Multimodal Product Keyword Generation with Semantic ID-Based Attribute Augmentation

Anonymous ACL submission

Abstract

Product keyword generation plays a critical role in advertising retrieval systems by expanding product-query matching and mitigating zero-result queries. Existing approaches predominantly rely on product textual descriptions and historical query logs, which often omit fine-grained product attributes and fail to capture diverse user expressions, especially for mid- and low-frequency queries. In this work, we propose a multimodal keyword generation framework that leverages product images to recover missing attributes and introduces semantic ID-based clustering to expand keyword diversity across semantically similar products. Specifically, we first extract multi-attribute keywords from multimodal product information, including text and images. We then use RQ-VAE to generate semantic IDs for each attribute and cluster products by these IDs to mine additional attribute-consistent keywords for training data augmentation. The augmented dataset is used to fine-tune a multimodal large language model for product keyword generation. Experiments on a large-scale real-world food delivery dataset show that incorporating multimodal attributes significantly improves keyword coverage, while semantic clustering further enhances keyword diversity and retrieval performance. Our method achieves substantially higher matching rates than recent state-of-the-art approaches and demonstrates competitive results on public benchmarks while requiring a significantly simpler inference pipeline.

1 Introduction

Keyword generation is a core component of advertising retrieval systems, aiming to generate keywords for queries or products to enhance recall coverage and mitigate issues such as missed or zero-result queries(Li et al., 2019; Long et al., 2025). On the query side, traditional approaches typically rely on query rewriting or expansion models to produce variant queries, thereby broadening the scope of relevant product retrieval(Lee et al., 2018; Wang et al., 2024; Lei et al., 2024, 2025; Lian et al., 2019). On the product side, existing methods often leverage product attribute information and online behavioral metrics, such as click-through and conversion rates, to generate descriptive, packaging-oriented keywords that summarize product characteristics(Li et al., 2019; Zhou et al., 2019; Wang et al., 2025b; Chen et al., 2025).

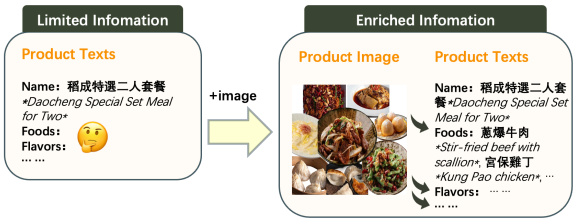

Figure 1: Original product information (**left**) versus product information enhanced with image understanding (**right**). It can be observed that relying solely on textual descriptions fails to capture the detailed contents of product sets. In contrast, incorporating image information enables the extraction of fine-grained content details and additional product attributes, such as foods and flavors.

However, existing product-side keyword generation models predominantly rely on product textual descriptions and historical user click data, which often omit critical product attributes. For example, OKG (Wang et al., 2025b) generates keywords based on historical search trends and product attributes, while OMS (Chen et al., 2025) leverages advertiser-provided descriptions and high-frequency search terms. RARE (Liu et al., 2025) employs fine-tuned LLMs to produce commercial intent keywords from product descriptions. Despite these advances, such methods remain constrained by incomplete product texts and fail to adequately cover diverse user queries, leading to

Attributes	Food	Ingredients	Categories	Flavors	Cooking_methods	Functional_uses	Synonyms	Similar_foods
Examples	葱爆牛肉 *Stir-fried beef with scallion*, 宫保雞丁 *Kung Pao chicken*, ···	牛肉 *Beef*, 三文魚 *Salmon*, 豆腐 *Tofu* ,···	四川菜 *Sichuan cuisine*, 健康餐 *Healthy meals*, ···	麻辣 *Spicy and numbing*, 酸爽 *Sour and refreshing*, ···	爆炒 *Stir-frying*, 乾煎 *Pan-frying*, 水煮 *Boiling*, ···	聚餐分享 *Sharing in group meals*, 減脂瘦身 *Weight loss*, ···	燒烤 *Barbecue*, 烤串 *Skewered grilled meat*, ···	雞胸肉沙律 *Chicken breast salad*, 生菜沙律 *Lettuce salad*, ···

Figure 2: Analysis of click log samples where user queries failed to match product text reveals eight distinct categories of challenging queries, the figure contains the names of eight categories and some examples.

missed recall opportunities.

As illustrated in the left panel of Figure 1, many product texts lack essential details such as foods and flavors, limiting the accuracy of keyword generation for queries that involve these missing attributes. We hypothesize that these gaps can be addressed by utilizing product images, which often contain richer and more detailed information, as shown in the right panel.We validate this hypothesis in the following pre-experiment section, where we explore the effectiveness of integrating product images to fill in missing attributes.

Nevertheless, relying solely on images is insufficient to address the full spectrum of user queries, particularly those varying across regions and cultures. Thus, in addition to filling missing attributes with multimodal data, it is crucial to generate diverse keywords that capture regional and cultural variations.

To address these challenges, we propose a two-pronged framework: (1) generating product keywords by leveraging both textual and image data to recover missing attributes, and (2) augmenting keyword sets by mining additional keywords from semantically similar products, thereby capturing more diverse expressions. Specifically, we define multiple attribute dimensions, including foods, ingredients, categories, etc, and employ GPT-4.1 (OpenAI et al., 2024) to generate keywords for each attribute. To facilitate cross-product keyword augmentation, we generate SIDs via RQ-VAE, cluster products based on these IDs to form shared keyword pools, and augment the training set by filtering keywords according to semantic similarity. Finally, a multimodal large model is fine-tuned on this augmented dataset to generate diverse and accurate keywords across all attributes.

The main contributions of this work are summarized as follows:

1. We introduce attribute-specific keyword categories for product keyword generation, enabling standardized and customizable keyword production. We also construct a large-scale multimodal dataset from a Hong Kong food delivery platform, where keyword annotations are generated using GPT-4.1 and refined through human evaluation.

2. We propose a multimodal keyword generation framework that integrates semantic ID-based clustering with external keyword mining to augment training data and improve keyword diversity and attribute coverage.

3. Extensive experiments demonstrate that our method generalizes well across multiple LLM architectures, significantly outperforming the recent SOTA method on query-keyword-product matching and achieving competitive performance on public benchmarks with a simpler inference pipeline.

2 Pre-Experiment

To investigate the impact of missing product attributes on keyword generation, we conducted preliminary experiments that revealed key challenges in existing methods. We sampled approximately 200,000 high-click-through-rate product listings from the Hong Kong region. Our analysis found that about **40%** of the products lacked detailed descriptions. For example, a listing like "*NVN South Vietnamese Baguette: liver pate, smoked ham, pork sausage, pig's ear, pickled radish, cucumber, mayonnaise, scallion oil, and herbs*" omits ingredients like "smoked ham" or "pickled radish." Without these details, the product is only retrievable by exact search terms like "*NVN South Vietnamese Baguette*", causing missed retrievals for queries with ingredients like "*smoked ham*". This leads to gaps in keyword coverage, preventing users from finding products with specific attributes.

To mitigate this, we incorporated product images alongside titles into GPT-4.1 to extract missing attribute information. This approach recovered approximately **17%** of previously unmatched

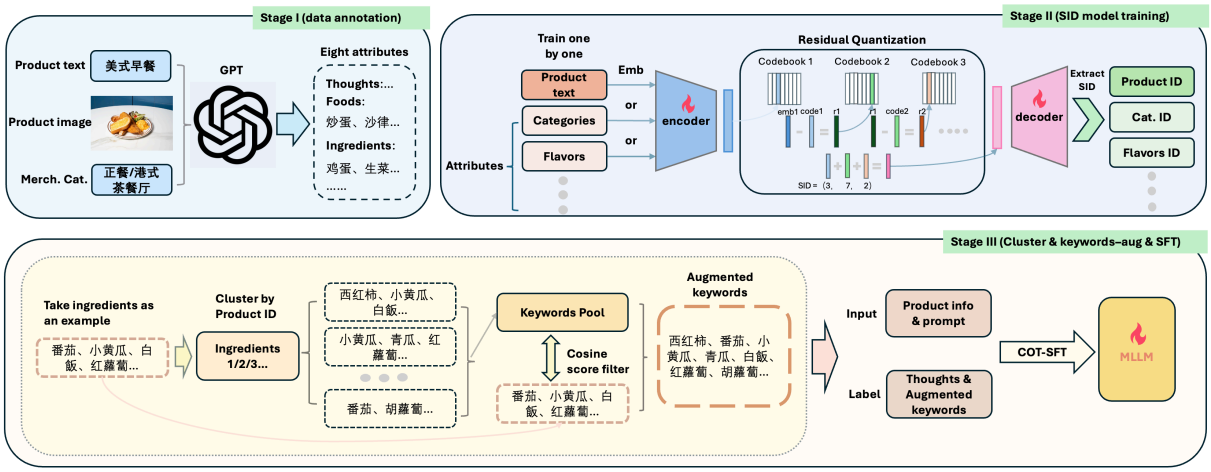

Figure 3: The overall framework of our method comprises three stages. In **Stage I**, multimodal product data (text and images) are fed into GPT-4.1 to generate multi-attribute keywords, which are refined via LLM-based evaluation and human annotation. In **Stage II**, the product text and selected attribute keywords are used to train semantic ID models, assigning semantic IDs to products and keywords. In **Stage III**, SIDs of attribute keywords are clustered, and additional keywords are retrieved from the cluster keyword pool to augment the training data. The augmented dataset is then used to fine-tune a local multimodal LLM for improved keyword generation.

queries; however, a substantial fraction remained unresolved.

By analyzing the remaining unmatched samples, we identified eight attribute types (Figure 2) where user queries are most prone to retrieval failures. Guided by these insights, we employed a multimodal LLM to generate keywords for each attribute types, underscoring the need for a comprehensive keyword generation framework.

3 Methods

3.1 Dataset Construction

We define eight attribute-specific categories for product keywords: foods, ingredients, categories, flavors, cooking methods, functional uses, synonyms, and similar foods. In Stage I (illustrated in Figure 3), we use GPT-4.1 (OpenAI et al., 2024) to process the product image, product text, and the merchants main business category (which may differ from the products category but serves as reference). GPT-4.1 is prompted to generate a detailed description of the image, explain its reasoning, and summarize keywords for the eight attributes.

To reduce noise in the generated keywords, we apply a two-stage cleaning process. First, we prompt GPT-4.1 to re-evaluate and filter out irrelevant or incorrect keywords, ensuring the quality of the generated list. Then we manually inspect each entry to refine the keywords and remove inaccuracies. To ensure diversity in the test set, crucial for evaluating similar-product keyword mining, we conduct multiple sampling iterations to increase label diversity.

Finally, we compiled a dataset with **30K** training samples and **5K** test samples for Hong Kong takeaway food products.

3.2 Semantic ID Model Training

We formulate a discrete quantization and reconstruction task to generate semantic ID representations for product texts and their eight types of keywords. However, the SID model performs poorly on the attributes of food, ingredients, synonyms, and similar foods due to their large vocabularies and high semantic dispersion, which leads to weakly aligned SID assignments and introduces noise in subsequent clustering and keyword enhancement steps. Therefore, we train SIDs only for product text and the other four attributes, while the remaining categories use the IDs derived from the product text.

To generate stable semantic ID representations, we construct a text reconstruction model based on RQ-VAE (Lee et al., 2022), as shown in Stage II of Figure 3. RQ-VAE applies residual quantization to refine semantic representations, improving the granularity of product text and enabling the model to capture fine-grained semantic details.

RQ-VAE first obtains the embedding vector x of a text document through a text encoder (e.g., BERT (Devlin et al., 2019)). The embedding is then passed through a learnable DNN encoder to

obtain a latent representation z, and the initial residual is defined as $r_0 = z$.

For each quantization level d, a codebook $\mathcal{C}_d = \{e_k\}_{k=1}^{N}$ is maintained. Quantization selects the nearest codebook entry.

$$c_d = \arg\min_k \|r_d - e_k\|_2. \tag{1}$$

The residual is updated as $r_{d+1} = r_d - e_{c_d}$. After D quantization levels, the quantized vectors $(e_{c_0}, \ldots, e_{c_{D-1}})$ are obtained, and their indices $(c_0, \ldots, c_{D-1})$ form the semantic ID (SID). The final quantized latent representation is $\hat{z} = \sum_{d=0}^{D-1} e_{c_d}$.

The decoder reconstructs the input representation from $\hat{z}$. The overall RQ-VAE training objective is

$$L = L_{\text{recon}} + \beta L_{\text{commit}}, \tag{2}$$

where $L_{\text{recon}} = \|x - \hat{x}\|_2^2$ denotes the reconstruction loss and β controls the weight of the commitment term. The commitment loss across quantization levels is

$$L_{\text{commit}} = \sum_{d=0}^{D-1} \|r_d - \text{sg}[e_{c_d}]\|_2^2, \tag{3}$$

where sg$[\cdot]$ denotes the stop-gradient operator.

The codebooks are updated using exponential moving average (EMA)(Morales-Brotons et al., 2024), $e'_k = \lambda e_k + (1-\lambda)\frac{1}{N_k}\sum_{z \in S_k} z$, where λ is the decay rate, N_k is the number of assignments to e_k, and S_k denotes the set of vectors assigned to that code.

Following Tiger (Rajput et al., 2023), we initialize the codebooks using cluster centers from the first batch of samples to mitigate codebook collapse.

Compared with clustering directly on dense embeddings (e.g., k-means), discrete semantic IDs provide explicit clustering boundaries that lead to more stable product grouping. Products sharing the same SID prefix tend to exhibit similar semantic attributes, which facilitates reliable keyword mining from semantically related products and improves the effectiveness of keyword augmentation.

3.3 Semantic Clustering and Keyword Augmentation

To enhance the diversity of keywords in the training set, we augment each product's attribute-specific keywords by mining terms from products with semantically similar attributes.

For each attribute, we first cluster products based on the first m layers of their attribute-specific semantic IDs (SIDs). Let $S = \{s_1, \ldots, s_n\}$ denote the set of attribute keyword sets for all n products, where s_i contains all keywords associated with the i-th product for this attribute. The cluster C_k is defined as:

$$C_k = \{s_j \mid f_m(SID(x_j, attr)) = k,\ s_j \in S\}, \tag{4}$$

where f_m extracts the first m characters of the attribute-specific SID, and products with the same $f_m(SID)$ prefix are assigned to the same cluster C_k.

Next, for each cluster C_k, we construct a **keywords pool** $V(C_k)$ that contains all keywords for the given attribute across products in the cluster. For the current product x, let $W(x)$ denote its existing set of keywords for this attribute. We then compute the cosine similarity $S(w, v)$ between each keyword $w \in W(x)$ and each candidate keyword $v \in V(C_k) \setminus W(x)$.

To select additional keywords, we take the top-a keywords from $V(C_k)$ that have the highest similarity to any keyword in $W(x)$ but are not already included in $W(x)$:

$$A(x) = \underset{v \in V(C_k) \setminus W(x)}{\arg\text{top-}a} \left\{ \max_{w \in W(x)} S(w, v) \right\}. \tag{5}$$

Here, $A(x)$ is the set of augmented keywords for the current product and attribute, which are then added to the products label in the training set.

This procedure is applied to all attribute dimensions of every product. By leveraging semantic clustering and similarity-based selection, the method expands the attribute-specific keyword sets, improving both the diversity and coverage of the training data. The workflow is illustrated in Stage III of Figure 3.

3.4 Fine-tuning of the Keyword Generation Model

Finally, based on the augmented training set, we perform COT-SFT on the MLLM and optimize the model via the next-token prediction loss:

$$P(x) = \prod_{t=1}^{T} P(x_t | x_{<t}), \tag{6}$$

During inference, the model already possesses the required multimodal and diverse generation capabilities. By simply providing the product's text,

Model	Setting	Macro		Micro	
		Recall	F1	Recall	F1
Closed-source models					
Qwen-VL-Plus	–	33.57	34.21	26.36	29.92
GPT-4o-2024-11-20	–	31.99	37.51	26.67	33.32
Open-source models					
Qwen3-VL-2B-Instruct	zero-shot	13.82	14.24	10.30	11.51
	SFT	34.78	36.86	28.35	31.77
	aug_data SFT	37.35	36.55	31.30	32.23
Qwen3-VL-4B-Instruct	zero-shot	26.80	28.40	20.87	23.82
	SFT	36.71	38.94	30.36	34.12
	aug_data SFT	39.18	38.31	32.73	34.02
Qwen3-VL-8B-Instruct	zero-shot	27.57	29.77	21.67	25.23
	SFT(Text)	34.16	35.08	28.16	30.68
	SFT(Image)	30.54	31.65	25.32	27.55
	SFT(I+T)	37.31	**39.52**	30.71	34.42
	aug_data SFT	**40.13**	39.12	**33.59**	**34.81**

Table 1: Main results on the HK dataset. We apply the proposed data augmentation strategy to the Qwen3-VL model series and further compare unimodal and multimodal supervised fine-tuning settings on **Qwen3-VL-8B-Instruct**. Best results in each column are highlighted in bold.

Method	Full Match	Partial Match
Text	41.21	59.78
Image	23.85	47.66
I+T	42.71	60.32
aug_data	**44.22**	**61.43**

Table 2: Evaluation of query-product matching effectiveness on offline product click logs. Queries are segmented using Jieba. **Full match** indicates that all query segments appear in the product string, while **partial match** means at least one segment is matched.

Matching Criterion	Precision
Exact match	0.52
Semantic match	0.94

Table 3: Agreement between GPT-4.1 and human annotations on 200 randomly sampled instances. Exact matching requires identical keywords, while semantic matching allows paraphrased but equivalent expressions.

image, and merchant category, we can retrieve the product's keywords across the eight specified attribute dimensions.

4 Experiments

4.1 Experimental Setup

Datasets. Our dataset is constructed from product click logs in the Hong Kong region of a large food delivery platform. We select high click-through rate products, yielding approximately **35k products**. Keyword annotations are generated with GPT-4.1, producing an average of **36 keywords per product**, and refined by automatic GPT-4.1 filtering and subsequent human verification.

To assess annotation reliability, we randomly sample 200 instances for manual inspection. As shown in Table 3, GPT-4.1 achieves a strict matching precision of 0.52 and a semantic precision of 0.94 compared with human annotations. Although wording differences occasionally occur, the high semantic agreement indicates GPT-4.1 provides reliable supervision for keyword generation. Detailed per-attribute statistics, including Recall and F1, are provided in Appendix A.6.

Evaluation Metrics. Keyword generation is evaluated using **Recall** and **F1-score**. True positives (TP), false positives (FP), and false negatives (FN) are computed via exact keyword matching between predicted and ground-truth sets.

Unless otherwise specified, Recall and F1 denote **micro-averaged** metrics, computed by aggregating TP, FP, and FN across all attributes. We also report **macro-averaged** metrics, where scores are calculated independently for each attribute and then averaged.

Implementation Details. For fair comparison, identical hyperparameters are used across models. Semantic IDs are generated with RQ-VAE trained

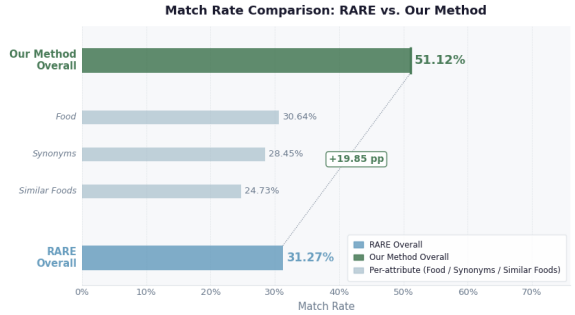

Figure 4: Comparison of the Matching Rate between **RARE Method** and **Our Method** on Click Log Data. Note: The percentages of the three attributes sum to more than the total Match rate because some samples match multiple attributes.

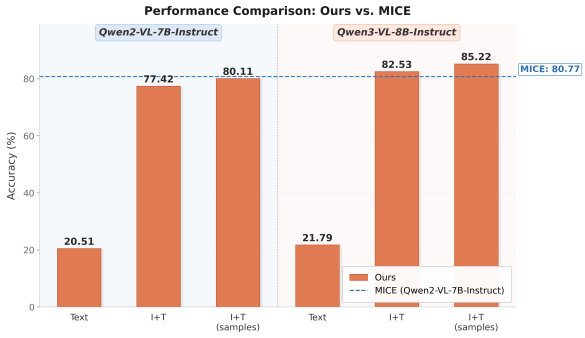

Figure 5: Comparison of **Our Method** against the **MICE Method** on the Food subset of the **ImplicitAVE** benchmark. *"samples"* represents the addition of similar samples clustered using SID in the prompt.

on 768-dim embeddings from a pre-trained BERT encoder. The RQ-VAE consists of a DNN encoder (three hidden layers: 512, 256, 128 with ReLU), a residual quantizer, and a symmetric decoder, producing 32-dim latent representations. Four residual quantization levels are used with codebooks of size 256; commitment coefficient $\beta = 0.25$. The model is trained for 500 epochs with AdamW (lr=0.001, batch size 1024).

Codebooks are initialized via clustering on the first batch and employ dead-code reactivation for codes unused for more than 20 steps. Semantic IDs need not be unique; similar SIDs indicate shared product semantics. During keyword augmentation, cosine similarity is computed on BERT embeddings.

For supervised fine-tuning (SFT), we use LoRA with AdamW (lr=2×10^{-4}) and a cosine annealing scheduler, batch size 4, gradient accumulation 8, for one epoch. Experiments are implemented in PyTorch on two Tesla A100-80GB GPUs.

Clustering Method	Recall	F1
k-means	32.45	33.82
SID	**33.59**	**34.81**

Table 4: Ablation study on clustering methods.

Codebook Layers	ID Layer	Recall	F1
3 layers	layer2	31.65	34.50
4 layers	layer2	**33.59**	34.81
	layer3	31.61	**34.97**
5 layers	layer2	33.59	34.55
	layer3	31.42	34.72
	layer4	31.23	34.73

Table 5: Ablation on the number of codebook layers used in the semantic ID model. "layer2/3/4" indicates clustering based on the first 2, 3, or 4 layers of semantic IDs.

Baselines. To evaluate the general applicability of our proposed method, we conduct fine-tuning and evaluation on multiple MLLMs. These include InternVL3_5-2B, InternVL3_5-8B(Wang et al., 2025a), Qwen2.5-VL-3B-Instruct, Qwen2.5-VL-7B-Instruct(Bai et al., 2025b), Qwen3-VL-2B-Instruct, Qwen3-VL-4B-Instruct, and Qwen3-VL-8B-Instruct(Bai et al., 2025a). Concurrently, we also evaluate the performance on the test set using several closed-source models, including: Qwen3-VL-Plus, GPT-4o-2024-11-20.

4.2 Main Results

Keyword generation results. Table 1 reports keyword generation performance on the Hong Kong takeaway platform dataset using Qwen3-VL-8B-Instruct. Results for other model variants are provided in Appendix A.3.

SFT on our dataset improves performance across all models, demonstrating the effectiveness and generalizability of the constructed dataset. For Qwen3-VL-8B-Instruct, comparisons across text, image, and image+text modalities show that incorporating images helps recover missing product attributes, leading to more accurate keyword generation. Furthermore, augmented-data SFT confirms that the semantic ID clustering strategy increases label diversity by mining semantically similar keywords, encouraging the model to generate a broader range of accurate keywords.

Downstream task results. Table 2 reports query-keyword matching rates evaluated on a held-out set of **30k** product click logs sampled from

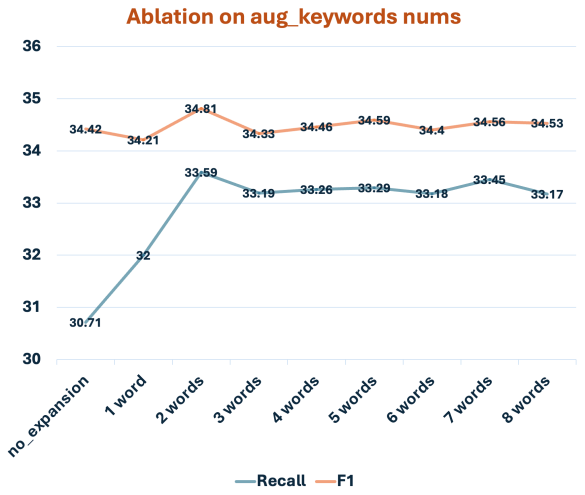

Figure 6: Ablation study on increasing the number of keywords. Compared the model's **Recall** and **F1** evaluation results without data augmentation and with the augmentation of 1 to 8 words.

Figure 7: Examples of keywords generation. **Green** denotes correct keywords predictions, whereas **red** denotes incorrect keywords predictions.

the test set. Both visual information and the proposed data augmentation strategy significantly improve matching performance. However, since the click logs originate from a text-based recall system, incorporating images provides only limited additional gains.

Comparison with the SOTA method. We compare our approach with **RARE** (Liu et al., 2025), a state-of-the-art baseline. To approximate the RARE framework, we implement its core idea where a Multimodal LLM (MLLM) generates Commercial Intent (CI) terms to construct a CI term bank. Constrained beam search then decodes queries while restricting outputs to words in the bank, establishing query-CI-product mappings (details in Appendix A.7).

In our method, multi-attribute keywords are organized into attribute-specific term banks (specifically *food*, *synonyms*, and *similar_foods*, as other attributes alone are insufficient for retrieval). Constrained beam search is applied to decode queries against these banks to construct query-keyword-product mappings, which directly reflect the quality of generated keywords in retrieving the correct products. As shown in Figure 4, our multi-attribute approach achieves a matching rate of 0.5112, substantially outperforming RARE (0.3127).

Results on other benchmarks. We further evaluate our method on the Food subset of the **ImplicitAVE** dataset (Zou et al., 2024). As shown in Figure 5, our approach achieves competitive performance compared with the previous SOTA method, **MICE** (Gong et al., 2025). Unlike MICE, which requires a complex pipeline involving multiple MLLMs for caption generation and summarization before inference, our method uses only a single MLLM. By enriching the prompt with three similar training samples retrieved via Semantic ID (SID) clustering, our approach achieves comparable results while significantly simplifying the inference pipeline and reducing computational cost (details in Appendix A.7).

4.3 Ablation Study

Clustering Method. To evaluate the advantage of the Semantic ID-based clustering strategy, we compare it with a k-means clustering baseline. For fairness, both methods use the same number of clusters per attribute, and the training set is augmented with up to two keywords per attribute label. As shown in Table 4, SID-based clustering consistently outperforms k-means in both Recall and F1 on Qwen3-VL-8B-Instruct, demonstrating the advantage of semantic ID representations for keyword augmentation.

Number of Codebook Layers and ID Layers. We further analyze the impact of the RQ-VAE codebook depth and the number of ID layers used for clustering. As shown in Table 5, the best balance between Recall and F1 is achieved when `codebook layer=4` and `ID layer=2`. Although setting `codebook layer=5` yields the highest Recall, its lower F1 indicates reduced precision and an inferior overall trade-off.

Number of Augmented Keywords. We also study the effect of the number of augmented keywords in the SID-based keyword augmentation

module. As shown in Figure 6, the model achieves the best Recall and F1 when two augmented keywords are used. While increasing the number of augmented keywords improves diversity, excessive augmentation introduces low-similarity noise, leading to performance degradation.

Qualitative Analysis. Figure 7 illustrates typical generation examples. When product texts lack detailed information, text-only models often produce incorrect keywords. In contrast, multimodal inputs and augmented training data help recover missing attributes and reduce mismatches caused by different lexical expressions.

5 Related Work

Product Keyword Generation. Product keyword generation plays an important role in advertising retrieval systems by improving product-query matching and recall. Early approaches relied on word-level statistical methods (Robertson and Zaragoza, 2009; Ramos, 2003), leveraging query logs and co-occurrence statistics to generate candidate keywords. With the development of neural models (Zhou et al., 2019; Li et al., 2019), keyword generation evolved to utilize product descriptions and user queries for more context-aware keyword prediction. More recently, large language models (LLMs) (Liu et al., 2025; Cai et al., 2026) have been applied to keyword generation due to their strong semantic understanding and instruction-following capabilities. However, most existing approaches rely primarily on textual information, overlooking rich semantic signals contained in product images. The emergence of multimodal models, such as GPT-4/5 (OpenAI et al., 2024), Qwen-VL (Bai et al., 2025b,a), and Gemini (Team et al., 2025), provides strong visual reasoning capabilities, enabling the extraction of fine-grained product attributes from images and opening new possibilities for multimodal keyword generation.

Product Semantic Clustering. Product semantic representations are commonly modeled using either dense embeddings or discrete identifiers. Dense representations, such as those produced by BERT (Devlin et al., 2019), capture rich semantic information from product descriptions. In contrast, traditional discrete IDs serve primarily as indexing keys and lack semantic structure, limiting their usefulness for similarity-based clustering. Recent work, such as Tiger (Rajput et al., 2023), addresses this limitation by introducing semantic IDs (SIDs) derived from the residual quantization mechanism of RQ-VAE (Lee et al., 2022). These hierarchical IDs encode product semantics at multiple levels of granularity, allowing semantically similar products to share similar ID patterns. Such representations provide a natural and efficient mechanism for product clustering and semantic grouping.

6 Conclusion

We propose a multimodal product keyword generation framework addressing two key challenges in existing systems: incomplete textual descriptions and limited keyword diversity. By incorporating visual information, our method recovers missing product attributes and enriches product representations, while semantic ID-based clustering discovers additional keywords from semantically related products, expanding the training set and increasing keyword diversity. Experiments on a large-scale Hong Kong food delivery dataset show that our framework significantly improves keyword recall, generalizes across different LLM architectures, and enhances downstream query-product matching. Our approach consistently outperforms the state-of-the-art method RARE in query-keyword matching and achieves competitive performance on the ImplicitAVE benchmark using only a single multimodal model, simplifying inference. These results demonstrate the effectiveness and practicality of our framework for real-world keyword generation and retrieval applications.

7 Limitations

Dataset: The current dataset is generated by GPT-4.1, followed by filtering using GPT-4.1 and manual review. However, the keyword generation criteria in the prompt are manually defined. Establishing stricter and more standardized criteria could further improve keyword quality, especially for attributes such as synonyms and similar products, where model performance remains suboptimal. Additionally, since most samples already contain detailed text, the contribution of the image modality may be limited.

References

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang,

Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025a. Qwen3-vl technical report. *Preprint*, arXiv:2511.21631.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025b. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Jingwen Cai, Sara Leckner, and Johanna Björklund. 2026. From precision to perception: Human-in-the-loop evaluation of keyword extraction for internet-scale contextual advertising. *Information Systems*, 138:102665.

Bowen Chen, Zhao Wang, and Shingo Takamatsu. 2025. OMS: On-the-fly, multi-objective, self-reflective ad keyword generation via LLM agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 18583–18601, Suzhou, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jiaying Gong, Hongda Shen, and Janet Jenq. 2025. MICE: Mixture of image captioning experts augmented e-commerce product attribute value extraction. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 1151–1160, Vienna, Austria. Association for Computational Linguistics.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. 2022. Autoregressive image generation using residual quantization. *Preprint*, arXiv:2203.01941.

Mu-Chu Lee, Bin Gao, and Ruofei Zhang. 2018. Rare query expansion through generative adversarial networks in search advertising. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 500508, New York, NY, USA. Association for Computing Machinery.

Yibin Lei, Yu Cao, Tianyi Zhou, Tao Shen, and Andrew Yates. 2024. Corpus-steered query expansion with large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 393–401, St. Julian's, Malta. Association for Computational Linguistics.

Yibin Lei, Tao Shen, and Andrew Yates. 2025. ThinkQE: Query expansion via an evolving thinking process. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 17772–17781, Suzhou, China. Association for Computational Linguistics.

Zhipeng Li, Jianwei Wu, Lin Sun, and Tao Rong. 2019. Combinatorial keyword recommendations for sponsored search with deep reinforcement learning. *Preprint*, arXiv:1907.08686.

Yijiang Lian, Zhijie Chen, Jinlong Hu, Kefeng Zhang, Chunwei Yan, Muchenxuan Tong, Wenying Han, Hanju Guan, Ying Li, Ying Cao, Yang Yu, Zhigang Li, Xiaochun Liu, and Yue Wang. 2019. An end-to-end generative retrieval method for sponsored search engine –decoding efficiently into a closed target domain. *Preprint*, arXiv:1902.00592.

Tongtong Liu, Zhaohui Wang, Meiyue Qin, Zenghui Lu, Xudong Chen, Yuekui Yang, and Peng Shu. 2025. Real-time ad retrieval via LLM-generative commercial intention for sponsored search advertising. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 28948–28960, Suzhou, China. Association for Computational Linguistics.

Hou-Wan Long, Yicheng Song, Zidong Wang, and Tianshu Sun. 2025. Kp-agent: Keyword pruning in sponsored search advertising via llm-powered contextual bandits. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, CIKM '25, page 49914995, New York, NY, USA. Association for Computing Machinery.

Daniel Morales-Brotons, Thijs Vogels, and Hadrien Hendrikx. 2024. Exponential moving average of weights in deep learning: Dynamics and benefits. *Preprint*, arXiv:2411.18704.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan H. Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Maheswaran Sathiamoorthy. 2023. Recommender systems with generative retrieval. *Preprint*, arXiv:2305.05065.

Juan Enrique Ramos. 2003. Using tf-idf to determine word relevance in document queries.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333389.

Gemini Team, Rohan Anil, and Sebastian Borgeaud. 2025. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, Zhaokai Wang, Zhe Chen, Hongjie Zhang, Ganlin Yang, Haomin Wang, Qi Wei, Jinhui Yin, Wenhao Li, Erfei Cui, and 56 others. 2025a. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *Preprint*, arXiv:2508.18265.

Yang Wang, Zheyi Sha, Kunhai Lin, Chaobing Feng, Kunhong Zhu, Lipeng Wang, Xuewu Jiao, Fei Huang, Chao Ye, Dengwu He, Zhi Guo, Shuanglong Li, and Lin Liu. 2024. One-step reach: Llm-based keyword generation for sponsored search advertising. In *Companion Proceedings of the ACM Web Conference 2024*, WWW '24, page 16041608, New York, NY, USA. Association for Computing Machinery.

Zhao Wang, Briti Gangopadhyay, Mengjie Zhao, and Shingo Takamatsu. 2025b. OKG: On-the-fly keyword generation in sponsored search advertising. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pages 115–127, Abu Dhabi, UAE. Association for Computational Linguistics.

Di Wu, Da Yin, and Kai-Wei Chang. 2024. KPEval: Towards fine-grained semantic-based keyphrase evaluation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1959–1981, Bangkok, Thailand. Association for Computational Linguistics.

Hao Zhou, Minlie Huang, Yishun Mao, Changlei Zhu, Peng Shu, and Xiaoyan Zhu. 2019. Domain-constrained advertising keyword generation. In *The World Wide Web Conference*, WWW '19, page 24482459, New York, NY, USA. Association for Computing Machinery.

Henry Peng Zou, Vinay Samuel, Yue Zhou, Weizhi Zhang, Liancheng Fang, Zihe Song, Philip S. Yu, and Cornelia Caragea. 2024. ImplicitAVE: An open-source dataset and multimodal LLMs benchmark for implicit attribute value extraction. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 338–354, Bangkok, Thailand. Association for Computational Linguistics.

	Food	Ingredients	Categories	Flavors	Cooking_methods	Functional_uses	Synonyms	Similar_foods
Qwen-VL-Plus	27.62	36.95	56.78	39.78	45.16	32.34	18.07	11.88
InternVL3_5-2B(zero-shot)	13.24	21.5	46.43	13.18	25.1	13.1	2.6	0.84
InternVL3_5-2B(SFT)	27.41	30.3	74.74	27.05	33.26	20.24	8.7	3.15
InternVL3_5-2B(aug_data SFT)	27.29	33.36	73.81	29.18	34.43	21.08	10.45	3.52
InternVL3_5-8B(zero-shot)	20.36	29.25	51.22	22.13	33.24	16.39	7.9	3.61
InternVL3_5-8B(SFT)	32.48	36.67	76.98	31.59	37.76	22.84	11.75	6.09
InternVL3_5-8B(aug_data SFT)	32.87	39.44	77.45	32.47	40.37	23.54	14	6.5
Qwen2.5-VL-3B(zero-shot)	13.64	20.54	54.56	16.25	27.68	10.71	4.9	1.5
Qwen2.5-VL-3B(SFT)	34.07	40.41	83.11	36.8	40.93	22.12	14.07	8.23
Qwen2.5-VL-3B(aug_data SFT)	36.04	45.32	82.53	41.05	43.68	24.78	17.14	8.22
Qwen2.5-VL-7B(zero-shot)	10.41	17.43	54.56	12.6	26.08	11.75	3.67	1.7
Qwen2.5-VL-7B(SFT)	37.41	43.25	82.19	37.27	43.49	24.2	15.43	9.45
Qwen2.5-VL-7B(aug_data SFT)	39.54	47.83	80.82	43.22	44.93	26.59	18.8	10.09
Qwen3-VL-2B(zero-shot)	10.95	17.76	37.36	13.99	17.67	5.68	5.9	1.27
Qwen3-VL-2B(SFT)	38.1	42.62	79.18	35.02	40.12	21.47	14.28	7.43
Qwen3-VL-2B(aug_data SFT)	40.59	47.58	80.8	38.76	41.42	23.77	17.63	8.22
Qwen3-VL-4B(zero-shot)	20.7	31.31	62.27	27.43	35.53	19.67	12.62	4.9
Qwen3-VL-4B(SFT)	41.53	45.16	78.67	37.1	42.22	23.1	16.1	9.84
Qwen3-VL-4B(aug_data SFT)	43.26	49.46	79.56	41.82	43.61	25.3	19.94	10.45
Qwen3-VL-8B(zero-shot)	22.84	32.89	50.66	31.03	39.43	23.92	13.01	6.81
Qwen3-VL-8B(SFT/text)	37.63	41.13	80.75	37.18	42.33	22.78	13.66	8.9
Qwen3-VL-8B(SFT/image)	39.38	34.42	76.41	36.39	41.96	21.98	14.96	8.89
Qwen3-VL-8B(SFT/text+mage)	41.76	45.4	81.08	35.73	43.51	22.66	16.6	9.95
Qwen3-VL-8B(aug_data SFT)	45.15	51.01	80.54	41.79	44.94	24.83	21.24	11.57

Figure 8: Evaluation Results of **Recall** for Eight Attribute Keywords under Various Models and Experimental Settings.

	Food	Ingredients	Categories	Flavors	Cooking_methods	Functional_uses	Synonyms	Similar_foods
Qwen-VL-Plus	33.56	43.43	52.32	36.28	41.01	30.59	21.52	14.96
InternVL3_5-2B(zero-shot)	17.64	22.61	45.6	14.13	25.41	13.78	3.33	1.21
InternVL3_5-2B(SFT)	27.32	30.96	68.77	27.52	33.76	22.03	10.56	4.22
InternVL3_5-2B(aug_data SFT)	22.23	28.83	63.86	25.59	31.77	21.65	10.24	4.17
InternVL3_5-8B(zero-shot)	28.03	32.32	47.94	23.41	35.22	17.49	10.23	4.99
InternVL3_5-8B(SFT)	32.16	37.93	71.21	32.06	38.63	24.62	14.12	8.07
InternVL3_5-8B(aug_data SFT)	26.67	34.53	67.22	28.1	37.15	23.52	13.4	7.58
Qwen2.5-VL-3B(zero-shot)	20.32	22.98	53.66	18.7	29.37	12.95	6.39	2.24
Qwen2.5-VL-3B(SFT)	35.28	44.12	79.07	39.78	45.69	25.31	18.29	11.4
Qwen2.5-VL-3B(aug_data SFT)	34.35	44.79	77.2	38.29	45.36	25.91	18.64	10.35
Qwen2.5-VL-7B(zero-shot)	15.5	20.36	51.78	14.63	29.71	13.56	4.87	2.4
Qwen2.5-VL-7B(SFT)	38.29	46.92	77.98	40.22	48.14	27.73	20.02	13.02
Qwen2.5-VL-7B(aug_data SFT)	37.22	46.97	75.66	40.2	47.41	27.77	20.65	12.75
Qwen3-VL-2B(zero-shot)	15.96	19.38	33.32	13.92	17.57	5.57	6.48	1.69
Qwen3-VL-2B(SFT)	38.37	45.84	75.37	37.59	44.38	24.54	18.48	10.3
Qwen3-VL-2B(aug_data SFT)	36.6	45.92	75.28	36.03	43.31	25.3	19.53	10.4
Qwen3-VL-4B(zero-shot)	28.15	35.47	57.87	27.11	37.51	19.32	15.05	6.7
Qwen3-VL-4B(SFT)	41.4	48.56	74.84	39.54	46.65	26.19	20.86	13.46
Qwen3-VL-4B(aug_data SFT)	38.67	47.45	74.44	38.29	45.37	27.53	21.66	13.05
Qwen3-VL-8B(zero-shot)	30.61	38.1	48.18	32.38	41.17	24.04	14.9	8.81
Qwen3-VL-8B(SFT/text)	38.57	44.71	76.28	38.32	45.75	25.31	17.45	12.14
Qwen3-VL-8B(SFT/image)	40.39	37.13	72.34	39.2	46.27	25.13	19.44	12.18
Qwen3-VL-8B(SFT/text+mage)	41.77	48.71	77.06	40.09	47.71	25.78	21.39	13.64
Qwen3-VL-8B(aug_data SFT)	41.26	49.56	74.58	39.05	45.69	27.16	22.6	13.91

Figure 9: Evaluation Results of **F1** for Eight Attribute Keywords under Various Models and Experimental Settings.

A Appendix

A.1 Main Results by Attribute

Figures 8 and 9 present attribute-level performance across all model variants. After fine-tuning on the augmented training set, Qwen3-VL-8B-Instruct achieves the best or near-best results on most attribute categories. An exception is the *functional_uses* attribute, where performance is notably lower than that of Qwen-VL-Plus and the gains from data augmentation are limited. This may be due to the inherent difficulty of modeling functional-use semantics and the relatively limited training data available for this attribute.

A.2 Qualitative Analysis of RQ-VAE SIDs

Figure 10 visualizes selected product text SIDs using multidimensional scaling. Products with sim-

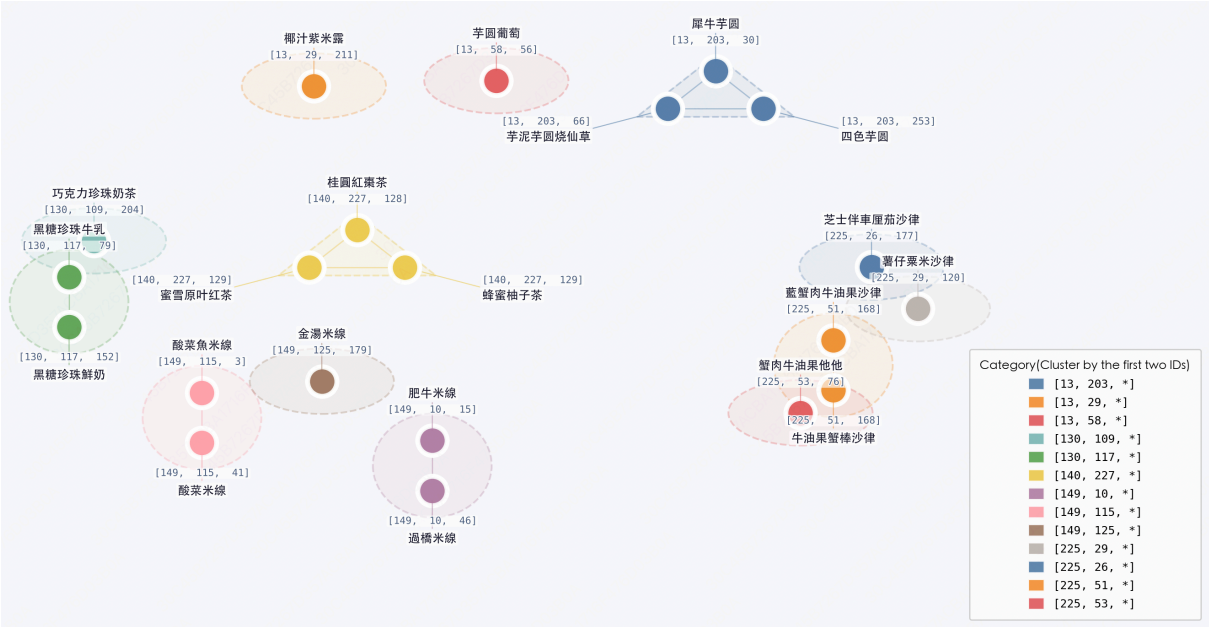

Figure 10: Visualization of selected product text SIDs using Multidimensional Scaling (MDS).

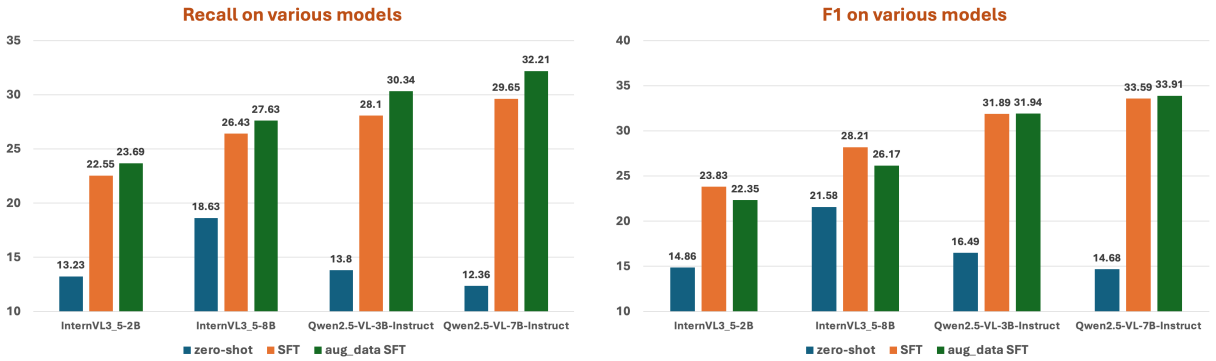

Figure 11: The performance of our method evaluated on **InternVL3_5** series and **Qwen2.5-VL** model series.

Codebook Layers	ID Layer	Recall	F1
3 layers	layer2	29.41	32.07
4 layers	layer2	**31.03**	32.23
	layer3	29.16	**32.25**
5 layers	layer2	30.93	31.97
	layer3	29.25	32.27
	layer4	29.04	32.34

Table 6: Ablation study on the number of code layers used in the semantic ID model (**Qwen3-VL-2B-Instruct**).

Codebook Layers	ID Layer	Recall	F1
3 layers	layer2	30.86	33.63
4 layers	layer2	**32.73**	34.02
	layer3	30.84	34.07
5 layers	layer2	32.67	33.71
	layer3	30.73	33.97
	layer4	30.64	**34.15**

Table 7: Ablation study on the number of code layers used in the semantic ID model (**Qwen3-VL-4B-Instruct**).

ilar semantic IDs appear close to each other in the projection and correspond to semantically related items, such as *"Blue Crab Meat and Avocado Salad"* and *"Avocado and Crab Stick Salad"*. In contrast, products with dissimilar SIDs are positioned farther apart; for example, *taro ball*, *salad*, and *rice noodle* products form clearly separated regions. These observations indicate that the learned semantic IDs effectively capture product-level semantic similarity, providing a reliable basis for clustering.

Data Size	Recall	F1
5k	30.11	31.45
1w	31.22	32.61
3w	**33.59**	**34.81**

Table 8: Ablation study on the amount of data.

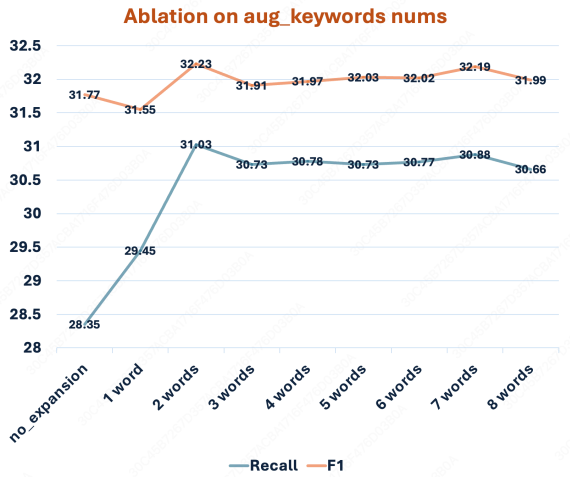

Figure 12: A ablation study on increasing the number of keywords on the **Qwen3-VL-2B-Instruct** model.

A.3 Main Results on Other Models

We further evaluate our method on the InternVL3_5 and Qwen2.5-VL model series, as shown in Figure 11. The results demonstrate consistent performance improvements across different architectures and model scales, indicating strong robustness of the proposed framework.

A.4 Scaling Law

Table 8 analyzes the impact of training data size by evaluating models trained on 5K, 10K, and 30K samples. Together with the main results, the findings show that model performance consistently improves with increased training data, suggesting that the proposed framework follows a clear scaling trend with respect to both data volume and model capacity.

A.5 Ablation Results on Other Models

Tables 6 and 7 present ablation studies on the number of codebook layers in the semantic ID model and the number of ID layers used for clustering, conducted on Qwen3-VL-2B-Instruct and Qwen3-VL-4B-Instruct. Both models exhibit the same optimal configuration as Qwen3-VL-8B-Instruct, achieving the best balance between Recall and F1 when `codebook_layer`=4 and `ID_layer`=2.

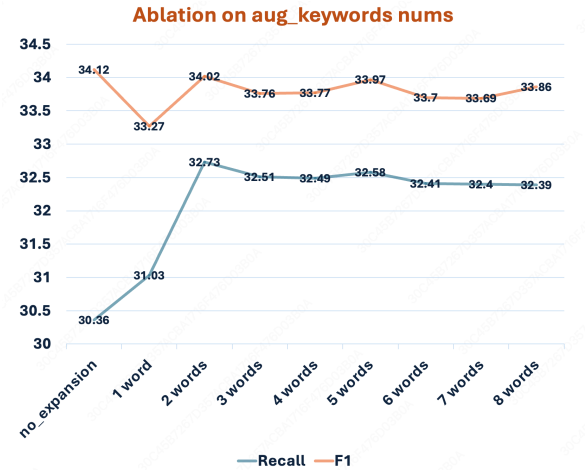

Figure 13: A ablation study on increasing the number of keywords on the **Qwen3-VL-4B-Instruct** model.

Figures 12 and 13 further analyze the number of augmented keywords used in the keyword enhancement module. The optimal configuration is consistent across models: using two augmented keywords yields the best results in both Recall and F1.

Overall, these results indicate that the selected hyperparameters generalize well across different model sizes rather than being specific to a particular architecture.

A.6 Data Quality Verification

To assess the reliability of GPT-4.1-generated annotations, we randomly sampled 200 products and asked human annotators to independently label the keywords for each attribute. We compared GPT-4.1-generated labels with human annotations.

Table 9 reports both exact-match metrics (Precision, Recall, F1) and semantic-aware metrics (Sem-Precision, Sem-Recall, Sem-F1) (Wu et al., 2024). While exact lexical matching yields relatively lower scores, we observe strong semantic agreement between GPT-4.1 labels and human annotations across all attributes.

Manual inspection of mismatched cases shows that most discrepancies arise from paraphrases or stylistic differences. For example, ingredient descriptions may appear as crab meat versus crab, or dish variations such as avocado crab salad versus crab and avocado salad. Such variations are semantically equivalent but penalized by strict lexical matching.

The semantic-aware metrics, which account for synonymy and paraphrasing, therefore provide a more appropriate evaluation for keyword genera-

Attribute	Precision	Recall	F1	Sem-P	Sem-R	Sem-F1
food	0.5566	0.6330	0.5794	0.9241	0.9496	0.9358
ingredients	0.6223	0.6903	0.6476	0.9387	0.9516	0.9448
flavors	0.4726	0.5967	0.5119	0.9276	0.9552	0.9408
categories	0.6278	0.6361	0.6280	0.9803	0.9751	0.9776
cooking_methods	0.6028	0.7439	0.6374	0.9330	0.9605	0.9455
functional_uses	0.4774	0.5100	0.4824	0.9217	0.9267	0.9237
synonyms	0.3512	0.4197	0.3517	0.9410	0.9523	0.9464
similar_foods	0.4194	0.5647	0.4595	0.9172	0.9336	0.9250
macro_average	**0.5163**	**0.5993**	**0.5397**	**0.9355**	**0.9506**	**0.9424**

Table 9: Evaluation of the effectiveness of GPT-4.1 compared to human annotation on 200 sampled samples.

tion. The high semantic scores indicate that GPT-4.1 labels are largely consistent with human annotations at the meaning level.

Overall, these results suggest that GPT-4.1-generated labels provide reliable semantic supervision and can serve as a practical reference standard for training.

A.7 Implementation Details

Experimental Setup for Comparison with RARE. To reproduce the RARE baseline, we use Qwen3-VL-8B-Instruct to generate Commercial Intent (CI) terms from both product images and textual descriptions, producing 25 CIs per product. During constrained beam search decoding, we set `beam_size`=50, `max_step`=12, and the target output length to 25 words. A `length_penalty` of 1.2 is applied to reduce the bias toward shorter outputs.

For our method, we construct attribute-specific keyword banks and select the three most relevant attributes-*food*, *synonyms*, and *similar_foods*for constrained beam search. Based on the average keyword counts of these attributes, the target output lengths are set to 7, 7, and 11, respectively (25 words in total). The final matching rate is computed as the joint query-attribute-product matching rate across these attributes.

Experimental Setup on the ImplicitAVE Food Subset. Since each sample in the ImplicitAVE dataset contains only a single attribute with one value, the label augmentation strategy used in our main experiments cannot be directly applied. Instead, after incorporating multimodal inputs and generating Semantic IDs (SIDs) for clustering, we modify the inference procedure accordingly.

Specifically, we augment the prompt with three semantically similar training samples (including their ground-truth labels) retrieved through SID-based clustering. We also append the complete attribute vocabulary to the prompt, constraining the model to generate outputs from this predefined set. No additional training is performed; the model directly performs zero-shot inference on the test set, and performance is evaluated using accuracy.

During SID model training for this dataset, we retain the four-layer codebook structure. Due to the smaller dataset size (approximately 3K products), the codebook size of each layer is reduced to 64, while all other hyperparameters remain consistent with the main experiments.