OneVision: An End-to-End Generative Framework for Multi-view E-commerce Vision Search

Anonymous ACL submission

Abstract

Traditional vision search follows the multi-stage cascading architecture (MCA) paradigm to balance efficiency and conversion, yet optimization objectives collide across stages, making it difficult to achieve Pareto optimality. Meanwhile, pronounced multi-view discrepancies of the same item undermine retrieval accuracy, as embedding-similarity methods fail to bridge large appearance variations. In this paper, we propose **OneVision**, an end-to-end generative framework that addresses both challenges. OneVision introduces VRQ, a vision-aligned residual quantization method that assigns consistent shallow-level codes to visually similar images of the same item while preserving fine-grained differences at deeper levels. It further employs a multi-stage training pipeline to retrieve all relevant items across views for a query image and incorporates user-specific signals for personalized preference modeling. In offline evaluations, OneVision performs on par with online MCA. Dynamic pruning further reduces its inference latency by 18% over the unpruned variant. In A/B tests, it achieves significant improvements: +3.61% item CTR, +2.43% CVR, and +5.89% order volume. The code will be publicly available.

1 Introduction

A vision search system takes an image as an input query and retrieves results with high visual relevance. Similar to other search engines, such systems are organized as a multi-stage pipeline, as depicted in Figure 1: (i) feature extraction, which obtains visual representations from regions of interest; (ii) the recall stage, which constructs a large candidate pool by retrieving visually similar items; (iii) the pre-ranking stage, which filters and refines the candidate pool; and (iv) the ranking stage, which orders the remaining candidates to deliver personalized results.

Recent years have seen substantial progress in vision search, driven by advances in object detection (Cheng et al., 2024; Tian et al., 2025; Wang et al., 2025a), category prediction (Zhu et al., 2022; Noor et al., 2023, 2024), and visual representation learning (Lee et al., 2022b; Shao et al., 2023; Liang et al., 2025; Siméoni et al., 2025), enabling large-scale industrial deployment on platforms such as eBay and Alibaba (Yang et al., 2017; Zhang et al., 2018; Chen et al., 2024). Despite this progress, traditional vision search pipelines suffer from several structural limitations: (i) *Multi-view discrepancy.* Images of the same item often vary in viewpoint, making it difficult to align embeddings across views and leading to missed recalls. (ii) *Stage-wise objective misalignment.* Optimization goals across stages conflict, preventing Pareto-optimal solutions and degrading both retrieval accuracy and conversion efficiency. (iii) *Multi-stage overhead.* Cascading architectures incur additional computation and cross-module communication, increasing system latency and reducing resource efficiency.

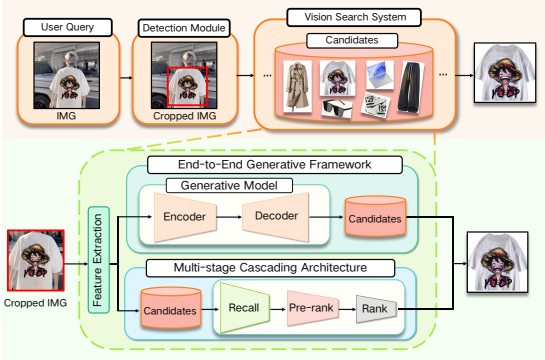


Figure 1: Generative vs. Traditional Multi-Stage Architectures for E-Commerce Vision Search.

Generative retrieval (GR) has emerged as a promising alternative by redefining retrieval as a sequence-to-sequence generation task that directly produces all matching item identifiers from a query and user-behavior sequences (Deng et al., 2025; Chen et al., 2025). For image retrieval, IR-Gen (Zhang et al., 2024), the first generative ap-

proach for image retrieval, formulates the task as seq2seq generation of nearest-neighbor IDs, and GENIUS (Kim et al., 2025) introduces modality-decoupled semantic quantization with query augmentation. However, both methods overlook the multi-view discrepancy of the same item, fine-grained discriminative encoding, and user-specific information, making them suitable only as a single recall channel rather than as standalone end-to-end vision search systems.

To address these limitations, we propose OneVision, an end-to-end vision search system for e-commerce that unifies recall, pre-ranking, and ranking. Our approach captures multi-view representations and incorporates user behavioral signals, enabling candidates that are both visually relevant and aligned with user preferences. Specifically, OneVision includes the following components:

1) **Vision-aligned residual quantization (VRQ) encoding.** VRQ employs multi-view contrastive learning to unify shallow-level codes across views, while applying OPQ at deeper levels to preserve fine-grained item-specific distinctions. It further incorporates statistical business attributes to enhance codebook utilization and support personalized generative ranking. Category information is further incorporated to align codes among items of the same category.

2) **Multi-stage pipeline and personalized generation.** To enable preference-aware generative retrieval, we adopt a unified multi-stage training paradigm: Pretraining, Multi-view SFT, and Personalized Post-training. Pretraining establishes initial coarse semantic alignment. Multi-view SFT encourages cross-view collaborative representation learning, leading to more complete candidate generation. Finally, user behavior sequences and listwise DPO are used to steer generation toward items better aligned with personalized ranking objectives.

3) **Dynamic pruning for efficient inference.** To reduce redundancy in image tokens, we prune less informative visual tokens before the generative encoder using K-means++ clustering. Curriculum learning and distillation are further applied to preserve retrieval quality. Beyond accelerating inference and reducing computational cost, this strategy even slightly improves retrieval quality by suppressing redundant visual tokens.

Extensive offline evaluations show that OneVision matches the performance of online MCA while substantially reducing end-to-end latency. Dynamic pruning further reduces inference latency by 18% over the unpruned variant. Large-scale online A/B tests on our industrial search platform further reveal notable improvements: item CTR +3.61%, PV CTR +3.43%, CVR +2.43%, OPM +4.53%, and Orders +5.89%. These results demonstrate that OneVision effectively unifies retrieval and personalization, delivering measurable gains while simplifying the traditional serving pipeline.

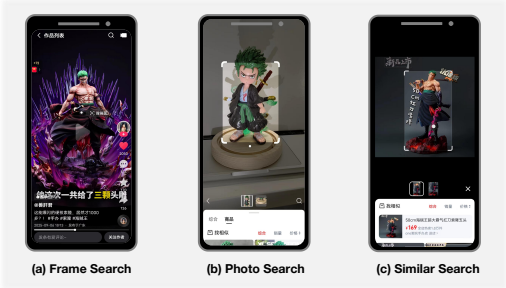


Figure 2: The major vision search scenarios in e-commerce on our industrial platform.

2 Related Works

2.1 Vision Search System

Modern image-based search systems build on advances in object detection, category prediction, and representation learning to produce compact visual embeddings. Large-scale approximate nearest neighbor (ANN) retrieval constructs candidate pools that are then refined through pre-ranking and ranking (Nadkarni and Dasararaju, 2022; Dagan et al., 2023; You et al., 2024). This paradigm has been widely adopted industrially: eBay uses deep region-based features with ANN for item discovery (Yang et al., 2017); Alibaba integrates category-aware embeddings with cascaded ranking (Zhang et al., 2018); and JD.com supports real-time visual search via distributed indexing (Li et al., 2018).

2.2 Generative Retrieval

Instead of querying an external index, GR trains models to directly generate item identifiers (e.g., semantic IDs), outperforming dense-retrieval baselines such as EBR (Huang et al., 2020) and RocketQA (Qu et al., 2020). Foundational work includes DSI (Tay et al., 2022), which generates learned identifiers for direct retrieval, and Tiger (Rajput et al., 2023), which derives semantic IDs from item content for end-to-end recommendation. LC-REC (Zheng et al., 2024) further adapts LLMs by integrating collaborative semantics.

A key design choice in GR is the encoding method. RQ-VAE (Lee et al., 2022a) employs

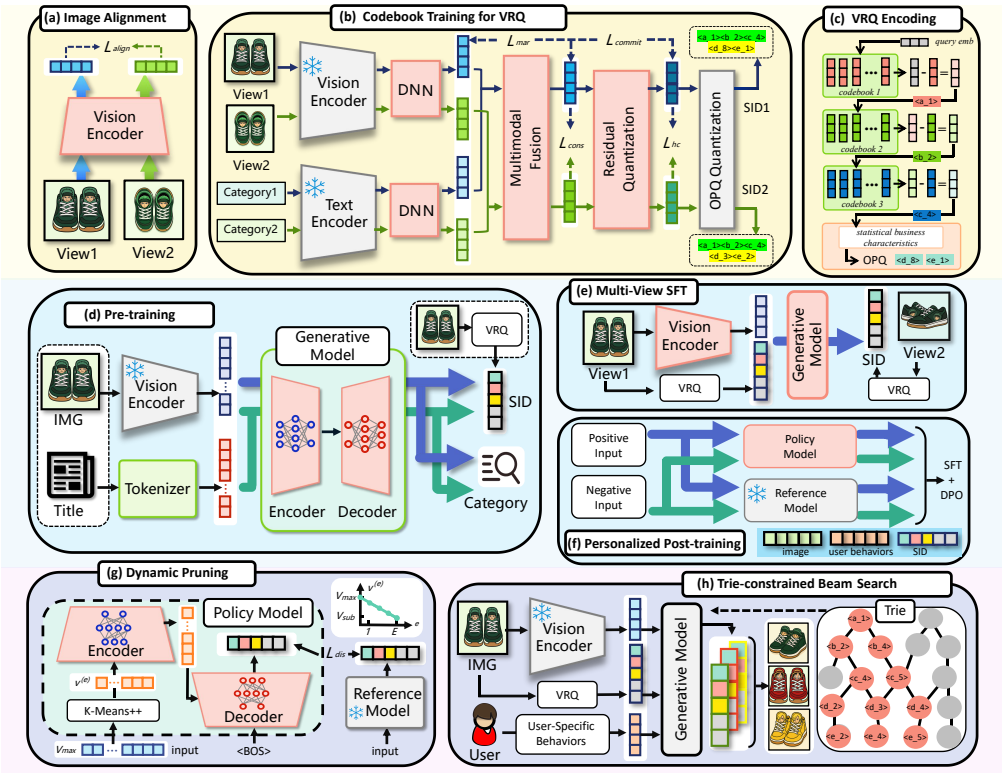


Figure 3: Overall Framework of OneVision. The first row illustrates our VRQ codebook training and encoding process. The second row shows our multi-stage generative training pipeline, including Pretraining, Multi-view SFT, and Personalized Post-training. The third row presents our dynamic pruning and end-to-end inference process.

residual quantization for compact yet high-capacity codebooks. QARM (Luo et al., 2024) aligns multi-modal representations into learnable SIDs through residual K-means. FSQ (Mentzer et al., 2023) replaces vector quantization with bounded scalar quantizers to improve codebook utilization.

Recent work has demonstrated GR's viability as a replacement for traditional recall-ranking stacks. OneRec (Deng et al., 2025) unifies recall and ranking via session-wise generation and DPO. MMQ (Xu et al., 2025) learns semantic IDs from multimodal content via behavior-aware fine-tuning. In e-commerce, OneSearch (Chen et al., 2025) shows that end-to-end generative models can unify retrieval and personalization. For image retrieval, IRGen (Zhang et al., 2024), the first such approach, formulates the task as seq2seq generation of nearest-neighbor IDs, and GENIUS (Kim et al., 2025) introduces modality-decoupled semantic quantization with query augmentation. Building on this line, OneVision further addresses multi-view alignment, fine-grained encoding, and personalized ranking.

3 Methodology

OneVision addresses multi-view discrepancy, stage-wise objective misalignment, and inference efficiency through three components: (1) VRQ, which aligns shallow-level codes across views while injecting item-specific signals into deeper levels for fine-grained discrimination (§ 3.1); (2) a multi-stage generative training pipeline that builds cross-view robustness and incorporates user behavior for personalized generation (§ 3.2); and (3) dynamic pruning, which removes redundant visual tokens to reduce computation without sacrificing performance (§ 3.3). Figure 3 illustrates the overall framework. Further training and inference details are provided in Appendix A.

3.1 Vision-aligned Residual Quantization

Semantic encoding of images is crucial for generative vision search; however, existing quantization methods for generative vision search face two challenges: (i) *Multi-view inconsistency.* Images of the same item under different viewpoints are assigned divergent SIDs, hindering generative model learning. For example, FSQ (Mentzer et al., 2023) and OPQ (Ge et al., 2013) lack hierarchical multi-view modeling. (ii) *Low discriminability.* Visually similar images of the same item share identical SIDs, resulting in low codebook utilization. Methods such as VQ-VAE (Van Den Oord et al., 2017), RQ-VAE (Lee et al., 2022a), and RQ-Kmeans (Luo

et al., 2024) are limited by insufficient code diversity. To address both, we propose VRQ, which employs multi-view contrastive learning to optimize trainable shallow RQ codebooks and integrates statistical business signals into deeper residual OPQ encoding for fine-grained discrimination. Category information is further incorporated to suppress noisy SIDs caused by main-object misrecognition.

3.1.1 Multi-View Contrastive Learning

Effective vision search requires aligning similar and multi-view item images while preserving category consistency. We first apply the alignment loss $\mathcal{L}_{\text{align}}$ (Chen et al., 2020a) to train the backbone $\mathcal{V}$ on e-commerce search logs, learning contrastive visual representations from visually similar image pairs. These aligned embeddings provide a foundation for subsequent VRQ encoding.

For VRQ encoding, given an item image and its category label, we extract the visual feature x using the image encoder $\mathcal{V}$ and the category feature y using a pretrained BGE model (Xiao et al., 2024). We then learn latent representations $v := \mathcal{E}_v(x)$ and $t := \mathcal{E}_t(y)$, where $\mathcal{E}_v$ and $\mathcal{E}_t$ denote the visual and textual DNN encoders. The fused embedding is computed as (Baldrati et al., 2023):

$$f = (1-\alpha)\cdot v + \alpha \cdot t + f_{\text{concat}}, \tag{1}$$

where f_{concat} is derived from the concatenated image-text embedding via an MLP, and α is learned by another MLP with sigmoid activation. Our multi-view contrastive objective comprises fused feature consistency and cross-modal alignment:

$$\mathcal{L}_{\text{cons}} = -\frac{1}{N}\sum_{i=1}^{N}\Bigg[\log\frac{\exp\left(f_i^{(1)}\cdot f_i^{(2)}/\tau\right)}{\sum_{j=1}^{N}\exp\left(f_i^{(1)}\cdot f_j^{(2)}/\tau\right)} + \log\frac{\exp\left(v_i^{(1)}\cdot f_i^{(2)}/\tau\right)}{\sum_{j=1}^{N}\exp\left(v_i^{(1)}\cdot f_j^{(2)}/\tau\right)}\Bigg], \tag{2}$$

where $f_i^{(1)}$ and $f_i^{(2)}$ denote fused embeddings of the i-th positive pair, N is the batch size of positive pairs, and τ is a temperature parameter. To prevent the model from relying solely on visual features and to strengthen category awareness (Chen et al., 2023; Liang et al., 2025), we add a margin loss:

$$\mathcal{L}_{\text{mar}} = \frac{1}{N}\sum_{i=1}^{N}\max\left(0,\ \gamma + s_{ii}^{v2f} - s_{ii}^{f2f}\right), \tag{3}$$

where s_{ii}^{f2f} and s_{ii}^{v2f} denote fused-fused and visual-fused similarity respectively, and γ is a margin parameter.

3.1.2 Hierarchical Quantization

The fused embedding f is quantized by an L-level RQ-VAE with codebook $E_l = \{e_k^{(l)} \in \mathbb{R}^d \mid k = 1,\ldots,K_l\}$ at level l. The code c_l is determined from the residual r_l ($r_0 = f$) as

$$c_l = \arg\min_k \|r_l - e_k^{(l)}\|, \tag{4}$$

with $r_{l+1} = r_l - e_{c_l}^{(l)}$. The resulting SID is $(c_0,\ldots,c_{L-1})$ and the quantized representation at level l is $\hat{f}^{(l)} = \sum_{l'=0}^{l} e_{c_{l'}}^{(l')}$. The commitment loss in RQ-VAE (Lee et al., 2022a) is

$$\mathcal{L}_{\text{commit}} = \sum_{l=0}^{L-1}\|r_l - \text{sg}(e_{c_l}^{(l)})\|^2, \tag{5}$$

where sg$(\cdot)$ is the stop-gradient operator and codebooks are updated via an exponential moving average (EMA) (Razavi et al., 2019). A hierarchical consistency loss enforces encoding consistency between the two views of a multi-view pair:

$$\mathcal{L}_{\text{hc}} = \sum_{l=0}^{L-1}\|\hat{f}^{(l,1)} - \hat{f}^{(l,2)}\|^2, \tag{6}$$

and the overall VRQ training loss is:

$$\mathcal{L}_{\text{rq}} = \lambda_1\mathcal{L}_{\text{cons}} + \lambda_2\mathcal{L}_{\text{mar}} + \lambda_3\mathcal{L}_{\text{commit}} + \lambda_4\mathcal{L}_{\text{hc}}, \tag{7}$$

where λ_i are adjustable weights. Notably, we initialize RQ-VAE codebooks with RQ-Kmeans cluster centroids for more robust training.

While RQ-VAE captures coarse-to-mid level semantics, to better distinguish visually identical items, we further enhance SID uniqueness by fusing the RQ-VAE residual embeddings that capture fine-grained visual attributes with statistical business attributes (e.g., clicks, price, purchase counts) and applying OPQ for residual encoding. This improves codebook diversity and enables unified personalized generation.

3.2 Multi-stage Training Pipeline

Traditional retrieval pipelines struggle with multi-view variation, limiting the joint optimization of relevance and personalization. OneVision addresses this issue with a multi-stage training pipeline and a two-stage architecture composed of a vision encoder and a GR model. The vision encoder extracts query visual tokens, while the GR model performs semantic reasoning and candidate item generation. For practical deployment, we instantiate the GR model with an encoder-decoder backbone for efficient inference.

3.2.1 Pretraining for Visual Alignment

Since the backbone models are pretrained on textual corpora, while vision search relies on image representations, we introduce a pretraining stage to establish initial alignment among item images, titles, categories, and SIDs. We design four related tasks: (i) predicting the item SID from its image; (ii) predicting the SID from both image and title; (iii) predicting the item category from the image; and (iv) predicting the category from both image and title. The first task aligns visual representations with SIDs, while the others reinforce cross-modal semantic relevance and category awareness. We adopt the standard next-token prediction loss (Lewis et al., 2019; Xue et al., 2020; Yang et al., 2025) for training.

3.2.2 Multi-View Supervised Fine-Tuning

Existing GR methods mainly retrieve items with the same SID as the query, but this ignores viewpoint variations and user interaction signals, degrading retrieval quality and conversion. VRQ partially addresses this by assigning consistent shallow-level codes across visually similar images of the same item. However, under large cross-view visual discrepancies, contrastive objectives alone cannot guarantee reliable shallow-level alignment. To address this, we introduce a multi-view SFT stage. Using training pairs constructed from main–auxiliary item images, item-to-item (i2i) relations, and user-interaction logs, the model is trained to predict same-item SIDs across views, learning a view-complete SID set that supports consistent item retrieval and subsequent personalized ranking.

3.2.3 Personalized Post-training

To model user behavior, we incorporate two complementary signals: a long-term representation aggregated from SIDs of recently clicked or purchased items, and a short-term sequence of SIDs from the user's five most recent image-search clicks (Chen et al., 2025). The model is then trained with personalized SFT under the NTP loss.

To further align generation with user preferences, we adopt a list-wise DPO objective that assigns higher likelihoods to preferred items. Within each session s with context x_s, user interactions are partitioned into purchased, clicked-but-not-purchased, and exposed-but-not-clicked items, forming a natural preference order. Let the session item set be $\mathcal{Y}_s = \{y^+, y_1^-, y_2^-, \ldots, y_m^-\}$, where y^+ denotes an item from a higher-preference tier and y_j^- denotes less-preferred items. The DPO objective is:

$$\mathcal{L}_{\mathrm{DPO}} = \mathbb{E}_s \left[\log \left(1 + \sum_{y^- \in \mathcal{Y}_s^-} \exp\big(z(y^-, y^+)\big) \right) \right], \tag{8}$$

where $\mathcal{Y}_s^-$ is the negative set, and $z(y^-, y^+) = \beta \left(\log \frac{\pi_\theta(y^-|x_s)}{\pi_{\mathrm{ref}}(y^-|x_s)} - \log \frac{\pi_\theta(y^+|x_s)}{\pi_{\mathrm{ref}}(y^+|x_s)} \right)$. Here, β is the inverse temperature parameter, π_θ is the policy model, and π_{ref} is the fixed reference model.

3.3 Dynamic Pruning for Efficient Inference

3.3.1 Dynamic Pruning

Image token encoding in the visual encoder is often redundant, and beam search decoding, which expands up to B hypotheses at each step, further amplifies this overhead. We propose a dynamic pruning strategy that retains only the most informative visual tokens, reducing inference latency while maintaining accuracy.

Inspired by (Omri et al., 2025; Wang et al., 2025b), given a query image with $V_{\max}$ visual tokens, we apply K-means++ (Arthur and Vassilvitskii, 2006) to form V_{sub} clusters and retain only the token closest to each cluster center. To mitigate information loss, we adopt a distillation framework where a frozen post-trained model serves as the teacher and the pruned model as the student. We also retain the SFT objective during pruning to stabilize generation. The distillation term is:

$$\mathcal{L}_{\mathrm{dis}} = \sum_{k=1}^{T} \mathrm{KL}\big(\pi_{\mathrm{ref}_p}(\cdot|u, t_{<k}) \,\|\, \pi_{\theta_p}(\cdot|u, t_{<k})\big), \tag{9}$$

where π_{ref_p} and π_{θ_p} denote the teacher and student models, KL is computed over next-token distributions, and T is the output length. Following curriculum learning (Wang et al., 2021b), the number of retained tokens $v^{(e)}$ decreases over training epochs:

$$v^{(e)} = \left\lfloor V_{\max} - \frac{e}{E}\big(V_{\max} - V_{\mathrm{sub}}\big) \right\rfloor, \; e = 1, \ldots, E, \tag{10}$$

where E is the total epochs and $v^{(E)} = V_{\mathrm{sub}}$.

3.3.2 End-to-End Inference

Given a query image and user context, the vision encoder first extracts visual tokens, which are then compressed by dynamic pruning. The GR model autoregressively generates candidate item SIDs via beam search. The top-B sequences are retained by cumulative log-probability, and constrained decoding with a Trie structure (Fredkin, 1960; De Cao

Table 1: Comparison of Encoding Strategies and Retrieval Paradigms on CLICK and ORDER Benchmarks. Ours (w/o P.) and Ours (w/ P.) denote the results without and with the incorporation of personalization signals and additional business information.

Methods		CLICK (8.6K)					ORDER (8.6K)				
		HR@1	HR@4	MRR@4	HR@10	MRR@10	HR@1	HR@4	MRR@4	HR@10	MRR@10
Traditional Retrieval											
	DINOv3	49.68%	72.11%	54.78%	77.66%	55.63%	44.95%	70.92%	54.24%	76.80%	55.15%
	Online MCA	51.01%	74.82%	60.16%	**83.89%**	61.37%	50.84%	74.92%	59.83%	**82.85%**	60.51%
Generative Retrieval											
RQ-Kmeans	IRGen	38.29%	57.29%	46.19%	66.56%	47.72%	37.22%	56.26%	46.19%	66.13%	46.70%
	GENIUS	42.43%	62.50%	50.93%	70.72%	52.30%	41.19%	61.70%	49.74%	69.95%	51.15%
	Ours (w/o P.)	45.30%	63.70%	52.95%	72.10%	54.36%	44.61%	63.05%	52.18%	72.04%	53.68%
	Ours (w/ P.)	48.91%	68.67%	57.14%	77.41%	58.60%	48.05%	68.38%	56.37%	77.38%	57.88%
VRQ	IRGen	44.13%	64.72%	52.90%	71.04%	53.96%	41.71%	63.48%	50.86%	70.13%	52.01%
	GENIUS	45.22%	65.51%	54.00%	71.22%	54.98%	42.84%	64.05%	51.91%	70.43%	52.98%
	Ours (w/o P.)	50.56%	73.51%	57.91%	78.50%	59.07%	48.57%	72.22%	58.20%	76.27%	58.47%
	Ours (w/ P.)	**51.46%**	**77.28%**	**61.77%**	82.29%	**62.46%**	**51.42%**	**76.28%**	**60.89%**	80.23%	**61.71%**

Table 2: Ablation Study of OneVision with Different Encoding Designs and Additional Information Signals.

Configuration	CLICK (8.6K)		ORDER (8.6K)	
	HR@10	MRR@10	HR@10	MRR@10
Online MCA	**83.89%**	61.37%	**82.85%**	60.51%
—w/o ranking	81.45%	56.87%	80.14%	55.47%
RQ–Kmeans	77.41%	58.60%	77.38%	57.88%
RQ–OPQ (2/256)	80.72%	61.51%	79.82%	60.19%
VRQ (w/o Stat. Emb.)	78.98%	59.72%	77.58%	58.02%
VRQ (Ours)	82.29%	**62.46%**	80.23%	**61.71%**

et al., 2020) built from all valid SIDs ensures that only valid sequences are produced.

4 Experiments

In this section, we evaluate OneVision on industrial-scale e-commerce datasets with online A/B testing to validate its effectiveness and practical applicability for real-world generative vision search.

4.1 Implementation Details

Datasets. We construct training datasets from large-scale industrial e-commerce search logs, totaling about 50 million samples, including query–item image pairs annotated with SIDs, item-to-item associations, and 10 million user search logs spanning 4 months. In addition, the vision-alignment and VRQ codebook stages use separate same-item image-pair data, as detailed in Appendix A.2. For evaluation, we use three test sets: a real-world visual search interaction set (8.6K click and 8.6K purchase events), a large-scale semantic benchmark (67K query–item pairs), and a fashion-industry multi-view benchmark (50K main–auxiliary image pairs). Full details are provided in Appendix A.2.

Evaluation Metrics. We evaluate recall and ranking performance with HitRate (HR) and Mean Reciprocal Rank (MRR), averaged over all experiments. For VRQ quality, we also report Quantized ANN Score (QAS) for visual clustering and Independent Code Occupancy (ICO) for code utilization. Higher QAS and lower ICO indicate better encodings, and detailed definitions are provided in Appendix A.4. We further benchmark OneVision against the online MCA, which comprises recall, pre-ranking, and ranking stages, to validate its end-to-end image-based search performance.

Implementation of OneVision. We use a 12-layer ViT encoder (hidden size 768) to extract visual representations, fed into a BART model with 6 encoder and 6 decoder layers (hidden size 768). The VRQ SID design uses 5 levels (three RQ-VAE levels and two residual OPQ levels), with codebook sizes $K = (2048, 512, 256 \mid 256, 256)$. We adopt constrained beam search with beam size 10 for offline evaluation and 256 for online inference. Full training details are provided in Appendix A.5.

4.2 Real-World Offline Evaluation

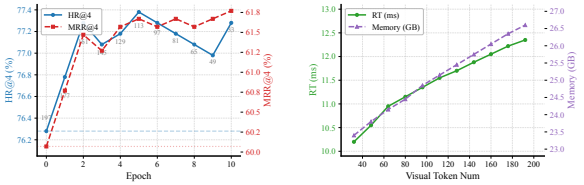
Experiment on the CLICK and ORDER Benchmarks. As shown in Table 1, our method consistently improves retrieval performance on the real-world search interaction set, reaching and slightly surpassing online MCA. While RQ-Kmeans-based GR underperforms systems using DINOv3 features with ANN retrieval, switching the SID encoder to our VRQ yields average relative gains of 8.83% (HR@1) and 11.06% (HR@4) on CLICK and ORDER sets. This improvement stems from replacing unsupervised RQ–Kmeans with a contrastively trained RQ-VAE, which reduces SID imbalance and cluster underuse, while residual OPQ refines visual residuals for more uniform, semantically aligned SIDs. Building on VRQ, OneVision out-

Table 3: Comparison of Retrieval Methods on the Semantic Benchmark across Top, Middle, and Long-tail industries. The benchmark is constructed via online ANN recall, so the unquantized online model attains near-100% HR by construction; QAM denotes its quantized counterpart (Quantized ANN Method). VLM adopts InternVL3.5-1B.

Methods	22K Top			22K Middle			22K Long-tail		
	HR@1	HR@4	MRR@4	HR@1	HR@4	MRR@4	HR@1	HR@4	MRR@4
Embedding-based Retrieval									
DINOv3	79.12%	87.56%	84.14%	78.61%	88.86%	82.97%	77.46%	87.41%	82.10%
QAM	81.45%	90.34%	85.08%	72.77%	85.10%	77.74%	69.57%	83.54%	75.23%
Generative Retrieval									
IRGen	88.49%	94.92%	91.28%	84.19%	92.65%	87.81%	80.42%	91.93%	85.32%
GENIUS	86.63%	94.06%	89.84%	82.49%	91.78%	86.45%	78.88%	91.08%	84.09%
VLM	85.77%	91.80%	88.41%	77.01%	88.51%	81.92%	71.24%	85.94%	77.29%
Ours	**89.29%**	**96.53%**	**93.30%**	**85.19%**	**93.70%**	**89.55%**	**81.89%**	**92.15%**	**86.09%**

Table 4: Comparison of Encoding Methods on Independent Code Occupancy (ICO), Quantized ANN Score (QAS), and Generative Retrieval (GR).

Method	ICO ↓	QAS ↑		GR ↑	
		HR@4	MRR@4	HR@4	MRR@4
RQ-Kmeans	4.84	81.76%	74.66%	89.98%	85.42%
FSQ	**2.17**	**98.47%**	**96.99%**	36.56%	23.32%
OPQ	4.83	84.70%	79.05%	86.07%	78.83%
RQ-VAE (Tiger)	4.91	69.48%	61.41%	79.60%	68.80%
RQ-VAE (GENIUS)	4.23	83.75%	77.18%	92.34%	87.15%
RQ-OPQ (2/256)	3.92	85.15%	78.23%	93.28%	88.80%
VRQ (Ours)	3.78	86.33%	79.35%	**94.13%**	**89.65%**



(a) Accuracy vs. token count (b) Efficiency vs. token count

Figure 4: Impact of Visual Token Number on Retrieval Accuracy (HR@4, MRR@4) and System Efficiency.

performs IRGen and GENIUS, improving HR@1/4 by 13.57% and MRR@10 by 9.92% on average. Integrating personalization signals and statistical business information further adds +3.22 HR and +3.30 MRR. Notably, our best model VRQ (w/ P.) slightly exceeds online MCA (e.g., CLICK HR@1 51.46% vs 51.01%, ORDER HR@1 51.42% vs 50.84%, and averaged +1.15pt MRR@10 across CLICK and ORDER), showing that end-to-end generative retrieval can rival and even exceed well-tuned ANN-based industrial systems.

Ablation Study of OneVision. As shown in Table 2, removing the ranking stage from online MCA degrades retrieval quality, confirming its necessity. Replacing RQ-Kmeans with RQ-OPQ yields an average relative improvement of 4.10% in HR and MRR, as residual OPQ enhances SID stability and fine-grained representation. Consequently, the VRQ with multi-stage training achieves the highest scores, rivaling the deployed online system. Adding business statistics (Stat. Emb.) yields +2.98 HR and +3.22 MRR by helping separate near-duplicate items.

The Effectiveness of Curriculum-based Dynamic Pruning. As shown in Figure 4 (a), HR@4 and MRR@4 are maintained or even slightly improved when visual tokens are aggressively reduced from 197 to 33, indicating that many tokens are redundant and that pruning noisy patches helps refine visual representations. Figure 4 (b) further shows that this strategy reduces real-time inference latency by about 18% and memory consumption by about 13%. Overall, dynamic pruning yields both efficiency gains and improved retrieval robustness by suppressing noisy tokens while keeping cleaner, more discriminative visual features.

4.3 Additional Semantic Offline Evaluation

Experiment on the Semantic Benchmark. Table 3 compares embedding-based and generative retrieval methods across the Top, Middle, and Long-tail industries. Quantizing the strong online embedding model for ANN retrieval (QAM) results in a pronounced drop in accuracy (e.g., averaged across the three tiers, HR@1 drops to 74.60% and MRR@4 drops to 79.35%), highlighting the sensitivity of embedding-based systems to aggressive quantization. Within generative retrieval, our image-centric GR model consistently outperforms other GR baselines: compared with IRGen and GENIUS, HR@1 improves by 1.99% on top industries and 2.82% on long-tail industries in relative terms, with similar gains in HR@4 and MRR@4. Despite their generality, current VLMs underperform at this scale: even with InternVL3.5-1B (Wang

Table 5: Ablation study of different training strategies. PT denotes pretraining, MV-SFT denotes multi-view supervised fine-tuning.

Model	HR@1	HR@4	HR@10	MRR@4	MRR@10
PT	34.19%	55.72%	61.67%	43.11%	44.09%
MV-SFT Only	36.48%	59.96%	65.96%	46.00%	47.37%
PT + MV-SFT	**38.96%**	**62.70%**	**68.92%**	**48.79%**	**49.80%**

Table 6: Online A/B Test Results (All Improvements Significant, $p < 0.05$). PV CTR denotes page-view click-through rate; OPM denotes orders per mille.

Method	Item CTR	PV CTR	CVR	OPM	Orders
Online MCA w/o ranking	-5.01%	-5.51%	-10.73%	-10.39%	-12.65%
OneVision	**+3.61%**	**+3.43%**	**+2.43%**	**+4.53%**	**+5.89%**

et al., 2025b), the model remains less effective than task-specific generative retrieval for large-scale e-commerce search.

Experiment on Different Encoding Methods. Table 4 compares encoding approaches under ICO, QAS, and GR. Methods focused on reconstruction quality but ignoring semantic alignment struggle under generative decoding: FSQ achieves strong QAS (HR@4 98.47%, MRR@4 96.99%) but performs poorly under GR (HR@4 36.56%, MRR@4 23.32%). Among RQ-VAE variants, the Tiger baseline without contrastive objectives yields HR@4 79.60% and MRR@4 68.80%, while GENIUS with contrastive learning improves to 92.34% and 87.15%. By adding OPQ-based residual decomposition on top, our VRQ further achieves HR@4 94.13% and MRR@4 89.65%, confirming that residual encoding with contrastive learning yields more discriminative and generation-friendly SIDs.

The Effectiveness of the Multi-view Modeling Capability. We evaluate the multi-view modeling capability of OneVision using 50K apparel item pairs from the fashion industry with diverse viewpoints. As shown in Table 5, Pretraining + Multi-View SFT achieves the best performance (HR@1 / HR@10 = 38.96 / 68.92), validating the effectiveness of multi-view supervised fine-tuning.

Multi-view Logits Visualization. To better illustrate the effect of multi-view SFT, Figure 5 visualizes OneVision's logits distributions as heatmaps, comparing R1 (pretraining-only, mapping a query image to its own SID) with R2 (R1 plus multi-view SFT, predicting SIDs of the same item across viewpoints). Across three representative examples, R1 yields high logits mainly for the query's own view, whereas R2 produces consistently high logits across multiple views, confirming that multi-view SFT enables view-complete SID generation.

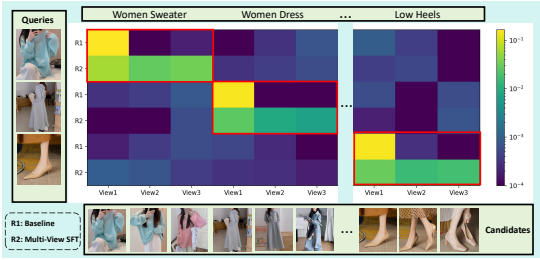


Figure 5: Logits Heatmap Comparison Between Baseline and Multi-view SFT. R1 predicts only the query SID, while R2 generalizes SID prediction across views.

4.4 Online A/B Tests

Online Performance Evaluation. We conduct online A/B tests on our industrial e-commerce search system to validate OneVision in real-world deployment. As shown in Table 6, we compare (1) online MCA (baseline), (2) online MCA without ranking (recall and pre-ranking only), and (3) OneVision, where items sharing the same code are further ranked using lightweight business signals. OneVision consistently outperforms online MCA, with +3.61% Item CTR, +2.43% CVR, and +5.89% in orders. In contrast, removing the ranking stage from online MCA causes substantial degradation (e.g., Orders drop by 12.65%), underscoring the central role of ranking in image-based search.

Online Efficiency Evaluation. The traditional online MCA pipeline incurs substantial latency across recall ($\sim$44 ms), pre-ranking ($\sim$10 ms), and ranking ($\sim$200 ms). In contrast, OneVision replaces the cascaded retrieval, pre-ranking, and ranking stages with a generative SID decoding pass followed by a lightweight SID lookup, reducing the end-to-end latency to about 83 ms. OneVision thus delivers stronger effectiveness and online efficiency while alleviating stage-wise misalignment. Full deployment and latency-breakdown details are provided in Appendix B.

5 Conclusion

We present OneVision, an end-to-end generative framework that bridges retrieval and personalization in e-commerce vision search. With vision-aligned residual quantization, multi-view semantic alignment, and semantic ID generation, OneVision handles multi-view discrepancies while preserving item distinctiveness. Offline evaluations and online A/B tests show that it simplifies the traditional multi-stage retrieval pipeline while improving accuracy and efficiency in industrial deployment.

Limitations

OneVision targets the recall–ranking back-end of visual search and does not subsume the upstream object-detection step, which we treat as an interactive disambiguation module. Empirically, OneVision narrows but does not fully close the gap to online MCA on two regimes: long-tail items, where embedding-based ANN can recover any visually similar item but generative decoding must *produce* an under-trained SID; and OCR-heavy industries such as books, where five-level discrete codes cannot preserve the fine textual cues that dense visual features retain. Closing these gaps likely requires either substantially larger generative backbones with stronger emergent generalization, or a hybrid design that re-introduces a dense-feature channel for text-dependent industries. A full discussion is provided in Appendix C.2.

Ethical Considerations

OneVision is developed for e-commerce vision search using large-scale industrial search logs and user behavior signals. All training, evaluation, and online deployment data are processed under strict privacy protection, anonymization, access control, and data retention policies. Since the model directly affects item retrieval and exposure, we monitor its performance across industries, item popularity groups, and business scenarios to reduce potential exposure bias, especially for long-tail items. Online evaluation and human auditing are also conducted to ensure that improvements in efficiency and conversion are achieved without compromising user experience, fairness, or platform trust.

References

David Arthur and Sergei Vassilvitskii. 2006. k-means++: The advantages of careful seeding. Technical report, Stanford.

Alberto Baldrati, Marco Bertini, Tiberio Uricchio, and Alberto Del Bimbo. 2023. Composed image retrieval using contrastive learning and task-oriented clip-based features. *ACM Transactions on Multimedia Computing, Communications and Applications*.

Ben Chen, Xian Guo, Siyuan Wang, Zihan Liang, Yue Lv, Yufei Ma, Xinlong Xiao, Bowen Xue, Xuxin Zhang, Ying Yang, et al. 2025. Onesearch: A preliminary exploration of the unified end-to-end generative framework for e-commerce search. *arXiv preprint arXiv:2509.03236*.

Ben Chen, Linbo Jin, Xinxin Wang, Dehong Gao, Wen Jiang, and Wei Ning. 2023. Unified vision-language representation modeling for e-commerce same-style products retrieval. In *Companion Proceedings of the ACM Web Conference 2023*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020a. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. 2020b. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*.

Xinlei Chen, Saining Xie, and Kaiming He. 2021. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*.

Xu Chen, Zida Cheng, Yuangang Pan, Shuai Xiao, Xiaoming Liu, Jinsong Lan, Qingwen Liu, and Ivor W Tsang. 2024. Branches, assemble! multi-branch cooperation network for large-scale click-through rate prediction at taobao. *arXiv preprint arXiv:2411.13057*.

Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. 2024. Yolo-world: Real-time open-vocabulary object detection. *arXiv preprint arXiv:2401.17270*.

Arnon Dagan, Ido Guy, and Slava Novgorodov. 2023. Shop by image: characterizing visual search in e-commerce. *Information Retrieval Journal*.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2020. Autoregressive entity retrieval. *arXiv preprint arXiv:2010.00904*.

Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. 2025. Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment. *arXiv preprint arXiv:2502.18965*.

Edward Fredkin. 1960. Trie memory. *Communications of the ACM*.

Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. 2013. Optimized product quantization. *IEEE transactions on pattern analysis and machine intelligence*.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based retrieval in facebook search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Sungyeon Kim, Xinliang Zhu, Xiaofan Lin, Muhammet Bastan, Douglas Gray, and Suha Kwak. 2025. Genius: A generative framework for universal multimodal search. In *Proceedings of the Computer Vision and Pattern Recognition Conference*.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. 2022a. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Seongwon Lee, Hongje Seong, Suhyeon Lee, and Euntai Kim. 2022b. Correlation verification for image retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Jie Li, Haifeng Liu, Chuanghua Gui, Jianyu Chen, Zhenyuan Ni, Ning Wang, and Yuan Chen. 2018. The design and implementation of a real time visual search system on jd e-commerce platform. In *Proceedings of the 19th International Middleware Conference Industry*.

Zihan Liang, Yufei Ma, ZhiPeng Qian, Huangyu Dai, Zihan Wang, Ben Chen, Chenyi Lei, Yuqing Ding, and Han Li. 2025. Uniecs: Unified multimodal e-commerce search framework with gated cross-modal fusion. *arXiv preprint arXiv:2508.13843*.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. 2024. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*.

Xinchen Luo, Jiangxia Cao, Tianyu Sun, Jinkai Yu, Rui Huang, Wei Yuan, Hezheng Lin, Yichen Zheng, Shiyao Wang, Qigen Hu, et al. 2024. Qarm: Quantitative alignment multi-modal recommendation at kuaishou. *arXiv preprint arXiv:2411.11739*.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. 2023. Finite scalar quantization: Vq-vae made simple. *arXiv preprint arXiv:2309.15505*.

Prajit Nadkarni and Narendra Varma Dasararaju. 2022. Visually similar products retrieval for shopsy. *arXiv preprint arXiv:2210.04560*.

Khondaker Tasrif Noor, Antonio Robles-Kelly, Leo Yu Zhang, and Mohamed Reda Bouadjenek. 2023. A bottom-up capsule network for hierarchical image classification. In *2023 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*.

Khondaker Tasrif Noor, Antonio Robles-Kelly, Leo Yu Zhang, Mohamed Reda Bouadjenek, and Wei Luo. 2024. A consistency-aware deep capsule network for hierarchical multi-label image classification. *Neurocomputing*.

Yasmine Omri, Parth Shroff, and Thierry Tambe. 2025. Token sequence compression for efficient multimodal computing. *arXiv preprint arXiv:2504.17892*.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2020. Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2010.08191*.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. 2023. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems*.

Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. 2019. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*.

Shihao Shao, Kaifeng Chen, Arjun Karpur, Qinghua Cui, André Araujo, and Bingyi Cao. 2023. Global features are all you need for image retrieval and reranking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. 2025. Dinov3. *arXiv preprint arXiv:2508.10104*.

Yi Tay, Vinh Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, et al. 2022. Transformer memory as a differentiable search index. *Advances in Neural Information Processing Systems*.

Yunjie Tian, Qixiang Ye, and David Doermann. 2025. Yolov12: Attention-centric real-time object detectors. *arXiv preprint arXiv:2502.12524*.

Aaron Van Den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems*.

Ao Wang, Lihao Liu, Hui Chen, Zijia Lin, Jungong Han, and Guiguang Ding. 2025a. Yoloe: Real-time seeing anything. *arXiv preprint arXiv:2503.07465*.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. 2025b. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*.

Xiaohui Wang, Ying Xiong, Yang Wei, Mingxuan Wang, and Lei Li. 2021a. Lightseq: A high performance inference library for transformers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*.

Xin Wang, Yudong Chen, and Wenwu Zhu. 2021b. A survey on curriculum learning. *IEEE transactions on pattern analysis and machine intelligence*.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*.

Yi Xu, Moyu Zhang, Chenxuan Li, Zhihao Liao, Haibo Xing, Hao Deng, Jinxin Hu, Yu Zhang, Xiaoyi Zeng, and Jing Zhang. 2025. Mmq: Multimodal mixture-of-quantization tokenization for semantic id generation and user behavioral adaptation. *arXiv preprint arXiv:2508.15281*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Fan Yang, Ajinkya Kale, Yury Bubnov, Leon Stein, Qiaosong Wang, Hadi Kiapour, and Robinson Piramuthu. 2017. Visual search at ebay. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*.

Xiaoyu You, Mi Zhang, Jianwei Xu, and Min Yang. 2024. I can retrieve more than images: Contrastive stealing attack against deep hashing models. In *2024 IEEE International Conference on Web Services (ICWS)*.

Yanhao Zhang, Pan Pan, Yun Zheng, Kang Zhao, Yingya Zhang, Xiaofeng Ren, and Rong Jin. 2018. Visual search at alibaba. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*.

Yidan Zhang, Ting Zhang, Dong Chen, Yujing Wang, Qi Chen, Xing Xie, Hao Sun, Weiwei Deng, Qi Zhang, Fan Yang, et al. 2024. Irgen: Generative modeling for image retrieval. In *European Conference on Computer Vision*.

Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. 2024. Adapting large language models by integrating collaborative semantics for recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*.

Lvxing Zhu, Hao Chen, Chao Wei, and Weiru Zhang. 2022. Enhanced representation with contrastive loss for long-tail query classification in e-commerce. In *Proceedings of the Fifth Workshop on e-Commerce and NLP (ECNLP 5)*.

A Training and Experimental Details

A.1 End-to-End Inference Details

This appendix elaborates on the end-to-end inference pipeline described in § 3.3.2, covering both the offline SID indexing and the online query-time generation.

Offline SID Indexing. Prior to online serving, all catalog items are encoded using the VRQ procedure in Figure 3 (c) into 5-level SIDs $(c_0, c_1, c_2 \mid c_3, c_4)$; the production indexing pipeline is described in Appendix B.2. The resulting SIDs are aggregated into a Trie of depth 5, where each root-to-leaf path represents a valid SID, and each leaf stores the list of items that share that SID. The Trie is rebuilt at each catalog refresh and acts as the constraint structure for decoding.

Online Inference Pipeline. Given a query image and user-specific context, OneVision performs the following steps:

1. *Visual encoding with dynamic pruning.* The query image is passed through the ViT encoder to obtain $V_{\max}$ visual tokens, which are reduced to V_{sub} representative tokens via K-means++ clustering, retaining only the token closest to each cluster centroid.

2. *User context fusion.* The pruned visual tokens are concatenated with user behavior tokens, comprising a long-term aggregated representation and a short-term sequence of SIDs from the user's five most recent image-search clicks.

3. *Constrained beam search decoding.* The BART decoder autoregressively generates SIDs token-by-token. At each step, the next-token distribution is masked by the Trie so that only valid prefixes are considered. Starting from $t_{[\text{BOS}]}$, the decoder maintains the top-B partial sequences ranked by cumulative log-probability, and continues until all beams reach $t_{[\text{EOS}]}$ or the maximum length of 5 is reached.

4. *Item lookup and ranking.* Each generated SID is mapped through the Trie back to its associated items, and items sharing the same SID

are further ordered by lightweight business signals.

Beam Configuration. We use $B = 10$ for offline evaluation and $B = 256$ for online serving; the rationale for the production beam size is given in Appendix B.3. Constrained decoding with the Trie guarantees that all B beams correspond to real catalog items, so no post-hoc filtering is needed.

The complete training and inference procedures are summarized in Algorithm 1 and Algorithm 2 (Appendix A.6).

A.2 Dataset Details

This appendix expands on the datasets introduced in § 4.1. All data are drawn from production search logs at our industrial e-commerce platform, with approximately 50M training samples in total.

Dataset accounting. The 50M samples refer to a shared generative training corpus that is reused (with stage-specific task formats) across Pretraining, Multi-view SFT, and Personalized Post-training. The image-alignment and VRQ codebook training stages use a separate same-item image-pair corpus, which is only used to train the visual alignment backbone and the VRQ codebooks. Therefore, the 30M image-pair corpus and the 50M generative training corpus correspond to different training stages, and should not be interpreted as conflicting dataset counts.

Training Data. The training corpus consists of three components: (1) query images and item images, each annotated with their corresponding SIDs; (2) item-to-item (i2i) associations representing semantically similar item pairs; and (3) approximately 10M user search logs covering 4 months of click and purchase interactions. For (2) and (3), each item is expanded with its main and auxiliary images so that every pair contributes multiple viewpoints of the same item.

Evaluation Data. We use three test sets covering retrieval quality, semantic coverage, and multi-view robustness: (1) *Real-world visual search interaction set:* 8.6K click events and 8.6K purchase events from 30 days of online visual search logs; (2) *Large-scale semantic benchmark:* 67K query–item pairs retrieved via online ANN recall, with approximately 22K pairs each from top, middle, and long-tail industries; (3) *Fashion-industry multi-view benchmark:* 50K main–auxiliary image pairs of the same item.

A.3 Evaluation Protocol

Multiple items can map to the same 5-level SID. We first introduce the deterministic rule for ordering items *within* the same SID, then explain how the specific value of K used in offline evaluation is chosen.

Item Ordering within Shared SIDs. Consider the degenerate case in which the codebooks collapse and all items share a single SID. Without a within-SID ordering rule, every query would trivially retrieve every item and report 100% recall, which is meaningless. A strict ordering protocol turns SID quality into a measurable property: if many items collapse to the same SID, only a small fraction of them can be matched to the corresponding queries, and recall will drop accordingly. For each SID we fix a deterministic ordering of its associated items prior to evaluation. When a query generates a SID containing k items, only the top-ranked one is counted as a hit for the matching query; later items at the same SID can be hits only for their own queries. Concretely, if items A and B both map to a SID and A is ordered first, then a query for A counts as a hit while a query for B does not, despite B being recalled by the model.

Choice of K in Offline Evaluation. All offline benchmarks in this paper report HR@K and MRR@K for $K \in \{1, 4, 10\}$. We do not extend K further; the reasons are practical rather than methodological.

1. *User Behavior on the Image-Search Surface.* On our image-search result page, the median position of a user's first click falls within the top ten results. Items ranked beyond position ten are rarely examined, so improvements in HR@K for $K > 10$ are not reflected in actual user-perceived quality. Top-10 therefore captures the part of the ranking that matters for production.

2. *Offline Pool Size.* Our offline evaluation pools contain on the order of a few million items, roughly two orders of magnitude smaller than the full online catalog. At this pool size, Top-10 already provides ample surface area to discriminate strong from weak retrieval systems; pushing K much higher would primarily reward systems that produce loosely related fillers and dilute the gap between methods.

3. *Relation to the Online Beam Size.* The Top-10 evaluation horizon is unrelated to the much

larger online beam size of $B = 256$ used at serving time (Appendix B.3). Online retrieval must surface several hundred candidates to feed downstream business-policy ranking over the full catalog of hundreds of millions of items, whereas offline evaluation measures retrieval quality directly on a fixed, smaller pool.

A.4 Evaluation Metrics for VRQ

A good SID encoding should both group visually similar items together and keep distinct items separable. We measure these two properties with Quantized ANN Score (QAS) and Independent Code Occupancy (ICO).

Quantized ANN Score (QAS). QAS measures how faithfully VRQ preserves semantic similarity after quantization. Given a set of visually similar item pairs whose pretrained representations already exhibit high ANN similarity, we recompute their ANN similarity using the quantized representations produced by VRQ. QAS is the average similarity over all pairs; a higher QAS indicates less quantization loss.

Independent Code Occupancy (ICO). ICO measures code utilization as the average number of items assigned to each unique SID. A large ICO implies that many items collapse to the same code, reducing discriminability; a small ICO indicates that the codebook is well utilized at the item level.

A.5 Training Details

This appendix complements § 3.1–§ 3.3 with the full training configurations of OneVision. All stages use AdamW with weight decay 0.01. The ViT learning rate is set to one-tenth of the BART learning rate to stabilize visual fine-tuning. The full training pipeline is summarized in Algorithm 1 (Appendix A.6).

Image Alignment. We pre-train the ViT-Base backbone on ∼30M same-item image pairs from search logs (main–auxiliary item images and click-through pairs), combining a MoCo-style (He et al., 2020; Chen et al., 2020b, 2021) contrastive loss (queue size 65,536, momentum 0.99, temperature $\tau = 0.07$) with a triplet loss (margin 0.4). A 2-layer MLP projection head (768→256→64) is attached. Trained with learning rate 1×10^{-4}, effective batch size 256×8, 10 epochs.

VRQ Codebook. The fused embedding (Eq. 1) takes as input the 64-dim aligned visual feature and the 64-dim BGE-encoded category feature, with the gating coefficient α initialized to 0.9. We set λ_1=λ_2=λ_3=λ_4=1.0, contrastive temperature $\tau = 0.07$, and margin $\gamma = 0.2$ (Eq. 3). Codebooks are initialized from RQ-Kmeans centroids and updated via EMA. Trained with learning rate 1×10^{-4}, batch size 4,096, 150 epochs under the SimCLR paradigm. The two OPQ levels operate on the third-level RQ-VAE residual after fusing it with PCA-reduced (128→64) statistical business attributes (clicks, price, purchase counts) at a 9:1 residual-to-statistical ratio; the conservative weighting injects item-specific signals while preserving visual semantics.

Stage 1: Pretraining. We mix four next-token-prediction tasks with a sampling ratio of 7:1:1:1: SID prediction from images, SID prediction from image-title pairs, category prediction from images, and category prediction from image-title pairs. This prioritizes the core image-to-SID mapping. The model is trained for 10 epochs with a batch size of 2,048, 5% linear warmup, a BART learning rate of 8×10^{-5}, and a ViT learning rate of 8×10^{-6}.

Stage 2: Multi-View SFT. We construct ∼50M paired training samples from three sources: main–auxiliary item images, item-to-item (i2i) associations, and user query-image / clicked-item pairs. Pairs are filtered by the cosine similarity of the production image-recall ViT embeddings: main–auxiliary pairs use a low threshold of 0.75 (the relation is established by catalog metadata and the threshold mainly screens out mislabeled assets), while i2i and query-click pairs are filtered at a stricter 0.82 to suppress weakly related noise. Both thresholds were chosen by case-by-case validation on sampled pairs at each candidate cutoff. Because i2i and query-click signals are mined from search logs rather than human-curated, residual label noise is intrinsic to the corpus: MV-SFT is therefore already trained under weakly supervised conditions, and the threshold gate functions as a quality-control mechanism that absorbs most of this noise before it reaches the model. Because MV-SFT is a supervised next-token-prediction objective rather than a contrastive one, no explicit negative sampling is required—the training signal comes entirely from predicting both SIDs of each positive pair. When an item happens to have no paired view available (for example, when only a main image is listed), the construction naturally degenerates to the self-SID prediction task already used in Stage 1 pretraining, so missing-view items remain trainable without any

special handling. Given either side as input (assignment is randomized at every step), the model is trained to predict the SIDs of *both* sides, encouraging view-complete generation. Formally, with I_{in} drawn uniformly from the pair and ctx denoting the conditioning context,

$$\mathcal{L}_{\text{MV-SFT}} = -\mathbb{E}_{\substack{(I_a,I_b)\sim\mathcal{D}_{\text{mv}} \\ I_{\text{in}}\sim\text{Unif}\{I_a,I_b\}}} \Big[\log \pi_\theta\big(\text{SID}(I_a), \text{SID}(I_b) \mid I_{\text{in}}, \text{ctx}\big)\Big]. \quad (11)$$

We optimize this objective for 10 epochs with a batch size of 2,048, using learning rates of 2×10^{-5} for BART and 2×10^{-6} for ViT.

Stage 3: Personalized Post-training. Personalized SFT first appends user long-term and short-term behavior tokens to the input. List-wise DPO (Eq. 8) is then applied with candidate lists of size 2 (pairwise) or 3 (one positive, two negatives), an overall positive-to-negative ratio of ∼1:1.2, and inverse temperature $\beta = 0.05$. BART learning rate 2×10^{-5} (ViT 2×10^{-6}), batch size 1,024. We note that personalization plays a deliberately auxiliary role in visual search. Unlike feed-style recommendation, where the user query is implicit and personalization carries most of the relevance signal, image search is an intent-anchored task: the query image itself, together with the multi-view supervision of Stage 2, already resolves the bulk of the retrieval decision. User-behavior signals act as a fine-grained disambiguator on top of that decision (e.g., preferring the gender variant or price tier that matches the user's history) rather than as the primary ranking signal, which is consistent with the modest but stable lifts attributed to personalization signals in the ablations.

Finalization: Dynamic Pruning. The curriculum linearly decreases the number of retained visual tokens from $V_{\max} = 197$ to $V_{\text{sub}} = 33$ over $E = 10$ epochs (Eq. 10). To prevent forgetting under aggressive token compression, we retain a downstream SFT loss alongside the KL distillation term (Eq. 9):

$$\mathcal{L}_{\text{prune}} = \mathcal{L}_{\text{SFT}} + 0.5\,\mathcal{L}_{\text{dis}}, \quad (12)$$

corresponding to a 2:1 SFT-to-distillation weighting; this auxiliary signal is empirically critical for matching the teacher's accuracy. BART learning rate 2×10^{-5} (ViT 2×10^{-6}), batch size 1,024.

Component	Setting
Architecture	
Visual encoder	ViT-Base (768d, 12L)
Generative model	BART-Base (6+6L, 768d)
VRQ codebook sizes	$(2048, 512, 256 \mid 256, 256)$
Dynamic Pruning	
Tokens (max / sub)	$V_{\max}$=197, V_{sub}=33
Curriculum	linear over E=10 epochs
K-means++ device	CPU, per-image, max iter 10
Beam Search	
Offline beam size	$B = 10$
Online beam size	$B = 256$
Max decode length	5
Latency Measurement	
ViT+MLP	A10 + TensorRT
BART decoder	RTX 4090 + LightSeq
Input resolution	224×224
Operating QPS / replica	10

Table 7: Compact summary of implementation settings for OneVision (architecture, dynamic pruning, beam search, and latency measurement).

A.6 Algorithmic Procedures

We summarize the complete OneVision training pipeline in Algorithm 1 and the end-to-end inference pipeline in Algorithm 2. The training procedure consists of two preparatory stages that produce the VRQ encoder Φ_{VRQ}, three training stages that progressively shape the generative model π_θ, and a final curriculum-based pruning step. Inference consists of an offline catalog-indexing pass and an online query-time generation pass.

A.7 Baseline Implementation

We describe the implementation of all baselines reported in Tables 1 and 3. To isolate methodological differences, generative-retrieval baselines share the same backbone capacity and optimization budget as OneVision unless explicitly stated.

Embedding-based Retrieval. *DINOv3* (Siméoni et al., 2025) uses the released DINOv3-Base checkpoint to extract 768-dim image embeddings; retrieval is performed by cosine-similarity ANN search using Faiss. *Online MCA* is the multi-stage cascading system currently deployed at our industrial e-commerce platform, comprising recall, pre-ranking, and ranking. *Quantized ANN Method (QAM)* reuses the online embedding model but replaces continuous ANN with VRQ-quantized retrieval, serving as an upper bound on what code-based retrieval can preserve from the original embeddings.

Generative Retrieval Baselines. *IRGen* (Zhang

Algorithm 1 OneVision Training Pipeline

Require: Image-pair corpus $\mathcal{D}_{\text{pair}}$, multi-view SFT corpus $\mathcal{D}_{\text{mv}}$, user-behavior corpus $\mathcal{D}_{\text{usr}}$, statistical attributes $\mathcal{S}$
Ensure: VRQ encoder Φ_{VRQ}, generative model π_θ
Stage 0a: Image Alignment
Train ViT $\mathcal{V}$ + MLP head (768→256→64) on $\mathcal{D}_{\text{pair}}$ via $\mathcal{L}_{\text{align}} = \mathcal{L}_{\text{MoCo}} + \mathcal{L}_{\text{tri}}$
Stage 0b: VRQ Codebook Training
for paired batch $(I^{(1)}, I^{(2)}, y) \in \mathcal{D}_{\text{pair}}$ **do**
  $v^{(k)} \leftarrow \mathcal{E}_v(\mathcal{V}(I^{(k)}))$, $t \leftarrow \mathcal{E}_t(\text{BGE}(y))$
  $f^{(k)} \leftarrow (1-\alpha)v^{(k)} + \alpha t + f^{(k)}_{\text{cat}}$ ▷ Eq. 1
  Update RQ-VAE codebooks (levels 1–3) by $\mathcal{L}_{\text{rq}}$ ▷ Eq. 7
end for
Compute level-3 residuals; fuse with PCA-reduced $\mathcal{S}$ at 9:1
Train OPQ codebooks (levels 4–5) on the fused residuals
Form $\Phi_{\text{VRQ}} = \text{RQ-VAE} \circ \text{OPQ}$
Stage 1: Pretraining
Init π_θ: ViT $\mathcal{V}$ (no projection) → MLP → BART
Train under NTP loss on tasks {image→SID, image+title→SID, image→cat., image+title→cat.}, mix 7:1:1:1
Stage 2: Multi-View SFT
for pair $(I_a, I_b) \in \mathcal{D}_{\text{mv}}$ **do**
  Sample $I_{\text{in}} \in \{I_a, I_b\}$ uniformly
  Train π_θ to predict both $\text{SID}(I_a)$ and $\text{SID}(I_b)$
end for
Stage 3: Personalized Post-training
Append long-/short-term user behavior tokens for samples in $\mathcal{D}_{\text{usr}}$; run personalized SFT
Run list-wise DPO with $\mathcal{Y}_s = \{y^+, y_1^-, \ldots, y_m^-\}$, $m \in \{1, 2\}$ ▷ Eq. 8
Finalization: Curriculum Dynamic Pruning
Set teacher $\pi_{\text{ref}_p} \leftarrow \pi_\theta$ (frozen); init student $\pi_{\theta_p} \leftarrow \pi_\theta$
for epoch $e = 1, \ldots, E$ **do**
  $v^{(e)} \leftarrow \lfloor V_{\max} - \frac{e}{E}(V_{\max} - V_{\text{sub}}) \rfloor$ ▷ Eq. 10
  Cluster ViT tokens via K-means++ to $v^{(e)}$ groups; keep nearest token per centroid
  Update π_{θ_p} by $\mathcal{L}_{\text{prune}} = \mathcal{L}_{\text{SFT}} + 0.5\,\mathcal{L}_{\text{dis}}$
end for
return Φ_{VRQ}, $\pi_\theta \leftarrow \pi_{\theta_p}$

Algorithm 2 OneVision End-to-End Inference

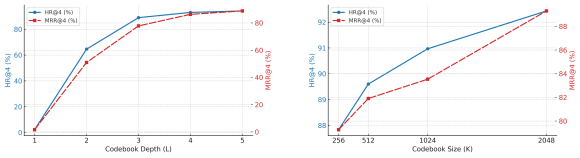
Require: Query image I_q, user context u, trained Φ_{VRQ} and π_θ, item catalog $\mathcal{C}$, beam size B
Ensure: Ranked retrieval list $\mathcal{R}$
Offline: Catalog Indexing ▷ performed once per refresh
for $c \in \mathcal{C}$ **do**
  $\text{SID}(c) \leftarrow \Phi_{\text{VRQ}}(c)$ ▷ 5-level encoding
end for
Build Trie $\mathcal{T}$ of depth 5 from $\{\text{SID}(c)\}_{c \in \mathcal{C}}$; attach item lists at leaves
Online: Query-time Generation
Encode I_q via ViT to $V_{\max}$ visual tokens $\mathbf{Z}$
Cluster $\mathbf{Z}$ to V_{sub} groups via K-means++; keep token nearest each centroid as $\mathbf{Z}'$
Concatenate $\mathbf{Z}'$ with user-behavior tokens of u; feed to BART encoder
Init beams $\mathcal{B} \leftarrow \{(t_{[\text{BOS}]}, 0)\}$
while not all beams ended and length < 5 **do**
  Expand each beam; mask logits with $\mathcal{T}$ to retain only valid prefixes
  Keep top-B partial sequences by cumulative log-probability
end while
For each completed SID in $\mathcal{B}$, fetch item list via $\mathcal{T}$; rank within-SID by lightweight business signals; concatenate across beams to form $\mathcal{R}$
return $\mathcal{R}$

et al., 2024) couples a vision encoder with an autoregressive decoder, forming an end-to-end seq2seq generator that resembles a BART-style architecture. *GENIUS* (Kim et al., 2025) introduces modality-decoupled semantic quantization with query augmentation. To match OneVision in capacity, both baselines are re-implemented with the same ViT-Base + BART backbone, yielding ∼230M parameters in total—on par with OneVision. We train them on the same 50M images, with the same task-mix proportions used in OneVision pretraining (Stage 1), but without the multi-view supervision: the model only predicts its own SID rather than the SID set of a paired view. Both baselines use AdamW, learning rate 8×10^{-5}, batch size 256×8, and 10 epochs—identical to OneVision pretraining. At inference, we use Trie-constrained beam search with $B = 10$ on offline benchmarks.

Vision–Language Model Baseline. *VLM* refers to InternVL3.5-1B (Wang et al., 2025b) fine-tuned with full parameters to autoregressively generate item SIDs from a query image, using the same SID space and Trie-constrained decoding as OneVision. Training uses AdamW with learning rate 8×10^{-5} on 50M samples for 5 epochs, with batch size 8×8; the smaller batch size and shorter training reflect the substantially higher per-step cost of the 1B-parameter model, which prevents matching the budget of the smaller GR baselines.

A.8 Reproducibility Statement

OneVision is developed for and deployed in a production e-commerce search system; its training and evaluation corpora are derived from proprietary user logs and cannot be released in full, so we are unable to provide a self-contained public benchmark for direct numerical replication. To support reproduction at a methodological level, we will release the complete training codebase—covering image alignment, VRQ codebook training, the three-stage generative training pipeline together with the curriculum pruning step, and Trie-constrained inference—together with the full set of hyperparameters and architectural details documented in Appendices A.5 and A.6. The release will additionally ship a small synthetic demo dataset on which the entire pipeline can be exercised end-to-end, so that practitioners can validate their setup before plug-



(a) Results across depth L (b) Results across size K

Figure 6: Impact of Different Codebook Depth L and Size K.

Table 8: Relative PV CTR Gains Across Top, Middle, and Long-Tail Industries.

Method	Top	Middle	Long-tail
OneVision	+2.58%	+3.64%	+15.29%

ging in private data. Practitioners with access to comparable e-commerce data should then be able to reproduce the framework from these resources.

A.9 Additional Ablation and Analysis

This appendix collects supplementary analyses that did not fit in the main text: beam-size sensitivity, codebook depth/size ablations, a drill-down of the online A/B test results, and retrieval visualizations.

Beam-size Sensitivity. In addition to the offline beam size $B = 10$ used throughout the main results, we have run sweeps over $B \in \{10, 32, 64, 128, 256, 512\}$. HR@10 and MRR@10 improve modestly when B grows from very small values and then plateau, with no measurable change across the larger half of the range. We therefore retain $B = 10$ for offline evaluation to keep the evaluation protocol simple and efficient, and keep $B = 256$ for online serving for the reasons discussed in Appendix B.3.

Experiment on Codebook Hyperparameters. We also ablate the RQ-VAE codebook depth L and size K under constrained beam search. As shown in Figure 6, increasing L from 1 to 4 improves HR@4 from 2.21% to 93.03% and MRR@4 from 1.58% to 86.33%, with only marginal gains beyond $L = 4$. Similarly, enlarging K from 256 to 2048 raises HR@4 from 87.84% to 92.50% and MRR@4 from 80.27% to 89.32%. Overall, within a practical codebook space and constrained decoding, retrieval accuracy shows consistent improvement with increasing depth and size.

Drill-down analysis. We further conduct a drill-down analysis of the online A/B test results. As shown in Table 8, multi-view modeling yields consistent PV CTR gains across industry tiers, with increases of +2.58%, +3.64%, and +15.29% for the top, middle, and long-tail industries, respectively. In particular, women's clothing and bags & accessories improve by approximately +3.3% and +4.8%, respectively. These gains stem from multi-view modeling, which generates a more comprehensive set of candidate SIDs and thereby improves recall under large viewpoint variations.

Visualization of Retrieval Results with OneVision. Finally, to better illustrate the encoding and retrieval effectiveness, we visualize query images and their corresponding Top-2 retrieved items produced by OneVision, together with their VRQ SIDs across different e-commerce categories. As shown in Figure 7, for images with small visual variations, VRQ leverages multi-view contrastive learning to maintain consistent shallow-level codes. For same-item images with large visual discrepancies, shallow-level alignment becomes challenging; nevertheless, with multi-view SFT, OneVision can still retrieve the correct items via beam search. Notably, the final two code levels differ across all examples, reflecting VRQ's fine-grained encoding that preserves visual details while incorporating statistical business attributes.

B Online Deployment Details

B.1 Deployment Architecture

Two-stage Model Serving. For practical reasons in our serving cluster, OneVision is split into two independently deployed services. The first service hosts the ViT encoder together with the MLP projection layer and is accelerated with TensorRT; it runs on a single NVIDIA A10 GPU. The second service hosts the BART encoder–decoder and runs constrained beam-search decoding; it is accelerated with LightSeq (Wang et al., 2021a) and runs on a single NVIDIA RTX 4090. Each service uses one GPU per A/B-test bucket, which is sufficient under current traffic.

Inter-service Communication and Token Reduction. The two services communicate over gRPC. Dynamic pruning is performed entirely on the ViT side: after K-means++ clustering, only $V_{\text{sub}} = 33$ tokens (32 cluster representatives plus the `[CLS]` token) are transmitted to the BART service, instead of the full 197 tokens produced by the visual encoder. This keeps the inter-service payload small and avoids transferring redundant visual signal across the network. K-means++ is executed on the CPU on a per-image basis with a maximum of 10 iterations, which converges in under 1 ms per query and is already accounted for in the

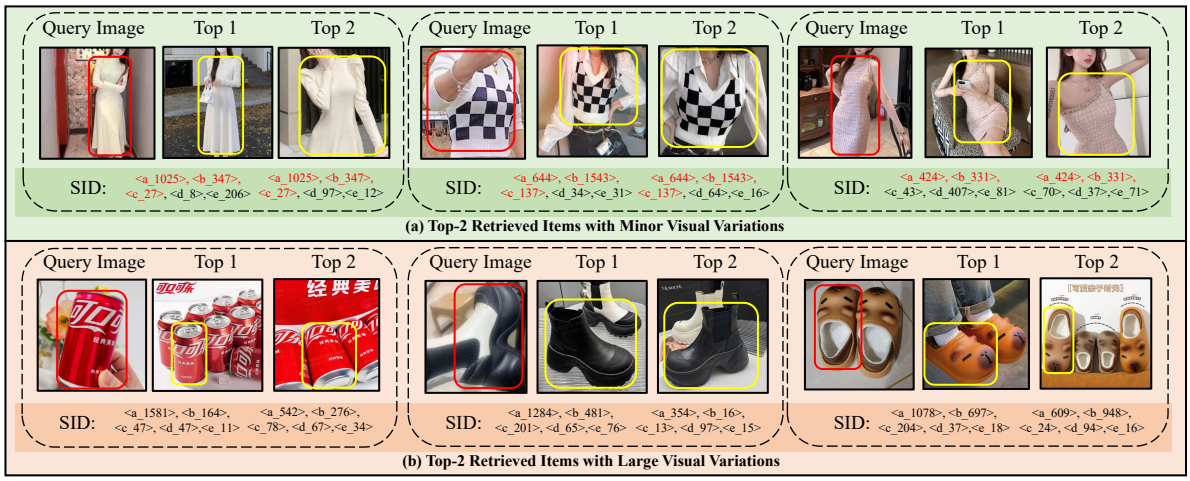


Figure 7: Visualization of OneVision retrieval and VRQ encoding. For visually similar items, VRQ enforces consistent shallow-level codes, while visually diverse images of the same item are retrieved via multi-view SFT. Differences in the last two deeper code levels indicate fine-grained encoding and improved code diversity.

ViT+MLP latency reported in Appendix B.4; no separate batched implementation is needed at the current QPS.

BART-side Input Assembly. The BART service concatenates the 33 incoming visual embeddings with the user-behavior tokens (long-term aggregated representation plus the SIDs of the user's five most recent image-search clicks), then runs constrained beam search to produce SIDs.

B.2 SID Indexing Pipeline

Reusing the Image-to-Image Recall Encoder. Our industrial e-commerce platform already operates an image-to-image recall service that produces visual embeddings for every item in the catalog and uses them for ANN-based retrieval. The visual encoder of this recall service *is* the contrastively aligned ViT described in Appendix A.5: in other words, the alignment stage doubles as the training procedure for the production image-recall encoder. Because VRQ is trained with this ViT frozen, the embeddings already stored in the ANN service share the exact feature space VRQ expects, so we reuse them directly when assigning SIDs rather than re-running the visual encoder over the catalog.

Object-Region Extraction during Offline Indexing. Before an item enters the recall service, each of its item images is processed by an e-commerce-finetuned GroundingDINO (Liu et al., 2024) detector to crop the actual item region. The detector is prompted with the item's rewritten English category text, treated as an entity description. This object-region step is shared with the recall pipeline and removes background clutter (props, packaging, lifestyle scene) before the ViT extracts the visual embedding. The cropped embedding is what the ANN service indexes and, accordingly, what VRQ consumes when producing SIDs.

SID Assignment. For each indexed item we pair its cropped visual embedding with the BGE-encoded category text and feed them through the latter half of the VRQ pipeline—the three-level RQ-VAE followed by the two-level OPQ encoding with statistical business attributes (Appendix A.5). The output is the 5-level SID. The catalog covers hundreds of millions of items and close to two billion item images; with eight RTX 4090 GPUs the indexing service can encode several million items (tens of millions of images) per day, and we refresh SIDs at a daily cadence to absorb newly listed and modified items. After each refresh, the updated SID set is aggregated into the Trie used for constrained decoding (Appendix A.1).

B.3 SID-to-Item Lookup and Deduplication

Image-level Granularity. VRQ encodes *images*, not items. Each catalog item carries on the order of half a dozen item images (main image, auxiliary images, lifestyle shots), each of which receives its own 5-level SID. An item can therefore appear under multiple SIDs, and a single SID can be shared by images of different items.

Lookup via the Feature Center. At inference time, each decoded SID is resolved against our industrial e-commerce platform's internal feature center: the SID is first mapped to the set of image IDs sharing that code, and each image ID is then resolved to its parent item ID through a maintained

image-to-item table. The feature center is heavily cached and contributes negligible latency relative to model inference.

Deduplication within a Result List. Because different images of the same item may decode to different SIDs, an item can surface multiple times in a single decoded result list. We therefore deduplicate at the item level: when an item appears more than once, only the highest-scoring occurrence is kept. Because the within-SID ranking score in Eq. 13 below is computed at the item level, all images of a given item tie on that score; ties are broken by the beam-search cumulative log-probability, which favors the image the generator was most confident about. Empirically, this step removes roughly 30% of decoded SIDs as duplicate items.

Online Serving. At serving time, items within the same SID are ranked by a lightweight statistical score computed over the past 30 days:

$$\text{score}(c) = 5 \cdot N^{30d}_{\text{order}}(c) + 0.01 \cdot \text{GMV}^{30d}(c) + 0.1 \cdot N^{30d}_{\text{click}}(c), \tag{13}$$

where N^{30d}_{order}, GMV^{30d}, and N^{30d}_{click} denote the number of orders, total gross merchandise value, and number of clicks within the last 30 days. Items are sorted in descending order of $\text{score}(c)$, surfacing the most commercially salient item among those sharing a SID.

Online Beam Size. Three factors jointly motivate the relatively large online beam size of $B = 256$. First, the recall channel itself must surface several hundred candidates to feed downstream business-policy ranking, comparable to the 200+ items the image-recall channel of online MCA produces. Second, image-level SID granularity means the same item may occupy several beam slots through its different images. Third, after the item-level deduplication above removes roughly 30% of decoded SIDs, the count of distinct candidates shrinks further. A smaller beam often fails to surface enough distinct items to populate the downstream ranking input. Beyond the offline retrieval-quality sweep reported in Appendix A.9, we did not run a systematic production-side sweep; in spot checks at serving traffic, increasing B from 256 to 512 changes BART latency only marginally (both remain in the 20-something millisecond range) and does not produce meaningful retrieval-quality differences, so we keep $B = 256$ as a conservative production setting.

Bucket-size Statistics. We estimate the SID collision behavior of the 5-level VRQ configuration used in this paper on the full production catalog. At the image level (the granularity of VRQ encoding itself), the duplicate rate sits around 55% with P50, P95, and P99 bucket sizes of roughly 2, 8, and 25 respectively. At the item level (after the image-to-item deduplication discussed above), the picture is markedly cleaner: only about 35% of occupied SIDs map to more than one item, with P50, P95, and P99 bucket sizes around 1, 3, and 8. The majority of queried codes therefore resolve to a single item, and the long tail is bounded at P99 by a small enough number of candidates for the within-SID statistical score to sort effectively.

B.4 Latency and Throughput

Throughout this paper, "efficiency" refers specifically to end-to-end query-time latency measured under the operating QPS reported below; it is not a measure of throughput or training compute. All numbers in this section are mean and tail latencies observed on production traffic with no synthetic batching.

Per-stage Latency. End-to-end latency is the sum of two model-inference segments plus a negligible lookup step. The ViT+MLP service (A10, TensorRT) currently averages 55 ms per query at the operating QPS of 10 per replica on 224×224 inputs, with a P99 of roughly 90 ms; this number is on the high side because kernel-level optimization is still in progress. The BART service (RTX 4090, LightSeq) averages 27 ms with a tail latency P99 of 45 ms, also at QPS 10 per replica, decoding with beam size $B = 256$ and maximum length 5. The downstream feature-center lookup adds approximately 1 ms and is dominated by cache hits.

End-to-end Latency. Aggregating the two stages and the lookup yields an end-to-end mean of roughly 83 ms and a P99 around 120 ms.[1] This compares favorably to the deployed online MCA cascade, whose end-to-end latency is dominated by the dense ranking stage and totals around 254 ms.

Stage-by-Stage Comparison with Online MCA. Table 9 compares the per-stage latency of the two pipelines. OneVision collapses the recall / pre-ranking / ranking back-end into a single generative pass and replaces ANN retrieval with a 1 ms SID lookup, eliminating the relevance-prediction and

[1] Stage-level P99 values do not sum directly to end-to-end P99 because the two services do not hit their respective tail-latency events on the same request; the reported end-to-end P99 is measured empirically.

CTR-scoring stages that dominate the online MCA timeline.

Stage	Online MCA	OneVision
ViT encoding†	40	—
Category prediction†	40	—
Retrieval	4 (ANN)	—
Relevance scoring	10	—
CTR scoring	200	—
Generative pass	—	82
SID lookup	—	1
Total	**254**	**83**

Table 9: Per-stage mean latency (ms). †ViT encoding and category prediction run in parallel in online MCA, so only the slower of the two (40 ms) contributes to the total. In OneVision, the 82 ms generative pass consists of ViT+MLP visual encoding (55 ms) and BART SID generation (27 ms), followed by a lightweight SID lookup.

Effect of Dynamic Pruning. The 18% end-to-end latency reduction reported in the main text is the speedup of OneVision *with* the curriculum-based dynamic pruning of § 3.3 relative to the same model *without* pruning; it is an intra-OneVision comparison and unrelated to the online MCA numbers above. Pruning reduces the visual token count from $V_{\max} = 197$ to $V_{\text{sub}} = 33$, which shortens both the BART encoder pass and the inter-service payload while leaving retrieval quality essentially unchanged (Figure 4).

C Discussion

C.1 Scope of the End-to-End Pipeline

What OneVision Replaces. Industrial visual search systems typically consist of an object-detection front-end followed by a multi-stage cascading retrieval back-end (recall, pre-ranking, ranking). OneVision unifies the back-end three stages into a single generative model, but does not subsume the object-detection step. We argue this choice is appropriate rather than incomplete.

Why Object Detection Is Kept Separate. First, object detection is not strictly required for visual search: in our deployed system, users may directly draw a bounding box over the object of interest after taking a photo, and the in-app same-item search uses the full image without any cropping. Second, when detection is invoked, it functions as an interactive disambiguation step rather than a representation step—the user confirms which object in a cluttered photo they are actually searching for, instead of having the system blindly retrieve similar items for every detected object. Bundling detection into a single end-to-end generator would remove this confirmation loop and degrade the retrieval intent signal. We therefore treat object detection as a separable, user-facing module orthogonal to the retrieval pipeline that OneVision targets.

Learned vs. Rule-based Boundary. A second clarification concerns where the learned model ends and rule-based post-processing begins. The recall, pre-ranking, and ranking stages of the previous cascade—each a separately trained scoring model—are subsumed by a single generative pass in OneVision. After SIDs are decoded, items that share a SID are ordered by the lightweight statistical score in Eq. 13, which is a rule-based bucket-level sort. This boundary is intentional and consistent with prior generative-retrieval work such as GENIUS (Kim et al., 2025), which similarly resolves within-bucket ambiguity through an ANN look-up rather than learning that order end-to-end. The rule-based step only orders items *within* a SID and does not affect what is recalled, so the recall-quality comparisons against online MCA reported in the main text remain apples-to-apples: both pipelines emit a ranked candidate list, and OneVision's list is produced by one learned generative pass rather than by a stack of separately trained scoring models.

C.2 Failure Modes and the Gap to Online MCA

OneVision matches or surpasses online MCA on most metrics (Tables 1 and 3) and delivers significant improvements in online A/B tests, but it does not uniformly dominate online MCA across all subpopulations. We discuss the regimes where the gap remains.

Long-tail Items. Embedding-based retrieval recovers an unseen item as long as its visual representation is close enough to the query in feature space; the matching is purely geometric and does not require the encoder to have seen the specific item during training. Generative retrieval, in contrast, must *produce* the SID of the target item token by token, and tokens corresponding to rarely seen items are systematically under-trained.

Categories with Strong OCR Dependence. Categories whose retrieval relevance hinges on text content within the image—books being the most representative example—are difficult for a SID generator that compresses each item into five discrete codes. The dense visual features used by online

MCA can preserve fine-grained textual cues that a low-bitrate code sequence cannot, leaving an accuracy gap on these categories.

These two regimes (long-tail items and OCR-heavy categories) jointly account for the residual relevance gap observed against online MCA. Closing them likely calls for complementary directions: larger generative backbones with stronger emergent generalization to amortize the under-trained SIDs of long-tail items, and a hybrid design that re-introduces a dense-feature channel alongside the discrete codes for text-dependent categories.