:2602.11518*.
- Liang, Y.; Zhang, Z.; Zhu, Y.; Zhang, K.; Guo, Z.; Zhou, W.; Yang, Z.; Wu, K.; Ni, Y.; Zeng, A.; et al. 2026. Rethinking Generative Recommender Tokenizer: Recsys-Native Encoding and Semantic Quantization Beyond LLMs. *arXiv preprint arXiv:2602.02338*.
- Liu, E.; Zheng, B.; Ling, C.; Hu, L.; Li, H.; and Zhao, W. X. 2024. Generative recommender with end-to-end learnable item tokenization. *arXiv preprint arXiv:2409.05546*.
- Luo, X.; Cao, J.; Sun, T.; Yu, J.; Huang, R.; Yuan, W.; Lin, H.; Zheng, Y.; Wang, S.; Hu, Q.; et al. 2025. Qarm: Quantitative alignment multi-modal recommendation at kuaishou. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, 5915–5922.
- Ni, J.; Abrego, G. H.; Constant, N.; Ma, J.; Hall, K. B.; Cer, D.; and Yang, Y. 2021. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. *arXiv preprint arXiv:2108.08877*.
- Rajput, S.; Mehta, N.; Singh, A.; Keshavan, R. H.; Vu, T.; Heldt, L.; Hong, L.; Tay, Y.; Tran, V. Q.; Samost, J.; et al. 2023. Recommender systems with generative retrieval. In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Shi, T.; Shen, C.; Yu, W.; Nie, S.; Li, C.; Zhang, X.; He, M.; Han, Y.; and Xu, J. 2025. LLaDA-Rec: Discrete Diffusion for Parallel Semantic ID Generation in Generative Recommendation. *arXiv preprint arXiv:2511.06254*.
- Singh, A.; Vu, T.; Mehta, N.; Keshavan, R.; Sathiamoorthy, M.; Zheng, Y.; Hong, L.; Heldt, L.; Wei, L.; Tandon, D.; et al. 2024. Better generalization with semantic ids: A case study in ranking for recommendations. In *Proceedings of the 18th ACM Conference on Recommender Systems*, 1039–1044.
- Sun, W.; Yan, L.; Chen, Z.; Wang, S.; Zhu, H.; Ren, P.; Chen, Z.; Yin, D.; Rijke, M.; and Ren, Z. 2023. Learning to tokenize for generative retrieval. *Advances in Neural Information Processing Systems*, 36: 46345–46361.
- Tay, Y.; Tran, V.; Dehghani, M.; Ni, J.; Bahri, D.; Mehta, H.; Qin, Z.; Hui, K.; Zhao, Z.; Gupta, J.; et al. 2022. Transformer Memory as a Differentiable Search Index. *Advances in Neural Information Processing Systems*, 35: 21831–21843.
- Tong, Z.; Liu, J.; Zhang, W.; Ruan, H.; Tang, D.; Zeng, Z.; Zeng, Q.; Zhang, P.; Lu, T.; and Gu, N. 2026. RQ-GMM: Residual Quantized Gaussian Mixture Model for Multimodal Semantic Discretization in CTR Prediction. *arXiv preprint arXiv:2602.12593*.
- Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).
- Wan, Q.; Yang, Z.; Yang, D.; Fan, Y.; Yan, X.; Liu, S.; Liu, Y.; Zhang, C.; Xu, W.; Qin, J.; et al. 2026. R3-VAE: Reference Vector-Guided Rating Residual Quantization VAE for Generative Recommendation. *arXiv preprint arXiv:2604.11440*.
- Wang, W.; Bao, H.; Lin, X.; Zhang, J.; Li, Y.; Feng, F.; Ng, S.-K.; and Chua, T.-S. 2024. Learnable item tokenization

for generative recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 2400–2409.

Wang, Y.; Hou, Y.; Wang, H.; Miao, Z.; Wu, S.; Chen, Q.; Xia, Y.; Chi, C.; Zhao, G.; Liu, Z.; et al. 2022. A neural corpus indexer for document retrieval. *Advances in Neural Information Processing Systems*, 35: 25600–25614.

Wang, Y.; Wu, B.; Su, Y.; Zhang, T.; Du, Y.; Yu, L.; Guo, J.; and Cheng, X. 2026. Distilling Large Embeddings via Hyper-spherical Householder Quantization. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 10562–10576.

Xia, M.; Zhou, Z.; Ma, G.; and Huang, D. 2026a. Unleash the Potential of Long Semantic IDs for Generative Recommendation. *arXiv preprint arXiv:2602.13573*.

Xia, T.; Zhang, J.; Liu, Y.; Dou, H.; Yin, T.; Cao, J.; Liang, X.; Xie, T.; Liu, L.; Chen, X.; et al. 2026b. QARM V2: Quantitative Alignment Multi-Modal Recommendation for Reasoning User Sequence Modeling. *arXiv preprint arXiv:2602.08559*.

Yin, J.; Kou, S.; Li, C.; Wang, S.; Huang, Y.; Wei, X.; Zhu, Y.; Wang, H.; and Wang, X. 2026. DOS: Dual-Flow Orthogonal Semantic IDs for Recommendation in Meituan. In *Proceedings of the ACM Web Conference 2026*, 8305–8308.

Zhang, F.; Liu, X.; Jia, X.; Zhang, Y.; Zhang, S.; Li, X.; Zhuang, F.; Lin, W.; and Zhang, Z. 2025a. Multi-level Relevance Document Identifier Learning for Generative Retrieval. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 10066–10080.

Zhang, Q.; Szymanski, L.; Zhang, H.; and Deng, J. D. 2026. How Reliable Are Semantic-ID Tokenizer Comparisons in Generative Recommendation? *arXiv preprint arXiv:2605.25330*.

Zhang, Y.; Li, M.; Long, D.; Zhang, X.; Lin, H.; Yang, B.; Xie, P.; Yang, A.; Liu, D.; Lin, J.; et al. 2025b. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Zhu, J.; Jin, M.; Liu, Q.; Qiu, Z.; Dong, Z.; and Li, X. 2024. Cost: Contrastive quantization based semantic tokenization for generative recommendation. In *Proceedings of the 18th ACM Conference on Recommender Systems*, 969–974.

A Theoretical Proofs

We first derive how representation-level carryover behaves within a cluster, including the case in which the centroid is the arithmetic mean of its assigned representations. We then prove the geometric properties of the projection residual used by PRQ-KMeans.

A.1 Residual-Carryover Identity

Consider a nonzero centroid $\mathbf{c}$, let $\rho = \|\mathbf{c}\|_2$ and $\hat{\mathbf{c}} = \mathbf{c}/\rho$, and fix a nonempty assigned set $\mathcal{I}_j$. For each $i \in \mathcal{I}_j$, write $\mathbf{r}_i = \alpha_i \hat{\mathbf{c}} + \mathbf{u}_i$, where $\alpha_i = \mathbf{r}_i^\top \hat{\mathbf{c}}$ and $\mathbf{u}_i \perp \hat{\mathbf{c}}$, and let $\bar{\alpha}_j = |\mathcal{I}_j|^{-1} \sum_{i \in \mathcal{I}_j} \alpha_i$. Because the carryover coefficient is $\alpha_i - \rho$, its mean squared magnitude satisfies

$$\frac{1}{|\mathcal{I}_j|} \sum_{i \in \mathcal{I}_j} (\alpha_i - \rho)^2 = \text{Var}_{\mathcal{I}_j}(\alpha) + (\bar{\alpha}_j - \rho)^2. \quad (15)$$

Here, $\text{Var}_{\mathcal{I}_j}(\alpha) = |\mathcal{I}_j|^{-1} \sum_{i \in \mathcal{I}_j} (\alpha_i - \bar{\alpha}_j)^2$ denotes the population variance within the assigned cluster. When $\mathbf{c}$ is the arithmetic mean of the assigned representations, $\bar{\alpha}_j = \hat{\mathbf{c}}^\top \mathbf{c} = \rho$. The mean squared carryover then equals the within-cluster variance of α_i and becomes zero only when all assigned representations have the same coefficient along the centroid direction. For a non-mean centroid, the second term in Eq. (15) captures the additional offset.

A.2 Proof of Proposition 1

The identity above characterizes why full-centroid subtraction can retain a component along the selected centroid. We now prove that the projection residual removes this component with the minimum Euclidean change, as stated in Proposition 1. Let $\mathbf{c} \in \mathbb{R}^d$ be nonzero and define

$$\mathbf{v}^* = \mathbf{r} - \frac{\mathbf{r}^\top \mathbf{c}}{\|\mathbf{c}\|_2^2} \mathbf{c}. \quad (16)$$

Exact orthogonality. By direct expansion,

$$\mathbf{c}^\top \mathbf{v}^* = \mathbf{c}^\top \mathbf{r} - \frac{\mathbf{r}^\top \mathbf{c}}{\|\mathbf{c}\|_2^2} \|\mathbf{c}\|_2^2 = 0. \quad (17)$$

If $\mathbf{v}^* \neq \mathbf{0}$, normalization multiplies it by a positive scalar and therefore preserves orthogonality.

Minimum Euclidean change. The feasible set $\mathcal{S} = \{\mathbf{v} : \mathbf{v}^\top \mathbf{c} = 0\}$ is a closed linear subspace. Because $\mathbf{v}^* \in \mathcal{S}$ and $\mathbf{v}^* - \mathbf{r} = -(\mathbf{r}^\top \mathbf{c} / \|\mathbf{c}\|_2^2) \mathbf{c} \in \text{span}(\mathbf{c}) = \mathcal{S}^\perp$, any feasible $\mathbf{v} \in \mathcal{S}$ satisfies

$$\|\mathbf{v} - \mathbf{r}\|_2^2 = \|\mathbf{v} - \mathbf{v}^*\|_2^2 + \|\mathbf{v}^* - \mathbf{r}\|_2^2. \quad (18)$$

The second term is fixed and the first is nonnegative, with equality only when $\mathbf{v} = \mathbf{v}^*$. Thus $\mathbf{v}^*$ is the unique solution to Eq. (6). This minimum-change property applies to the projection residual before normalization.

B Algorithm Details

Algorithm 1 presents the complete fitting and encoding procedure. The fitting stage learns the global mean and level-wise codebooks from the input embeddings. The encoding stage freezes these parameters and generates a hierarchical SID for each embedding.

Algorithm 1 PRQ-KMeans fitting and SID encoding

Input: embeddings $\{\mathbf{x}_i\}_{i=1}^N$, depth L , codebook sizes $\{K_\ell\}_{\ell=1}^L$, neighborhood size k , concentration β , and the number of refinement iterations

Output: global mean $\boldsymbol{\mu}$, codebooks $\{\mathcal{C}^{(\ell)}\}_{\ell=1}^L$, and hierarchical SIDs

Tokenizer Fitting

Global Component Removal

$\bar{\mathbf{x}}_i \leftarrow \text{norm}(\mathbf{x}_i)$ for all i

$\boldsymbol{\mu} \leftarrow N^{-1} \sum_i \bar{\mathbf{x}}_i$

$\mathbf{r}_i^{(1)} \leftarrow \text{norm}(\bar{\mathbf{x}}_i - (\bar{\mathbf{x}}_i^\top \boldsymbol{\mu} / \|\boldsymbol{\mu}\|_2^2) \boldsymbol{\mu})$ for all i

for $\ell = 1, \dots, L$ **do**

 Sample K_ℓ current residuals to initialize the centroids

Top-k Soft Centroid Refinement

for each prescribed refinement iteration **do**

$s_{ij}^{(\ell)} \leftarrow \cos(\mathbf{r}_i^{(\ell)}, \mathbf{c}_j^{(\ell)})$

$\mathcal{N}_{k,i}^{(\ell)} \leftarrow \text{TopK}_j s_{ij}^{(\ell)}$

$w_{ij}^{(\ell)} \leftarrow \frac{\exp(\beta s_{ij}^{(\ell)})}{\sum_{t \in \mathcal{N}_{k,i}^{(\ell)}} \exp(\beta s_{it}^{(\ell)})}$ for $j \in \mathcal{N}_{k,i}^{(\ell)}$; set it to zero

 otherwise

$\mathbf{c}_j^{(\ell)} \leftarrow \frac{\sum_i w_{ij}^{(\ell)} \mathbf{r}_i^{(\ell)}}{\sum_i w_{ij}^{(\ell)}}$

end for

Hard Assignment

$z_i^{(\ell)} \leftarrow \arg \max_j \cos(\mathbf{r}_i^{(\ell)}, \mathbf{c}_j^{(\ell)})$

$\mathbf{c}_i^{(\ell)} \leftarrow \mathbf{c}_{z_i^{(\ell)}}^{(\ell)}$

if $\ell < L$ **then**

Projection Residual

$\mathbf{r}_i^{(\ell+1)} \leftarrow \text{norm}\left(\mathbf{r}_i^{(\ell)} - \frac{\mathbf{r}_i^{(\ell)\top} \mathbf{c}_i^{(\ell)}}{\|\mathbf{c}_i^{(\ell)}\|_2^2} \mathbf{c}_i^{(\ell)}\right)$

end if

end for

SID Encoding

for each input embedding $\mathbf{x}_i$ **do**

$\bar{\mathbf{x}}_i \leftarrow \text{norm}(\mathbf{x}_i)$

$\mathbf{r}_i^{(1)} \leftarrow \text{norm}(\bar{\mathbf{x}}_i - (\bar{\mathbf{x}}_i^\top \boldsymbol{\mu} / \|\boldsymbol{\mu}\|_2^2) \boldsymbol{\mu})$

for $\ell = 1, \dots, L$ **do**

$z_i^{(\ell)} \leftarrow \arg \max_j \cos(\mathbf{r}_i^{(\ell)}, \mathbf{c}_j^{(\ell)})$

$\mathbf{c}_i^{(\ell)} \leftarrow \mathbf{c}_{z_i^{(\ell)}}^{(\ell)}$

if $\ell < L$ **then**

$\mathbf{r}_i^{(\ell+1)} \leftarrow \text{norm}\left(\mathbf{r}_i^{(\ell)} - \frac{\mathbf{r}_i^{(\ell)\top} \mathbf{c}_i^{(\ell)}}{\|\mathbf{c}_i^{(\ell)}\|_2^2} \mathbf{c}_i^{(\ell)}\right)$

end if

end for

 Store $(z_i^{(1)}, \dots, z_i^{(L)})$ as the SID of $\mathbf{x}_i$

end for

return $\boldsymbol{\mu}$, $\{\mathcal{C}^{(\ell)}\}_{\ell=1}^L$, and all stored SIDs

C Experimental Details

This section describes the public and industrial benchmark protocols, the baseline configurations, and the metrics and reproducibility settings used throughout the experiments.

C.1 Public Benchmark Setup

The public evaluation uses the GenRec data processing and TIGER encoder-decoder pipeline (Rajput et al. 2023; Wan et al. 2026) on Sports, Toys, Clothing, and LastFM. Tokenizers are fitted on 768-dimensional Sentence-T5-Base item representations (Ni et al. 2021), and every tokenizer produces a three-level SID with a 256-256-256 codebook. PRQ-KMeans uses $k = 2$ and $\beta = 15$.

All SIDs are evaluated with the same TIGER encoder-decoder, which has four encoder layers, four decoder layers, model dimension 128, feed-forward dimension 1024, and six attention heads. Training runs for at most 200 epochs with learning rate 10^{-4} , zero weight decay, training batch size 256, and inference batch size 96. The best validation checkpoint is selected with patience 10, and decoding uses beam size 30. The public results use item-level Recall and NDCG. While some previous work reports SID-level metrics, we use item-level Recall and NDCG to account for SID collisions (Zhang et al. 2026). Specifically, for each predicted SID with multiple mapped items, the evaluator randomly selects one representative item with fixed seed 42; duplicate item predictions are removed before metric computation.

C.2 Industrial Benchmark Setup

The industrial benchmark follows the OneSearch training and evaluation framework (Chen et al. 2025). Tokenizer fitting uses 7,797,542 items and 9,046,403 training-query records, for 16,843,945 representations in total. The 128-dimensional representations encode query text, item titles, prices, keywords, and OCR-derived text with a distilled BGE encoder aligned by query-query, item-item, and query-item objectives.

The three- and five-level settings use 1024-512-128 and 1024-512-128-64-64 SIDs, respectively. PRQ-KMeans uses $k = 5$ and $\beta = 15$. All tokenizers share BART-Base (Lewis et al. 2020) and the same three training stages: semantic content alignment with 35,689,588 examples, co-occurrence synchronization with 54,459,755 examples, and user personalization modeling with 45,916,427 examples. Training uses batch size 512 and learning rate 5×10^{-5} ; the three stages run for 4, 3, and 6 epochs, respectively.

Following OneSearch, we construct separate Click and Order test sets from search logs, each containing 30,000 query-item pairs with the corresponding observed behavior. Beam search returns 128 SIDs. For each SID, mapped items are ranked by availability and a composite score derived from historical click-through rate, conversion rate, click count, and order count, and the top five items are retained. The resulting item lists are expanded in beam order, stably deduplicated, and truncated to 50 items. We report HitRate and MRR under the cutoff defined in Section 5.1.

C.3 Baseline Configurations

The public and three-level industrial comparisons include RQ-VAE, R3-VAE, RQ-KMeans, RQ-GMM, PQ-KMeans, and OPQ-KMeans. In the five-level industrial comparison, RQ-OPQ uses two RQ-KMeans levels, a balanced K-Means third level, and two OPQ suffix tokens. ResKmeansFSQ uses

three RQ-KMeans levels and maps its L3 residual with a joint 12-bit FSQ code split into two 64-way suffix tokens. PRQ-OPQ replaces the first two RQ-KMeans levels of RQ-OPQ with PRQ-KMeans while retaining its balanced third-level K-Means and two OPQ suffix tokens.

For a fair comparison, each tokenizer uses its prescribed fitting or training schedule. RQ-VAE and R3-VAE are trained for 20 epochs with batch size 4,096 and learning rate 5×10^{-4} . RQ-KMeans and PQ-KMeans use 25 clustering iterations, OPQ-KMeans uses 25 alternating rotation and clustering iterations, and RQ-GMM uses at most 30 EM iterations with variance floor 10^{-6} . PRQ-KMeans uses 25 soft centroid-refinement iterations.

C.4 Metrics and Reproducibility

Let $\mathcal{S}$ be the set of distinct full SIDs assigned to items and let $n(s)$ be the number of items assigned to $s \in \mathcal{S}$. The independent-code ratio is

$$\text{ICR} = \frac{|\{s \in \mathcal{S} : n(s) = 1\}|}{|\mathcal{S}|}. \quad (19)$$

Its denominator is the number of distinct SIDs rather than the number of items. For level ℓ , let $\mathcal{P}_\ell$ be the set of occupied length- ℓ SID prefixes. Cumulative prefix utilization is

$$\text{Util}_\ell = \frac{|\mathcal{P}_\ell|}{\prod_{t=1}^\ell K_t}. \quad (20)$$

Used-prefix Gini is computed from the item frequencies of prefixes in $\mathcal{P}_\ell$ only. If $M_\ell = |\mathcal{P}_\ell|$ and $f_{(1)} \leq \dots \leq f_{(M_\ell)}$ are their sorted item frequencies, then

$$\text{Gini}_\ell = \frac{2 \sum_{q=1}^{M_\ell} q f_{(q)}}{M_\ell \sum_{q=1}^{M_\ell} f_{(q)}} - \frac{M_\ell + 1}{M_\ell}. \quad (21)$$

Unused prefixes are excluded, and a lower value indicates a more even distribution among the occupied prefixes. Table 1 reports ICR and utilization as percentages. In the three-level block of Table 1, ICR uses the complete three-token SID. In the five-level block, ICR uses the complete five-token SID, while utilization and Gini report each tokenizer’s L1–L3 prefix.

SID tokenizer fitting, SID encoding, and codebook analyses are conducted on a CPU server with two AMD EPYC 9654 processors (96 cores per socket) and 2.2 TB RAM. Downstream generative-retrieval training and evaluation are conducted on a separate server with eight NVIDIA H800 GPUs (80 GB HBM3 each) and two Intel Xeon Platinum 8558 processors (48 cores per socket).

D Additional Experimental Results

This section follows the order of the main analysis. It first provides the carryover measurement and controlled-update details used in the motivation, then reports the complete public ablations, public-benchmark sensitivity results, and two qualitative case studies.

D.1 Residual Carryover Analysis

For the industrial RQ-KMeans tokenizer, we average the carryover ratio in Eq. (4) over the full tokenizer-fitting population. After L2 normalization, the carryover ratio equals the absolute cosine between the next-level representation and the selected centroid. For two independent isotropic unit directions, the expected absolute cosine is

$$\mathbb{E}[|\mathbf{u}^\top \mathbf{v}|] = \frac{\Gamma(d/2)}{\sqrt{\pi}\Gamma((d+1)/2)}, \quad \mathbf{u}, \mathbf{v} \stackrel{\text{i.i.d.}}{\sim} \text{Unif}(\mathbb{S}^{d-1}). \quad (22)$$

For $d = 128$, this expectation is 7.07%, providing a dimensional scale for the values in Figure 2(a). Figure 2(a) reports $\bar{m}^{(\ell)}$ for the three levels.

The controlled update on the industrial dataset used in the main text can be written directly in terms of centroid subtraction and projection: let $\mathbf{c}_i = \mathbf{c}_{z_i^{(1)}}^{(1)}$, $\rho_i = \|\mathbf{c}_i\|_2$, $\hat{\mathbf{c}}_i = \mathbf{c}_i / \rho_i$, $\alpha_i = \mathbf{r}_i^{(1)\top} \hat{\mathbf{c}}_i$, and $\mathbf{u}_i = \mathbf{r}_i^{(1)} - \alpha_i \hat{\mathbf{c}}_i$.

$$\begin{aligned} \mathbf{r}_i^{(2)}(\eta) &= \text{norm} \left(\mathbf{r}_i^{(1)} - \left[\eta \mathbf{c}_i + (1 - \eta) \frac{\mathbf{r}_i^{(1)\top} \mathbf{c}_i}{\|\mathbf{c}_i\|_2^2} \mathbf{c}_i \right] \right) \\ &= \text{norm}(\eta(\alpha_i - \rho_i) \hat{\mathbf{c}}_i + \mathbf{u}_i). \end{aligned} \quad (23)$$

The second equality follows from Eq. (3) and shows that the operational update scales only the original carryover term before applying normalization. The L1 codebook and assignments remain fixed, and L2 is refitted for every value of η .

D.2 Public Benchmark Ablations

Table A1 reports the complete public results for the three ablations, consistent with the main-text ablation findings.

	Sports		Toys		Clothing		LastFM	
Variant	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG
PRQ-KMeans	0.0531	0.0219	0.0851	0.0361	0.0382	0.0151	0.0179	0.0076
– Global	0.0505	0.0202	0.0830	0.0359	0.0370	0.0151	0.0107	0.0044
– Soft	0.0515	0.0210	0.0815	0.0350	0.0371	0.0146	0.0043	0.0017
– Projection	0.0503	0.0201	0.0810	0.0345	0.0360	0.0141	0.0099	0.0044

Table A1: Ablation study on the public benchmarks.

D.3 Public-Benchmark Hyperparameter Sensitivity

We also evaluate hyperparameter sensitivity on Sports, Toys, Clothing, and LastFM. Figure A1 varies $k \in \{1, 2, 5, 10\}$ at fixed $\beta = 15$ and $\beta \in \{10, 15, 20, 25\}$ at fixed $k = 2$. For the neighborhood size, $k = 2$ gives the best or tied-best Recall and NDCG on all four benchmarks, while increasing k further provides no consistent gain. For the concentration parameter, $\beta = 15$ performs best on Sports, Clothing, and LastFM, whereas $\beta = 10$ performs best on Toys. Thus, the preferred concentration varies across datasets.

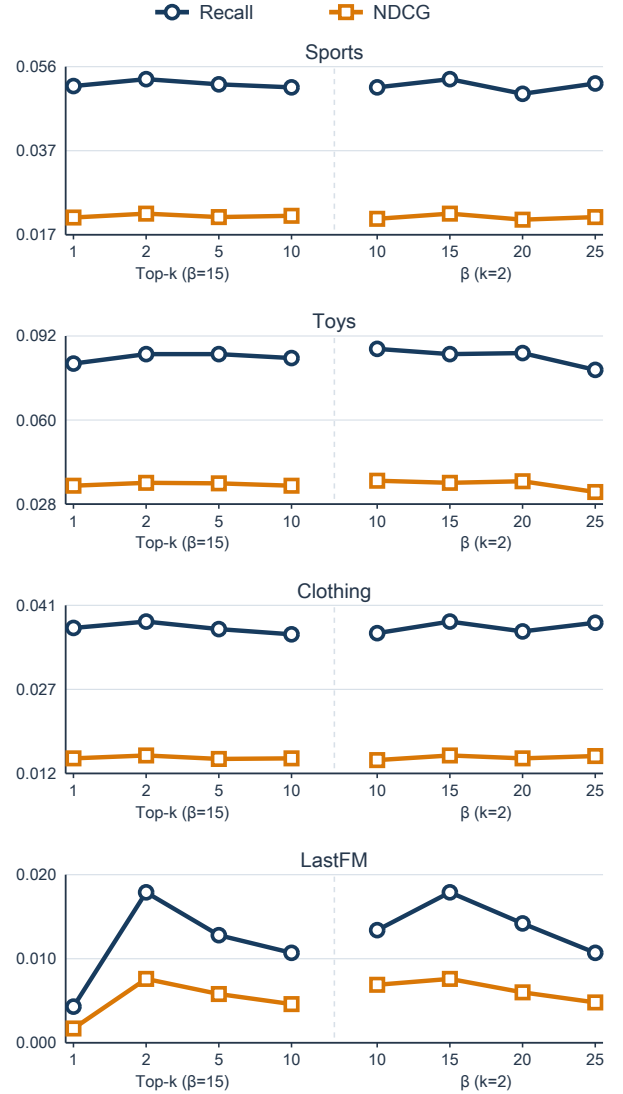


Figure A1: Hyperparameter sensitivity on the public benchmarks. Each dataset occupies one panel. The left segment varies Top- k with $\beta = 15$, and the right segment varies weight concentration with $k = 2$. Recall and NDCG follow the definitions in Section 5.1.

D.4 Qualitative Case Studies

Tables A2 and A3 examine two aspects of hierarchical SID organization. The collision case follows one fixed item group through the two tokenizers, and the weak-intent case compares the prefix neighborhoods associated with the same query.

For the collision case, we select a 12-item Kitty-themed slippers group using a fixed SID-structure criterion. The group shares one RQ-KMeans full SID. PRQ-KMeans retains one common level-1 prefix, separates the items into two level-2 prefixes, and produces three full SIDs with four

items each. Table A2 summarizes the keywords associated with the resulting PRQ-KMeans branches.

Full-SID branch	L2	Items	Keywords
A	A	4	Outdoor; dormitory; shower.
B	B	4	Indoor; soft-soled.
C	A	4	Bathroom; travel; home/outdoor.

Table A2: Keyword summaries for the selected 12-item collision group under PRQ-KMeans. Branch labels are anonymized.

For the pre-specified query translated as “hotel,” we identify the level-2 prefix assigned by each tokenizer and inspect the catalog items that share this prefix. The RQ-KMeans prefix contains 22 items and 14 distinct level-3 branches, while the PRQ-KMeans prefix contains 100 items and 29 branches. The latter neighborhood includes observed hotel-related contexts such as entrances, bathrooms, corridors, and commercial installations.

Method	Items under L2 prefix	Distinct L3 branches
RQ-KMeans	22	14
PRQ-KMeans	100	29

Table A3: Prefix-neighborhood statistics for the weak-intent query translated as “hotel.”