

Preference Shapes Relevance: Cross-component Hierarchical Semantic Alignment for Personalized Generative Retrieval

Gaoming Zhang¹ Angqing Jiang¹ Jianchun Song² Kena Qi²
Dayao Chen² Wei Lin² Defu Lian^{1*}

¹University of Science and Technology of China

²Meituan

{zzzgm, philipgaq}@mail.ustc.edu.cn liandefu@ustc.edu.cn

{songjianchun, qikena, chendayao, linwei31}@meituan.com

Abstract

Generative Retrieval (GR) has emerged as a promising paradigm by mapping queries directly to Semantic IDs (SIDs) with powerful representation capabilities for candidate items. However, existing SIDs derived solely from item content create a semantic gap, failing to align dynamic query intents with static item representations. Furthermore, current generative paradigms rarely model user behavior sequences and are always bottlenecked by the high inference latency of beam-search autoregressive decoding. To address these challenges, we propose Cross-component Hierarchical semantic Alignment for Personalized generative retrieval (CHAP), a novel personalized GR framework from a hierarchical perspective. First, we design a Hierarchical Semantic Alignment module to align query’s latent space with item’s quantization path and synchronize multi-granular semantics. Second, we construct a personalized GR framework that models user behavior by synergizing discrete SIDs for structural guidance and continuous representations for fine-grained semantic refinement. Notably, we introduce a Residual Cascading Generation mechanism to restrict the costly multi-step Transformer Decoder to a single-pass inference, boosting inference throughput while mitigating information loss. Extensive experiments on three public datasets, one proprietary industrial dataset, and online A/B tests demonstrate CHAP’s superiority, validating the effectiveness and practical value of our approach. The code is publicly available at <https://github.com/zzzgm/CHAP>.

1 Introduction

Modern Information Retrieval (IR) systems traditionally rely on the "Index-Retrieve-Rank" cascade pipeline (Wang et al., 2011; Zhou et al., 2021), employing sparse methods like BM25 (Robertson et al., 2009) for keyword matching and dense

methods like DPR (Karpukhin et al., 2020) for semantic vector mapping. However, static models often struggle to adapt to dynamic, personalized user intents (Liu et al., 2020). Recently, Generative Retrieval (GR) has emerged as a disruptive paradigm alongside the advancement of Large Language Models (LLMs) (Tang et al., 2023b; Li et al., 2025). By adopting an end-to-end approach that transforms retrieval into a sequence generation task which directly generates relevant item identifiers (ItemIDs) from user queries, GR eliminates the need for complex external indices. This paradigm significantly enhances scalability and demonstrates superior performance in long-tail queries and cold-start problems (Kuo et al., 2024; Yuan et al., 2024).

In the GR architecture, generating Semantic IDs (SIDs) with robust representational capabilities remains a core challenge (Zhang et al., 2025b; Liu et al., 2025). Existing models like TIGER (Rajput et al., 2023) and HierGR (Zhang et al., 2025a) train SIDs solely on item corpora, lacking explicit query perception and failing to bridge the inherent *semantic gap* between user intents and item descriptions, which leads to poor generalization on complex queries (Zhang et al., 2025c; Zheng et al., 2025). Furthermore, current methods exhibit critical limitations in structural utilization, personalization, and inference efficiency. Treating quantized SIDs merely as flat symbol sequences ignores the natural "Coarse-to-Fine" hierarchy, wasting its benefits for feature generalization and semantic stability (Williams et al., 2020; Si et al., 2024). Relying solely on discrete sparse IDs also discards fine-grained continuous vector details, hindering the integration of user behavior sequences (Yang et al., 2025). Crucially, standard autoregressive GR paradigms suffer from severe computational bottlenecks during online serving; repeatedly executing the heavy Transformer Decoder (especially the costly Cross-Attention) for each hierarchical step causes prohibitive latency, hindering large-scale

*Corresponding author.

deployment in latency-sensitive industrial systems.

To address these limitations, we propose **Cross-component Hierarchical semantic Alignment for Personalized generative retrieval (CHAP)**, a novel framework which performs hierarchical alignment on pre-trained SIDs and efficiently fuses sparse and dense representations in a sequence model.

Specifically, CHAP initializes SIDs using a Residual Quantized Variational Autoencoder (RQ-VAE) (Rajput et al., 2023). To bridge the query-item semantic gap while preventing codebook collapse, we design a multi-module **Hierarchical Semantic Alignment** paradigm which includes a cross-sample alignment module to proactively pull the query’s potential representation towards the target’s quantization space, a hierarchical-aware contrastive learning module to enforce coarse-to-fine alignment, and a soft probability distillation module to regularize the output distribution of the encoder via KL divergence. Integrating these modules ensures the generated SIDs can robustly comprehend complex query semantics.

Furthermore, CHAP constructs a **Personalized Cascading Generative Sequence Model** that implements multi-granular modeling of user queries and personalized behavior sequences by cross-cascading aligned sparse SIDs with its raw dense vector. Unlike standard autoregressive approaches, CHAP implements a **Residual Cascading Generation** strategy which decouples heavy historical intent routing from hierarchical code generation. By restricting the costly multi-step Transformer Decoder to a single-pass inference and offloading the layer-wise generation to lightweight residual blocks, this mechanism not only mitigates information loss but also acts as an efficient decoding accelerator, significantly boosting inference throughput. Our main contributions are as follows:

- We propose a Hierarchical Semantic Alignment paradigm, effectively bridging the inherent Semantic Gap between user intents and static item codes.
- We design a Personalized GR Architecture equipped with a Residual Cascading Generation mechanism, which resolves the inherent latency bottleneck of autoregressive GR while effectively mitigating information loss through coarse-to-fine structural guidance.
- Extensive experiments on three public datasets and one proprietary industrial dataset,

as well as online A/B tests demonstrate the superiority of our method.

2 Related Works

Sparse retrieval methods like BM25 (Robertson et al., 2009) and TF-IDF (Ramos et al., 2003) rely on exact term matching but suffer from vocabulary mismatch. Conversely, **dense retrieval** approaches (e.g., DPR (Karpukhin et al., 2020), ANCE (Xiong et al., 2020)) map inputs into a shared vector space, enhancing semantic matching but necessitating costly external indices like HNSW (Malkov and Yashunin, 2018). **Personalized retrieval** tailors results by incorporating user contexts. QEM (Ai et al., 2019a) investigates personalization mechanisms, while DREM (Ai et al., 2019b) models dynamic user-product relationships via knowledge graphs. Embedding-based approaches like ZAM (Ai et al., 2019a), HEM (Ai et al., 2017), and TEM (Bi et al., 2020) integrate user profiles and history into unified embeddings.

Generative retrieval internalizes corpus knowledge to directly generate ItemIDs. DSI (Tay et al., 2022) pioneered this paradigm, followed by semantic enhancements in SE-DSI (Tang et al., 2023a). TIGER (Rajput et al., 2023) first utilized RQ-VAE for SIDs. Regarding architecture, COBRA (Yang et al., 2025) integrates sparse and dense vectors, while CAT-ID² (Liu et al., 2025), and MERGE (Zhang et al., 2025b) optimize ID generation by exploiting multi-level correlations.

3 Preliminaries

3.1 Problem Definition

Given a user’s interaction history $S = \{i_1, \dots, i_N\}$ and a current query q , Personalized GR aims to retrieve items by directly generating their identifiers such as a unique hierarchical SID $\mathbf{c} = (c^0, \dots, c^{L-1})$ derived from a codebook. The objective is to learn a probabilistic model θ that maximizes the probability of generating the target SID given the context:

$$\begin{aligned} P(i|q, S; \theta) &= P(\mathbf{c}|q, S; \theta) \\ &= \prod_{l=0}^{L-1} P(c^l | c^{<l}, q, S; \theta). \end{aligned} \quad (1)$$

where $c^{<l}$ denotes the generated prefix tokens at layer l . Inference is to generate the target SID with the highest joint probability.

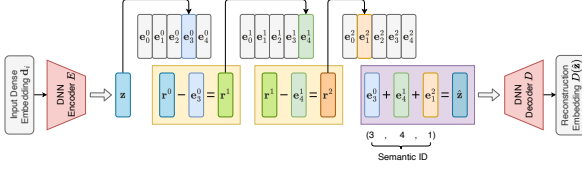


Figure 1: Schematic of SID generation via RQ-VAE.

3.2 Semantic ID Generation via RQ-VAE

To assign meaningful discrete identifiers to items effectively, we employ RQ-VAE (Rajput et al., 2023) (Fig. 1). This approach generates hierarchical SIDs by recursively quantizing item embeddings in a coarse-to-fine manner, which serves as the foundational representation for our proposed framework.

Given an item i , a pre-trained language encoder first encodes its textual content into a dense embedding $\mathbf{d}_i$. A deep neural network encoder $E(\cdot)$ then maps this embedding to a latent representation $\mathbf{z} = E(\mathbf{d}_i)$. The quantization process utilizes L residual layers, where each layer l is equipped with a codebook $\mathcal{C}^l = \{\mathbf{e}_k^l\}_{k=1}^K$ containing K code words. At each step l , quantization is performed by selecting the code word nearest to the current residual vector $\mathbf{r}^l$ (where $\mathbf{r}^0 = \mathbf{z}$). The *Semantic Token* c^l for layer l is determined as:

$$c^l = \arg \min_k \|\mathbf{r}^l - \mathbf{e}_k^l\|_2^2, \quad \mathbf{r}^{l+1} = \mathbf{r}^l - \mathbf{e}_{c^l}^l. \quad (2)$$

After L iterations, we obtain a discrete coarse-to-fine SID $(c^0, \dots, c^{L-1})$. The quantized latent representation $\hat{\mathbf{z}}$ is then approximated by the sum of the selected codebook vectors: $\hat{\mathbf{z}} = \sum_{l=0}^{L-1} \mathbf{e}_{c^l}^l$.

The training of RQ-VAE aims to reconstruct the original input while optimizing the codebooks. The decoder $D(\cdot)$ attempts to reconstruct the input $\mathbf{d}_i$ from $\hat{\mathbf{z}}$, resulting in the reconstruction loss:

$$\mathcal{L}_{\text{recon}} = \|\mathbf{d}_i - D(\hat{\mathbf{z}})\|_2^2. \quad (3)$$

To ensure the stability of quantization, RQ-VAE employs a split quantization objective consisting of a codebook loss to update codebook vectors towards the residuals and a commitment loss to constrain the encoder output:

$$\mathcal{L}_{\text{codebook}} = \sum_{l=0}^{L-1} \|sg[\mathbf{r}^l] - \mathbf{e}_{c^l}^l\|_2^2 \quad (4)$$

$$\mathcal{L}_{\text{commit}} = \beta \sum_{l=0}^{L-1} \|\mathbf{r}^l - sg[\mathbf{e}_{c^l}^l]\|_2^2, \quad (5)$$

where $sg[\cdot]$ denotes the stop-gradient operation which prevents gradient update during backpropagation, and β is a hyperparameter balancing the commitment loss. The final training objective is:

$$\mathcal{L}_{\text{RQ-VAE}} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{codebook}} + \mathcal{L}_{\text{commit}}. \quad (6)$$

4 Methodology

4.1 Hierarchical Semantic Alignment (HSA)

Directly applying an item-pretrained RQ-VAE to retrieval faces a semantic gap, as its encoder $E(\cdot)$ is optimized solely for item reconstruction. To bridge this, we propose a Hierarchical Semantic Alignment (HSA) paradigm (Fig. 2). To prevent semantic drift and catastrophic forgetting, we freeze the pre-trained codebook, forcing the query quantization path directly into the stable item manifold.

Cross-Sample Alignment (CSA). To pull the query’s latent representation into the item’s quantization space, we repurpose RQ-VAE losses by swapping input-target variables. Given a query q and target t , the *Cross-Commitment* loss forces the query’s residual $\mathbf{r}_q^l$ to align with the target’s selected code $\mathbf{e}_{c^l}^t$:

$$\mathcal{L}_{\text{CC}} = \beta \sum_{l=0}^{L-1} \|\mathbf{r}_q^l - sg[\mathbf{e}_{c^l}^t]\|_2^2. \quad (7)$$

The *Cross-Reconstruction* loss ensures the target’s raw content $\mathbf{d}_t$ is reconstructable from the query’s quantized representation $\hat{\mathbf{z}}_q$:

$$\mathcal{L}_{\text{CR}} = \|\mathbf{d}_t - D(\hat{\mathbf{z}}_q)\|_2^2. \quad (8)$$

The total CSA loss is $\mathcal{L}_{\text{CSA}} = \mathcal{L}_{\text{CC}} + \mathcal{L}_{\text{CR}}$.

Hierarchical-Aware Contrastive Learning (HCL). To explicitly exploit the “Coarse-to-Fine” SID structure, we align the query’s partial quantized representation $\hat{\mathbf{z}}_q^{(l)} = \sum_{i=0}^l \mathbf{e}_{c_q^i}^i$ with the target’s partial quantized representation $\hat{\mathbf{z}}_t^{(l)} = \sum_{i=0}^l \mathbf{e}_{c_t^i}^i$ accumulated up to depth l :

$$\begin{aligned} \mathcal{L}_{\text{HCL}} &= - \sum_{l=0}^{L-1} \log \frac{\exp(s_l^+ / \tau)}{\sum_{j \in \mathcal{B}} \exp(s_{lj} / \tau)}, \\ s_l^+ &= \text{sim}(\hat{\mathbf{z}}_q^{(l)}, sg[\hat{\mathbf{z}}_t^{(l)}]), \\ s_{lj} &= \text{sim}(\hat{\mathbf{z}}_q^{(l)}, sg[\hat{\mathbf{z}}_j^{(l)}]). \end{aligned} \quad (9)$$

where $\mathcal{B}$ is the batch containing the positive target and negatives, and τ is the temperature.

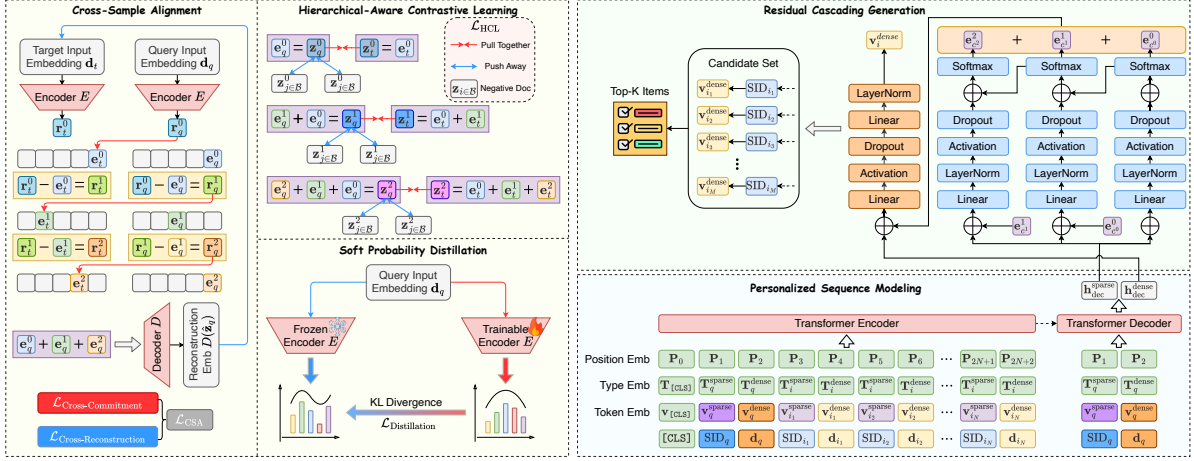


Figure 2: **Overall architecture of CHAP**, featuring HSA to bridge the query-item gap (left), and dual-view Personalized Sequence Modeling with Residual Cascading Generation for efficient coarse-to-fine retrieval (right).

Soft Probability Distillation. To prevent drastic latent shifts caused by hard quantization assignments, we distill the soft code assignment distribution from the frozen item-side quantization path of the target item to the trainable query encoder via KL divergence:

$$\mathcal{L}_{\text{Distill}} = \sum_{l=0}^{L-1} \text{KL} (P_{\text{item}}(\cdot|t) \parallel P_{\text{query}}(\cdot|q)), \quad (10)$$

where the soft assignment probability is defined by the Euclidean distance to the residual:

$$P(c^l = k|q) = \frac{\exp(-\|\mathbf{r}_q^l - \mathbf{e}_k^l\|_2^2)}{\sum_{j=1}^K \exp(-\|\mathbf{r}_q^l - \mathbf{e}_j^l\|_2^2)}. \quad (11)$$

The comprehensive HSA objective is optimized via a weighted sum:

$$\mathcal{L}_{\text{HSA}} = \mathcal{L}_{\text{CSA}} + \gamma \cdot \mathcal{L}_{\text{HCL}} + \delta \cdot \mathcal{L}_{\text{Distill}}. \quad (12)$$

4.2 Personalized Sequence Modeling

Dual-View Sequence Modeling. To capture the multi-granularity preferences of users and generate high-fidelity retrieval representations, we construct an input sequence where each interaction synergizes an aligned discrete SID and a continuous dense vector. For a user history $S = \{i_1, \dots, i_N\}$ and query q , the input is organized as:

$$X_{in} = [\mathbf{v}_{[\text{CLS}]}, \mathbf{v}_q^{\text{sparse}}, \mathbf{v}_q^{\text{dense}}, \mathbf{v}_{i_1}^{\text{sparse}}, \mathbf{v}_{i_1}^{\text{dense}}, \dots]. \quad (13)$$

The sparse view $\mathbf{v}^{\text{sparse}}$ maps the aggregated codebook vectors ($\sum \mathbf{e}_{c^l}^l$) into a unified dense space,

while $\mathbf{v}^{\text{dense}}$ corresponds to the raw language model embedding. To maintain training-inference consistency and prevent exposure bias, the unified decoder is initialized exclusively with the query’s dual representation ($\mathbf{v}_q^{\text{sparse}}, \mathbf{v}_q^{\text{dense}}$). This explicitly forces the decoder to “look back” via cross-attention to retrieve personalized results.

Hybrid Optimization. Our multi-task framework optimizes the **Sparse Generation Loss** for hierarchical SID classification across all L levels:

$$\mathcal{L}_{\text{sparse}} = - \sum_{l=0}^{L-1} \log \left(\frac{\exp(s_{c^l}^l)}{\sum_{k=1}^K \exp(s_k^l)} \right), \quad (14)$$

where $s^l \in \mathbb{R}^K$ is the logit vector at level l . Simultaneously, to avoid Mean Squared Error instability, we employ an InfoNCE-based **Dense Contrastive Loss** ($\mathcal{L}_{\text{dense}}$) over the batch set $\mathcal{B} = \{\mathbf{v}^+\} \cup \mathcal{B}^-$:

$$\mathcal{L}_{\text{dense}} = - \log \frac{\exp(\text{sim}(\hat{\mathbf{v}}, \mathbf{v}^+)/\tau)}{\sum_{\mathbf{v}' \in \mathcal{B}} \exp(\text{sim}(\hat{\mathbf{v}}, \mathbf{v}')/\tau)}, \quad (15)$$

treating the predicted dense vector $\hat{\mathbf{v}}$ (introduced in Sec. 4.3) as the anchor. The joint objective $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{sparse}} + \mathcal{L}_{\text{dense}}$ creates a robust curriculum learning effect, where sparse structural constraints simplify the dense ranking optimization.

4.3 Residual Cascading Generation and Retrieval

To achieve a balance between structural correctness and fine-grained semantic precision, we adopt a “Coarse-to-Fine” strategy to coordinate the prediction of discrete SIDs and continuous dense vectors. Formally, we decompose the generation probability of a target item i , given the context X , into

two stages: the joint probability of the L hierarchical Semantic Tokens, followed by the conditional probability of the dense vector (operationalized by a cosine-softmax score over generated candidates):

$$P(i|X) = \underbrace{\prod_{l=0}^{L-1} P(c_i^l | c_i^{<l}, X)}_{\text{Sparse Generation}} \cdot \underbrace{P(\mathbf{v}_i | c_i^{0:L-1}, X)}_{\text{Dense Refinement}}. \quad (16)$$

An entropy-based justification for this cascaded sparse-dense factorization is provided in App. B.

Coarse-to-Fine Residual Generation. To avoid the prohibitive latency of standard autoregressive decoding, we propose a Single-pass Residual Generation Mechanism. The Transformer Decoder executes only once over the query’s input tokens. Specifically, let $\mathbf{h}_{\text{dec}}^{\text{sparse}}$ and $\mathbf{h}_{\text{dec}}^{\text{dense}}$ denote the output hidden states corresponding to the discrete query SID position and the continuous query embedding position, respectively. Utilizing $\mathbf{h}_{\text{dec}}^{\text{sparse}}$ as a global structural context anchor, subsequent states for predicting level l are updated via lightweight residual blocks:

$$\mathbf{h}^{(l+1)} = \text{ResBlock} \left(\text{Concat}[\mathbf{h}_{\text{dec}}^{\text{sparse}}, \mathbf{e}_{c_l}^l] \right) + \mathbf{h}^{(l)}. \quad (17)$$

This decoupled design acts as a powerful decoding accelerator. To align the continuous vector with this discrete structure, we inject the reconstructed quantization $\hat{\mathbf{z}} = \sum_{l=0}^{L-1} \mathbf{e}_{c_l}^l$ into the dense prediction head conditioned on $\mathbf{h}_{\text{dec}}^{\text{dense}}$:

$$\hat{\mathbf{v}} = \text{PredHead} \left(\text{Concat}[\mathbf{h}_{\text{dec}}^{\text{dense}}, \hat{\mathbf{z}}] \right), \quad (18)$$

achieving an elegant balance between structural correctness and fine-grained semantic precision.

Candidate Scoring. During inference, we maximize throughput via a **parallel sampling strategy** to generate M candidate SIDs. We calculate the accumulated log-probability of the sampled hierarchical path as the sparse signal S_{sparse} , and the cosine similarity between the predicted dense vector $\hat{\mathbf{v}}$ corresponding to the sampled SID and the candidate item’s raw vector $\mathbf{v}_i$ as the dense signal S_{dense} . Both signals are normalized using a temperature-scaled Softmax over the candidate pool $\mathcal{I}_{\text{cand}}$:

$$\mathcal{N}(S) = \frac{\exp(S/\tau)}{\sum_{j \in \mathcal{I}_{\text{cand}}} \exp(S_j/\tau)}. \quad (19)$$

The final score fuses both dual-view intents:

$$S_{\text{final}}(i) = \mathcal{N}(S_{\text{sparse}}(i)) \cdot \mathcal{N}(S_{\text{dense}}(i)). \quad (20)$$

Table 1: Statistics of the datasets in our experiments.

Dataset	#Items	#Queries	#Q-I Pairs	Avg. Hist.
ESCI-us	1,215,854	97,346	1,818,825	-
KuaiSearch	6,634,118	295,561	682,228	10.37
Amazon	35,772	9,070	9,070	40.12
Local-Life	10,145,542	3,089,026	3,089,026	7.47

5 Experiments

We analyze CHAP’s effectiveness and address these research questions in this section: **RQ1:** How does CHAP perform against state-of-the-art baselines? **RQ2:** How does semantic alignment improve generated ItemID quality? **RQ3:** How do core components contribute across intent-wise query segments? **RQ4:** Does our residual architecture resolve GR latency bottlenecks? **RQ5:** How robust is CHAP to codebook and decoding hyperparameters? **RQ6:** Can CHAP drive tangible business growth in live online production?

5.1 Experimental Settings

5.1.1 Datasets

To validate the effectiveness of CHAP, we conduct extensive experiments on four real-world datasets. **ESCI-us** (Reddy et al., 2022) features authentic Amazon search queries with complex linguistic patterns and strict attribute constraints. **KuaiSearch** (Li et al., 2026) provides a massive e-commerce search corpus from Kuaishou, preserving real queries, long-term user behavior sequences, and long-tail product distributions. **Amazon** (Cai et al., 2025) integrates diverse user profiles and historical web behaviors to rigorously evaluate highly personalized search instructions. Lastly, **Local-Life** is a large-scale industrial dataset collected from 30 days of user interaction logs on a leading platform, covering heterogeneous search intents with raw natural language texts. Detailed descriptions and comprehensive statistics for all four datasets are provided in Tab. 1 and App. C.

5.1.2 Baselines

To comprehensively evaluate the performance of CHAP, we compare it against a diverse set of 14 representative baselines across four categories. For sparse retrieval, we select **BM25** (Robertson et al., 2009), **Doc2Query** (Nogueira et al., 2019), and **DeepCT** (Dai and Callan, 2019). For dense retrieval, we employ **DPR** (Karpukhin et al., 2020), **Sen-T5** (Ni et al., 2022), and **MPNet** (Song et al., 2020). For personalized retrieval scenarios,

Table 2: Performance comparison on four datasets. **Bold** indicates the result significantly outperforms all baselines with paired t-tests at $p < 0.05$ level, and underlined values denote the second-best results.

Type	Model	ESCI-us			KuaiSearch			Amazon			Local-Life		
		R@10	R@50	NDCG@50	R@10	R@50	HR@50	R@10	R@50	NDCG@50	R@10	R@50	MRR@10
Sparse	BM25	0.0480	0.0932	0.0811	0.0706	0.1564	0.2088	0.4807	0.6780	0.3887	0.2531	0.3415	0.0997
	Doc2Query	0.0551	0.1410	0.1015	0.0784	0.1772	0.2382	0.5064	0.6904	0.4039	0.2971	0.3959	0.1374
	DeepCT	<u>0.0626</u>	0.1486	0.0997	0.0760	0.1898	0.2531	0.5869	0.7622	0.4708	0.3718	0.4387	0.1913
Dense	DPR	0.0552	0.1765	0.1104	0.0826	0.2079	0.2769	0.2010	0.3413	0.1666	0.3153	0.4820	0.1582
	Sen-T5	0.0458	0.1363	0.0845	0.0673	0.1622	0.2227	0.5690	0.7967	0.4376	0.2360	0.3802	0.0956
	MPNet	0.0294	0.0879	0.0531	0.0619	0.1279	0.1970	0.5961	0.7755	0.4719	0.1820	0.3210	0.0815
Personalized	TEM	0.0213	0.0545	0.0402	0.0852	0.2131	0.2776	0.4814	0.7301	0.3535	0.3551	0.5401	0.1716
	CoPPS	0.0246	0.0878	0.0458	0.0877	0.2153	<u>0.2832</u>	0.4854	0.8004	0.3699	0.4265	0.5971	0.2403
Generative	DSI	0.0217	0.0552	0.0433	0.0623	0.1369	0.2018	0.4181	0.6202	0.3414	0.1868	0.3056	0.0836
	NCI	0.0432	0.1006	0.0803	0.0591	0.1270	0.1781	0.3765	0.5836	0.3139	0.2540	0.3951	0.1014
	LTRGR	0.0316	0.0944	0.0629	0.0688	0.1501	0.2184	0.5143	0.7025	0.4062	0.2608	0.4644	0.1120
	TIGER	0.0487	0.1335	0.0814	0.0731	0.1796	0.2427	0.5387	0.7292	0.4237	0.2659	0.4068	0.1073
	MERGE	0.0612	0.1903	0.1070	0.0858	0.2033	0.2625	0.5697	0.7819	0.4523	0.3850	0.5828	0.2389
	COBRA	0.0585	<u>0.1950</u>	<u>0.1126</u>	0.0890	<u>0.2074</u>	0.2803	<u>0.6205</u>	<u>0.8474</u>	<u>0.4906</u>	<u>0.5020</u>	<u>0.6057</u>	<u>0.2810</u>
	CHAP (Ours)	0.1127	0.2452	0.2454	0.0944	0.2289	0.2949	0.7358	0.9416	0.5191	0.5803	0.8085	0.3302

Table 3: Ablation study of CHAP on Local-Life.

Variant	R@10	R@50	MRR@10	QPS
CHAP (Full Model)	0.5803	0.8089	0.3302	78.9
(A) Hierarchical Semantic Alignment (HSA)				
w/o HSA	0.5115	0.7488	0.2845	78.9
w/o $\mathcal{L}_{CSA}$	0.5164	0.7602	0.2721	78.9
w/o $\mathcal{L}_{HCL}$	0.5218	0.7534	0.2858	78.9
w/o $\mathcal{L}_{Distill}$	0.5692	0.7895	0.3063	78.9
(B) Sequence Modeling				
w/o Dense Refinement	0.4352	0.6455	0.2241	81.2
(C) Residual Cascading Generation				
w/o Residual Gen. (Autoregressive)	0.5285	0.7610	0.2849	40.5
w/o Self-Attention (Cross-Attention)	0.5712	0.8062	0.3237	84.8
w/o Decoder (Pooling + MLP)	0.4651	0.6836	0.2584	112.4

we compare against **TEM** (Bi et al., 2020) and **CoPPS** (Dai et al., 2023). Finally, for Generative Retrieval (GR), we evaluate against state-of-the-art models including **DSI** (Tay et al., 2022), **NCI** (Wang et al., 2022), **TIGER** (Rajput et al., 2023), **LTRGR** (Li et al., 2024), **MERGE** (Zhang et al., 2025b), and **COBRA** (Yang et al., 2025). Detailed descriptions are provided in App. D.

5.1.3 Evaluation Metrics

We evaluate retrieval performance using four standard metrics: **Recall** (R@K) for the overall macroscopic coverage of relevant candidates; **Hit Ratio** (HR@K) for the success rate of retrieving at least one target in sparse or single-interaction scenarios; **Mean Reciprocal Rank** (MRR@K) for assessing strict positional ranking quality in single-target datasets; and **Normalized Discounted Cumulative Gain** (NDCG@K) for handling fine-grained permutations in multi-target datasets. Detailed mathematical formulations and selection justifications are provided in App. E.

5.1.4 Implementation Details

We utilize BERT-base (Devlin et al., 2019) for raw text representation and T5-base (Raffel et al., 2020) for the generative retrieval backbone (Chinese-BERT-base and mT5-base for Chinese datasets). In the SID learning stage, the RQ-VAE is configured with a codebook depth of $L = 3$ and a codebook size of $K = 512$. For the Hierarchical Semantic Alignment, the loss weights γ and δ are set to 0.01 and 0.001, respectively. During inference, we employ a parallel sampling strategy to efficiently generate $M = 50$ diverse candidate SIDs, and set the temperature $\tau = 1.1$ for the joint probability normalization operator $\mathcal{N}(\cdot)$. Comprehensive details and explicit implementation settings for all baseline models are provided in App. F.

5.2 Overall Results (RQ1)

Tab. 2 presents the comparative results across the four evaluated datasets. CHAP consistently establishes a new state-of-the-art, significantly outperforming strong baselines from all categories, including advanced GR models like COBRA and MERGE. Specifically, while standard GR methods like TIGER struggle with the inherent semantic gap—often lagging behind traditional BM25, our Hierarchical Semantic Alignment effectively bridges this gap, synergizing the semantic comprehension of dense retrievers with the structural precision of GR. Furthermore, traditional personalized baselines frequently suffer from relevance drift, allowing historical behaviors to overshadow current search intents. In contrast, CHAP dynamically refines query relevance through its dual-view sequence modeling. This prevents relevance drift

and ensures retrieved items are precisely aligned with both the explicit query constraints and the user’s fine-grained intents, yielding robust generalization across diverse, real-world search scenarios.

Notably, unlike COBRA (Yang et al., 2025), which mandates the generation of multiple dense representations—incurring excessive training costs and prohibitive real-time inference overhead, CHAP achieves superior efficiency by elegantly reusing the item’s raw representations for sequence modeling. Furthermore, MERGE (Zhang et al., 2025b) heavily relies on aligning multiple positive samples and varying relevance levels under a single query, exhibiting limited generalization on datasets with single positive targets or uniform relevance levels. In contrast, our semantic alignment paradigm is specifically designed to bridge the gap between complex, multi-intent queries and targets.

5.3 Ablation Study

Tab. 3 details our ablation study on Local-Life, incorporating Queries Per Second (QPS) to concurrently assess industrial efficiency. Detailed variant configurations are provided in App. G.

5.3.1 Analysis of Generated SIDs (RQ2)

Replacing our aligned encoder with a fixed pre-trained encoder (w/o HSA) causes significant degradation, confirming the critical necessity of proactive alignment in bridging the semantic gap. Furthermore, we observe a performance drop when removing the continuous vectors (w/o Dense Refinement). This validates our dual-view design intuition: sparse SIDs act strictly as structural skeletons for coarse retrieval, while dense vectors are indispensable for carrying the fine-grained semantics required for precise personalized matching.

Furthermore, visualization of the latent space encoded by DNN encoder of RQ-VAE after PCA dimensionality reduction (from ESCI-us, shown in Fig. 3) intuitively confirms this advantage: while vanilla RQ-VAE suffers from a visible semantic gap with loosely scattered relevant items, CHAP tightly clusters relevant items (orange) directly around their corresponding queries (red), as highlighted by the dashed boxes, while explicitly pushing irrelevant items (blue) further away. This demonstrates the efficacy of our alignment paradigm in drawing dynamic user intents and static item semantics into a highly unified space. Additional qualitative diagnostics are provided in App. I.

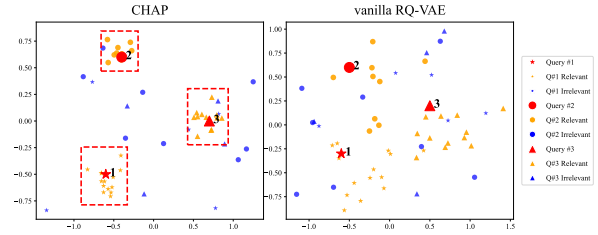


Figure 3: Visualization of Items under three queries.

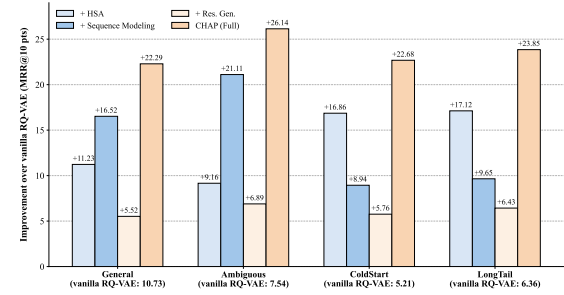


Figure 4: Intent-wise query segment analysis.

5.3.2 Intent-wise Query Analysis (RQ3)

To further dissect the source of CHAP’s superiority, we extract four intent-wise query segments from the Local-Life dataset: General, Ambiguous, ColdStart, and LongTail (detailed in App. H). We explicitly evaluate the absolute MRR@10 improvements brought by individually integrating each core component over the vanilla RQ-VAE baseline (Fig. 4).

The segment-wise results corroborate our design intuitions, revealing the orthogonal strengths of each module. Introducing the dual-view Sequence Modeling delivers the most dominant boost on the **Ambiguous** segment. This confirms that for highly generic queries, dense historical contextualization is indispensable for intent disambiguation. However, its isolated contribution drops on the **ColdStart** and **LongTail** segments, as cold entities inherently lack sufficient interaction histories for sequence models to exploit. Conversely, integrating the HSA module addresses these sparse segments, suggesting that proactive semantic mapping bridges the representation gap for rare queries and new items, independent of popularity biases. Meanwhile, the Residual Generation mechanism yields a stable, structural baseline uplift across all slices by mitigating layer-wise decoding decay.

5.3.3 Efficiency Analysis (RQ4)

We deeply deconstruct our decoding architecture to evaluate its necessity and efficiency. Degenerating to a standard autoregressive paradigm (w/o Residual Gen.) harms accuracy and halves infer-

Table 4: Efficiency comparison on Local-Life. ItemID and Seq. denote the training time for ItemID construction and the GR model, respectively. Total is their sum.

Model	ItemID (H)	Seq. (H)	Total (H)	QPS
DSI	1.3	16.5	17.8	37.5
NCI	1.3	34.0	35.3	26.2
LTRGR	1.5	31.9	33.4	34.7
TIGER	0.4	18.2	18.6	39.2
MERGE	0.8	19.3	20.1	36.8
COBRA	0.4	26.4	26.8	32.4
CHAP (Ours)	0.5	23.8	24.3	78.9

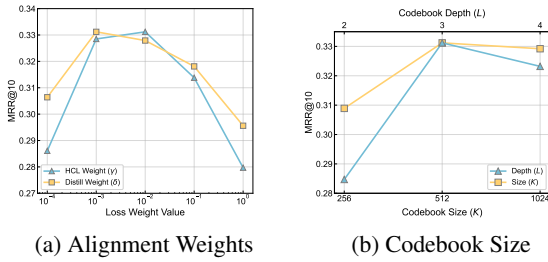


Figure 5: Effect of hyperparameters on retrieval tasks.

ence throughput (QPS), indicating our residual design is a powerful decoding accelerator. Removing the decoder’s self-attention (w/o Self-Attention) improves QPS with minimal precision loss, showing explicit residual pathways effectively replace self-attention for hierarchical modeling. However, entirely removing the decoder (w/o Decoder) maximizes QPS but collapses MRR. This highlights the indispensable role of cross-attention for dynamic history routing across lengthy user sequences.

To further contextualize the efficiency (Tab. 4), we evaluated the computational overhead using an $8 \times \text{A100}$ (80G) GPU cluster for training and a single A100 (80G) GPU for inference QPS. During training, advanced baselines like NCI and LTRGR incur excessive computational overhead due to massive query augmentation or complex list-wise optimization. In contrast, CHAP maintains a highly competitive duration, presenting a cost-effective trade-off for modeling dual-view sequences. During online inference, while standard GR models and dense-heavy frameworks (e.g., COBRA) are severely bottlenecked by repeated layer-wise decoding, our decoupled architecture, combined with parallel sampling, achieves substantially higher QPS, satisfying the strict low-latency requirements of large-scale industrial serving.

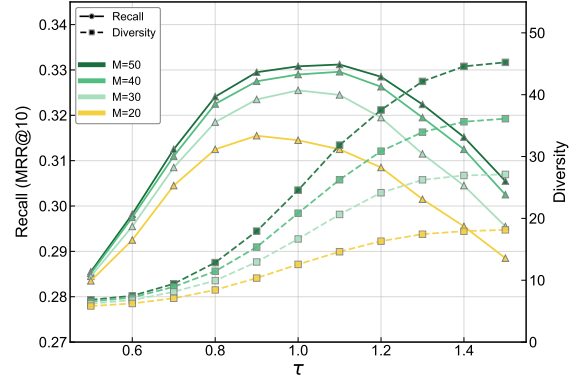


Figure 6: Impact of normalization temperature τ and candidate set width M on accuracy and diversity.

Table 5: Online A/B testing results over a 14-day period. All metrics are statistically significant with paired t-tests at $p < 0.05$ level.

Metric	Relative Improvement
UV-CTR (Click-Through Rate)	+0.77%
UV-CXR (Conversion Rate)	+2.09%
Pay Orders (Order Volume)	+2.98%

5.4 Hyperparameter Analysis (RQ5)

We analyze the sensitivity of critical hyperparameters on the Local-Life dataset to evaluate CHAP’s robustness. Regarding training objectives, the alignment weights exhibit an inverted U-shape trend, peaking at $\gamma = 0.01$ and $\delta = 0.001$ to achieve optimal semantic alignment without disrupting the primary reconstruction stability. For the codebook scale, performance peaks at depth $L = 3$ and size $K = 512$. Notably, CHAP demonstrates exceptional robustness to these structural parameters; while extreme settings induce minor metric fluctuations, our framework avoids the performance collapse typically observed in standard GR models under suboptimal codebook assignments. During inference, we regulate the accuracy-diversity equilibrium by varying the normalization temperature τ across parallel sampling sizes M . Ranking accuracy follows an inverted U-shape, suffering from over-confident sharp distributions at low τ and excessive exploration noise at high τ . Ultimately, CHAP achieves an optimal balance between semantic discrimination and candidate diversity at $\tau = 1.1$ with $M = 50$. Detailed analysis is provided in Fig. 5 and Fig. 6.

5.5 Online A/B Test (RQ6)

To verify CHAP’s practical effectiveness and scalability, we conducted a rigorous 14-day online A/B test on a leading production local-life service plat-

form, covering 20% of user traffic. CHAP was deployed via NVIDIA Triton to decouple modules for high-throughput real-time inference, rather than relying on query caching. Compared to the highly optimized online production retrieval system, CHAP achieved consistent improvements: **UV-CTR** increased by 0.77%, demonstrating that HSA enhances semantic relevance. Moreover, conversion metrics saw substantial lifts, with **UV-CXR** rising by 2.09% and **Pay Order Volume** growing by 2.98%, showing that CHAP not only attracts user clicks but, crucially, accurately captures latent user intent to drive final purchasing decisions, yielding tangible business growth in industrial scenarios.

6 Conclusion

In this work, we propose CHAP, a novel personalized GR framework designed to overcome the inherent semantic gap and inference latency bottlenecks of existing GR paradigms. By introducing the HSA module, CHAP effectively maps dynamic user intents directly into the static item quantization space. Furthermore, our dual-view sequence modeling, coupled with a single-pass Residual Cascading Generation mechanism, decouples historical intent routing from layer-wise decoding. This architecture not only ensures fine-grained semantic precision but also acts as a powerful decoding accelerator. Extensive experiments across four diverse real-world datasets, alongside a rigorous online A/B test on a leading local-life platform, demonstrate that CHAP establishes a new state-of-the-art. It achieves a balance between exact intent alignment and industrial-grade inference throughput.

Limitations

Although CHAP demonstrates strong empirical effectiveness and industrial efficiency, several limitations remain for future exploration. First, in the HSA stage, freezing the item-side codebook provides a stable quantization manifold and prevents semantic drift. However, this design may also prevent item representations from being dynamically updated with collaborative filtering signals or query-side feedback, which could limit the upper bound of personalization in highly dynamic environments. Second, our dual-view personalized sequence modeling is mainly validated with hierarchical SIDs generated by RQ-VAE-style quantization. While this setting is representative for recent generative retrieval systems, the generalization ability

of the overall framework to other types of ItemIDs, such as lexical identifiers, learned atomic IDs, or tree-based semantic IDs, still requires systematic investigation. Finally, the Residual Cascading Generation module substantially improves inference efficiency by avoiding repeated heavy decoder passes. Nevertheless, how the acceleration ratio evolves when scaling model size, codebook depth, and candidate budget remains an open question. Further compression techniques, such as model distillation for the residual generation blocks, may provide an additional path toward more efficient large-scale deployment.

Ethical Considerations

This research was reviewed and approved by the relevant technical ethics committee. All data collection procedures followed the platform’s compliance requirements and were conducted with user permission or authorization where applicable. Before being used for model training, evaluation, and analysis, all user-related data were anonymized or de-identified to remove personally identifiable information. Our experiments rely only on aggregated statistics and anonymized behavioral signals, and do not attempt to infer or recover individual identities. The online A/B test was conducted under the platform’s standard safety and compliance protocols to minimize potential risks to user experience. The open-source implementation of CHAP is released under a CC BY-NC-SA 4.0 license, explicitly disclaiming liability for direct commercial deployment to ensure strict non-commercial, research-only usage.

Acknowledgements

The work is supported by grants from New Generation Artificial Intelligence-National Science and Technology Major Project (2025ZD0123403), National Natural Science Foundation of China (No. U24A20253), Scientific Research Innovation Capability Support Project for Young Faculty, and the Meituan Research Fund.

References

- Qingyao Ai, Daniel N Hill, SVN Vishwanathan, and W Bruce Croft. 2019a. A zero attention model for personalized product search. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 379–388.

- Qingyao Ai, Yongfeng Zhang, Keping Bi, Xu Chen, and W Bruce Croft. 2017. Learning a hierarchical embedding model for personalized product search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 645–654.
- Qingyao Ai, Yongfeng Zhang, Keping Bi, and W Bruce Croft. 2019b. Explainable product search with a dynamic relation embedding model. *ACM Transactions on Information Systems (TOIS)*, 38(1):1–29.
- Vladimir Baikalov, Iskander Bagautdinov, and Sergey Muravyov. 2026. Mitigating collaborative semantic id staleness in generative retrieval. *arXiv preprint arXiv:2604.13273*.
- Keping Bi, Qingyao Ai, and W Bruce Croft. 2020. A transformer-based embedding model for personalized product search. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1521–1524.
- Hongru Cai, Yongqi Li, Wenjie Wang, Fengbin Zhu, Xiaoyu Shen, Wenjie Li, and Tat-Seng Chua. 2025. Large language models empowered personalized web agents. In *Proceedings of the ACM on Web Conference 2025*, pages 198–215.
- Tadeusz Caliński and Jerzy Harabasz. 1974. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27.
- Shitong Dai, Jiongnan Liu, Zhicheng Dou, Haonan Wang, Lin Liu, Bo Long, and Ji-Rong Wen. 2023. Contrastive learning for user sequence representation in personalized product search. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 380–389.
- Zhuyun Dai and Jamie Callan. 2019. Context-aware sentence/passage term importance estimation for first stage retrieval. *arXiv preprint arXiv:1910.10687*.
- David L Davies and Donald W Bouldin. 1979. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227.
- 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.
- Yan Fang, Jingtao Zhan, Qingyao Ai, Jiaxin Mao, Weihang Su, Jia Chen, and Yiqun Liu. 2024. Scaling laws for dense retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1339–1349.
- Junchen Fu, Xuri Ge, Alexandros Karatzoglou, Ioannis Arapakis, Suzan Verberne, Joemon M Jose, and Zhaochun Ren. 2026. Differentiable semantic id for generative recommendation. *arXiv preprint arXiv:2601.19711*.
- Zitian Guo, Yupeng Hou, Clark Mingxuan Ju, Neil Shah, and Julian McAuley. 2026. Mlps are efficient distilled generative recommenders. *arXiv preprint arXiv:2605.12617*.
- Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517.
- Angqing Jiang, Gaoming Zhang, Jianchun Song, Kena Qi, Dayao Chen, Wei Lin, and Defu Lian. 2026a. Think-to-personalize: Unifying reasoning and retrieval for user-centric personalized dense retrieval. *arXiv preprint arXiv:2608.18855*.
- Jie Jiang, Xinxun Zhang, Enming Zhang, Yuling Xiong, Jun Zhang, Jingwen Wang, Huan Yu, Yuxiang Wang, Hao Wang, Xiao Yan, and 1 others. 2026b. End-to-end semantic id generation for generative advertisement recommendation. *arXiv preprint arXiv:2602.10445*.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781.
- Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.
- Tzu-Lin Kuo, Tzu-Wei Chiu, Tzung-Sheng Lin, Sheng-Yang Wu, Chao-Wei Huang, and Yun-Nung Chen. 2024. A survey of generative information retrieval. *arXiv preprint arXiv:2406.01197*.
- Xiaoxi Li, Jiajie Jin, Yujia Zhou, Yuyao Zhang, Peitian Zhang, Yutao Zhu, and Zhicheng Dou. 2025. From matching to generation: A survey on generative information retrieval. *ACM Transactions on Information Systems*, 43(3):1–62.
- Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. 2024. Learning to rank in generative retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8716–8723.
- Yupeng Li, Ben Chen, Mingyue Cheng, Zhiding Liu, Xuxin Zhang, Chenyi Lei, and Wenwu Ou. 2026. [Kuaishow: A large-scale e-commerce search dataset for recall, ranking, and relevance](#). *arXiv preprint arXiv:2602.11518*.

- Jingjing Liu, Chang Liu, and Nicholas J Belkin. 2020. Personalization in text information retrieval: A survey. *Journal of the Association for Information Science and Technology*, 71(3):349–369.
- Jiongnan Liu, Zhicheng Dou, Guoyu Tang, and Sulong Xu. 2023. Jdsearch: A personalized product search dataset with real queries and full interactions. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2945–2952.
- Xiaoyu Liu, Fuwei Zhang, Yiqing Wu, Xinyu Jia, Zenghua Xia, Fuzhen Zhuang, Zhao Zhang, Fei Jiang, and Wei Lin. 2025. Cat-id²: Category-tree integrated document identifier learning for generative retrieval in e-commerce. *arXiv preprint arXiv:2511.01461*.
- Yu A Malkov and Dmitry A Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):824–836.
- Fengran Mo, Abbas Ghaddar, Kelong Mao, Mehdi Rezagholizadeh, Boxing Chen, Qun Liu, and Jian-Yun Nie. 2024. Chiq: Contextual history enhancement for improving query rewriting in conversational search. *arXiv preprint arXiv:2406.05013*.
- Thong Nguyen and Andrew Yates. 2023. Generative retrieval as dense retrieval. *arXiv preprint arXiv:2306.11397*.
- Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the association for computational linguistics: ACL 2022*, pages 1864–1874.
- Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375*.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.
- Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, and 1 others. 2023. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems*, 36:10299–10315.
- Juan Ramos and 1 others. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. New Jersey, USA.
- Chandan K. Reddy, Lluís Màrquez, Fran Valero, Nikhil Rao, Hugo Zaragoza, Sambaran Bandyopadhyay, Arnab Biswas, Anlu Xing, and Karthik Subbian. 2022. *Shopping queries dataset: A large-scale ESCI benchmark for improving product search*.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.
- Stephen Robertson, Hugo Zaragoza, and 1 others. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.
- Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.
- Zihua Si, Zhongxiang Sun, Jiale Chen, Guozhang Chen, Xiaoxue Zang, Kai Zheng, Yang Song, Xiao Zhang, Jun Xu, and Kun Gai. 2024. Generative retrieval with semantic tree-structured identifiers and contrastive learning. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pages 154–163.
- Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33:16857–16867.
- Zhongxiang Sun, Zihua Si, Xiaoxue Zang, Dewei Leng, Yanan Niu, Yang Song, Xiao Zhang, and Jun Xu. 2023. Kuaisar: A unified search and recommendation dataset. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pages 5407–5411.
- Yubao Tang, Ruqing Zhang, Jiafeng Guo, Jiangui Chen, Zuowei Zhu, Shuaiqiang Wang, Dawei Yin, and Xueqi Cheng. 2023a. Semantic-enhanced differentiable search index inspired by learning strategies. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4904–4913.
- Yubao Tang, Ruqing Zhang, Jiafeng Guo, and Maarten de Rijke. 2023b. Recent advances in generative information retrieval. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pages 294–297.
- Yi Tay, Vinh Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, and 1 others. 2022. Transformer memory as a differentiable search index. *Advances in Neural Information Processing Systems*, 35:21831–21843.
- Lidan Wang, Jimmy Lin, and Donald Metzler. 2011. A cascade ranking model for efficient ranked retrieval. In *Proceedings of the 34th international ACM SIGIR*

- conference on Research and development in Information Retrieval, pages 105–114.
- Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao, Shibin Wu, Qi Chen, Yuqing Xia, Chengmin Chi, Guoshuai Zhao, Zheng Liu, and 1 others. 2022. A neural corpus indexer for document retrieval. *Advances in Neural Information Processing Systems*, 35:25600–25614.
- Will Williams, Sam Ringer, Tom Ash, David MacLeod, Jamie Dougherty, and John Hughes. 2020. Hierarchical quantized autoencoders. *Advances in Neural Information Processing Systems*, 33:4524–4535.
- Yanjiang Wu, Yinfu Feng, Jian Wang, Wenji Zhou, Yunnan Ye, Rong Xiao, and Jun Xiao. 2024. Hi-gen: Generative retrieval for large-scale personalized e-commerce search. *arXiv preprint arXiv:2404.15675*.
- Ming Xia, Zhiqin Zhou, Guoxin Ma, and Dongmin Huang. 2026. Unleash the potential of long semantic ids for generative recommendation. *arXiv preprint arXiv:2602.13573*.
- Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.
- Yuhao Yang, Zhi Ji, Zhaopeng Li, Yi Li, Zhonglin Mo, Yue Ding, Kai Chen, Zijian Zhang, Jie Li, Shuanglong Li, and 1 others. 2025. Sparse meets dense: Unified generative recommendations with cascaded sparse-dense representations. *arXiv preprint arXiv:2503.02453*.
- Peiwen Yuan, Xinglin Wang, Shaoxiong Feng, Boyuan Pan, Yiwei Li, Heda Wang, Xupeng Miao, and Kan Li. 2024. Generative dense retrieval: Memory can be a burden. *arXiv preprint arXiv:2401.10487*.
- Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 1503–1512.
- Fuwei Zhang, Xiaoyu Liu, Xinyu Jia, Yingfei Zhang, Zenghua Xia, Fei Jiang, Fuzhen Zhuang, Wei Lin, and Zhao Zhang. 2025a. Hiergr: Hierarchical semantic representation enhancement for generative retrieval in food delivery search. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 444–455.
- Fuwei Zhang, Xiaoyu Liu, Xinyu Jia, Yingfei Zhang, Shuai Zhang, Xiang Li, Fuzhen Zhuang, Wei Lin, and Zhao Zhang. 2025b. Multi-level relevance document identifier learning for generative retrieval. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10066–10080.
- Yingchen Zhang, Ruqing Zhang, Jiafeng Guo, Wenjun Peng, Sen Li, Fuyu Lv, and Xueqi Cheng. 2025c. C2t-id: Converting semantic codebooks to textual document identifiers for generative search. In *Proceedings of the 2025 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pages 331–336.
- Carolina Zheng, Minhui Huang, Dmitrii Pedchenko, Kaushik Rangadurai, Siyu Wang, Fan Xia, Gaby Nahum, Jie Lei, Yang Yang, Tao Liu, and 1 others. 2025. Enhancing embedding representation stability in recommendation systems with semantic id. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, pages 954–957.
- Fan Zhou, Xovee Xu, Goce Trajcevski, and Kunpeng Zhang. 2021. A survey of information cascade analysis: Models, predictions, and recent advances. *ACM Computing Surveys (CSUR)*, 54(2):1–36.
- Yujia Zhou, Zhicheng Dou, and Ji-Rong Wen. 2023. Enhancing generative retrieval with reinforcement learning from relevance feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12481–12490.
- Jianbo Zhu, Xing Fang, Jing Wang, Mingmin Jin, Bokang Wang, Guangxin Song, Zhenyu Xie, and Junjie Bai. 2026. Efficient generative retrieval for e-commerce search with semantic cluster ids and expert-guided rl. *arXiv preprint arXiv:2605.14434*.

A Related Works

Sparse retrieval methods like BM25 (Robertson et al., 2009) and TF-IDF (Ramos et al., 2003) rely on exact term matching but suffer from vocabulary mismatch. Conversely, **dense retrieval** approaches (e.g., DPR (Karpukhin et al., 2020), ANCE (Xiong et al., 2020)) map inputs into a shared vector space, enhancing semantic matching but necessitating costly external indices like HNSW (Malkov and Yashunin, 2018). To address these limitations, STAR (Zhan et al., 2021) introduces stable negative sampling, while ColBERT (Khattab and Zaharia, 2020) and Sentence-BERT (Reimers and Gurevych, 2019) refine fine-grained similarity and semantic understanding. Recent work also investigates scaling laws in dense models (Fang et al., 2024).

Personalized retrieval tailors results by incorporating user contexts. QEM (Ai et al., 2019a) investigates personalization mechanisms, while DREM (Ai et al., 2019b) models dynamic user-product relationships via knowledge graphs. Embedding-based approaches like ZAM (Ai et al., 2019a), HEM (Ai et al., 2017), and TEM (Bi et al., 2020) integrate user profiles and history into unified embeddings. In conversational domains, CHIQ (Mo et al., 2024) enhances accuracy by leveraging historical search queries. TTP (Jiang et al., 2026a) introduces a unified reasoning and retrieval framework for personalized dense retrieval.

Generative retrieval internalizes corpus knowledge to directly generate item identifiers. DSI (Tay et al., 2022) pioneered this paradigm, followed by semantic enhancements in SE-DSI (Tang et al., 2023a). Various optimization strategies have emerged: SEATER (Si et al., 2024) utilizes balanced k-ary trees; Tied-Atomic (Nguyen and Yates, 2023) connects GR with dense retrieval; GenRRL (Zhou et al., 2023) employs reinforcement learning; and LTRGR (Li et al., 2024) incorporates ranking tasks. TIGER (Rajput et al., 2023) first utilized RQ-VAE for SIDs. Regarding architecture, COBRA (Yang et al., 2025) integrates sparse and dense vectors, while Hi-gen (Wu et al., 2024), CAT-ID² (Liu et al., 2025), and MERGE (Zhang et al., 2025b) optimize ID generation by exploiting multi-level correlations.

Recent and concurrent studies have further revisited Semantic ID based generative retrieval from several complementary perspectives. A line of work improves the construction of Semantic IDs beyond item-only reconstruction. For example,

CQ-SID introduces category-aware and query-item contrastive objectives for e-commerce generative retrieval (Zhu et al., 2026), while DIGER (Fu et al., 2026) and UniSID (Jiang et al., 2026b) explore differentiable or end-to-end Semantic ID learning to better align identifier assignment with downstream recommendation objectives. Another direction studies the dynamics and scalability of Semantic IDs, such as mitigating collaborative Semantic ID staleness under evolving user-item interactions (Baikalov et al., 2026) and exploiting longer Semantic IDs to enhance representation capacity while controlling decoding cost (Xia et al., 2026). Closely related to our efficiency motivation, MLP-based distilled generative recommenders show that attention-heavy autoregressive decoders can be distilled into lightweight MLP-based heads for faster Semantic ID generation (Guo et al., 2026), suggesting that much of the decoder-side computation is redundant once global user context and prefix structure are properly preserved. These studies are largely orthogonal to CHAP: they mainly focus on identifier learning, temporal ID maintenance, or decoder distillation, whereas CHAP targets personalized generative retrieval by explicitly aligning query representations with the item quantization path, jointly modeling sparse Semantic IDs and dense representations, and using residual cascading generation to reduce autoregressive latency while preserving fine-grained personalized relevance.

B Proof of Cascaded Sparse-Dense Modeling

Theorem 1 (Entropy reduction of CHAP’s cascaded modeling). Let X denote the personalized context encoded from X_{in} , and let $\mathbf{C}_i = (C_i^0, \dots, C_i^{L-1})$ and $\mathbf{V}_i$ denote the random variables of the target item’s hierarchical SID and dense vector, respectively. Consider an independent dual-view factorization

$$P_{\text{ind}}(\mathbf{C}_i, \mathbf{V}_i | X) = \prod_{l=0}^{L-1} P(C_i^l | X) \cdot P(\mathbf{V}_i | X), \quad (21)$$

and CHAP’s cascaded sparse-dense factorization

$$P_{\text{chap}}(\mathbf{C}_i, \mathbf{V}_i | X) = \prod_{l=0}^{L-1} P(C_i^l | C_i^{<l}, X) \cdot P(\mathbf{V}_i | \mathbf{C}_i, X). \quad (22)$$

Denote their total uncertainties by $\mathcal{H}_{\text{ind}}$ and $\mathcal{H}_{\text{chap}}$, where $H(\cdot)$ is used for discrete SIDs and $h(\cdot)$ for the continuous dense vector. Then

$$\mathcal{H}_{\text{chap}}(\mathbf{C}_i, \mathbf{V}_i | X) \leq \mathcal{H}_{\text{ind}}(\mathbf{C}_i, \mathbf{V}_i | X). \quad (23)$$

Equality holds if and only if each SID level is conditionally independent of its prefix given X , and $\mathbf{V}_i$ is conditionally independent of $\mathbf{C}_i$ given X .

Proof. Writing P_{ind} and P_{chap} for the two distributions above, the independent dual-view entropy decomposes as

$$\begin{aligned} \mathcal{H}_{\text{ind}} &= -\mathbb{E}_{P_{\text{ind}}}[\log P_{\text{ind}}] \\ &= \sum_{l=0}^{L-1} H(C_i^l | X) \\ &\quad + h(\mathbf{V}_i | X). \end{aligned} \quad (24)$$

For CHAP’s cascaded factorization, we have

$$\begin{aligned} \mathcal{H}_{\text{chap}} &= -\mathbb{E}_{P_{\text{chap}}}[\log P_{\text{chap}}] \\ &= \sum_{l=0}^{L-1} H(C_i^l | C_i^{<l}, X) \\ &\quad + h(\mathbf{V}_i | \mathbf{C}_i, X). \end{aligned} \quad (25)$$

By the non-negativity of conditional mutual information,

$$H(C_i^l | X) - H(C_i^l | C_i^{<l}, X) = I(C_i^l; C_i^{<l} | X) \geq 0, \quad (26)$$

and

$$h(\mathbf{V}_i | X) - h(\mathbf{V}_i | \mathbf{C}_i, X) = I(\mathbf{V}_i; \mathbf{C}_i | X) \geq 0. \quad (27)$$

Substituting these inequalities into the two entropy decompositions yields

$$\begin{aligned} \mathcal{H}_{\text{ind}} - \mathcal{H}_{\text{chap}} &= \sum_{l=0}^{L-1} I(C_i^l; C_i^{<l} | X) \\ &\quad + I(\mathbf{V}_i; \mathbf{C}_i | X) \geq 0. \end{aligned} \quad (28)$$

Therefore, CHAP’s coarse-to-fine SID generation followed by dense refinement does not increase the modeling uncertainty compared with independently predicting sparse and dense views. The equality condition follows directly from the equality condition of conditional mutual information: all terms above are zero only when $C_i^l \perp C_i^{<l} | X$ for every level l and $\mathbf{V}_i \perp \mathbf{C}_i | X$. This completes the proof.

C Dataset Details

In this section, we provide detailed descriptions of the four datasets used to evaluate the proposed CHAP framework. The detailed statistics are summarized in Tab. 6.

ESCI-us. We utilize the large version of the English subset from the ESCI¹ (Reddy et al., 2022) dataset. Unlike the widely used Amazon Product Reviews (He and McAuley, 2016) dataset, which necessitates constructing synthetic queries, ESCI contains authentic queries derived from actual user logs on Amazon, filled with the noise and complexity of real-world search traffic. Each query in ESCI is associated with products labeled by relevance levels: Exact (E), Substitute (S), Complement (C), and Irrelevant (I). This dataset is characterized by complex linguistic patterns, including negation conditions (e.g., “not” and “without”) and intricate attribute constraints (e.g., dimensions, price, functionality). These factors result in a low text overlap between queries and product descriptions, making it a rigorous benchmark for evaluating a model’s semantic understanding capabilities. During sequence modeling, we treat items labeled “E” and “S” as positive samples. Notably, unlike existing works (Liu et al., 2025; Zhang et al., 2025b) that utilized the small version of the dataset and applied specific cleaning and preprocessing steps, we directly adopt the large version with raw query and item texts. We strictly follow the official pre-defined train and test splits without any additional preprocessing to rigorously test the model’s robustness and generalization capabilities.

KuaiSearch. KuaiSearch² (Li et al., 2026) is a large-scale e-commerce search dataset built upon real user search interactions from the Kuaishou platform. In contrast to existing datasets that often rely on heuristically constructed queries or aggressively filter out inactive entities, KuaiSearch preserves authentic user queries and natural-language product texts without popularity-based filtering. This ensures the retention of cold-start users and long-tail products, accurately reflecting real-world e-commerce long-tail distributions. The whole dataset contains approximately 330,000 users, 18,000,000 products, and 2,500,000 real search queries. Furthermore, it provides extensive long-term user behavior sequences and systematically covers the core stages of industrial search pipelines.

¹<https://github.com/amazon-science/esci-data>

²<https://github.com/benchen4395/KuaiSearch>

Table 6: Detailed statistics of the four datasets.

Dataset	#Items	#Queries	#Q-I Pairs	Train		Test		Avg. Hist.
				#Queries	#Q-I Pairs	#Queries	#Q-I Pairs	
ESCI-us	1,215,854	97,346	1,818,825	74,888	1,393,063	22,458	425,762	-
KuaiSearch	6,634,118	295,561	682,228	266,004	613,696	29,557	68,532	10.37
Amazon	35,772	9,070	9,070	6,896	6,896	2,174	2,174	40.12
Local-Life	10,145,542	3,089,026	3,089,026	2,984,294	2,984,294	104,732	104,732	7.47

In our experiments, we specifically utilize the *recall* subset. During data construction, we treat both clicked and purchased items as positive samples. Strictly following the methodology described in the original paper, we apply identical data cleaning steps—filtering out entries that lacked explicit search queries or had no positive samples—and subsequently partition the cleaned dataset into training and testing sets using a random 9:1 split.

Amazon. To evaluate our model’s capacity for fine-grained personalized retrieval, we utilize the Amazon search dataset from the Personalized Web Agent Benchmark (PersonalWAB)³ (Cai et al., 2025). Unlike traditional static search benchmarks, PersonalWAB is explicitly designed to assess the integration of user profiles and historical web behaviors in understanding customized instructions. The dataset originates from real-world Amazon interaction logs and encompasses diverse user profiles alongside tens of thousands of historical web behaviors. It features highly personalized, human-synthesized natural language instructions tailored to each user’s profile, requiring models to dynamically infer specific user preferences. This dataset presents a unique challenge for generative retrieval models, as it necessitates a deep semantic fusion of the current query, user profiles, and heterogeneous historical behaviors to achieve accurate personalized product search. To fully preserve the complexity and noise of these customized instructions, we utilized this dataset in its original form and directly adopted the officially pre-defined train and test splits without performing any data cleaning or preprocessing.

Local-Life. To assess performance in a dynamic, personalized environment, we introduce a large-scale industrial dataset collected from 30 days of user interaction logs on a leading Local-Life service platform. Compared to public datasets like KuaiSAR (Sun et al., 2023) or JDSearch (Liu et al., 2023), which only provide encrypted token IDs

preventing semantic verification, this dataset comprises raw natural language text. This transparency is crucial for verifying our model’s ability to align semantics between queries and items. The dataset covers diverse scenarios (e.g., Dining, Entertainment, Travel) and heterogeneous search intents (e.g., Brand Search, Ambiguous Intent, Exact Item Search, and Long-tail or Cold-start Search), comprehensively reflecting real-world user behaviors. We construct interaction sequences from the first 29 days for training and the last day for testing.

D Baseline Descriptions

To rigorously evaluate the proposed CHAP framework, we comprehensively compare it against 14 representative baselines categorized into sparse retrieval, dense retrieval, personalized retrieval, and generative retrieval (GR) methods.

Sparse Retrieval Methods.

- **BM25** (Robertson et al., 2009): A classical probabilistic information retrieval model that relies on exact lexical matching based on term frequency and inverse document frequency.
- **Doc2Query** (Nogueira et al., 2019): A document expansion technique that utilizes a sequence-to-sequence model to predict potential queries for each document prior to indexing, thereby enriching the vocabulary and mitigating the vocabulary mismatch problem.
- **DeepCT** (Dai and Callan, 2019): A deep contextualized term weighting framework that leverages BERT-based text representations to estimate the document-specific semantic importance of terms, translating them into enhanced term weights for first-stage sparse retrieval.

Dense Retrieval Methods.

- **DPR** (Karpukhin et al., 2020): Dense Passage Retriever, a highly effective bi-encoder

³<https://github.com/HongruCai/PersonalWAB>

architecture optimized via in-batch and hard negative contrastive learning to map queries and items into a shared semantic vector space.

- **Sen-T5** (Ni et al., 2022): A scalable sentence encoding model built upon the T5 framework, pre-trained on massive text-to-text and contrastive tasks to generate high-quality, continuous text representations.
- **MPNet** (Song et al., 2020): A robust unified pre-training framework that successfully combines the advantages of masked language modeling (e.g., BERT) and permuted language modeling (e.g., XLNet) for enhanced document embeddings.

Personalized Retrieval Methods.

- **TEM** (Bi et al., 2020): Transformer-based Embedding Model, which explicitly encodes user search history alongside current queries to construct dynamic user profiles, aligning retrieval representations with personalized intents.
- **CoPPS** (Dai et al., 2023): Contrastive learning for Personalized Product Search, a framework that discriminates fine-grained user intents by applying multi-view contrastive learning over user behavior sequence representations.

Generative Retrieval (GR) Methods.

- **DSI** (Tay et al., 2022): A pioneering end-to-end differentiable search index that directly generates relevant document identifiers using a Transformer memory based on semantic clustering.
- **NCI** (Wang et al., 2022): Neural Corpus Indexer, a strong GR baseline that utilizes a prefix-aware decoder and a query generation strategy to optimize hierarchical semantic identifiers effectively.
- **TIGER** (Rajput et al., 2023): A generative recommender and retrieval model that leverages a Residual Quantized Variational AutoEncoder (RQ-VAE) to generate hierarchical, semantically rich discrete IDs strictly based on item textual features.

- **LTRGR** (Li et al., 2024): Learning-to-Rank in Generative Retrieval, which integrates ranking-aware principles into the generation process by optimizing the log-probabilities of generation according to target ranking metrics like ListMLE.
- **MERGE** (Zhang et al., 2025b): A recent generative retrieval framework that aims to optimize identifier assignments by aligning multiple positive target items across varying relevance signals for multi-level ranking.
- **COBRA** (Yang et al., 2025): A state-of-the-art cascading generative framework that unites sparse identifier generation and dense representation regression within a joint optimization landscape to improve retrieval coverage and accuracy.

E Evaluation Metrics Details

To comprehensively evaluate the retrieval accuracy and ranking quality of the proposed CHAP framework, we employ four widely adopted metrics: Recall, Hit Ratio (HR), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain (NDCG). Here, we detail their formulations and the specific rationales for selecting them based on the characteristics of different datasets.

Recall (R@K). Recall evaluates the model’s ability to retrieve all relevant items. It provides a macroscopic view of coverage, which is essential for measuring overall semantic alignment.

$$R@K = \frac{1}{|Q|} \sum_{q \in Q} \frac{|\mathcal{R}_q \cap \hat{\mathcal{I}}_{q,K}|}{|\mathcal{R}_q|}, \quad (29)$$

where Q is the set of queries, $\mathcal{R}_q$ denotes the ground-truth relevant items for query q , and $\hat{\mathcal{I}}_{q,K}$ represents the top- K retrieved candidates.

Hit Ratio (HR@K). Hit Ratio measures the probability that a user’s target item successfully appears in the top- K retrieved list. It is formulated as an indicator function:

$$HR@K = \frac{1}{|Q|} \sum_{q \in Q} \mathbb{I}(|\mathcal{R}_q \cap \hat{\mathcal{I}}_{q,K}| > 0), \quad (30)$$

where $\mathbb{I}(\cdot)$ evaluates to 1 if the condition is true and 0 otherwise. *Rationale:* In practical industrial scenarios (e.g., e-commerce purchases or local-life orders), user intent is often highly focused on a

single specific item. HR effectively reflects the success rate of fulfilling this distinct user intent, making it highly suitable for evaluating performance on datasets like Local-Life and KuaiSearch.

Mean Reciprocal Rank (MRR@K). MRR strictly assesses the ranking quality by focusing on the position of the *first* relevant item retrieved:

$$\text{MRR@K} = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{\text{rank}_q^*}, \quad (31)$$

where rank_q^* is the rank position of the first relevant item in the top- K list (if no relevant item is in the top K , the score is 0). *Rationale:* For single-target or exact-intent datasets (such as PersonalWAB and Local-Life), users typically stop browsing once they find the exact match. MRR rigorously penalizes models that place the target item lower down the list, directly correlating with user satisfaction and online CTR.

Normalized Discounted Cumulative Gain (NDCG@K). NDCG evaluates the ranking quality by considering both the presence and the position of multiple relevant items, factoring in multi-level relevance scores:

$$\text{DCG@K} = \sum_{i=1}^K \frac{2^{\text{rel}_i} - 1}{\log_2(i + 1)}, \quad (32)$$

$$\text{NDCG@K} = \frac{\text{DCG@K}}{\text{IDCG@K}}, \quad (33)$$

where rel_i is the graded relevance score of the item at rank i , and IDCG@K is the ideal DCG obtained by perfectly sorting all relevant items. *Rationale:* For datasets like ESCI-us, a single query often maps to multiple items with varying relevance levels (e.g., Exact, Substitute, Complement). NDCG is uniquely capable of capturing these fine-grained grading differences, rewarding models that place highly relevant items at the top while tolerating lower-ranked partially relevant items.

F Implementation Details for CHAP and Baselines

To ensure reproducibility and address concerns regarding experimental fairness, particularly about candidate generation budgets and ranking protocols, we provide the comprehensive implementation and optimization details for our proposed CHAP framework and all evaluated baseline models.

F.1 Training Details of CHAP

The training of CHAP consists of two primary stages. In the SID Learning Stage, we pre-train the RQ-VAE to discretize item embeddings. We use the AdamW optimizer with a learning rate of 1×10^{-4} and a batch size of 1024, training the autoencoder for 500 epochs with the commitment loss weight $\beta = 0.5$ to ensure semantic stability and avoid codebook collapse. In the Hierarchical Semantic Alignment and Generation Stage, we freeze the pre-trained item codebook and fine-tune the Transformer backbone along with the alignment modules. This stage is trained for 50 epochs using the AdamW optimizer with a reduced learning rate of 1×10^{-5} , a warm-up ratio of 0.1, and a linear learning rate decay schedule. The maximum sequence length for the dual-view user behavior input is truncated to 512 tokens to balance context modeling and computational memory. All experiments are implemented in PyTorch and conducted on eight NVIDIA A100 (80GB) GPUs.

F.2 Implementation Settings of Baselines

To ensure a strictly fair, apples-to-apples comparison, all deep learning-based baselines are equipped with backbones of comparable capacity to CHAP and are provided with identical raw textual features and historical sequence lengths. Specifically, for English datasets, we uniformly use BERT-base and T5-base; for Chinese datasets, we replace them with Chinese-BERT-base and mT5-base to handle the specific tokenization and vocabulary. BERT-base has about 110M parameters, Chinese-BERT-base about 102M, T5-base about 220M, and mT5-base about 580M. All GR baselines reported in Tab. 4 use the same-capacity T5-base or mT5-base backbone as CHAP, so the efficiency comparison mainly reflects framework-level design differences rather than backbone scale.

Sparse Retrieval. We utilize the Pyserini toolkit for BM25, setting parameters $k_1 = 1.2$ and $b = 0.75$. For Doc2Query, we generate 40 synthetic queries per document using a pre-trained T5 (or mT5) model before indexing. For DeepCT, we employ the respective BERT-base models to project contextualized token embeddings into term weights.

Dense and Personalized Retrieval. For DPR, Sen-T5, MPNet, TEM, and CoPPS, we reproduce their results on this dataset using the official open-source implementations and default training strate-

gies. For personalized models (TEM and CoPPS), we explicitly feed the same historical interaction sequences (up to the exact same length as used in CHAP) into their sequence encoders to construct the user profile embeddings. Crucially, their dual-tower inner-product search is accelerated using the FAISS library to perform exhaustive vector search over the entire corpus. This ensures they are evaluated under an unbounded candidate budget, making the comparison strictly fair against CHAP, which operates under a restricted generative candidate pool.

Generative Retrieval (GR) and Ranking Fairness. To eliminate discrepancies caused by model scale and evaluate the intrinsic capability of each framework, all GR baselines (DSI, NCI, TIGER, LTRGR, MERGE, COBRA) are implemented using the same T5-base (or mT5-base) backbone. For ItemID construction in DSI and NCI, we employ Hierarchical K-Means on BERT embeddings to build semantic tree IDs. For TIGER and COBRA, we utilize their default RQ-VAE architecture configured with depth $L = 3$ and codebook size $K = 512$, strictly matching our CHAP’s dimensionality.

To directly address potential concerns regarding evaluation fairness, particularly the candidate generation budgets: no external heavy rerankers (e.g., Cross-Encoders) are applied to any baseline or our CHAP model. During inference, we ensure a rigorously fair evaluation by restricting all GR models to the exact same candidate budget ($M = 50$). Crucially, rather than enforcing an identical decoding algorithm that might artificially impair baseline performance (e.g., generating invalid IDs), we allow all GR baselines to employ their optimal Constrained Beam Search with a beam width of 50. In contrast, leveraging our decoupled architecture, CHAP efficiently generates 50 candidates via a Parallel Sampling strategy. For pure generative baselines, their final ranking relies exclusively on the beam search probabilities over their 50 candidates; for hybrid baselines (e.g., COBRA, MERGE), they utilize their internal scoring mechanisms over the same pool. Similarly, CHAP simply fuses its internally generated sparse probabilities and dense similarities in a parameter-free manner within its 50 samples.

By aligning the language backbone, enforcing identical candidate budgets ($M = 50$), maintaining the optimal decoding integrity of baselines, and restricting all models to internal parameter-free scor-

ing, we guarantee that CHAP’s superior ranking accuracy and remarkable inference throughput (QPS) stem entirely from our core architectural innovations.

G Ablation Variant Definitions

In our ablation study (Sec. 5.3), we systematically deconstruct CHAP to validate the contribution of each module. The specific configurations of the ablated variants are defined as follows:

(A) Hierarchical Semantic Alignment (HSA) Variants:

- *w/o HSA*: Replaces our trainable, aligned query encoder with a standard RQ-VAE encoder whose weights are strictly frozen after the item-side pre-training stage.
- *w/o $\mathcal{L}_{CSA}$* : Excludes the cross-sample alignment losses (both Cross-Commitment and Cross-Reconstruction), relying only on contrastive and distillation objectives.
- *w/o $\mathcal{L}_{HCL}$* : Removes the hierarchical-aware contrastive learning objective, eliminating the explicit multi-granular alignment penalty.
- *w/o $\mathcal{L}_{Distill}$* : Omits the soft probability distillation module that regularizes the fine-tuned encoder’s output distribution.

(B) Sequence Modeling Variants:

- *w/o Dense Refinement*: Eliminates the dense continuous vector from both the input sequence and the generation target. The model relies entirely on discrete sparse SIDs for sequence modeling and ranking.

(C) Residual Cascading Generation Variants:

- *w/o Residual Gen. (Autoregressive)*: Replaces the lightweight layer-wise residual fusion mechanism with a standard flat autoregressive generation, forcing the heavy Transformer Decoder to execute sequentially for all L layers.
- *w/o Self-Attention*: Retains the residual cascading mechanism but structurally removes all Masked Self-Attention sub-layers within the Transformer Decoder, retaining only the Cross-Attention mechanism to interact with the encoded history.

- *w/o Decoder (Pooling + MLP)*: Entirely removes the Transformer Decoder. Instead, it relies on mean pooling over the encoder’s outputs to derive a static global context vector, which is then fed directly into the residual blocks for SID and dense vector prediction.

H Definitions of Intent-wise Query Segments

In industrial Local-Life search scenarios, user intents are highly heterogeneous. To rigorously evaluate the model’s robustness and the specific contribution of each architectural component, we extract four intent-wise query segments from the Local-Life dataset based on query attributes and historical interaction frequencies. Notably, to ensure a rigorous and realistic evaluation, the Ambiguous, ColdStart, and LongTail subsets share the exact same entire candidate item pool (i.e., over 10 million items) as the General set during retrieval. The detailed statistics of these segments are summarized in Tab. 7.

- **General**: This segment is exactly the entire set of the Local-Life dataset. It encompasses the natural flow of daily user search traffic, reflecting the macroscopic ranking ability of the model across all scenarios.
- **Ambiguous**: This segment comprises short, highly ambiguous queries with generalized intents (e.g., “Dinner”, “Nearby attractions”, “Weekend entertainment”). These queries typically map to a vast number of potential candidate items across multiple categories. Excelling in this segment rigorously tests the model’s *Personalized Sequence Modeling* capability, as accurate retrieval strictly requires deducing explicit user preferences from historical behaviors to disambiguate the broad intent.
- **ColdStart**: This segment targets queries that exclusively lead to items introduced to the platform within the last 14 days. These items have extremely sparse or zero historical click logs. Standard generative models typically struggle on these data because they lack collaborative training signals for the new SIDs and fail to integrate dense representations for semantic fallback. This scenario directly evaluates the model’s pure zero-shot semantic matching capability.
- **LongTail**: This segment focuses on queries whose historical interaction frequencies fall into the bottom 10% of the entire search logs (e.g., highly specific long-tail constraints, niche store names, or complex negated conditions). Similar to the ColdStart segment, standard GR models often exhibit poor performance on these data because they do not combine dense representations to capture precise textual semantics, falling back instead on memorized popularity bias. In contrast, excelling in this segment heavily relies on our model’s intrinsic *Hierarchical Semantic Alignment (HSA)* and dual-view design to accurately map complex linguistics.

I Additional Diagnostics for Semantic Alignment

In addition to visualization, we use clustering-oriented quantitative diagnostics to measure whether the aligned representation space becomes more compact and separable. We consider Silhouette Score (Rousseeuw, 1987), Calinski-Harabasz Index (Caliński and Harabasz, 1974), and Davies-Bouldin Index (Davies and Bouldin, 1979) over coarse product-category clusters constructed from the sampled encoder representations. Higher Silhouette and Calinski-Harabasz scores, together with a lower Davies-Bouldin score, indicate better intra-category compactness and inter-category separation. Tab. 8 reports the corresponding metrics computed from 10,000 sampled encoder representations in ESCI-us using product-category labels as cluster assignments.

Following the matrix-style diagnostic in prior cascaded retrieval analysis, Fig. 7 further provides a similarity-matrix view of the same trend. RQ-VAE shows a relatively noisy off-diagonal background, suggesting that item-only quantization leaves residual similarity between unrelated product categories. MERGE produces a clearer diagonal structure but still contains coarse diagonal blocks, indicating that its multi-level merging improves local consistency while retaining some coarse-grained semantic entanglement. CHAP exhibits a sharper local-neighborhood diagonal and a lighter off-diagonal background, suggesting that HSA better concentrates category-consistent representations while reducing spurious cross-category similarity.

Table 7: Detailed statistics of the intent-wise query segments extracted from the Local-Life dataset.

Segment	#Items	#Queries	#Q-I Pairs	Train		Test		Avg. Hist.
				#Queries	#Q-I Pairs	#Queries	#Q-I Pairs	
General	10,145,542	3,089,026	3,089,026	2,984,294	2,984,294	104,732	104,732	7.47
Ambiguous	10,145,542	10,000	10,000	-	-	10,000	10,000	6.21
ColdStart	10,145,542	5,000	5,000	-	-	5,000	5,000	5.74
LongTail	10,145,542	10,000	10,000	-	-	10,000	10,000	10.16

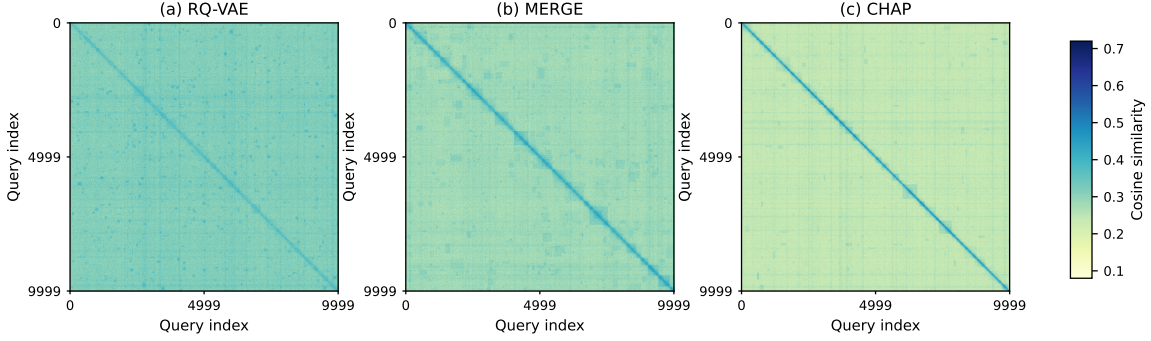


Figure 7: Similarity matrix diagnostics over 10,000 sampled queries from ESCI-us. The figure compares pairwise similarity matrices of RQ-VAE, MERGE, and CHAP, where CHAP exhibits a clearer local-neighborhood structure and fewer coarse diagonal blocks after semantic alignment.

Method	Silhouette $\uparrow$	CH Index $\uparrow$	DB Index $\downarrow$
RQ-VAE	0.218	612.4	1.684
MERGE	0.331	948.7	1.153
CHAP	0.387	1136.2	0.912

Table 8: Semantic alignment diagnostics computed from 10,000 sampled encoder representations in ESCI-us.

J SID-Level Visualization Examples

Following the qualitative SID analyses in MERGE (Zhang et al., 2025b), we further inspect how CHAP organizes items across hierarchical SID layers. We use the real item-to-SID mappings from RQ-VAE, MERGE, and CHAP on ESCI-us, and select representative queries whose candidate items contain multiple relevance levels.

Fig. 8 visualizes the layer-wise SID concentration across six representative ESCI-us queries. For each query, we count the number of unique SID prefixes occupied by query-relevant items at each layer. Lower curves therefore indicate that relevant items are assigned to fewer hierarchical branches. Compared with RQ-VAE and MERGE, CHAP consistently produces more concentrated layer-wise distributions, especially at deeper SID layers. This trend suggests that HSA encourages the SID hierarchy to better follow query-conditioned relevance rather than only item-side lexical similarity.

To examine the semantic meaning of SID layers, Fig. 9 summarizes frequent product-title keywords along one aligned camera-related SID path. The first-layer prefix (133) captures a coherent coarse-grained visual-device cluster, including camera, security camera, webcam, night vision, and motion detection products. The second-layer prefix (133, 121) narrows this cluster toward webcam and USB video-camera products, while the third-layer prefix (133, 121, 293) further concentrates on webcam products with microphone, desktop, laptop, and streaming attributes. This coarse-to-fine keyword evolution is consistent with CHAP’s hierarchical alignment objective.

Tab. 9 provides item-level examples for a separate query, “crustless sandwich maker”. Since CHAP is initialized from and fine-tuned upon the RQ-VAE codebook, the RQ-VAE and CHAP SID columns are comparable. In contrast, MERGE learns a separate codebook with its alignment loss, so its numeric SIDs should only be interpreted within MERGE’s own codebook space rather than compared token by token with RQ-VAE or CHAP. In these selected examples, exact sandwich cutter and sealer products are mapped to the same CHAP SID, whereas irrelevant items such as generic cookie cutters, handheld sandwich grills, and craft makers remain separated. This example illustrates

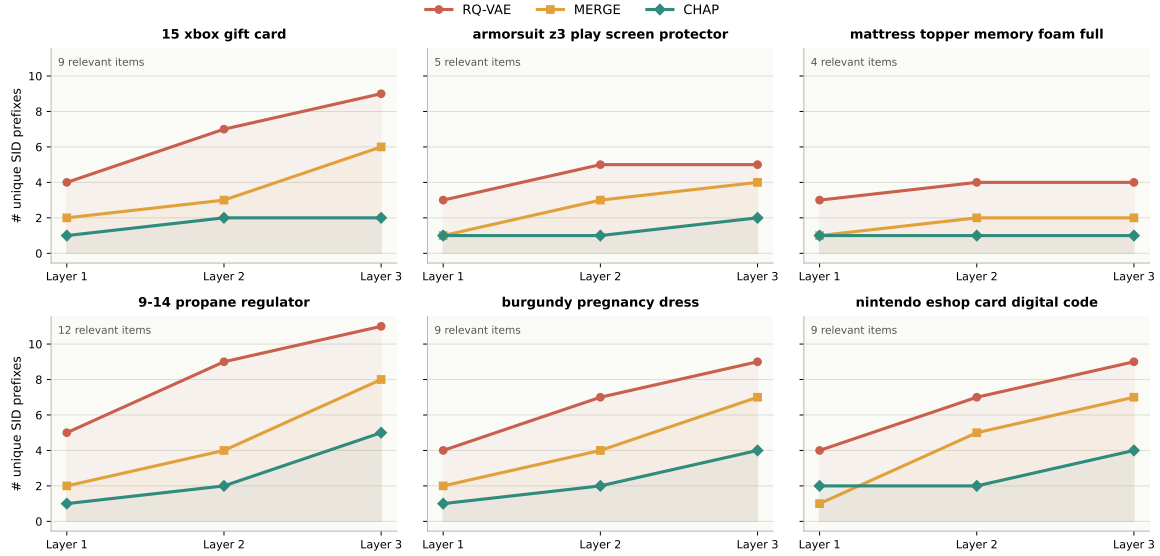


Figure 8: Layer-wise SID concentration over multiple representative ESCI-us queries. The curves are computed from the real item-to-SID mappings of RQ-VAE, MERGE, and CHAP. The y-axis reports the number of unique SID prefixes among query-relevant items, where lower values indicate more compact hierarchical assignments.

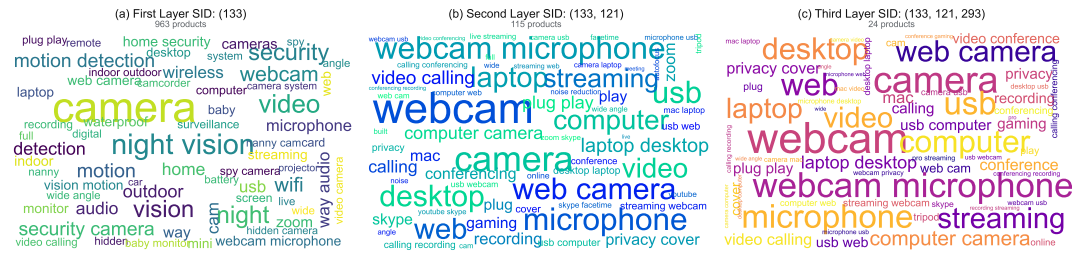


Figure 9: Product-title keyword-cloud visualization along an aligned camera-related SID path. The prefixes become progressively more specific from a broad camera-related cluster to webcam-oriented products and then to webcam products with microphone and streaming attributes.

how aligned SIDs improve the consistency of generated ItemIDs for query-relevant items.

the original work of the human authors.

K AI Usage Disclosure

In compliance with the conference guidelines regarding the use of artificial intelligence tools, we transparently disclose our light usage of AI assistants during the preparation of this manuscript. Specifically, AI tools were strictly limited to serving as utility aids for language polishing (e.g., correcting grammar, refining sentence structures, and improving readability for non-native speakers) and basic programming support (e.g., generating boilerplate scripts for data visualization).

The AI assistants did not generate any novel scientific claims, experimental designs, or core methodology. All research conceptualization, model implementation, data analysis, and the intellectual contributions of this paper are exclusively

Label	Product Title	RQ-VAE SID	MERGE SID	CHAP SID
E	Sandwich Cutter and Sealer for Kids	(14, 127, 331)	(287, 375, 142)	(14, 127, 294)
E	Decruster Sandwich Maker Set	(14, 502, 331)	(287, 64, 411)	(14, 127, 331)
E	DIY Uncrustables Sandwich Cutters	(160, 127, 393)	(287, 64, 411)	(14, 127, 331)
E	Fun Sandwich Cutters for Lunch Box	(14, 127, 441)	(287, 64, 156)	(14, 127, 226)
I	Cookie Cutters with Irreverent Phrases	(435, 359, 375)	(93, 382, 27)	(435, 241, 88)
I	Toas-Tite Handheld Sandwich Grill	(472, 154, 266)	(316, 45, 390)	(472, 308, 119)
I	Cricut Champagne Maker	(46, 491, 265)	(446, 220, 108)	(46, 73, 402)
Q	crustless sandwich maker	(14, 127, 393)	(287, 64, 411)	(14, 127, 331)

Table 9: Comparison of generated SIDs under the query “crustless sandwich maker” using RQ-VAE, MERGE, and CHAP.