GenSAR: Unifying Balanced Search and Recommendation with Generative Retrieval

Teng Shi
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
shiteng@ruc.edu.cn

Jun Xu*
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
junxu@ruc.edu.cn

Xiao Zhang
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
zhangx89@ruc.edu.cn

Xiaoxue Zang
Kuaishou Technology Co., Ltd.
Beijing, China
xxic666@126.com

Kai Zheng
Kuaishou Technology Co.. Ltd.
Beijing, China
zhengk92@gmail.com

Yang Song
Kuaishou Technology Co., Ltd.
Beijing, China
ys@sonyis.me

Enyun Yu
Independent
Beijing, China
yuenyun@126.com

Abstract

Many commercial platforms provide both search and recommendation (**S&R**) services to meet different user needs. This creates an opportunity for joint modeling of S&R. Although many joint S&R studies have demonstrated the advantages of integrating S&R, they have also identified a trade-off between the two tasks. That is, when recommendation performance improves, search performance may decline, or vice versa. This trade-off stems from the different information requirements: search prioritizes the semantic relevance between the queries and the items, while recommendation heavily relies on the collaborative relationship between users and items. To balance semantic and collaborative information and mitigate this trade-off, two main challenges arise: (1) How to incorporate both semantic and collaborative information in item representations. (2) How to train the model to understand the different information requirements of S&R. The recent rise of generative retrieval based on Large Language Models (LLMs) for S&R offers a potential solution. Generative retrieval represents each item as an identifier, allowing us to assign multiple identifiers to each item to capture both semantic and collaborative information. Additionally, generative retrieval formulates both S&R as sequence-to-sequence tasks, enabling us to unify different tasks through varied prompts, thereby helping the model better understand the requirements of each task. Based on this, we propose **GenSAR**, a method that unifies balanced S&R through generative retrieval. We design joint S&R identifiers and training tasks to address the above challenges, mitigate the trade-off between S&R, and further improve both tasks. Experimental results on a public dataset and a commercial dataset validate the effectiveness of GenSAR.

CCS Concepts

• **Information systems** → **Recommender systems**; **Personalization**.

Keywords

Recommendation; Search; Large Language Model

ACM Reference Format:
Teng Shi, Jun Xu, Xiao Zhang, Xiaoxue Zang, Kai Zheng, Yang Song, and Enyun Yu. 2025. GenSAR: Unifying Balanced Search and Recommendation with Generative Retrieval. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems (RecSys '25), September 22–26, 2025, Prague, Czech Republic.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3705328.3748071

1 Introduction

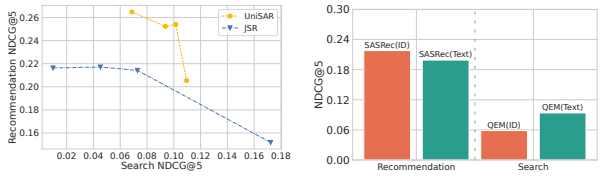
To facilitate the diverse ways of information access, many commercial platforms, such as e-commerce, video, and music platforms, offer both search [29, 31, 37] and recommendation [8, 10, 42, 43, 58–63] (**S&R**) services. This provides an opportunity for joint modeling of S&R, enabling better user interest modeling and enhancing the performance of both tasks.

Many studies have explored joint modeling of S&R, including: leveraging recommendation to enhance search [2, 3, 6, 7], using search to enhance recommendation [17, 38, 39, 46], and unified S&R modeling [36, 50, 53, 56, 57]. Although these studies have demonstrated that S&R can mutually enhance each other, they have also identified a trade-off when the model serves both tasks simultaneously [35, 36]. Specifically, when the recommendation

*Jun Xu is the corresponding author. Work partially done at Engineering Research Center of Next-Generation Intelligent Search and Recommendation, Ministry of Education. Work done when Teng Shi was an intern at Kuaishou.

RecSys '25, Prague, Czech Republic

ACM ISBN 979-8-4007-1364-4/25/09
https://doi.org/10.1145/3705328.3748071



(a) Trade-off between S&R (b) Performance of different embeddings

Figure 1: Empirical analysis on the Commercial dataset: (a) A trade-off between S&R is observed in representative joint S&R methods, JSR [56] and UniSAR [36]. (b) The performance of the sequential recommendation model SASRec [21] and the product search model QEM [2], using ID and text embeddings, respectively.

performance improves, the search performance tends to degrade, and vice versa. Empirical analysis of the representative methods of JSR [56] and UniSAR [36] based on a S&R dataset collected from a real commercial platform also confirmed the performance trade-off, as shown in Figure 1(a). More details please refer to Section 4.1.1.

Analysis also showed that the trade-off is rooted in the different information requirements of S&R. Search typically focuses more on the semantic relevance between queries and items, with traditional search models often based on pre-trained language models [20, 49, 51]. In contrast, recommendation heavily relies on collaborative information, where ID-based recommendation can yield excellent results [16, 21, 54]. Figure 1(b) shows an empirical validation where the S&R performances with ID- and Text-only embeddings are shown. The ID embeddings are randomly initialized and trained, containing collaborative information, while the Text embeddings are trained with BGE [49] and then reduced to the same dimensionality as that of the ID embeddings, containing semantic information. From Figure 1(b), we found that recommendation relies more on collaborative information while search focuses more on semantic information.

Therefore, balancing the semantic information required for search and the collaborative information needed for recommendation becomes a key issue in joint S&R modeling. It is non-trivial and faces two major challenges: (1) How to incorporate both semantic and collaborative information in item representations. Existing joint S&R models typically assign a single representation to each item, making it difficult to capture both types of information effectively; (2) How to let the model understand the difference in information requirements of S&R during training. Current joint models often treat S&R tasks identically, without differentiating them during training. This makes it challenging for the model to grasp their distinct requirements.

Recently, Large Language Model (LLM) [67]-based generative retrieval for search [44, 71] and recommendation [13, 33, 68] have garnered significant attention. This provides a solution to the aforementioned challenges: (1) Generative retrieval assigns an identifier (a sequence of tokens) to each item, allowing us to assign multiple identifiers to each item to balance semantic and collaborative information; (2) Generative retrieval formulates both S&R as sequence-to-sequence (Seq2Seq) tasks, enabling the unification of different S&R tasks and helping the model better understand the distinct requirements of each task.

Based on this, we propose **GenSAR**, which unifies balanced search and recommendation through generative retrieval, thereby alleviating the trade-off between S&R to better enhance each other. Firstly, we design a joint S&R identifier that integrates both semantic and collaborative information. Building on the RQ-VAE [33, 68] method, we employ shared codebooks for both semantic and collaborative information, alongside specific codebooks for each. As a result, items from search are represented by semantic codes, while items from recommendation are represented by collaborative codes. These two codes share a common portion to capture shared information while also retaining distinct parts to preserve the unique characteristics of semantic and collaborative information. Secondly, we design the joint S&R training tasks. We prepend a token representing the behavior type to the item identifier and then input the user's S&R history into the LLM (with the user query also provided for search). Different prompts are used to guide LLMs to predict the next recommended item, the next searched query, and the next searched item, enabling the model to understand the distinct requirements for S&R.

The major contributions of the paper are summarized as follows:
• We verified the existence of the trade-off between S&R, and identified that this trade-off arises from the different information requirements of S&R. Additionally, we have analyzed the challenges in balancing semantic and collaborative information needed for S&R.
• We propose GenSAR, which unifies balanced S&R through generative retrieval. We designed a joint S&R identifier to balance semantic and collaborative information, and developed joint training tasks to help the model understand the different requirements of each task.
• Experimental results on two datasets validate the effectiveness of GenSAR. GenSAR not only surpasses traditional S&R models but also outperforms generative S&R models.

2 Related Work

Joint Search and Recommendation. Joint modeling of S&R has attracted increasing attention in recent years and can be broadly categorized into three types: (1) Enhancing search with recommendation [2, 3, 6, 7], such as TEM [6], which uses Transformers to model user preferences, and CoPPS [7], which applies contrastive learning to address data sparsity. (2) Enhancing recommendation with search [17, 38, 39, 46], e.g., SESRec [39], which disentangles similar and dissimilar interests from both histories. (3) Unified modeling of S&R [36, 50, 53, 56, 57, 64–66], such as JSR [56, 57] with joint loss and UniSAR [36], which models behavior transitions. While these works show mutual benefits between S&R, they also reveal a trade-off. This paper addresses that trade-off within a generative retrieval framework.

Generative Search and Recommendation. With the rise of Large Language Models (LLMs) [67], LLM-based generative retrieval has been widely explored for both search [5, 23, 28, 32, 41, 44, 47, 70, 71] and recommendation [13, 19, 27, 30, 33, 52, 68]. These methods represent items as identifiers and input the user query (for search) or user history (for recommendation) into the LLM to generate the target item. Identifier designs can be grouped into: (1) Text-based,

using item titles [9, 25] or substrings [5, 24]; (2) Non-learnable ID-based, with early methods assigning random IDs [13], and later ones using clustering to encode semantic or collaborative structure [19, 44, 47]; (3) Learnable codebook-based, applying techniques like RQ-VAE [33, 68] to learn identifiers from semantic or collaborative embeddings. However, most existing approaches design identifiers tailored to either search or recommendation, focusing solely on semantic or collaborative information. In joint S&R, balancing both is essential for strong performance across tasks.

3 Our Approach

This section introduces our proposed method, GenSAR. Section 3.1 defines the Joint Search and Recommendation task. Section 3.2 presents the Joint Identifier module, where we design separate semantic and collaborative identifiers to balance the different needs of search and recommendation. Section 3.3 describes task-specific training objectives to help the model capture both types of information. Finally, Section 3.4 details the training and inference process of GenSAR.

3.1 Problem Formulation

Let $\mathcal{U}, \mathcal{V}, \mathcal{Q}$ denote the sets of users, items, and queries, respectively. Each user $u \in \mathcal{U}$ has a chronologically ordered interaction history $S_u = [(b_1, x_1), (b_2, x_2), \ldots, (b_N, x_N)]$ that includes her historical S&R behaviors, where N denotes the number of u's historical behaviors. $b_i \in \{\langle \mathrm{R_I} \rangle, \langle \mathrm{S_Q} \rangle, \langle \mathrm{S_I} \rangle\}$ represents the type of the i-th behavior: $\langle \mathrm{R_I} \rangle$ indicates an item clicked by the user after a recommendation, $\langle \mathrm{S_Q} \rangle$ represents a query searched by the user, and $\langle \mathrm{S_I} \rangle$ denotes an item clicked by the user after searching a query. x_i denotes the i-th behavior:

$$x_i = \begin{cases} v_i, & \text{if } b_i = \langle \mathrm{R_I} \rangle \text{ or } b_i = \langle \mathrm{S_I} \rangle, \\ q_i, & \text{if } b_i = \langle \mathrm{S_Q} \rangle, \end{cases} \tag{1}$$

where $v_i \in \mathcal{V}$ denotes the i-th interacted item and $q_i \in \mathcal{Q}$ is the i-th searched query. Our goal is to enable the model to understand user interests and predict the next item v_{N+1} for search when $b_{N+1} = \langle \mathrm{S_I} \rangle$ or recommendation when $b_{N+1} = \langle \mathrm{R_I} \rangle$.

3.2 Joint Search and Recommendation Identifier

This section introduces the design of the joint S&R identifier (Figure 2). We first extract semantic and collaborative embeddings for each item. Using RQ-VAE [22, 33, 68], we apply both shared and separate codebooks to learn two identifiers per item—one semantic, one collaborative. The identifiers share common parts to capture shared information, while retaining unique parts to reflect task-specific features.

3.2.1 Embedding Extraction. For each item $v \in \mathcal{V}$, we can input its textual information, such as the title and description, into a pre-trained retrieval model (e.g., BERT [12], BGE [49]) to obtain an embedding $\mathbf{v}_s \in \mathbb{R}^{d_s}$ that contains its semantic information. Meanwhile, we can also obtain an embedding $\mathbf{v}_c \in \mathbb{R}^{d_c}$ containing its collaborative information from a pre-trained recommendation model (e.g., SASRec [21], BERT4Rec [40]). d_s and d_c represent the dimensions of the semantic and collaborative embeddings, respectively. We map the semantic and collaborative embeddings to the same-dimensional latent space using two encoders:

$$\mathbf{z}_s = \mathrm{Encoder}_s(\mathbf{v}_s), \quad \mathbf{z}_c = \mathrm{Encoder}_c(\mathbf{v}_c), \tag{2}$$

where $\mathrm{z}_s \in \mathbb{R}^d, \mathrm{z}_c \in \mathbb{R}^d$ and d is the dimension of the latent embeddings, $\mathrm{Encoder}_s(\cdot)$ and $\mathrm{Encoder}_c(\cdot)$ are two MLPs (Multilayer Perceptrons).

3.2.2 Residual Quantization. To integrate both semantic and collaborative information, we use L_m-level shared codebooks, along with L_n-level specific codebooks for semantic and collaborative information, respectively. First, the latent embeddings for semantic and collaborative information, $\mathbf{z}_s$ and $\mathbf{z}_c$, are concatenated to form $\mathbf{r}_0^m = [\mathbf{z}_s; \mathbf{z}_c] \in \mathbb{R}^{2d}$. This $\mathbf{r}_0^m$ is then passed through the L_m-level shared codebooks to obtain the shared codes I_m and the residual embedding $\mathbf{r}_{L_m}^m$. Then, we extract the semantic part $\mathbf{r}_0^s = \mathbf{r}_{L_m}^m\,[1{:}d] \in \mathbb{R}^d$ and the collaborative part $\mathbf{r}_0^c = \mathbf{r}_{L_m}^m\,[d{:}2d] \in \mathbb{R}^d$ from $\mathbf{r}_{L_m}^m$, and input them separately into the semantic and collaborative codebooks to learn their specific codes I_s and I_c, respectively. Finally, the shared and specific codes are concatenated, resulting in two identifiers, I_{m+s} and I_{m+c}, for each item. Next, we will introduce the residual quantization process for both the shared and specific codebooks.

• **Shared Codebooks.** We have L_m-level shared codebooks. At each level $i \in \{1, 2, \ldots, L_m\}$, we have a shared codebook $C_i^m = \{\mathbf{e}_k\}_{k=1}^K$, where K is the size of each codebook and $\mathbf{e}_k \in \mathbb{R}^{2d}$ is a learnable code embedding. The residual quantization process for the shared codebooks is as follows:

$$\begin{aligned} c_i^m &= \arg\min_k ||\mathbf{r}_{i-1}^m - \mathbf{e}_k||_2^2, \quad \mathbf{e}_k \in C_i^m, \\ \mathbf{r}_i^m &= \mathbf{r}_{i-1}^m - \mathbf{e}_{c_i^m}, \quad \mathbf{r}_0^m = [\mathbf{z}_s; \mathbf{z}_c] \in \mathbb{R}^{2d}, \end{aligned} \tag{3}$$

where c_i^m is the assigned code from the i-th level of the shared codebook. $\mathbf{r}_{i-1}^m$ is the residual from last level. Through the recursive quantization in Eq. (3), we can obtain the shared codes $I_m = \left[c_1^m, c_2^m, \ldots, c_{L_m}^m\right]$ and the residual embedding $\mathbf{r}_{L_m}^m$.

• **Specific Codebooks.** We can extract the semantic part $\mathbf{r}_0^s = \mathbf{r}_{L_m}^m\,[1{:}d] \in \mathbb{R}^d$ and the collaborative part $\mathbf{r}_0^c = \mathbf{r}_{L_m}^m\,[d{:}2d] \in \mathbb{R}^d$ from the residual embedding $\mathbf{r}_{L_m}^m$ outputted by the shared codebooks. We then pass them separately through the L_n-level semantic and collaborative specific codebooks C_i^s and C_i^c, where $i \in \{1, 2, \ldots, L_n\}$. Please note that, unlike the shared codebook whose code embeddings are $2d$-dimensional, the code embeddings of the specific codebooks are d-dimensional. The residual quantization process for the specific codebooks can be formulated as follows:

$$\begin{aligned} c_i^s &= \arg\min_k ||\mathbf{r}_{i-1}^s - \mathbf{e}_k||_2^2, \quad \mathbf{e}_k \in C_i^s, \\ c_i^c &= \arg\min_k ||\mathbf{r}_{i-1}^c - \mathbf{e}_k||_2^2, \quad \mathbf{e}_k \in C_i^c, \\ \mathbf{r}_i^s &= \mathbf{r}_{i-1}^s - \mathbf{e}_{c_i^s}, \quad \mathbf{r}_i^c = \mathbf{r}_{i-1}^c - \mathbf{e}_{c_i^c}, \end{aligned} \tag{4}$$

where c_i^s and c_i^r represent the codes assigned by the i-th level semantic-specific and collaborative-specific codebooks, respectively. Through the recursive quantization in Eq. (4), we can obtain the semantic-specific and collaborative-specific codes as follows:

$$I_s = \left[c_1^s, c_2^s, \ldots, c_{L_n}^s\right], \quad I_c = \left[c_1^c, c_2^c, \ldots, c_{L_n}^c\right].$$

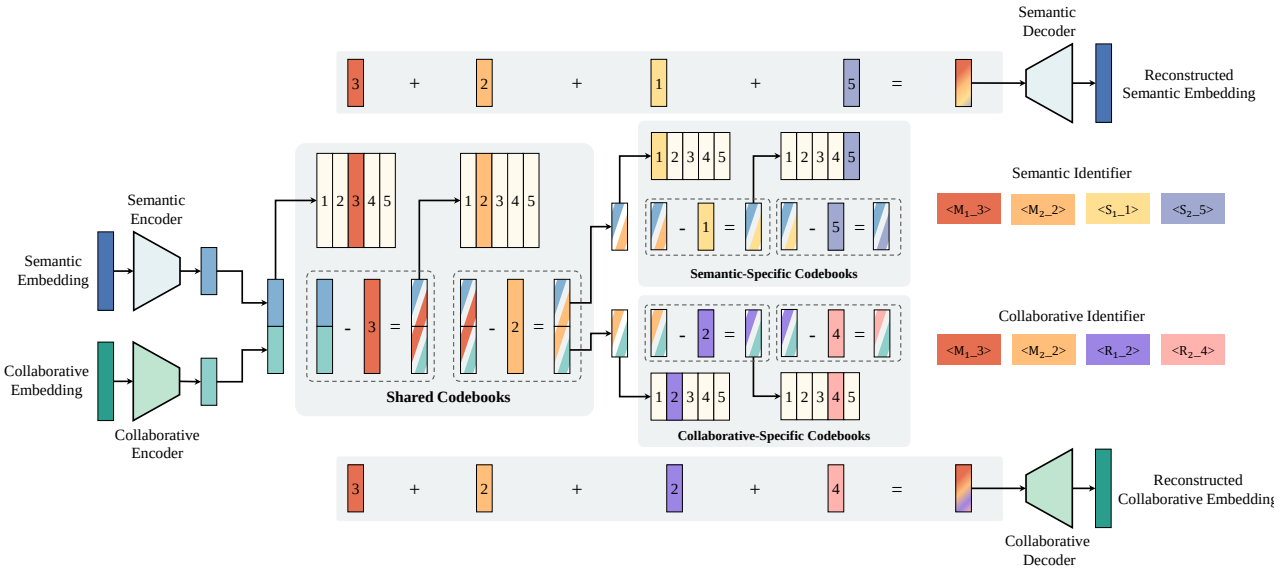


Figure 2: The joint search and recommendation identifier. We extract the semantic and collaborative embeddings for each item. These two embeddings are first concatenated and passed through the shared codebooks to learn shared codes. Then, the semantic and collaborative embeddings are separately processed through specific codebooks to learn specific codes. Finally, these codes are concatenated to form two identifiers for each item: one for semantics and one for collaboration.

Finally, by concatenating the shared codes and the specific codes, we can obtain the semantic identifier I_{m+s} and collaborative identifier I_{m+c} for item v:

$$\begin{aligned} I_{m+s} &= \left[c_1^m, c_2^m, \ldots, c_{L_m}^m, c_1^s, c_2^s, \ldots, c_{L_n}^s\right], \\ I_{m+c} &= \left[c_1^m, c_2^m, \ldots, c_{L_m}^m, c_1^c, c_2^c, \ldots, c_{L_n}^c\right]. \end{aligned} \tag{5}$$

3.2.3 Identifier Training. After passing through the shared and specific codebooks, we can obtain the semantic and collaborative quantized embeddings as follows:

$$\hat{\mathbf{z}}_s = \sum_{i=1}^{L_m} \mathbf{e}_{c_i^m}\,[1{:}d] + \sum_{i=1}^{L_n} \mathbf{e}_{c_i^s}, \quad \hat{\mathbf{z}}_c = \sum_{i=1}^{L_m} \mathbf{e}_{c_i^m}\,[d{:}2d] + \sum_{i=1}^{L_n} \mathbf{e}_{c_i^c}, \tag{6}$$

where $\mathbf{e}_{c_i^m} \in \mathbb{R}^{2d}$ is the code embedding of the shared codebooks, $\mathbf{e}_{c_i^s} \in \mathbb{R}^d$ and $\mathbf{e}_{c_i^c} \in \mathbb{R}^d$ are the code embeddings of the semantic and collaborative specific codebooks. The quantized semantic embedding $\hat{\mathbf{z}}_s \in \mathbb{R}^d$ and collaborative embedding $\hat{\mathbf{z}}_c \in \mathbb{R}^d$ will be used to reconstruct the original semantic and collaborative embeddings, $\mathbf{v}_s$ and $\mathbf{v}_c$:

$$\hat{\mathbf{v}}_s = \text{Decoder}_s(\hat{\mathbf{z}}_s), \quad \hat{\mathbf{v}}_c = \text{Decoder}_c(\hat{\mathbf{z}}_c), \tag{7}$$

where $\text{Decoder}_s(\cdot)$ and $\text{Decoder}_c(\cdot)$ are two MLPs. We can compute the reconstruction loss used for training the encoder and decoder as follows:

$$\mathcal{L}_{\text{Recon}} = ||\mathbf{v}_s - \hat{\mathbf{v}}_s||_2^2 + ||\mathbf{v}_c - \hat{\mathbf{v}}_c||_2^2. \tag{8}$$

We can also compute the loss for residual quantization as follows:

$$\begin{aligned} \mathcal{L}_{\text{RQ}}^m &= \sum_{i=1}^{L_m} ||\text{sg}[\mathbf{r}_{i-1}^m] - \mathbf{e}_{c_i^m}||_2^2 + \alpha||\mathbf{r}_{i-1}^m - \text{sg}[\mathbf{e}_{c_i^m}]||_2^2, \\ \mathcal{L}_{\text{RQ}}^s &= \sum_{i=1}^{L_n} ||\text{sg}[\mathbf{r}_{i-1}^s] - \mathbf{e}_{c_i^s}||_2^2 + \alpha||\mathbf{r}_{i-1}^s - \text{sg}[\mathbf{e}_{c_i^s}]||_2^2, \\ \mathcal{L}_{\text{RQ}}^c &= \sum_{i=1}^{L_n} ||\text{sg}[\mathbf{r}_{i-1}^c] - \mathbf{e}_{c_i^c}||_2^2 + \alpha||\mathbf{r}_{i-1}^c - \text{sg}[\mathbf{e}_{c_i^c}]||_2^2, \\ \mathcal{L}_{\text{RQ}} &= \mathcal{L}_{\text{RQ}}^m + \mathcal{L}_{\text{RQ}}^s + \mathcal{L}_{\text{RQ}}^c, \end{aligned} \tag{9}$$

where $\text{sg}[\cdot]$ denotes the stop-gradient operation and α is a hyper-parameter. $\mathcal{L}_{\text{RQ}}$ is used to train the code embeddings in both the shared and specific codebooks. Finally, the total loss for training the identifier is as follows:

$$\mathcal{L}_{\text{RQ-VAE}} = \mathcal{L}_{\text{Recon}} + \mathcal{L}_{\text{RQ}}. \tag{10}$$

3.2.4 Behavior-aware Identifier. After learning the semantic and collaborative identifiers for each item, we can represent each user interaction (b_i, x_i) as shown in Eq. (1). To help the model understand different behaviors in the user's interaction history, we prepend a token indicating the behavior type to each interaction's identifier. For interactions involving items, we prepend the corresponding behavior token to the identifier of each item. For interactions involving queries, we prepend the behavior token to the word sequence of the query. It can be formulated as follows:

$$\text{ID}(b_i, x_i) = \begin{cases} \left[\langle \text{R}_\text{I} \rangle, c_1^m, c_2^m, \ldots, c_{L_m}^m, c_1^c, c_2^c, \ldots, c_{L_n}^c\right], & \text{if } b_i = \langle \text{R}_\text{I} \rangle, \\ \left[\langle \text{S}_\text{Q} \rangle, w_1, w_2, \ldots, w_{|q_i|}\right], & \text{if } b_i = \langle \text{S}_\text{Q} \rangle, \\ \left[\langle \text{S}_\text{I} \rangle, c_1^m, c_2^m, \ldots, c_{L_m}^m, c_1^s, c_2^s, \ldots, c_{L_n}^s\right], & \text{if } b_i = \langle \text{S}_\text{I} \rangle, \end{cases} \tag{11}$$

where $[w_1, w_2, \ldots, w_{|q_i|}]$ are the words of query q_i. ID$(\cdot)$ denotes the function for obtaining the identifier of each interaction.

3.3 Joint Search and Recommendation Training

To better adapt the LLM to joint S&R tasks, we design training objectives that help it understand user behaviors and effectively learn both semantic and collaborative identifiers.

3.3.1 Next Recommendation Item Prediction. To enable the LLM to perform well on the recommendation task, we let it predict the next recommended item. Unlike previous generative recommendation models [13, 33, 68] that only use the user's recommendation history, our approach incorporates search history as well. This allows the LLM to better leverage the user's historical information and understand the relationship between S&R behaviors. A sample of the data is shown below:

Next Recommendation Item Prediction

Instruction: Below is the user's interaction history: ⟨S_Q⟩ Piano; ⟨S_I⟩ <M_1_247> <M_2_197> <S_1_184> <S_2_110>; ...; ⟨R_I⟩ <M_1_30> <M_2_147> <R_1_247> <R_2_229>. Please recommend the next item the user is likely to click.
Response: ⟨R_I⟩ <M_1_10> <M_2_25> <R_1_52> <R_2_37>

Here, "<M_1_10> <M_2_25>" represents the shared semantic and collaborative identifier of the item, "<S_1_184> <S_2_110>" represents the semantic-specific identifier, and "<R_1_52> <R_2_37>" represents the collaborative-specific identifier.

3.3.2 Next Search Query Prediction. Some works focus on query recommendation [4, 14, 48], where they predict the next query a user is likely to search. Since our user interaction history also includes search queries, we introduce a task that allows the LLM to predict the user's next intended search query based on their history. This helps the model better understand user search intent and the relationship between S&R behaviors. A sample of the data for this task is as follows:

Next Search Query Prediction

Instruction: Below is the user's interaction history: ⟨R_I⟩ <M_1_199> <M_2_175> <R_1_1> <R_2_44>; ⟨R_I⟩ <M_1_209> <M_2_235> <R_1_159> <R_2_80>; ...; ⟨R_I⟩ <M_1_147> <M_2_68> <R_1_118> <R_2_85>. Please predict the next query the user might want to search.
Response: ⟨S_Q⟩ Artificial Intelligence

3.3.3 Next Search Item Prediction. To enable the model to perform well on the search task, we have it predict the next search item. Previous generative search models [44, 71] only input the user's query into the LLM to predict the target item, which considers only the correlation between the query and the item, without taking the user's preferences into account. To address this, we include the user's S&R history in the input to reflect their preferences. A sample of the data for this task is as follows:

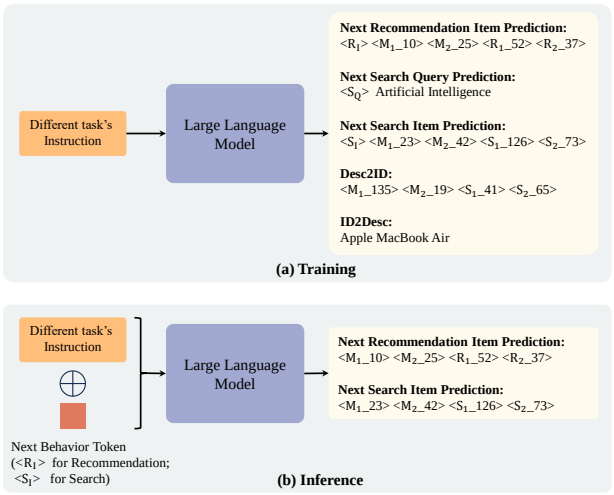


Figure 3: Training and Inference Process of GenSAR. During training, we provide LLM with different instructions to generate corresponding responses. During inference, we append a token at the end of the instruction to indicate the type of behavior to be predicted, enabling the LLM to be applied to either search or recommendation tasks.

Next Search Item Prediction

Instruction: Below is the user's interaction history: ⟨R_I⟩ <M_1_199> <M_2_175> <R_1_1> <R_2_44>; ⟨R_I⟩ <M_1_209> <M_2_235> <R_1_159> <R_2_80>; ...; ⟨R_I⟩ <M_1_147> <M_2_68> <R_1_118> <R_2_85>. The user's search query is ⟨S_Q⟩ Artificial Intelligence. Please predict the next item the user might click.
Response: ⟨S_I⟩ <M_1_23> <M_2_42> <S_1_126> <S_2_73>

Here, "⟨S_Q⟩ Artificial Intelligence" denotes the query that the user is currently searching for.

3.3.4 Identifier-Language Alignment. To enhance the LLM's understanding of both the collaborative and semantic identifiers of each item, we designed an identifier-language alignment task. This task enables the LLM to generate a corresponding description based on an item's identifier and, conversely, to generate the appropriate identifier from the item's description.

First, we have the Desc2ID task, which enables the LLM to generate the corresponding item identifier based on its description.

Desc2ID

Instruction: Using the provided description "Apple MacBook Air", predict the corresponding item.
Response: <M_1_135> <M_2_19> <S_1_41> <S_2_65>

Then, we have the ID2Desc task, which enables the LLM to generate the corresponding item description based on its identifier.

ID2Desc

Instruction: Please provide a description for the item <M_1_135> <M_2_19> <S_1_41> <S_2_65>.
Response: Apple MacBook Air.

Table 1: Comparison of different generative search or recommendation methods. "S." and "R." denote search and recommendation respectively.

Methods	Scale	Backbone	Task		Identifier	
			S.	R.	Semantic	Collaborative
P5 [13, 19]	60M/220M	T5-small/T5-base	✗	✓	✗	✓
TIGER [33]	60M	T5-small	✗	✓	✓	✗
LC-Rec [68]	7B	LLaMA	✗	✓	✓	✗
DSI-QG[71]	220M	T5-base	✓	✗	✓	✗
WebUltron [70]	220M	T5-base	✓	✗	✓	✗
GenRet [41]	220M	T5-base	✓	✗	✓	✗
GenSAR (Ours)	60M	T5-small	✓	✓	✓	✓

Please note that for both semantic and collaborative identifiers, we include the Desc2ID and ID2Desc training tasks. Since the input and output of these two tasks do not involve user history, we do not prepend a token indicating the behavior type to the identifier.

3.4 Training and Inference

This section introduces how to train the LLM for joint S&R, and how to use the trained LLM during inference to generate the target item for either the search or recommendation task. The training and inference process of GenSAR is shown in Figure 3.

3.4.1 Training. As previously mentioned, each interaction in the user's history is represented as an identifier, allowing us to formulate the task as a sequence-to-sequence problem. We train the model using next token prediction, optimizing the negative log-likelihood of generating the target as follows:

$$\mathcal{L} = -\sum_{t=1}^{T} log\ P(y_t|y_{<t},\ \text{Ins}). \tag{12}$$

Here, y represents the behavior-aware identifier of the target to be predicted, as defined in Eq. (11). T is the length of the identifier of the target item. Ins refers to the various instructions described in Section 3.3, which are used as inputs for the LLM.

3.4.2 Inference. During training, we train the LLM according to the input-output format described in Section 3.3. During inference, to apply the LLM to search and recommendation tasks, we append a behavior token, either "$\langle S_I \rangle$" for search or "$\langle R_I \rangle$" for recommendation, to the input of the LLM to prompt it to generate the corresponding next item for search or recommendation, respectively. The other tasks mentioned in Section 3.3 are used as auxiliary tasks during training to help the model better understand user S&R behaviors. During generation, to ensure that the items generated by the LLM are within the candidate set, we follow previous works [19, 68] and use constrained beam search.

3.5 Discussion

As shown in Table 1, we compare GenSAR with various generative search or recommendation methods in terms of scale (number of parameters), backbone architecture used, and applicable tasks. GenSAR adopts T5-small as its backbone, resulting in a relatively small number of parameters while being capable of serving both S&R tasks. Compared with existing methods, it achieves an optimal balance between efficiency and effectiveness.

Table 2: Statistics of the datasets used in this paper. "S" and "R" denote search and recommendation, respectively.

Dataset	#Users	#Items	#Queries	#Interaction-R	#Interaction-S
Amazon	192,403	62,883	983	1,266,903	1,081,934
Commercial	10,000	782,225	135,206	4,286,866	383,465

In terms of novelty, unlike existing methods that focus solely on either semantic or collaborative information in identifier design, our approach incorporates both the semantic information required for search and the collaborative signals essential for recommendation. This joint consideration helps alleviate the trade-off between S&R.

4 Experiments

We conducted experiments to evaluate the performance of GenSAR. The source code and experimental details are available [1].

4.1 Experimental Setup

4.1.1 Dataset. We conducted experiments on the following datasets: (1) **Amazon**[2] [15, 26]: Following previous works [2, 3, 36, 39], we use the semi-synthetic dataset based on Amazon recommendation data as the public dataset for our experiments. [3] (2) **Commercial**: To thoroughly evaluate the effectiveness of GenSAR, we collected a dataset from a Chinese commercial app, containing S&R interactions from 5,000 users over two weeks. For details on data processing and train/validation/test splitting, please see the code link.

4.1.2 Baselines. In this work, we use the following representative methods as baselines for comparison with GenSAR.

First, we compare with the following recommendation models: (1) *Sequential Recommendation*: **GRU4Rec** [18]; **SASRec** [21]; **FMLP-Rec** [69]; **LRURec** [55]. (2) *Generative Recommendation*: **P5-CID** [13, 19]; **TIGER** [33]; **LC-Rec** [68]. Next, we compare with the following search models: (1) *Personalized Search*: **QEM** [2]; **TEM** [6]; **CoPPS** [7]. (2) *Dense Retrieval*: **E5**[4] [45]; **BGE**[5] [49]. (3) *Generative Retrieval*: **DSI-QG** [71]; **WebUltron** [70]; **GenRet** [41]. Finally, we compare with the following joint S&R models: **JSR** [56]; **SESRec** [39]; **UnifiedSSR** [50]; **UniSAR** [36]. For more details on the baselines, please see the code link.

4.1.3 Evaluation Metrics & Implementation Details . Following previous works [36, 39, 69], we use ranking metrics including top-k *Hit Ratio* (HR) and top-k *Normalized Discounted Cumulative Gain* (NDCG). We report the results for k values of {1, 5, 10}, and since NDCG@1 is the same as HR@1, we do not report it. For more details on the evaluation and model implementation, please see the code link.

[1] https://github.com/TengShi-RUC/GenSAR

[2] https://cseweb.ucsd.edu/~jmcauley/datasets/amazon/links.html, https://github.com/QingyaoAi/Amazon-Product-Search-Datasets

[3] Please note that 70% of the items in the "Kindle Store" subset used in previous works [36, 39] lack textual information, so we use the "Electronics" subset, where less than 1% of the items lack text.

[4] https://huggingface.co/intfloat/multilingual-e5-base

[5] https://huggingface.co/BAAI/bge-base-en-v1.5, https://huggingface.co/BAAI/bge-base-zh-v1.5

Table 3: The *recommendation* performance of different methods on the two datasets. The best and the second-best methods are highlighted in bold and underlined fonts, respectively. The improvements over the second-best methods are statistically significant (t-test, p-value< 0.05). Following commonly used settings [36, 39, 69], we pair the ground-truth item with 99 randomly sampled items that the user has not interacted with to form the candidate list.

Datasets	Metrics	Recommendation							Joint Search and Recommendation				
		GRU4Rec	SASRec	FMLP-Rec	LRURec	P5-CID	TIGER	LC-Rec	JSR	SESRec	UnifiedSSR	UniSAR	**GenSAR**
Amazon	HR@1	0.0440	0.0544	0.0534	0.0544	0.0881	0.1073	0.1063	0.0657	0.0627	0.0477	0.0680	**0.1261**
	HR@5	0.1716	0.1887	0.1898	0.1890	0.1874	0.2046	0.1973	0.2075	0.2083	0.1667	0.2171	**0.2228**
	HR@10	0.2884	0.2992	0.3041	0.3001	0.2790	0.2852	0.2760	0.3188	0.3209	0.2707	**0.3319**	0.3063
	NDCG@5	0.1074	0.1216	0.1217	0.1218	0.1380	0.1565	0.1522	0.1371	0.1359	0.1071	0.1432	**0.1748**
	NDCG@10	0.1449	0.1571	0.1584	0.1575	0.1674	0.1824	0.1774	0.1729	0.1721	0.1405	0.1802	**0.2015**
Commercial	HR@1	0.1022	0.1519	0.1442	0.1363	0.2843	0.2630	0.2703	0.1576	0.1890	0.1515	0.2214	**0.2997**
	HR@5	0.2526	0.2812	0.2711	0.2637	0.3305	0.3013	0.3001	0.2685	0.2845	0.2844	0.3228	**0.3496**
	HR@10	0.3527	0.3716	0.3584	0.3525	0.3830	0.3448	0.3333	0.3529	0.3690	0.3870	**0.4056**	0.4031
	NDCG@5	0.1787	0.2179	0.2093	0.2021	0.3072	0.2819	0.2849	0.2142	0.2370	0.2195	0.2727	**0.3241**
	NDCG@10	0.2110	0.2470	0.2373	0.2306	0.3240	0.2958	0.2955	0.2413	0.2641	0.2524	0.2993	**0.3411**

4.2 Overall Performance

Table 3 and Table 4 show the S&R results on two datasets, respectively. From the results, we can observe that:

• Firstly, it can be seen that compared to existing search or recommendation models, GenSAR achieves state-of-the-art results. This validates the effectiveness of GenSAR in alleviating the trade-off between S&R through generative retrieval, by designing joint identifiers and training tasks for both tasks.

• Secondly, we can observe that most joint S&R methods (e.g., JSR, UniSAR, GenSAR) outperform traditional methods that using only item IDs, such as sequential recommendation (e.g., SASRec, FMLP-Rec) and personalized search methods (e.g., QEM, TEM, CoPPS). This demonstrates the advantages of jointly modeling of S&R, as it enhances the performance of both tasks.

• Thirdly, it can be observed that for search, dense retrieval (e.g., E5, BGE) and generative retrieval (e.g., GenRet, GenSAR) methods that rely on semantic information outperform personalized search models (e.g., QEM, TEM, CoPPS) that rely solely on ID information. This also confirms that for search, semantic information is more important than collaborative information.

4.3 Ablation Study

We conducted ablation study on the Commercial dataset to validate the effectiveness of the various training tasks in GenSAR, as shown in Table 5.

Impact of Behavior Token. As shown in Section 3.2.4, we prepended a token indicating the type of behavior to the identifier of each user interaction, enabling the LLM to recognize different behavior types. To evaluate its impact, we removed this behavior token, as shown in Table 5 ("w/o Behavior Token"). The results indicate that removing the behavior token degrades performance, validating that adding this token helps the LLM better understand the relationship between user S&R behaviors.

Next Recommendation Item Prediction (NRIP). As shown in Section 3.3.1, we incorporated the training task "Next Recommendation Item Prediction" (NRIP), which enables the LLM to predict the next item to recommend based on user history. To evaluate its impact, we removed this task, as shown in Table 5 ("w/o NRIP"). The results demonstrate that removing this task significantly degrades recommendation performance and slightly reduces search performance, highlighting the importance of NRIP. Additionally, this demonstrates that recommendation training tasks can enhance search performance, verifying that recommendation can benefit search.

Next Search Query Prediction (NSQP). We included the training task "Next Search Query Prediction" (NSQP) to enable the LLM to better understand user intent by predicting the next query a user might want to search, as described in Section 3.3.2. To evaluate its impact, we observed the results after removing this task, as shown in Table 5 ("w/o NSQP"). The results indicate that removing this task significantly degrades search performance and also affects recommendation performance, demonstrating that NSQP helps the model better understand user search intent.

Next Search Item Prediction (NSIP). In Section 3.3.3, we introduced the training task "Next Search Item Prediction" (NSIP), which allows the LLM to predict the next item a user might click based on their history and input query. We analyzed the impact of this task, as shown in Table 5 ("w/o NSIP"). The results indicate that removing this task significantly degrades search performance, while also slightly affecting recommendation performance. This demonstrates the importance of NSIP for search and further highlights that search training tasks can enhance recommendation performance, validating that search can assist recommendation.

Identifier-Language Alignment. In Section 3.3.4, we introduced two tasks, Desc2ID and ID2Desc, for identifier-language alignment, which help the LLM better understand the semantic and collaborative identifiers of each item. We observed the impact of removing these two tasks, as shown in Table 5 (w/o "Desc2ID" and w/o "ID2Desc"). It can be seen that removing these tasks leads to a decrease in both S&R performance, indicating the effectiveness of these tasks in helping the LLM better understand item identifiers.

4.4 Experimental Analysis

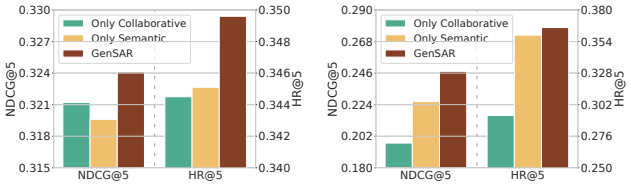
We conducted further experiments on the Commercial dataset to analyze the effectiveness of different modules in GenSAR.

Table 4: The *search* performance of different methods on the two datasets. Since search relies on semantic relevance, previous works [36, 50] that randomly sample negatives often produce overly easy examples, leading to inflated performance and poor model differentiation. To address this, we follow prior personalized search methods [1, 11] and use BM25 [34] to retrieve 99 harder negatives, forming a candidate list with the positive sample for more accurate evaluation.

Datasets	Metrics	Search							Joint Search and Recommendation			
		QEM	TEM	CoPPS	E5	BGE	DSI-QG	WebUltron	GenRet	JSR	UnifiedSSR	UniSAR	**GenSAR**
Amazon	HR@1	0.1512	0.0839	0.0943	0.3289	0.4030	0.3558	0.3432	0.4173	0.0835	0.0799	0.1122	**0.5262**
	HR@5	0.3101	0.3471	0.3380	0.5945	0.6264	0.5848	0.5464	0.6513	0.2407	0.2476	0.3129	**0.7529**
	HR@10	0.4657	0.5181	0.4909	0.7203	0.7475	0.6897	0.6216	0.7339	0.3463	0.3614	0.4333	**0.8217**
	NDCG@5	0.2311	0.2173	0.2154	0.4662	0.5219	0.4764	0.4507	0.5399	0.1623	0.1662	0.2143	**0.6485**
	NDCG@10	0.2809	0.2722	0.2647	0.5069	0.5613	0.5103	0.4748	0.5667	0.1962	0.2028	0.2533	**0.6710**
Commercial	HR@1	0.0311	0.0328	0.0265	**0.1277**	0.1267	0.1016	0.0804	0.1171	0.0273	0.0119	0.0511	0.1249
	HR@5	0.0870	0.1106	0.0998	0.3108	0.3184	0.2831	0.2619	0.3320	0.1202	0.0470	0.1810	**0.3655**
	HR@10	0.1539	0.1925	0.1792	0.4044	0.4194	0.4132	0.3992	0.4666	0.2137	0.0873	0.3231	**0.5250**
	NDCG@5	0.0586	0.0715	0.0626	0.2230	0.2258	0.1940	0.1721	0.2273	0.0728	0.0292	0.1144	**0.2472**
	NDCG@10	0.0799	0.0977	0.0880	0.2533	0.2584	0.2359	0.2164	0.2708	0.1026	0.0420	0.1597	**0.2987**

Table 5: Ablation study on the Commercial dataset, where "w/o" denotes the removal of the corresponding module in GenSAR.

Model	Recommendation		Search	
	HR@5	NDCG@5	HR@5	NDCG@5
GenSAR	**0.3496**	**0.3241**	**0.3655**	**0.2472**
w/o Behavior Token	0.3430	0.3193	0.3298	0.2224
w/o NRIP	0.0665	0.0392	0.3456	0.2342
w/o NSQP	0.3401	0.3163	0.3089	0.2053
w/o NSIP	0.3390	0.3152	0.1668	0.1113
w/o Desc2ID	0.3416	0.3188	0.3355	0.2278
w/o ID2Desc	0.3458	0.3220	0.3398	0.2308



(a) Recommendation Performance (b) Search Performance

Figure 4: Performance of GenSAR using different identifiers.

4.4.1 Impact of Different Identifier. To balance the semantic information needed for search and the collaborative information needed for recommendation, we designed the joint S&R identifier in Section 3.2. To validate its effectiveness, we compared it with identifiers learned directly from semantic embeddings or collaborative embeddings using RQ-VAE [33, 68], as shown in Figure 4. "Only Collaborative" represents using only collaborative embeddings, while "Only Semantic" represents using only semantic embeddings. The results show that identifiers derived solely from semantic or collaborative information lead to degraded performance. Furthermore,

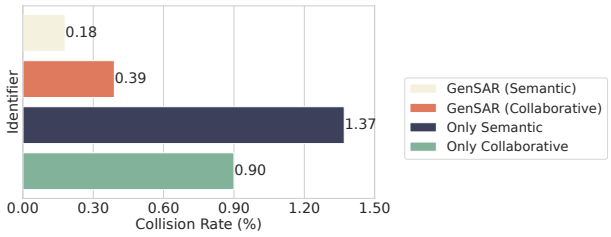


Figure 5: Collision rate of different identifiers.

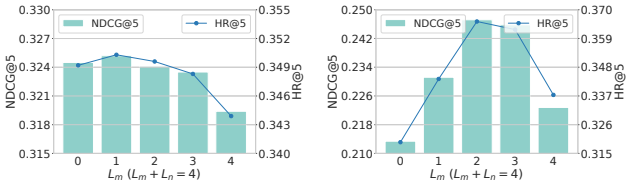
using only collaborative information results in worse search performance, which aligns with the fact that search relies more on semantic information.

4.4.2 Collision Rate of Different Identifier. Additionally, we analyzed the advantages of different identifiers from the perspective of collision rate. The formula for calculating the collision rate is as follows:

$$\text{Collision Rate} = (1 - \frac{\text{\# Unique Identifier}}{\text{\# Unique Item}}) * 100\%,$$

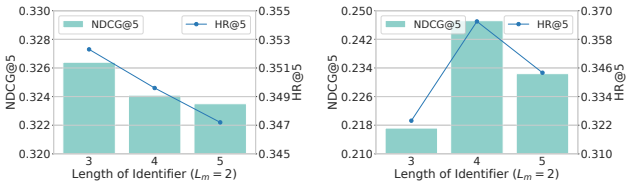
where # Unique Identifier represents the number of unique identifiers, and # Unique Item represents the number of unique items. Since RQ-VAE does not guarantee a unique identifier for each item during the learning process, collisions may occur where different items share the same identifier [33, 68]. A higher collision rate can negatively impact the model's performance. From Figure 5, it can be observed that the two identifiers assigned to each item in GenSAR, incorporating both semantic and collaborative information, have a lower collision rate of 0.18% and 0.39%, respectively. In contrast, identifiers derived solely from semantic embeddings or collaborative embeddings exhibit higher collision rates of 1.37% and 0.90%, respectively. This further validates the advantage of the identifiers in GenSAR, as their lower collision rate enables the model to achieve better performance.

4.4.3 Impact of Hyper-parameters. As described in Section 3.2, we have L_m-level shared codebooks and L_n-level specific codebooks. Here, we analyze the impact of the number of shared and specific



(a) Recommendation Performance (b) Search Performance

Figure 6: Performance under different numbers of shared codebooks L_m. We fix $L_m + L_n = 4$ and vary L_m to observe the results.



(a) Recommendation Performance (b) Search Performance

Figure 7: Performance under different length of the identifier. We fix $L_m = 2$ and vary L_n to adjust the identifier length.

codebooks (L_m and L_n) on the results, as shown in Figure 6. We fix $L_m + L_n = 4$ and observe the results. It can be seen that having too few ($L_m = 0$) or too many ($L_m = 4$) shared codebooks fails to achieve strong performance in both S&R. This indicates that L_m needs to be properly set so that the identifier can capture both the shared information between semantics and collaboration as well as their specific characteristics. Only in this way can we achieve better performance in both S&R.

Additionally, we analyzed the impact of identifier length on performance, as shown in Figure 7. We fix $L_m = 2$ and vary L_n to adjust the identifier length and observe the results. It can be seen that both shorter ($L_m + L_n = 3$) and longer ($L_m + L_n = 5$) identifiers lead to performance degradation. This is because, when the identifier is too short, the identifiers learned through RQ-VAE are more prone to collisions, resulting in a higher collision rate and making it difficult for the model to distinguish between different items. On the other hand, when the identifier is too long, the model requires more decoding steps during item generation, leading to accumulated errors and ultimately deteriorating performance. Therefore, it is essential to properly set the identifier length to achieve better performance.

5 Conclusion

In this paper, we propose GenSAR, which unifies balanced search and recommendation through generative retrieval to alleviate the trade-off between the two tasks and improve their performance. To balance the semantic information required for search and the collaborative information needed for recommendation, we design the joint S&R identifier and different training tasks. First, we learn two identifiers for each item to represent semantic and collaborative information, respectively. These identifiers share a common part to capture the information shared between semantics and collaboration while retaining distinct parts to preserve specific information. Second, we design different training tasks to help the model better understand the requirements of S&R tasks. We also validate the effectiveness of GenSAR through extensive experiments.

Acknowledgments

This work was funded by the National Key R&D Program of China (2023YFA1008704), the National Natural Science Foundation of China (No. 62472426, 62376275), Beijing Key Laboratory of Research on Large Models and Intelligent Governance, fund for building world-class universities (disciplines) of Renmin University of China, Kuaishou Technology.

References

[1] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. 2018. Multi-task learning for document ranking and query suggestion. In *International conference on learning representations*.

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

[3] 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*. 645–654.

[4] Jinheon Baek, Nirupama Chandrasekaran, Silviu Cucerzan, Allen Herring, and Sujay Kumar Jauhar. 2024. Knowledge-augmented large language models for personalized contextual query suggestion. In *Proceedings of the ACM on Web Conference 2024*. 3355–3366.

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

[6] 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*. 1521–1524.

[7] 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, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*. ACM, 380–389.

[8] Sunhao Dai, Changle Qu, Sirui Chen, Xiao Zhang, and Jun Xu. 2024. Recode: Modeling repeat consumption with neural ode. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2599–2603.

[9] Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. 2023. Uncovering chatgpt's capabilities in recommender systems. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 1126–1132.

[10] Sunhao Dai, Ninglu Shao, Jieming Zhu, Xiao Zhang, Zhenhua Dong, Jun Xu, Quanyu Dai, and Ji-Rong Wen. 2024. Modeling user attention in music recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 761–774.

[11] Chenlong Deng, Yujia Zhou, and Zhicheng Dou. 2022. Improving personalized search with dual-feedback network. In *Proceedings of the fifteenth ACM international conference on web search and data mining*. 210–218.

[12] 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)*.

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

[14] Yulong Gu, Wentian Bao, Dan Ou, Xiang Li, Baoliang Cui, Biyu Ma, Haikuan Huang, Qingwen Liu, and Xiaoyi Zeng. 2021. Self-supervised learning on users' spontaneous behaviors for multi-scenario ranking in e-commerce. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 3828–3837.

[15] 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*. 507–517.

[16] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[17] Zhankui He, Handong Zhao, Zhaowen Wang, Zhe Lin, Ajinkya Kale, and Julian Mcauley. 2022. Query-Aware Sequential Recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management* (Atlanta, GA, USA) *(CIKM '22)*. Association for Computing Machinery, New York, NY, USA, 4019–4023.

[18] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.).

[19] Wenyue Hua, Shuyuan Xu, Yingqiang Ge, and Yongfeng Zhang. 2023. How to index item ids for recommendation foundation models. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*. 195–204.

[20] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118* (2021).

[21] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 197–206.

[22] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. 2022. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11523–11532.

[23] Xiaoxi Li, Jiajie Jin, Yujia Zhou, Yuyao Zhang, Peitian Zhang, Yutao Zhu, and Zhicheng Dou. 2024. From matching to generation: A survey on generative information retrieval. *arXiv preprint arXiv:2404.14851* (2024).

[24] Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. 2023. Multiview Identifiers Enhanced Generative Retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 6636–6648.

[25] Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. 2024. Llara: Large language-recommendation assistant. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1785–1795.

[26] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 43–52.

[27] Gustavo Penha, Ali Vardasbi, Enrico Palumbo, Marco De Nadai, and Hugues Bouchard. 2024. Bridging Search and Recommendation in Generative Retrieval: Does One Task Help the Other?. In *Proceedings of the 18th ACM Conference on Recommender Systems*. 340–349.

[28] Weicong Qin, Zelin Cao, Weijie Yu, Zihua Si, Sirui Chen, and Jun Xu. 2024. Explicitly Integrating Judgment Prediction with Legal Document Retrieval: A Law-Guided Generative Approach. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2210–2220.

[29] Weicong Qin, Yi Xu, Weijie Yu, Chenglei Shen, Ming He, Jianping Fan, Xiao Zhang, and Jun Xu. 2025. MAPS: Motivation-Aware Personalized Search via LLM-Driven Consultation Alignment. *arXiv preprint arXiv:2503.01711* (2025).

[30] Weicong Qin, Yi Xu, Weijie Yu, Chenglei Shen, Xiao Zhang, Ming He, Jianping Fan, and Jun Xu. 2024. Enhancing Sequential Recommendations through Multi-Perspective Reflections and Iteration. *arXiv preprint arXiv:2409.06377* (2024).

[31] Weicong Qin, Yi Xu, Weijie Yu, Teng Shi, Chenglei Shen, Ming He, Jianping Fan, Xiao Zhang, and Jun Xu. 2025. Similarity= Value? Consultation Value Assessment and Alignment for Personalized Search. *arXiv preprint arXiv:2506.14437* (2025).

[32] Weicong Qin, Weijie Yu, Kepu Zhang, Haiyuan Zhao, Jun Xu, and Ji-Rong Wen. 2025. Uncertainty-aware evidential learning for legal case retrieval with noisy correspondence. *Information Sciences* (2025), 121915.

[33] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. 2023. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems* 36 (2023), 10299–10315.

[34] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.

[35] Chenglei Shen, Xiao Zhang, Teng Shi, Changshuo Zhang, Guofu Xie, and Jun Xu. 2024. A survey of controllable learning: Methods and applications in information retrieval. *arXiv preprint arXiv:2407.06083* (2024).

[36] Teng Shi, Zihua Si, Jun Xu, Xiao Zhang, Xiaoxue Zang, Kai Zheng, Dewei Leng, Yanan Niu, and Yang Song. 2024. UniSAR: Modeling User Transition Behaviors between Search and Recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1029–1039.

[37] Teng Shi, Jun Xu, Xiao Zhang, Xiaoxue Zang, Kai Zheng, Yang Song, and Han Li. 2025. Retrieval Augmented Generation with Collaborative Filtering for Personalized Text Generation. *arXiv preprint arXiv:2504.05731* (2025).

[38] Zihua Si, Xueran Han, Xiao Zhang, Jun Xu, Yue Yin, Yang Song, and Ji-Rong Wen. 2022. A Model-Agnostic Causal Learning Framework for Recommendation Using Search Data. In *Proceedings of the ACM Web Conference 2022* (Virtual Event, Lyon, France) *(WWW '22)*. Association for Computing Machinery, New York, NY, USA, 224–233.

[39] Zihua Si, Zhongxiang Sun, Xiao Zhang, Jun Xu, Xiaoxue Zang, Yang Song, Kun Gai, and Ji-Rong Wen. 2023. When search meets recommendation: Learning disentangled search representation for recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1313–1323.

[40] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (Beijing, China) *(CIKM '19)*. ACM, New York, NY, USA, 1441–1450.

[41] Weiwei Sun, Lingyong Yan, Zheng Chen, Shuaiqiang Wang, Haichao Zhu, Pengjie Ren, Zhumin Chen, Dawei Yin, Maarten Rijke, and Zhaochun Ren. 2024. Learning to tokenize for generative retrieval. *Advances in Neural Information Processing Systems* 36 (2024).

[42] Jiakai Tang, Sunhao Dai, Teng Shi, Jun Xu, Xu Chen, Wen Chen, Wu Jian, and Yuning Jiang. 2025. Think before recommend: Unleashing the latent reasoning power for sequential recommendation. *arXiv preprint arXiv:2503.22675* (2025).

[43] Jiakai Tang, Sunhao Dai, Zexu Sun, Xu Chen, Jun Xu, Wenhui Yu, Lantao Hu, Peng Jiang, and Han Li. 2024. Towards Robust Recommendation via Decision Boundary-aware Graph Contrastive Learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2854–2865.

[44] Yi Tay, Vinh Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, et al. 2022. Transformer memory as a differentiable search index. *Advances in Neural Information Processing Systems* 35 (2022), 21831–21843.

[45] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672* (2024).

[46] Yuening Wang, Man Chen, Yaochen Hu, Wei Guo, Yingxue Zhang, Huifeng Guo, Yong Liu, and Mark Coates. 2024. Enhancing Click-through Rate Prediction in Recommendation Domain with Search Query Representation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 2462–2471.

[47] Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao, Shibin Wu, Qi Chen, Yuqing Xia, Chengmin Chi, Guoshuai Zhao, Zheng Liu, et al. 2022. A neural corpus indexer for document retrieval. *Advances in Neural Information Processing Systems* 35 (2022), 25600–25614.

[48] Yu Wang, Zhengyang Wang, Hengrui Zhang, Qingyu Yin, Xianfeng Tang, Yinghan Wang, Danqing Zhang, Limeng Cui, Monica Cheng, Bing Yin, et al. 2023. Exploiting intent evolution in e-commercial query recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 5162–5173.

[49] Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 641–649.

[50] Jiayi Xie, Shang Liu, Gao Cong, and Zhenzhong Chen. 2024. UnifiedSSR: A Unified Framework of Sequential Search and Recommendation. In *Proceedings of the ACM on Web Conference 2024*. 3410–3419.

[51] 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* (2020).

[52] Yi Xu, Weicong Qin, Weijie Yu, Ming He, Jianping Fan, and Jun Xu. 2025. Decoding Recommendation Behaviors of In-Context Learning LLMs Through Gradient Descent. *arXiv preprint arXiv:2504.04386* (2025).

[53] Jing Yao, Zhicheng Dou, Ruobing Xie, Yanxiong Lu, Zhiping Wang, and Ji-Rong Wen. 2021. USER: A Unified Information Search and Recommendation Model Based on Integrated Behavior Sequence. In *Proceedings of the 30th ACM International Conference on Information J& Knowledge Management* (Virtual Event, Queensland, Australia) *(CIKM '21)*. Association for Computing Machinery, New York, NY, USA, 2373–2382.

[54] Zheng Yuan, Fajie Yuan, Yu Song, Youhua Li, Junchen Fu, Fei Yang, Yunzhu Pan, and Yongxin Ni. 2023. Where to go next for recommender systems? id- vs. modality-based recommender models revisited. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2639–2649.

[55] Zhenrui Yue, Yueqi Wang, Zhankui He, Huimin Zeng, Julian McAuley, and Dong Wang. 2024. Linear recurrent units for sequential recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. 930–938.

[56] Hamed Zamani and W. Bruce Croft. 2018. Joint Modeling and Optimization of Search and Recommendation. In *Proceedings of the First Biennial Conference on Design of Experimental Search & Information Retrieval Systems, Bertinoro, Italy, August 28-31, 2018 (CEUR Workshop Proceedings, Vol. 2167)*. CEUR-WS.org, 36–41.

[57] Hamed Zamani and W. Bruce Croft. 2020. Learning a Joint Search and Recommendation Model from User-Item Interactions. In *Proceedings of the 13th International Conference on Web Search and Data Mining* (Houston, TX, USA) *(WSDM '20)*. Association for Computing Machinery, New York, NY, USA, 717–725.

[58] Changshuo Zhang, Sirui Chen, Xiao Zhang, Sunhao Dai, Weijie Yu, and Jun Xu. 2024. Reinforcing Long-Term Performance in Recommender Systems with User-Oriented Exploration Policy. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1850–1860.

[59] Changshuo Zhang, Teng Shi, Xiao Zhang, Qi Liu, Ruobing Xie, Jun Xu, and Ji-Rong Wen. 2024. Modeling Domain and Feedback Transitions for Cross-Domain Sequential Recommendation. *arXiv preprint arXiv:2408.08209* (2024).

[60] Changshuo Zhang, Teng Shi, Xiao Zhang, Yanping Zheng, Ruobing Xie, Qi Liu, Jun Xu, and Ji-Rong Wen. 2024. QAGCF: Graph Collaborative Filtering for Q&A Recommendation. *arXiv preprint arXiv:2406.04828* (2024).

[61] Changshuo Zhang, Xiao Zhang, Teng Shi, Jun Xu, and Ji-Rong Wen. 2025. Test-Time Alignment for Tracking User Interest Shifts in Sequential Recommendation. *arXiv preprint arXiv:2504.01489* (2025).

[62] Kepu Zhang, Teng Shi, Sunhao Dai, Xiao Zhang, Yinfeng Li, Jing Lu, Xiaoxue Zang, Yang Song, and Jun Xu. 2024. SAQRec: Aligning Recommender Systems to User Satisfaction via Questionnaire Feedback. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 3165–3175.

[63] Xiao Zhang, Teng Shi, Jun Xu, Zhenhua Dong, and Ji-Rong Wen. 2024. Model-Agnostic Causal Embedding Learning for Counterfactually Group-Fair Recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2024).

[64] Yuting Zhang, Yiqing Wu, Ruidong Han, Ying Sun, Yongchun Zhu, Xiang Li, Wei Lin, Fuzhen Zhuang, Zhulin An, and Yongjun Xu. 2024. Unified Dual-Intent Translation for Jont Modeling of Search and Recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 6291–6300.

[65] Jujia Zhao, Wenjie Wang, Chen Xu, Xiuying Chen, Zhaochun Ren, and Suzan Verberne. 2025. Unifying Search and Recommendation: A Generative Paradigm Inspired by Information Theory. *arXiv preprint arXiv:2504.06714* (2025).

[66] Kai Zhao, Yukun Zheng, Tao Zhuang, Xiang Li, and Xiaoyi Zeng. 2022. Joint Learning of E-Commerce Search and Recommendation with a Unified Graph Neural Network. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining* (Virtual Event, AZ, USA) *(WSDM '22)*. Association for Computing Machinery, New York, NY, USA, 1461–1469.

[67] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223* (2023).

[68] Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. 2024. Adapting large language models by integrating collaborative semantics for recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 1435–1448.

[69] Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Filter-Enhanced MLP is All You Need for Sequential Recommendation. In *Proceedings of the ACM Web Conference 2022* (Virtual Event, Lyon, France) *(WWW '22)*. Association for Computing Machinery, New York, NY, USA, 2388–2399.

[70] Yujia Zhou, Jing Yao, Ledell Wu, Zhicheng Dou, and Ji-Rong Wen. 2023. WebUltron: An Ultimate Retriever on Webpages Under the Model-Centric Paradigm. *IEEE Transactions on Knowledge and Data Engineering* (2023).

[71] Shengyao Zhuang, Houxing Ren, Linjun Shou, Jian Pei, Ming Gong, Guido Zuccon, and Daxin Jiang. 2022. Bridging the gap between indexing and retrieval for differentiable search index with query generation. *arXiv preprint arXiv:2206.10128* (2022).