# Sparse Meets Dense: Unified Generative Recommendations with Cascaded Sparse-Dense Representations

Yuhao Yang yangyuhao01@baidu.com Baidu Inc. Beijing, China

> Yi Li liyi01@baidu.com Baidu Inc. Beijing, China

Kai Chen chenkai23@baidu.com Baidu Inc. Beijing, China Zhi Ji jizhi@baidu.com Baidu Inc. Beijing, China

Zhonglin Mo mozhonglin@baidu.com Baidu Inc. Beijing, China

Zijian Zhang zhangzijian02@baidu.com Baidu Inc. Beijing, China Zhaopeng Li lizhaopeng@baidu.com Baidu Inc. Beijing, China

Yue Ding dingyue03@baidu.com Baidu Inc. Beijing, China

Jie Li lijie06@baidu.com Baidu Inc. Beijing, China

Shuanglong Li lishuanglong@baidu.com Baidu Inc. Beijing, China Lin Liu liulin03@baidu.com Baidu Inc. Beijing, China

# **Abstract**

Generative models have recently gained attention in recommendation systems by directly predicting item identifiers from user interaction sequences. However, existing methods suffer from significant information loss due to the separation of stages such as quantization and sequence modeling, hindering their ability to achieve the modeling precision and accuracy of sequential dense retrieval techniques. Integrating generative and dense retrieval methods remains a critical challenge. To address this, we introduce the Cascaded Organized Bi-Represented generAtive retrieval (COBRA) framework, which innovatively integrates sparse semantic IDs and dense vectors through a cascading process. Our method alternates between generating these representations by first generating sparse IDs, which serve as conditions to aid in the generation of dense vectors. End-to-end training enables dynamic refinement of dense representations, capturing both semantic insights and collaborative signals from user-item interactions. During inference, COBRA employs a coarse-to-fine strategy, starting with sparse ID generation and refining them into dense vectors via the generative model. We further propose BeamFusion, an innovative approach combining

beam search with nearest neighbor scores to enhance inference flexibility and recommendation diversity. Extensive experiments on public datasets and offline tests validate our method's robustness. Online A/B tests on a real-world advertising platform with over 200 million daily users demonstrate substantial improvements in key metrics, highlighting COBRA's practical advantages.

## **CCS Concepts**

• Information systems  $\rightarrow$  Recommender systems.

#### **ACM Reference Format:**

## 1 Introduction

Recommendation systems are vital components of modern digital ecosystems, providing personalized item suggestions that align with user preferences across e-commerce platforms, streaming services, and social networks [5, 12, 31]. Recent advancements have focused on sequential recommendation methods, which leverage the sequential nature of user interactions to enhance recommendation performance [4, 9, 36, 51]. Notable models like SASRec [18] and BERT4Rec [37] have demonstrated the effectiveness of sequence models in capturing user behavior patterns.

The emergence of generative models has further expanded the capabilities of recommendation systems [2, 47, 50]. Unlike traditional sequential recommendation methods, generative models can directly predict target items based on user behavior sequences [6,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference'17, Washington, DC, USA

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 978-x-xxxx-x/YYYY/MM https://doi.org/10.1145/nnnnnnnnnnnn

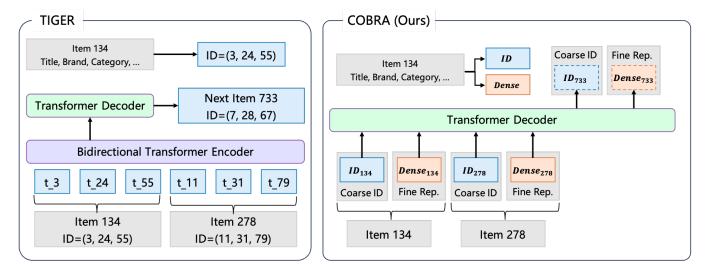


Figure 1: Comparison of generative recommendation paradigms. The left section illustrates traditional generative retrieval approaches, exemplified by TIGER, which utilize a sequence of sparse IDs as input within a Transformer encoder-decoder architecture to directly predict the sparse ID of the next item. The right section depicts the proposed COBRA framework, which employs Cascaded Organized Bi-Represented generAtive retrieval. This approach integrates sparse IDs to capture coarse-grained semantic information and dense vectors to encapsulate fine-grained detail. The cascaded representation is processed by a Transformer decoder that sequentially predicts the sparse ID followed by the dense vector.

35, 56]. These models handle complex user-item interactions and offer emerging abilities such as reasoning and few-shot learning, which significantly improve recommendation accuracy and diversity [3, 41, 43]. Among these, TIGER [33] is a pioneering approach in generative retrieval for recommendation systems. As depicted in Figure 1(Lower Left), TIGER leverages a Residual Quantized Variational AutoEncoder (RQ-VAE) [20] to encode item content features into hierarchical semantic IDs, allowing the model to share knowledge across semantically similar items without the need for individual item embeddings. Beyond TIGER, several other methods have been proposed to further explore the integration of generative models with recommendation systems. LC-Rec [53] aligns semantic and collaborative information using RQ-VAE with a series of alignment tasks. ColaRec [45] combines collaborative filtering signals with content information by deriving generative identifiers from a pretrained recommendation model. IDGenRec [38] leverages large language models to generate unique, concise, and semantically rich textual identifiers for recommended items, showing strong potential in zero-shot settings.

Despite these innovations, existing generative recommendation methods still face several challenges compared to sequential dense retrieval methods [7, 23]. Sequential dense retrieval methods, which rely on dense embeddings for each item, offer high accuracy and robustness but require substantial storage and computational resources. In contrast, generative methods, while efficient, often struggle with fine-grained similarity modeling [48]. To effectively leverage the strengths of both retrieval paradigms, we propose Cascaded Organized Bi-Represented generAtive retrieval (COBRA), a framework that synergizes generative and dense retrieval. Figure 1(Right) illustrates the cascaded sparse-dense representations in COBRA. The proposed method introduces a cascaded generative retrieval

framework alternating between generating sparse IDs and dense vectors. This approach mitigates information loss inherent in IDbased methods. Specifically, COBRA's input is a sequence of cascaded representations composed of sparse IDs and dense vectors corresponding to items in the user's interaction history. During training, the dense representations are learned through contrastive learning objectives in and end-to-end manner. By first generating the sparse ID and then the dense representation, COBRA reduces the learning difficulty of dense representations and promotes mutual learning between the two representations. During inference, COBRA employs a coarse-to-fine generation process, starting with sparse ID that provides a high-level categorical sketch capturing the categorical essence of the item. The generated ID is then appended to the input sequence and fed back into the model to predict the dense vector that captures the fine-grained details, enabling more precise and personalized recommendations. To ensure flexible inference, we introduce BeamFusion, a sampling technique combining beam search with nearest neighbor retrieval scores, ensuring controllable diversity in the retrieved items. Unlike TIGER, which relies solely on sparse IDs, COBRA harnesses the strengths of both sparse and dense representations.

Our main contributions are as follows:

 Cascaded Bi-Represented Retrieval Framework: We introduce COBRA, a novel cascading framework that alternates between generating sparse semantic IDs and dense vectors. By incorporating dense representations into the ID sequence, COBRA supplements the information loss inherent in ID-based methods. Using sparse IDs as conditions for generating dense vectors reduces the learning difficulty of dense representations.

- Learnable Dense Representations via End-to-End Training: COBRA leverages the original item data as input to generate dense representations through end-to-end training. Unlike static embeddings, COBRA's dense vectors are dynamically learned, capturing semantic information and fine-grained details.
- Coarse-to-Fine Generation Process: During inference, COBRA employs a coarse-to-fine generation process. It first generates sparse IDs, which are then fed back into the model to produce refined dense representations, enhancing the granularity of the dense vectors. Additionally, we propose BeamFusion for flexible and diverse recommendation.
- Comprehensive Empirical Validation: Through extensive experiments on multiple benchmark datasets, we demonstrate that COBRA achieves superior performance in terms of recommendation accuracy compared to existing state-of-the-art methods. These results validate the effectiveness of COBRA and its ability to balance precision and diversity in recommendation tasks.

#### 2 Related Work

Sequential Dense Recommendation. Sequential dense recommendation systems leverage user interaction sequences to learn dense representations for users and items [18, 21, 55], capturing both long-term preferences and short-term dynamics [10, 22, 32, 40]. Early models such as GRU4Rec [14] utilized Recurrent Neural Networks (RNNs) [49] to capture temporal dependencies in user behavior. Caser [39] applied Convolutional Neural Networks (CNNs) [54] to treat sequences as "images" for spatial feature extraction. The advent of Transformer-related models, such as SASRec [18] and BERT4Rec [37], has significantly advanced this field. These models employ self-attention mechanisms to capture complex user behaviors, with SASRec focusing on autoregressive tasks and BERT4Rec on bidirectional context modeling. More advanced models like PinnerFormer [30] and FDSA [52] have further enhanced user representation by leveraging Transformers for long-term behavior modeling and feature integration. Recent works, including ZESRec [8], UniSRec [15], and RecFormer [21], have emphasized cross-domain transferability by incorporating textual features and employing contrastive learning techniques. RecFormer, in particular, has unified language understanding and sequence recommendation through bidirectional Transformers.

Generative Recommendation. The increasing popularity of generative models across various domains has led to a paradigm shift in recommendation systems from discriminative to generative models [1, 17, 24, 25, 27, 46]. Generative models directly generate item identifiers rather than computing ranking scores for each item [16, 26, 33, 44]. P5 [11] has transformed various recommendation tasks into natural language sequences, providing a universal framework for recommendation completion through unique training objectives and prompts. TIGER [33] pioneered the application of generative retrieval to recommendations by using a residual quantized autoencoder to create semantically rich indexing identifiers. These identifiers are then utilized by a transformer-based model to generate item identifiers from user histories. LC-Rec [53] enhanced this approach by aligning semantic identifiers with collaborative

filtering techniques through additional alignment tasks. IDGen-Rec [38] merged generative systems with large language models to generate unique, semantically dense textual identifiers, demonstrating strong performance even in zero-shot settings. SEATER [34] focused on maintaining semantic consistency through balanced k-ary tree-structured indexes refined by contrastive and multi-task learning. ColaRec [45] aligned content-based semantic spaces with collaborative interaction spaces to improve recommendation efficacy. However, existing generative methods face several challenges. For instance, methods based on discrete IDs may lack fine-grained details and suffer from information loss, which can limit their ability to accurately capture user preferences [42]. Additionally, approaches that rely on natural language may struggle to align linguistic expressions with the requirements of recommendation tasks, potentially leading to suboptimal performance [19]. To address these issues, LIGER [48] proposes a hybrid model that combines advantages of generative and dense retrieval methods. It simultaneously generates sparse IDs and dense representations, treating them as complementary representations of the same object granularity. This hybrid approach effectively narrows the gap between generative and dense retrieval methods to some extent. However, LIGER's IDs and dense representations share the same granularity, and the dense representations are pre-trained and fixed. Therefore, how to more flexibly combine generative and dense retrieval methods is still an open question that needs further exploration.

# 3 Methodology

This section introduces the Cascaded Organized Bi-Represented generAtive Retrieval (COBRA) framework, which integrates cascaded sparse-dense representations and coarse-to-fine generation to enhance recommendation performance. Figure 2 illustrates the overall framework of COBRA.

## 3.1 Sparse-Dense Representation

3.1.1 Sparse Representation. COBRA generates sparse IDs using a Residual Quantized Variational Autoencoder (RQ-VAE), inspired by the approach in TIGER [33]. For each item, we extract its attributes to generate a textual description, which is embedded into a dense vector space and quantized to produce sparse IDs. These IDs capture the categorical essence of items, forming the basis for subsequent processing. For the sake of brevity, the subsequent methodology descriptions will assume that the sparse ID consists of a single level. However, it should be noted that this approach can be easily extended to accommodate scenarios involving multiple levels.

3.1.2 Dense Representation. To capture nuanced attribute information, we develop an end-to-end trainable dense encoder, encoding item textual contents. Each item's attributes are flattened into a text sentence, prefixed with a [CLS] token, and fed into a Transformer-based text encoder **Encoder**. The dense representation  $\mathbf{v}_t$  is extracted from the output corresponding to the [CLS] token, capturing fine-grained details of the item's textual content. As illustrated in the lower part of Figure 2, we incorporate position embeddings and type embeddings to model the positional and context of tokens within the sequence. These embeddings are added to the token embeddings in an additive manner, enhancing the

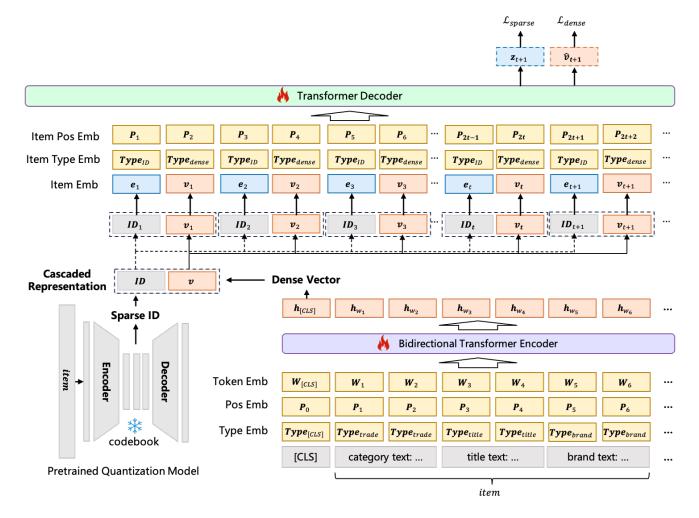


Figure 2: The architecture of COBRA. The model employs a cascaded sparse-dense representation approach, where sparse IDs are generated via Residual Quantization and dense vectors are produced by a trainable Transformer Encoder. These representations serve as inputs to a Transformer Decoder, which alternates between predicting sparse IDs and dense vectors. The predicted outputs are used to compute the loss functions  $\mathcal{L}_{\text{sparse}}$  and  $\mathcal{L}_{\text{dense}}$ . For the sake of simplicity, the figure illustrates an example with a single level of sparse ID.

model's ability to distinguish between different tokens and their positions in the sequence.

3.1.3 Cascaded Representation. The cascaded representation integrates sparse IDs and dense vectors within a unified generative model. Specifically, for each item, we combine its sparse ID  $ID_t$  and dense vector  $\mathbf{v}_t$  to form a cascaded representation ( $ID_t$ ,  $\mathbf{v}_t$ ). This approach leverages the strengths of both representations, providing a more comprehensive characterization of items: sparse IDs provide a stable categorical foundation through discrete constraints, while dense vectors maintain continuous feature resolution, ensuring that the model captures both high-level semantics and fine-grained details.

#### 3.2 Sequential Modeling

3.2.1 Probabilistic Decomposition. The probability distribution modeling of the target item is factorized into two stages, leveraging the complementary strengths of sparse and dense representations. Specifically, instead of directly predicting the next item  $s_{t+1}$  based on the historical interaction sequence  $S_{1:t}$ , COBRA predicts the sparse ID  $ID_{t+1}$  and the dense vector  $\mathbf{v}_{T+1}$  separately:

$$P(ID_{t+1}, \mathbf{v}_{t+1}|S_{1:t}) = P(ID_{t+1}|S_{1:t})P(\mathbf{v}_{t+1}|ID_{t+1}, S_{1:t})$$
(1)

where  $P(ID_{t+1}|S_{1:t})$  represents the probability of generating the sparse ID  $ID_{t+1}$  based on the historical sequence  $S_{1:t}$ , capturing the categorical essence of the next item.  $P(\mathbf{v}_{t+1}|ID_{t+1},S_{1:t})$  represents the probability of generating the dense vector  $\mathbf{v}_{t+1}$  given the sparse ID  $ID_{t+1}$  and the historical sequence  $S_{1:t}$ , capturing the fine-grained details of the next item. This decomposition allows COBRA to

leverage both the categorical information provided by sparse IDs and the fine-grained details captured by dense vectors.

3.2.2 Sequential Modeling with a Unified Generative Model. For sequential modeling, we utilize a unified generative model based on the Transformer architecture to effectively capture sequential dependencies in user-item interactions. The Transformer receives an input sequence of cascaded representations, with each item represented by its sparse ID and dense vector.

**Embedding Sparse IDs** The sparse ID, denoted as  $ID_t$ , is transformed into a dense vector space through an embedding layer:  $\mathbf{e}_t = \mathbf{Embed}(ID_t)$ . This embedding  $\mathbf{e}_t$  is concatenated with the dense vector  $\mathbf{v}_t$  to form the model's input at each time step:

$$\mathbf{h}_t = [\mathbf{e}_t; \mathbf{v}_t] \tag{2}$$

Transformer Modeling Our Transformer Decoder model comprises multiple layers, each featuring self-attention mechanisms and feedforward networks. As depicted in the upper part of Figure 2, the input sequence to the Decoder consists of cascaded representations. To enhance modeling of sequential and contextual information, these representations are augmented with item position and type embeddings. For brevity, mathematical formulations in the following sections focus on the cascaded sequence representation, omitting explicit notation for position and type embeddings. The Decoder processes this enriched input to generate contextualized representations for predicting the subsequent sparse ID and dense vector.

**Sparse ID Prediction** Given history interaction sequence  $S_{1:t}$ , to predict the sparse ID  $ID_{t+1}$ , the Transformer input sequence is:

$$S_{1:t} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_t]$$
  
=  $[\mathbf{e}_1, \mathbf{v}_1, \mathbf{e}_2, \mathbf{v}_2, \dots, \mathbf{e}_t, \mathbf{v}_t]$  (3)

where each  $\mathbf{h}_i$  is a concatenation of the sparse ID embedding and the dense vector for the i-th item. The Transformer processes this sequence to generate contextualized representations, subsequently used to predict the next sparse ID and dense vector. Specifically, the Transformer decoder processes the sequence  $\mathbf{S}_{1:t}$ , producing a sequence of vectors  $\mathbf{y}_t = \mathbf{TransformerDecoder}(\mathbf{S}_{1:t})$ . The logits for sparse ID prediction are derived as:

$$\mathbf{z}_{t+1} = \mathbf{SparseHead}(\mathbf{y}_t) \tag{4}$$

where  $\mathbf{z}_{t+1}$  represents the logits for the predicted sparse ID  $ID_{t+1}$ . **Dense Vector Prediction** For predicting the dense vector  $\mathbf{v}_{t+1}$ , the Transformer input sequence is:

$$\bar{\mathbf{S}}_{1:t} = [\mathbf{S}_{1:t}, \mathbf{e}_{t+1}] 
= [\mathbf{e}_1, \mathbf{v}_1, \mathbf{e}_2, \mathbf{v}_2, \dots, \mathbf{e}_t, \mathbf{v}_t, \mathbf{e}_{t+1}]$$
(5)

The Transformer decoder processes  $\bar{S}_{1:t}$  to output the predicted dense vector:

$$\hat{\mathbf{v}}_{t+1} = \mathbf{TransformerDecoder}(\bar{\mathbf{S}}_{1:t})$$
 (6)

# 3.3 End-to-End Training

In COBRA, the end-to-end training process is designed to optimize both sparse and dense representation prediction jointly. The training process is governed by a composite loss function that combines losses for sparse ID prediction and dense vector prediction. The sparse ID prediction loss, denoted as  $\mathcal{L}_{\text{sparse}}$ , ensures the model's proficiency in predicting the next sparse ID based on the historical sequence  $S_{1:t}$ :

$$\mathcal{L}_{\text{sparse}} = -\sum_{t=1}^{T-1} \log \left( \frac{\exp(z_{t+1}^{ID_{t+1}})}{\sum_{j=1}^{C} \exp(z_{t+1}^{j})} \right)$$
(7)

where T is the length of the historical sequence,  $ID_{t+1}$  is the sparse ID corresponding to interacted item at time step t+1,  $z_{t+1}^{ID_{t+1}}$  represents the predicted logit of groundtruth sparse ID  $ID_{t+1}$  at time step t+1, generated by the Transformer Decoder, and C denotes set of all sparse IDs.

The dense vector prediction loss  $\mathcal{L}_{dense}$  focuses on refining the dense vectors, enabling them to discern between similar and dissimilar items. The loss is defined as:

$$\mathcal{L}_{\text{dense}} = -\sum_{t=1}^{T-1} \log \frac{\exp(\cos(\hat{\mathbf{v}}_{t+1} \cdot \mathbf{v}_{t+1}))}{\sum_{item_j \in \text{Batch}} \exp(\cos(\hat{\mathbf{v}}_{t+1}, \mathbf{v}_{item_j}))}$$
(8)

where  $\hat{\mathbf{v}}_t$  is the predicted dense vector,  $\mathbf{v}_t$  is the ground truth dense vector for the positive item, and  $\mathbf{v}_j$  represents the dense vectors of items within the batch. The term  $\cos(\hat{\mathbf{v}}_{t+1}\cdot\mathbf{v}_{t+1})$  represents the cosine similarity between the predicted and ground truth dense vectors. A higher cosine similarity indicates that the vectors are more similar in direction, which is desirable for accurate dense vector prediction. The dense vectors are generated by an end-to-end trainable encoder **Encoder**, which is optimized during the training process. This ensures that the dense vectors are dynamically refined and adapted to the specific requirements of the recommendation task

The overall loss function is formulated as:

$$\mathcal{L} = \mathcal{L}_{\text{sparse}} + \mathcal{L}_{\text{dense}} \tag{9}$$

The dual-objective loss function enables a balanced optimization process, where the model dynamically refines dense vectors guided by sparse IDs. This end-to-end training approach captures both high-level semantics and feature-level information, optimizing sparse and dense representations jointly for superior performance.

# 3.4 Coarse-to-Fine Generation

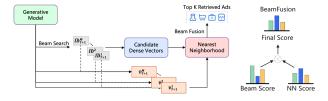


Figure 3: Illustration of the Coarse-to-Fine Generation process. During inference, M sparse IDs are generated via Beam Search, and appended to the sequence. Dense vectors are then generated and used in ANN to obtain candidate items. Beam-Fusion combines beam scores and similarity scores to rank candidates, from which the top K items are selected.

During the inference phase, COBRA implements the coarse-tofine generation procedure, involving the sequential generation of sparse IDs followed by the refinement of dense vectors in a cascaded manner, as illustrated in Figure 3. The coarse-to-fine generation process in COBRA is designed to capture both the categorical essence and fine-grained details of user-item interactions. This process involves two main stages:

**Sparse ID Generation**: Given a user sequence  $S_{1:T}$ , we utilize the ID probability distribution modeled by the Transformer Decoder,  $\hat{ID}_{T+1} \sim P(i_{T+1}|S_{1:T})$ , and employ the BeamSearch algorithm to derive the top M IDs. The formulation is as follows:

$$\{\hat{\mathbf{ID}}_{T+1}^k\}_{k=1}^M = \text{BeamSearch}(\mathbf{TransformerDecoder}(\mathbf{S}_{1:T}), M)$$

where  $k \in \{1,2,\ldots,M\}$ . Each generated ID is associated with a beam score  $\phi_{\hat{\Pi}\hat{D}^k_{T+1}}$ .

**Dense Vector Refinement**: Each generated sparse ID  $\hat{\mathbf{ID}}_{T+1}^k$  is subsequently converted into an embedding and appended to the previous cascaded sequence embedding  $\mathbf{S}_{1:T}$ . Then the corresponding dense vector  $\hat{\mathbf{v}}_{T+1}^k$  is generated:

$$\hat{\mathbf{v}}_{T+1}^{k} = \mathbf{TransformerDecoder}([\mathbf{S}_{1:T}, \mathbf{Embed}(\hat{\mathbf{ID}}_{T+1}^{k})]) \quad (11)$$

After that, we employ Approximate Nearest Neighbor (ANN) search to retrieve the top N candidate items:

$$\mathcal{A}_k = \text{ANN}(\hat{\mathbf{ID}}_{T+1}^k, C(\hat{\mathbf{ID}}_{T+1}^k), N)$$
 (12)

where  $C(\hat{\mathbf{ID}}_{T+1}^k)$  is the set of candidate items associated with sparse ID  $\hat{\mathbf{ID}}_{T+1}^k$ , and N represents the number of top items to be retrieved.

**BeamFusion Mechanism** In order to achieve a balance between precision and diversity, we devise a globally comparable score for items corresponding to each sparse ID. This score is capable of reflecting both the differences among different sparse IDs and the fine-grained difference among items under the same sparse ID. To accomplish this, we propose the BeamFusion mechanism:

$$\Phi^{(\hat{\mathbf{v}}_{T+1}^k,\mathbf{I}\hat{\mathbf{D}}_{T+1}^k,\mathbf{a})} = \mathrm{Softmax}(\tau\phi_{\hat{\mathbf{I}}\hat{\mathbf{D}}_{T+1}^k}) \times \mathrm{Softmax}(\psi\cos(\hat{\mathbf{v}}_{T+1}^k,\mathbf{a})) \tag{13}$$

where a represents the candidate item,  $\tau$  and  $\psi$  are coefficients, and  $\phi_{\hat{\Pi}\hat{D}^k_{T+1}}$  denotes the beam score obtained during the beam search process.

Finally, we rank all candidate items based on their BeamFusion Scores and select the top K items as the final recommendations:

$$\mathcal{R} = \text{TopK}\left(\bigcup_{k=1}^{M} \mathcal{A}_k, \Phi, K\right)$$
 (14)

where  $\mathcal{R}$  denotes the set of final recommendations, and TopK represents the operation of selecting the top K items with the highest BeamFusion Scores.

#### 4 Experiment

This section presents a comprehensive evaluation of the COBRA framework using both public and industrial datasets. Our experiments focus on assessing COBRA's ability to improve recommendation accuracy and diversity, while also validating its practical effectiveness through offline and online evaluations.

Dataset	# Users	# Items	Sequence Length		
			Mean	Median	
Beauty	22,363	12,101	8.87	6	
Sports and Outdoors	35,598	18,357	8.32	6	
Toys and Games	19,412	11,924	8.63	6	

**Table 1: Dataset Statistics** 

Table 2: Performance comparison on public datasets. The best metric for each dataset is highlighted in bold, while the second-best is underlined.

	Method	R@5	N@5	R@10	N@10
Beauty	P5	0.0163	0.0107	0.0254	0.0136
	Caser	0.0205	0.0131	0.0347	0.0176
	HGN	0.0325	0.0206	0.0512	0.0266
	GRU4Rec	0.0164	0.0099	0.0283	0.0137
	BERT4Rec	0.0203	0.0124	0.0347	0.0170
	FDSA	0.0267	0.0163	0.0407	0.0208
	SASRec	0.0387	0.0249	0.0605	0.0318
	S <sup>3</sup> -Rec	0.0387	0.0244	0.0647	0.0327
	TIGER	0.0454	0.0321	0.0648	0.0384
	COBRA[Ours]	0.0537	0.0395	0.0725	0.0456
	P5	0.0061	0.0041	0.0095	0.0052
	Caser	0.0116	0.0072	0.0194	0.0097
	HGN	0.0189	0.0120	0.0313	0.0159
Sports	GRU4Rec	0.0129	0.0086	0.0204	0.0110
	BERT4Rec	0.0115	0.0075	0.0191	0.0099
	FDSA	0.0182	0.0122	0.0288	0.0156
	SASRec	0.0233	0.0154	0.0350	0.0192
	S <sup>3</sup> -Rec	0.0251	0.0161	0.0385	0.0204
	TIGER	0.0264	0.0181	0.0400	0.0225
	COBRA[Ours]	0.0305	0.0215	0.0434	0.0257
	P5	0.0070	0.0050	0.0121	0.0066
	Caser	0.0166	0.0107	0.0270	0.0141
	HGN	0.0321	0.0221	0.0497	0.0277
S	GRU4Rec	0.0097	0.0059	0.0176	0.0084
Toys	BERT4Rec	0.0116	0.0071	0.0203	0.0099
	FDSA	0.0228	0.0140	0.0381	0.0189
	SASRec	0.0463	0.0306	0.0675	0.0374
	S <sup>3</sup> -Rec	0.0443	0.0294	0.0700	0.0376
	TIGER	0.0521	0.0371	0.0712	0.0432
	COBRA[Ours]	0.0619	0.0462	0.0781	0.0515

## 4.1 Public Dataset Experiments

4.1.1 Datasets and Evaluation Metrics. In our experiments, we evaluate the performance of COBRA using the Amazon Product Reviews dataset [13, 29], which is a well-established benchmark for recommendation tasks. This dataset encompasses product reviews and associated metadata collected from May 1996 to September 2014. Our analysis focuses on three specific subsets: "Beauty," "Sports and Outdoors," and "Toys and Games." For each subset, we construct item embeddings leveraging attributes such as title, price, category, and description. To ensure data quality, we apply a 5-core filtering

Method	R@50	R@100	R@200	R@500	R@800
COBRA	0.1180	0.1737	0.2470	0.3716	0.4466
COBRA w/o ID	0.0611	0.0964	0.1474	0.2466	0.3111
COBRA w/o Dense	0.0690	0.1032	0.1738	0.2709	0.3273
COBRA w/o BeamFusion	0.0856	0.1254	0.1732	0.2455	0.2855

Table 3: Performance comparison on industrial dataset

process, eliminating items with fewer than five user interactions and users with fewer than five item interactions. Detailed statistics of the datasets are presented in Table 1. For the evaluation of recommendation accuracy and ranking quality, we employ Recall@K and NDCG@K, specifically at K=5 and K=10. These metrics provide insights into the system's ability to accurately recommend relevant items and maintain a high-quality ranking order.

- 4.1.2 Baselines. To comprehensively evaluate the performance of our proposed COBRA method, we compare it with several state-of-the-art recommendation methods:
  - P5 [11]: Transforms recommendations into natural language sequences.
  - Caser [39]: Captures sequential patterns using convolutional layers.
  - HGN [28]: Hierarchical gating networks for long-/shortterm user interests.
  - GRU4Rec [14]: Models user behavior with gated recurrent units
  - SASRec [18]: Transformer-based model for long-term dependencies.
  - FDSA [52]: Self-attentive model for item-feature transitions.
  - BERT4Rec [37]: Bidirectional self-attention with cloze objective
  - **S**<sup>3</sup>-**Rec** [55]: Contrastive learning for recommendation.
  - TIGER[33]: Uses RQ-VAE to encode item content features and Transformer for generative retrieval.

These methods are chosen to represent a variety of recommendation techniques, including sequential dense recommendation and generative recommendation.

- 4.1.3 Implementation Details. In our approach, we adopt a method for generating semantic IDs similar to the one used in [33]. However, unlike [33], which uses a different configuration, we employ a 3-level semantic ID structure, where each level corresponds to a codebook size of 32. These semantic IDs are generated using the T5 model. COBRA is implemented with a lightweight architecture, featuring a 1-layer encoder and a 2-layer decoder.
- 4.1.4 Results. COBRA consistently surpasses all baseline models across various metrics, as presented in Table 2. On the "Beauty" dataset, COBRA achieves a Recall@5 of 0.0537 and a Recall@10 of 0.0725, exceeding the previous best model (TIGER) by 18.3% and 11.9%, respectively. For the "Sports and Outdoors" dataset, COBRA records a Recall@5 of 0.0305 and an NDCG@10 of 0.0215, outperforming TIGER by 15.5% and 18.8%, respectively. On the "Toys and Games" dataset, COBRA attains a Recall@10 of 0.0462

and an NDCG@10 of 0.0515, surpassing TIGER by 24.5% and 19.2%, respectively.

## 4.2 Industrial-scale Experiments

4.2.1 Dataset and Evaluation Metrics. To comprehensively evaluate the proposed COBRA method, we conduct experiments on the Baidu Industrial Dataset, a large-scale dataset derived from user interaction logs on the Baidu advertising platform. The dataset encompasses diverse recommendation scenarios, including list-page, dual-column, and short-video. It consists of five million users and two million advertisements, providing a comprehensive representation of real-world user behavior and advertising content. Advertisers and advertisements are represented through attributes such as title, industry labels, brand, and campaign text. These attributes are processed and encoded into two-level sparse IDs and dense vectors, capturing both coarse-grained and fine-grained semantic information. This dual representation enables COBRA to effectively model user preferences and item characteristics. The dataset is divided into two parts:  $D_{\text{train}}$  and  $D_{\text{test}}$ . The training set,  $D_{\text{train}}$ , includes user interaction logs collected over the first 60 days, covering recommendation content interactions during this period. The test set,  $D_{\text{test}}$ , is constructed from logs of the subsequent day following the  $D_{\text{train}}$  period, serving as a benchmark to assess model performance. For offline evaluation, we employ Recall@K as the evaluation metric, testing with  $K \in \{50, 100, 200, 500, 800\}$ . This metric provides a measure of the model's ability to accurately retrieve relevant recommendations at various thresholds.

## 4.2.2 Baselines. We compare COBRA against its variants:

- **COBRA w/o ID**: Removes sparse IDs, relying solely on dense vectors. This variant resembles RecFormer [21], using lightweight transformers for sequence modeling.
- COBRA w/o Dense: Removes dense vectors, using only sparse IDs. Due to the coarse-grained nature of IDs, this variant adopts a generative retrieval method similar to TIGER [33], leveraging semantic IDs for retrieval.
- COBRA w/o BeamFusion: Removes the BeamFusion module, using top-1 sparse ID and nearest-neighbor retrieval for top-k results.
- 4.2.3 Implementation Details. COBRA is built upon a Transformer-based architecture. In this framework, the text encoder processes advertisement text into sequences, which are then handled by the sparse ID head to predict 2-level semantic IDs configured as  $32 \times 32$ . For more fine-grained modeling of advertisements, the variant COBRA w/o Dense employs 3-level semantic IDs ( $256 \times 256 \times 256$ ).

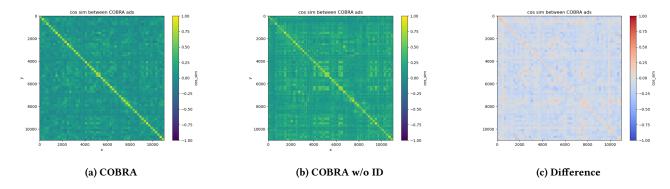


Figure 4: Cosine similarity matrices for advertisement dense embeddings. (a) COBRA's dense embeddings exhibit strong intra-ID cohesion and inter-ID separation. (b) COBRA w/o ID shows weaker category separation. (c) The difference matrix quantifies the enhancement in cohesion and separation when sparse IDs are incorporated.

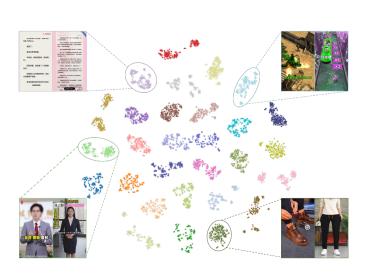


Figure 5: Embedding Visualization using t-SNE. The plot illustrates the distribution of 10,000 randomly sampled advertisement embeddings in a two-dimensional space for COBRA. Distinct clustering centers are observed for various IDs.

4.2.4 Results. As shown in Table 3, COBRA consistently outperforms all its variants across all evaluated metrics. At K=500, COBRA achieves a Recall@500 of 0.3716, representing a 42.2% improvement over the COBRA w/o Dense variant. When K=800, COBRA attains a Recall@800 of 0.4466, reflecting a 43.6% improvement over the COBRA w/o ID variant and a 36.1% enhancement compared to COBRA w/o BeamFusion. Notably, at relatively smaller values of K, the absence of Dense or ID representations results in more pronounced performance declines, underscoring the importance of cascaded representations for achieving granularity and precision. Conversely, as the recall size K increases, the performance

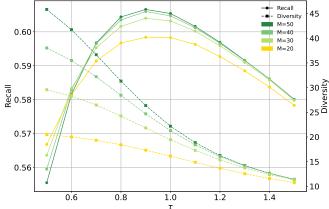


Figure 6: Recall-Diversity Curves. The x-axis represents the coefficient  $\tau$ , and the y-axis shows the Recall@2000 and Diversity metrics.

advantages associated with BeamFusion become increasingly evident, demonstrating its effectiveness in practical industrial recall systems.

The results further underscore the contributions of specific components:

- Excluding sparse IDs (COBRA w/o ID) leads to a recall reduction ranging from 26.7% to 41.5%, highlighting the critical role of semantic categorization.
- The removal of 3-level semantic IDs (COBRA w/o Dense) results in a performance drop between 30.3% and 48.3%, underscoring the importance of fine-grained modeling.
- Eliminating BeamFusion results in a recall decrease of 27.5% to 36.1%, emphasizing its significance in the integration of sparse signals.

# 4.3 Further Analysis

4.3.1 Analysis of Representation Learning. To evaluate the representation learning capabilities of the COBRA model, we construct

similarity matrices for the dense embeddings of advertisements, as illustrated in Figure 4. The COBRA model exhibits significant intra-ID cohesion and inter-ID separation, as demonstrated in the top heatmap of Figure 4a. This suggests that COBRA's dense embeddings proficiently capture detailed item characteristics while preserving semantic consistency within categories. Conversely, the model variant without sparse IDs (Figure 4b) shows weaker category separation, underscoring the importance of sparse IDs in maintaining semantic structure. The difference matrix in Figure 4c quantitatively confirms that incorporating sparse IDs enhances both cohesion and separation.

Further validation of COBRA's embeddings is achieved through visualizing the distribution of advertisement embeddings in a two-dimensional space using t-SNE. By randomly sampling 10,000 advertisements, distinct clustering centers for various categories are observed. Figure 5 reveals that advertisements are effectively clustered by category, indicating strong cohesion within categories. The clusters in purple, teal, light green, and dark green correspond primarily to advertisements for novels, games, legal services, and clothing, respectively. This demonstrates that the advertisement representations effectively capture semantic information.

4.3.2 Recall-Diversity Equilibrium. Balancing accuracy and diversity is a challenge in the retrieval stage of recommendation systems. To analyze this trade-off in COBRA, we examine recall-diversity curves, which depict how recall and diversity metrics evolve with the coefficient  $\tau$  in the BeamFusion mechanism under a fixed  $\phi = 16$ . Specifically, the curves in Figure 6 illustrate how Recall@2000 and diversity change as  $\tau$  varies. As shown in the figure, increasing  $\tau$ generally leads to a decrease in diversity. COBRA achieves an optimal balance between recall and diversity at  $\tau = 0.9$  and  $\phi = 16$ . Here, the model maintains high accuracy while ensuring that the recommendations cover a sufficiently diverse set of items. The diversity metric, defined as the number of different IDs in the recalled items, reflects the model's ability to avoid redundancy and provide users with a broader range of options. This fine-grained control over  $\tau$  and  $\phi$  allows practitioners to adjust the emphasis on accuracy or diversity based on specific business objectives. For instance, platforms prioritizing exploration can decrease  $\tau$  to enhance diversity. This flexibility distinguishes COBRA from models with fixed retrieval strategies, making it adaptable to diverse recommendation scenarios.

## 4.4 Online Results

To validate COBRA's real-world effectiveness, we conducted online A/B tests on the Baidu Industrial Dataset in January 2025. The test covered 10% of user traffic, ensuring statistical significance. The primary evaluation metrics were conversion and Average Revenue Per User (ARPU), which directly reflect user engagement and economic value. In the field covered by our proposed strategy, COBRA achieved a 3.60% increase in conversion and a 4.15% increase in ARPU. The results demonstrate that COBRA's hybrid architecture not only enhances recommendation quality in offline evaluations but also drives measurable business outcomes in production environments.

#### 5 Conclusions

In this work, we introduced COBRA, a generative recommendation framework that integrates cascaded sparse and dense representations for improved accuracy and diversity. COBRA employs a coarse-to-fine generation process, first generating sparse ID to capture the categorical essence of an item, then refining it with a dense vector. Extensive experiments demonstrate that COBRA outperforms state-of-the-art methods in both accuracy and diversity. Evaluations on public and industrial datasets, along with online A/B tests, confirm its effectiveness and practical applicability. By leveraging both sparse and dense representations, COBRA offers a robust solution for large-scale recommendation tasks.

#### References

- Keqin Bao, Jizhi Zhang, Wenjie Wang, Yang Zhang, Zhengyi Yang, Yancheng Luo, Chong Chen, Fuli Feng, and Qi Tian. 2023. A bi-step grounding paradigm for large language models in recommendation systems. ACM Transactions on Recommender Systems (2023).
- [2] Junyi Chen, Lu Chi, Bingyue Peng, and Zehuan Yuan. 2024. Hllm: Enhancing sequential recommendations via hierarchical large language models for item and user modeling. arXiv preprint arXiv:2409.12740 (2024).
- [3] Jiaju Chen, Chongming Gao, Shuai Yuan, Shuchang Liu, Qingpeng Cai, and Peng Jiang. 2024. DLCRec: A Novel Approach for Managing Diversity in LLM-Based Recommender Systems. arXiv preprint arXiv:2408.12470 (2024).
- [4] Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, and Wenwu Ou. 2019. Behavior sequence transformer for e-commerce recommendation in alibaba. In Proceedings of the 1st international workshop on deep learning practice for high-dimensional sparse data. 1-4.
- [5] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In Proceedings of the 1st workshop on deep learning for recommender systems. 7–10.
- [6] Yashar Deldjoo, Zhankui He, Julian McAuley, Anton Korikov, Scott Sanner, Arnau Ramisa, René Vidal, Maheswaran Sathiamoorthy, Atoosa Kasirzadeh, and Silvia Milano. 2024. A review of modern recommender systems using generative models (gen-recsys). In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 6448–6458.
- [7] Yashar Deldjoo, Zhankui He, Julian McAuley, Anton Korikov, Scott Sanner, Arnau Ramisa, Rene Vidal, Maheswaran Sathiamoorthy, Atoosa Kasrizadeh, Silvia Milano, et al. 2024. Recommendation with generative models. arXiv preprint arXiv:2409.15173 (2024).
- [8] Hao Ding, Yifei Ma, Anoop Deoras, Yuyang Wang, and Hao Wang. 2021. Zeroshot recommender systems. arXiv preprint arXiv:2105.08318 (2021).
- [9] Hao Fan, Mengyi Zhu, Yanrong Hu, Hailin Feng, Zhijie He, Hongjiu Liu, and Qingyang Liu. 2024. TiM4Rec: An Efficient Sequential Recommendation Model Based on Time-Aware Structured State Space Duality Model. arXiv preprint arXiv:2409.16182 (2024).
- [10] Ziwei Fan, Zhiwei Liu, Yu Wang, Alice Wang, Zahra Nazari, Lei Zheng, Hao Peng, and Philip S Yu. 2022. Sequential recommendation via stochastic self-attention. In Proceedings of the ACM web conference 2022. 2036–2047.
- [11] 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.
- [12] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. arXiv preprint arXiv:1703.04247 (2017).
- [13] 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.
- [14] B Hidasi. 2015. Session-based Recommendations with Recurrent Neural Networks. arXiv preprint arXiv:1511.06939 (2015).
- [15] Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2022. Towards universal sequence representation learning for recommender systems. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 585–593.
- [16] 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.

- [17] Jianchao Ji, Zelong Li, Shuyuan Xu, Wenyue Hua, Yingqiang Ge, Juntao Tan, and Yongfeng Zhang. 2024. Genrec: Large language model for generative recommendation. In European Conference on Information Retrieval. Springer, 494–502.
- [18] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In 2018 IEEE international conference on data mining (ICDM). IEEE, 197–206.
- [19] Sein Kim, Hongseok Kang, Kibum Kim, Jiwan Kim, Donghyun Kim, Minchul Yang, Kwangjin Oh, Julian McAuley, and Chanyoung Park. 2025. Lost in Sequence: Do Large Language Models Understand Sequential Recommendation? arXiv preprint arXiv:2502.13909 (2025).
- [20] 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.
- [21] Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. 2023. Text is all you need: Learning language representations for sequential recommendation. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 1258–1267.
- [22] Muyang Li, Zijian Zhang, Xiangyu Zhao, Wanyu Wang, Minghao Zhao, Runze Wu, and Ruocheng Guo. 2023. Automlp: Automated mlp for sequential recommendations. In Proceedings of the ACM web conference 2023. 1190–1198.
- [23] Yongqi Li, Xinyu Lin, Wenjie Wang, Fuli Feng, Liang Pang, Wenjie Li, Liqiang Nie, Xiangnan He, and Tat-Seng Chua. 2024. A survey of generative search and recommendation in the era of large language models. arXiv preprint arXiv:2404.16924 (2024).
- [24] Yaoyiran Li, Xiang Zhai, Moustafa Alzantot, Keyi Yu, Ivan Vulić, Anna Korhonen, and Mohamed Hammad. 2024. Calrec: Contrastive alignment of generative llms for sequential recommendation. In Proceedings of the 18th ACM Conference on Recommender Systems. 422–432.
- [25] Xinyu Lin, Wenjie Wang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. 2023. A multi-facet paradigm to bridge large language model and recommendation. arXiv preprint arXiv:2310.06491 3 (2023).
- [26] Han Liu, Yinwei Wei, Xuemeng Song, Weili Guan, Yuan-Fang Li, and Liqiang Nie. 2024. Mmgrec: Multimodal generative recommendation with transformer model. arXiv preprint arXiv:2404.16555 (2024).
- [27] Zihan Liu, Yupeng Hou, and Julian McAuley. 2024. Multi-behavior generative recommendation. In Proceedings of the 33rd ACM International Conference on Information and Knowledge Management. 1575–1585.
- [28] Chen Ma, Peng Kang, and Xue Liu. 2019. Hierarchical gating networks for sequential recommendation. In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 825–833.
- [29] 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.
- [30] Nikil Pancha, Andrew Zhai, Jure Leskovec, and Charles Rosenberg. 2022. Pinner-former: Sequence modeling for user representation at pinterest. In Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining. 3702–3712.
- [31] 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.
- [32] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. 2022. Contrastive learning for representation degeneration problem in sequential recommendation. In Proceedings of the fifteenth ACM international conference on web search and data mining. 813–823.
- [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] 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. 154–163.
- [35] Anima Singh, Trung Vu, Nikhil Mehta, Raghunandan Keshavan, Maheswaran Sathiamoorthy, Yilin Zheng, Lichan Hong, Lukasz Heldt, Li Wei, Devansh Tandon, et al. 2024. Better generalization with semantic ids: A case study in ranking for recommendations. In Proceedings of the 18th ACM Conference on Recommender Systems, 1039–1044.
- [36] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. Autoint: Automatic feature interaction learning via self-attentive neural networks. In Proceedings of the 28th ACM international conference on information and knowledge management. 1161–1170.
- [37] 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. 1441–1450.

- [38] Juntao Tan, Shuyuan Xu, Wenyue Hua, Yingqiang Ge, Zelong Li, and Yongfeng Zhang. 2024. Idgenrec: Llm-recsys alignment with textual id learning. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. 355–364.
- [39] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In Proceedings of the eleventh ACM international conference on web search and data mining. 565–573.
- [40] Viet Anh Tran, Guillaume Salha-Galvan, Bruno Sguerra, and Romain Hennequin. 2023. Attention mixtures for time-aware sequential recommendation. In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval. 1821–1826.
- [41] Zhizhong Wan, Bin Yin, Junjie Xie, Fei Jiang, Xiang Li, and Wei Lin. 2024. LARR: Large Language Model Aided Real-time Scene Recommendation with Semantic Understanding. In Proceedings of the 18th ACM Conference on Recommender Systems. 23–32.
- [42] Hangyu Wang, Jianghao Lin, Xiangyang Li, Bo Chen, Chenxu Zhu, Ruiming Tang, Weinan Zhang, and Yong Yu. 2024. FLIP: Fine-grained Alignment between IDbased Models and Pretrained Language Models for CTR Prediction. In Proceedings of the 18th ACM Conference on Recommender Systems. 94–104.
- [43] Shirui Wang, Bohan Xie, Ling Ding, Xiaoying Gao, Jianting Chen, and Yang Xiang. 2024. SeCor: Aligning Semantic and Collaborative Representations by Large Language Models for Next-Point-of-Interest Recommendations. In Proceedings of the 18th ACM Conference on Recommender Systems. 1-11.
- [44] Wenjie Wang, Honghui Bao, Xinyu Lin, Jizhi Zhang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. 2024. Learnable item tokenization for generative recommendation. In Proceedings of the 33rd ACM International Conference on Information and Knowledge Management. 2400–2409.
- [45] Yidan Wang, Zhaochun Ren, Weiwei Sun, Jiyuan Yang, Zhixiang Liang, Xin Chen, Ruobing Xie, Su Yan, Xu Zhang, Pengjie Ren, et al. 2024. Content-Based Collaborative Generation for Recommender Systems. In Proceedings of the 33rd ACM International Conference on Information and Knowledge Management. 2420–2430.
- [46] Ye Wang, Jiahao Xun, Minjie Hong, Jieming Zhu, Tao Jin, Wang Lin, Haoyuan Li, Linjun Li, Yan Xia, Zhou Zhao, et al. 2024. EAGER: Two-Stream Generative Recommender with Behavior-Semantic Collaboration. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 3245–3254.
- [47] Shuyuan Xu, Wenyue Hua, and Yongfeng Zhang. 2024. Openp5: An open-source platform for developing, training, and evaluating Ilm-based recommender systems. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. 386–394.
- [48] Liu Yang, Fabian Paischer, Kaveh Hassani, Jiacheng Li, Shuai Shao, Zhang Gabriel Li, Yun He, Xue Feng, Nima Noorshams, Sem Park, et al. 2024. Unifying Generative and Dense Retrieval for Sequential Recommendation. arXiv preprint arXiv:2411.18814 (2024).
- [49] Wojciech Zaremba. 2014. Recurrent neural network regularization. arXiv preprint arXiv:1409.2329 (2014).
- [50] Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Michael He, et al. 2024. Actions speak louder than words: Trillion-parameter sequential transducers for generative recommendations. arXiv preprint arXiv:2402.17152 (2024).
- [51] Gaowei Zhang, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, and Ji-Rong Wen. 2024. Scaling law of large sequential recommendation models. In Proceedings of the 18th ACM Conference on Recommender Systems. 444–453.
- [52] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, Xiaofang Zhou, et al. 2019. Feature-level deeper selfattention network for sequential recommendation.. In IJCAI. 4320–4326.
- [53] 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.
- [54] Lei Zheng, Vahid Noroozi, and Philip S Yu. 2017. Joint deep modeling of users and items using reviews for recommendation. In Proceedings of the tenth ACM international conference on web search and data mining. 425–434.
- [55] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In Proceedings of the 29th ACM international conference on information & knowledge management. 1893–1902.
- [56] Jieming Zhu, Mengqun Jin, Qijiong Liu, Zexuan Qiu, Zhenhua Dong, and Xiu Li. 2024. CoST: Contrastive Quantization based Semantic Tokenization for Generative Recommendation. In Proceedings of the 18th ACM Conference on Recommender Systems. 969–974.