MoMoREC: A Multi-agent Motivation Generation Framework for Residual Semantic ID-Aware Recommendation

Yige Wang[1], Mingming Li[1], Li Wang[1], Kaichen Zhao[1], Wangming Li[1], Weipeng Jiang[2], Xueying Li[1] *

[1]Taobao & Tmall Group of Alibaba
[2]School of Cyber Science and Engineering, Xi'an Jiaotong University
{tianxuan.wyg, mingcong.lmm, qianyue.wl, zhaokaichen.zkc, liwangming.lwm, xiaoming.lxy}@alibaba-inc.com
lenijwp@mail.xjtu.edu.cn

Abstract

Recent advances in the field of sequential recommendation have highlighted the potential of Large Language Models (LLMs) in enhancing item embeddings and improving user understanding. However, existing approaches face three major limitations: 1) insufficient understanding of the reasons behind users' purchase decisions, 2) the high-dimensional embeddings directly produced by LLMs are not well compatible with traditional low-dimensional ID embeddings and 3) reliance on additional fine-tuning and high inference overhead to adapt LLMs to the recommendation task. In this paper, we propose **MoMoREC**, a simple yet effective user-understanding-based recommendation strategy. This method leverages the intrinsic comprehension capabilities of LLMs combined with residual semantic IDs to better understand users. Specifically, starting from common user purchasing behaviors and incorporating item characteristics, we employ a multi-agent framework to utilize LLMs in analyzing user shopping motivations and extracting high-dimensional dense embeddings. These embeddings are then transformed into low-dimensional IDs using a residual semantic ID approach via clustering and residual dimensionality reduction, which can be fed into the recommendation model. **MoMoREC** effectively integrates the understanding power of LLMs with the strengths of recommendation systems, preserving rich semantic language embeddings while reducing or eliminating the need for auxiliary trainable modules. As a result, it seamlessly adapts to any sequential recommendation framework. Experiments on three benchmark datasets show that **MoMoRec** significantly improves traditional recommendation models, demonstrating its effectiveness and flexibility.

Introduction

Sequential recommendation systems are classical frameworks that infer user preferences from their historical purchase behavior and predict the next item a user is likely to interact with. Conventional recommendation models (Wang et al. 2021; Zhou et al. 2018; Sheng et al. 2021) assign unique IDs to items and learn user preferences through ID-based embeddings, effectively capturing interaction behavior (Zhang et al. 2024a). While efficient and parameter-light, these methods lack interpretability due to their reliance on learned latent representations. With the success of LLMs (Bai et al. 2023; Moonshot-AI 2024), utilizing their ability to generate item lists offers a straightforward approach to this end (Liao et al. 2025; Wu et al. 2024). As Figure 2 shows, directly generating item IDs incurs significantly higher parameter costs than traditional recommendation models, requiring additional training. As evidenced by Figure 2, employing LLMs' generative capabilities for item sequence generation incurs an order-of-magnitude latency increase (10× or even more) and parameter count differences reaching hundreds of times. This results in substantially higher storage costs and operational overheads, representing nearly prohibitive drawbacks in real recommendation scenarios. Moreover, since recommendation tasks fundamentally differ from conventional pre-training objectives, additional adaptation training is required to align the model with recommendation scenarios, inevitably introducing extra training overhead (Zhang et al. 2025b).

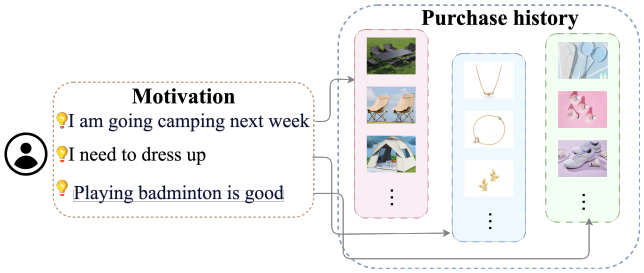

Figure 1: Users are typically driven by certain underlying motivations when purchasing a sequence of items, a process that is inherently latent and unobservable; we aim to infer these hidden representations through the analysis of observable behavioral patterns.

To harness the generative power of LLMs in natural language without additional training, we introduce user motivation—a latent driver of behavior underlying purchase decisions (Ozaki and Sevastyanova 2011). As illustrated in Figure 1, motivation captures the intent behind actions; for example, purchasing a pendant may be driven by the pursuit of beauty. We propose a multi-agent framework that leverages LLMs' intrinsic knowledge to infer motivations rather than

*Corresponding Author

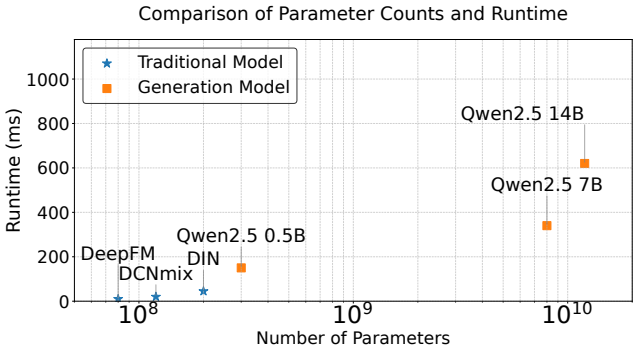

Figure 2: Comparison of Model Parameters and Recommendation Latency Across Different Recommendation Models

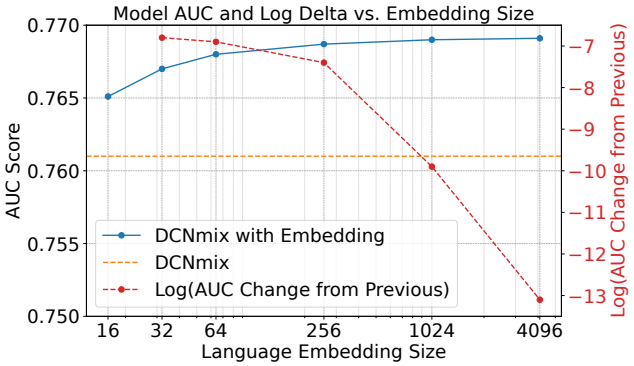

Figure 3: AUC (Blue) and Relative AUC Change (ΔAUC, Red Line) of DCNmix in Recommendation with Different Embedding Sizes

directly generating recommendations. Built on a cooperative paradigm (Guo et al. 2024), the system employs three specialized agents: (1) a user analyst identifying candidate motivations, (2) a summarizer integrating these into coherent themes, and (3) an arbiter selecting the most representative motivation. Grounded in users' real-world decisions, the analysis of co-purchased item patterns enhances motivation reliability (Alcan et al. 2022) and reduces the number of LLM inferences, although this approach favors frequent items and overlooks long-tail coverage (Zhang et al. 2023). To mitigate this issue, we propagate motivations from high-frequency to low-frequency items via a semantic similarity-based spreading module, the effectiveness of which is demonstrated in subsequent experiments.

The next challenge lies in effectively incorporating the user motivations derived from LLM analysis into the recommendation system, as textual information cannot be directly integrated in its raw form. A straightforward solution involves employing a semantic encoder (Devlin et al. 2019; Zhang et al. 2025a) to convert textual data into continuous high-dimensional embeddings. These embeddings are then fused with the conventional feature matrix and integrated into traditional recommendation models (Zhang et al. 2024b), effectively combining the low-latency advantages of classical approaches with the superior comprehension capabilities of LLMs. However, as demonstrated in Figure 3, increasing the embedding dimension does not yield linear improvements in AUC. In fact, the marginal gains in AUC gradually diminish as the embedding size grows larger. Although a larger hidden size is expected to capture more information, it is evident that the recommendation model fails to fully utilize this potential. Inspired by Rajput et al. (2023), the residual semantic ID approach effectively mitigates this issue. The generated semantic IDs can be treated as conventional discrete features, enabling the training of a feature embedding matrix that reduces the out of distribution deviation between LLMs and smaller models.

In this paper, we propose **MoMoREC**, a multi-agent **Mo**tivation generation fra**me**wo**r**k for Residual Semantic ID-Aware **Rec**ommendation, which can be summarized as follows: (1) a multi-agent LLM architecture is proposed to generate user purchase motivations, achieving a profound understanding of user behavior; (2) a motivation spreading framework is designed to propagate motivations from high-frequency products to long-tail items, ensuring the quality of motivations for long-tail commodities; (3) by computing the cluster centroids in the motivation space to construct a codebook, continuous embeddings are discretized into semantic IDs, and a multi-dimensional ID vector is formed through residual embedding, which mitigates the latency and cost issues in LLM-based recommendations and enables real-time inference in online scenarios.

Preliminaries

Problem Formulation

Let $\mathcal{I} = \{\mathbf{i}_1, \ldots, \mathbf{i}_n\}$ denote a collection of n distinct items. The user's interaction sequence over time is represented as $s = [\mathbf{i}_1, \ldots, \mathbf{i}_{|s|}]$, where $\mathbf{i}_j$ signifies the j-th item in the sequence, and $|s|$ denotes the total number of items with which the user has interacted. Consistent with prior research (Lin et al. 2024), we focus mainly on item-associated side information. In the context of sequential recommendation with side information, each item $\mathbf{i} \in \mathcal{I}$ is characterized not only by its unique identifier but also by multiple attribute features. Formally, an item is defined as $\mathbf{i}_j = \{v_j, a_{1,j}, \ldots, a_{m,j}\}$, where v_j represents the item ID, $a_{k,j}$ denotes the value of the k-th attribute associated with the item, and m indicates the total number of attribute types considered. The objective of our model is to estimate the probability distribution over the next potential item interaction and identify the item that maximizes this probability, formulated as $\text{argmax}_{j \in \mathcal{I}} P(\mathbf{i}_{|s|+1} = j \mid s)$.

Methodology

Overview

As illustrated in Figure 4, our framework consists of three components: (a) Motivation Generation, (b) Motivation Spreading, and (c) Residual Semantic ID Retrieval. Motivation Generation leverages users' frequently purchased items and employs an LLM to analyze user purchasing motivations, thereby achieving a comprehensive understanding of

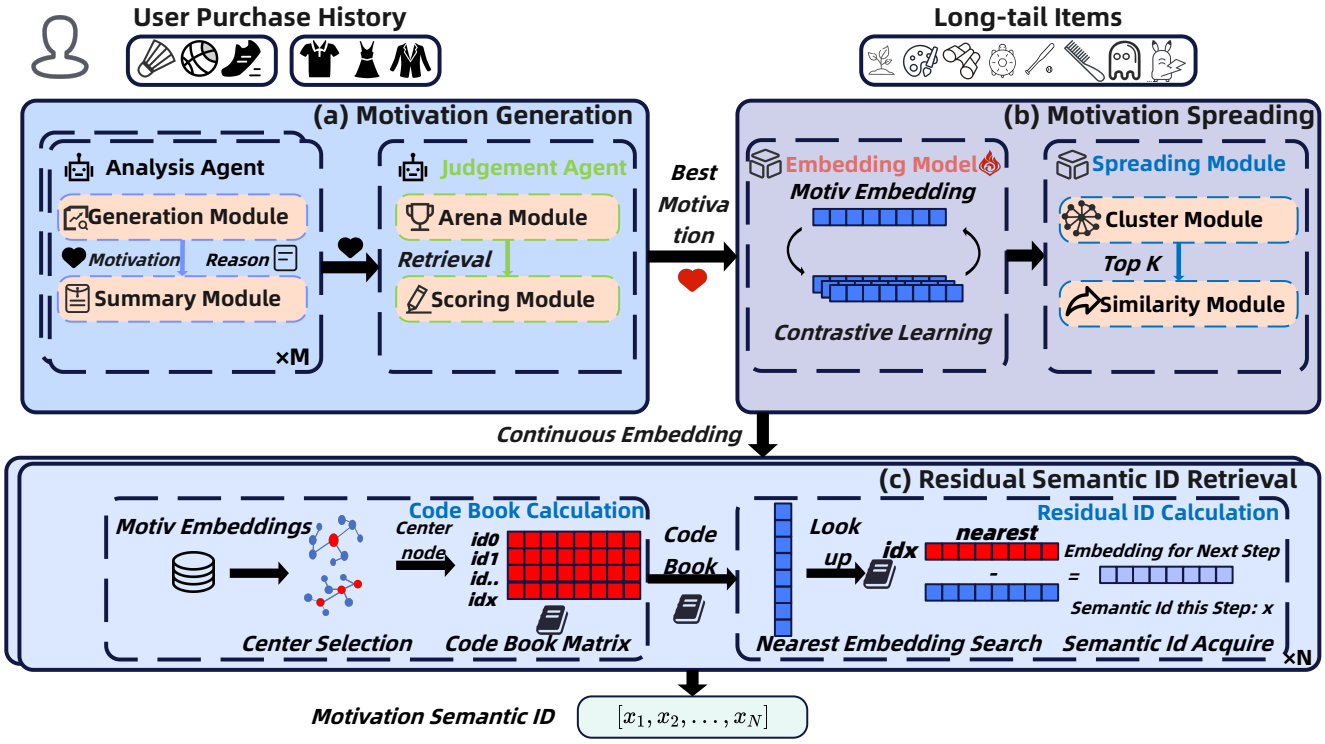

Figure 4: The Whole Structure of **MoMoREC** Framework

user preferences. Motivation Spreading aims to propagate the motivational features derived from frequently purchased items to long-tail or cold-start items, enhancing their semantic richness. Finally, Residual Semantic ID Retrieval is designed to transform continuous embedding features into discrete semantic IDs, enabling compatibility with classical recommendation models that operate on categorical input representations.

Motivation Generation

In this section, given a user's purchase sequence $s = [\mathbf{i}_1, \ldots, \mathbf{i}_{|s|}]$, we aim to generate a shared motivation $\mathbf{m}$ for all items in the list and assign it to each individual item. To ensure the sequence representation is more generalizable, we keep only those sequences that appear two or more times.

Analysis Agent The Analysis Agent is tasked with generating the motivation and comprises two key components:

The Generation Module is designed to produce a comprehensive motivation, accompanied by an explanation of its validity. Drawing inspiration from Chen et al. (2025), we acknowledge that engaging in extended reasoning during the inference phase often enhances performance; consequently, we impose minimal constraints on the length of the generated output at this stage. The quality of the generation is assessed along two dimensions: Relevance, which pertains to the extent to which the generated motivation aligns with the item's intended use, and Content Depth, referring to the degree of understanding exhibited regarding specific product attributes such as price, design aesthetics, brand significance, and other user-item interactions that contribute to a richer comprehension of the item sequence.

The Summary Module serves to condense lengthy motivation texts into more succinct forms. During this phase, excessively long outputs are trimmed and distilled into their core meaning. In contrast to the Generation Module, the length of the output is strictly controlled to ensure that the final motivation $\mathbf{m}$ is both concise and broadly generalizable. To mitigate potential information loss during compression, we increase the temperature τ to generate more diverse and comprehensive summaries.

Judgment Agent The Judgment Agent is employed to select the optimal solution from multiple motivations generated by various Generation Agents with multiple prompts and different base models. Inspired by Chatbot Arena (Chiang et al. 2024), we treat different Agents as candidates and adopt a pairwise comparison approach, scoring each candidate in terms of content depth and relevance, and subsequently providing an overall score. Inspired by Ouyang et al. (2022), directly assigning hard scores to each motivation instance would render instances within the same score bracket incomparable. Therefore, we adopt a pairwise comparison approach. After obtaining the ranking results, they are submitted to the Arena Module for final evaluation. For comparisons among Agents, we implement the ELO mechanism (Boubdir et al. 2024) according to the following formula, which helps reduce the number of pairwise comparisons:

$$\text{ELO}(R_i, R_j) = \frac{1}{1 + 10^{\frac{R_j - R_i}{\beta}}} \tag{1}$$

where R_i and R_j denote the current ratings of the two agents being compared, and the resulting value represents the expected probability of agent i outperforming agent j. After obtaining the scores R for the set of agents, we retain the agent with the highest ELO rating and its generated motivation as input for the subsequent stage.

Motivation Spreading

Since not all items in the product pool appear multiple times within interaction sequences s, and a large proportion of items have very low occurrence frequencies (Zhao et al. 2023), the requirement for sufficient co-occurrence across different s during motivation generation further reduces the number of eligible items. To address this issue, we attempt to propagate popular motivations to less frequent items in order to alleviate the long-tail problem.

Embedding Model Given a product $\mathbf{i}$ and its associated motivational factor $\mathbf{m}$, we obtain the embeddings of both through the model. Rather than relying on large pretrained models, we formulate this as a domain-specific representation learning task and employ a lightweight encoder, which we find sufficient to achieve high-quality alignment while maintaining computational efficiency. We adopt the contrastive learning paradigm, combining the InfoNCE loss (Oord, Li, and Vinyals 2018) and the CoSENT loss (Su 2022), to train different base models. It can be defined as follows:

$$\mathcal{L}_{InfoNCE} = -\sum_{i}^{m} \log \left[\frac{e^{\cos(\mathbf{m}_i, \mathbf{i}_i^+)/\tau}}{\sum_j^N e^{\cos(\mathbf{m}_i, \mathbf{i}_j^-)/\tau}} \right] \tag{2}$$

Where $\mathbf{m}_i$ denotes the embedding of the motivation factor, and $\mathbf{i}_i^+$ represents the embedding of the positive product candidate associated with $\mathbf{m}_i$. The term $\mathbf{i}_j^-$ denotes the embedding of a negative product candidate. The similarity between $\mathbf{m}_i$ and each product candidate is scaled by a temperature parameter τ.

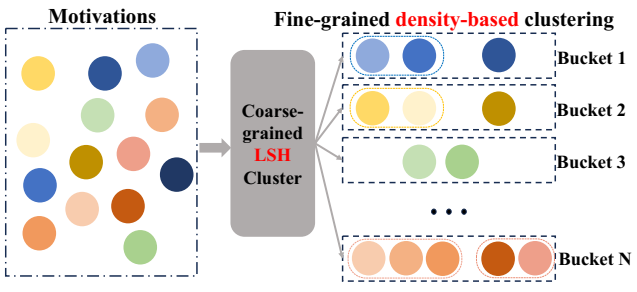

Figure 5: The clustering Module in **MoMoREC**. We use the LSH algorithm for bucketing and perform fine-grained clustering within each bucket.

$$\mathcal{L}_{Co} = \log \left[1 + \sum_{s(\mathbf{m}_i, \mathbf{i}_j) > s(\mathbf{m}_m, \mathbf{i}_n)} e^{\frac{\cos(\mathbf{m}_m, \mathbf{i}_n) - \cos(\mathbf{m}_i, \mathbf{i}_j)}{\tau}} \right] \tag{3}$$

Where $s(\mathbf{m}, \mathbf{i})$ is the similarity between $\mathbf{m}$ and $\mathbf{i}$, and cos means cosine similarity function. We utilize the items and motivations generated from the previous step as $(\mathbf{m}, \mathbf{i}^+)$, and treat samples from other instances within the same batch as $\mathbf{m}$ and $\mathbf{i}^-$ pairs.

Spreading Module The Spreading Module is designed to categorize similar motivations and assign generated motivations to long-tail items. To achieve this, we adopt a metric learning approach, mapping items $\mathbf{i}$ and motivations $\mathbf{m}$ into a shared semantic space, resulting in representations $h_\mathbf{i}$ and $h_\mathbf{m}$, where $h \in \mathbb{R}^d$ is the hidden state with embedding size d. Within this space, we compute the similarity between the item and motivation embeddings to identify semantically similar pairs.

The Cluster Module is used to assemble $\mathbf{m}$ with the same meanings. For instance, suppose that we acquire two motivations "Aim to develop strong and sculpted musculature" and "Desire to achieve well-defined, athletic muscles", acting as different formulations of the same underlying meaning. For million-scale or even larger item sets, such as Amazon-Books (Hou et al. 2024), computing semantic similarity directly results in a time complexity of $O(n^2)$, which is typically unacceptable. We surveyed common large-scale clustering algorithms. For K-means variants such as Mini Batch K-Means (Newling and Fleuret 2016), the number of clusters K must be specified a priori, yet determining an appropriate value is often challenging in practice. For density-based methods like hierarchical HDBSCAN (Malzer and Baum 2020), hyperparameter choices directly impact the final clustering outcome. Instead, as shown in Figure 5, we use the LSH algorithm (Rajaraman and Ullman 2011) to group the clustering objectives $\mathbf{m}$ into the bucket first. Within each bucket, we perform fine-grained density-based clustering for more detailed modeling. Based on this, the system has the capability to automatically discover new clusters without the need to specify the value of K. In each bucket, compute the adjacency matrix $M_{ij} = \mathbb{I}[\cos(\mathbf{m}_i, \mathbf{m}_j) > \tau]$. Connected components in this graph are identified through iterative graph coloring: starting from an uncolored node, all reachable nodes are assigned the same color. This process is repeated until all vertices are colored, with each color representing a distinct cluster.

The Similarity Module is designed to match the set of unmotivated items $\mathcal{I}_\mathbf{u}$ with the set of motivated items $\mathcal{I}_\mathbf{m}$, thereby obtaining their spreading motivation representations. Empirical observations show that $|\mathcal{I}_\mathbf{m}| \ll |\mathcal{I}_\mathbf{u}|$. Compared to $|\mathcal{I}_\mathbf{u}|$, the size of $|\mathcal{I}_\mathbf{m}|$ can be approximated as a constant, leading to a linear time complexity in this scenario. At this point, for each $\mathbf{i}_\mathbf{u} \in \mathcal{I}_\mathbf{u}$, the nearest neighbor items can be selected and stored using a min-heap to maintain the top-k nearest neighbors. Finally, based on the clustered $\mathbf{m}$ cor-

responding to the top-k nearest neighbor items, the majority voting method is applied, where the $\mathbf{m}$ with the highest number of votes is assigned as the motivation for the current long-tail item. In this way, the spreading step is completed.

Residual Semantic ID Retrieval

Considering the advantage of traditional recommendation models in achieving low response time, we adopt a paradigm that combines embeddings with traditional recommendation models. The Residual Semantic ID Retrieval module is specifically designed to convert the Continuous Motivation Embedding $h_{\mathbf{m}}$ into discrete ID information, enabling seamless integration with the traditional recommendation model. Also, the ID features can also be integrated with the CTR (Click-Through Rate) (Feng et al. 2019) model, not only augmenting end-to-end sequential recommendation models but also enhancing click-through rate prediction. We aim to obtain retrieved ID vector of length n, represented as a vector $\mathbf{E} = [id^1, id^2, \ldots, id^n] \in \mathbb{Z}^n$. For k-th layer's ID id^k, we execute the following procedure:

Code Book Calculation The codebook at the k-th layer, denoted as $\mathbf{C}^k \in \mathbb{R}^{l \times d}$, serves as a learnable dictionary for vector quantization, where each row corresponds to a centroid in the d-dimensional embedding space, and l is the predefined number of clusters. Given its explicit matrix parameterization, the codebook size scales linearly with both l and d, making it imperative to maintain a fixed and tractable dimensionality to ensure computational and memory efficiency. As a result, methods that generate embeddings or cluster assignments with variable dimensionality become impractical, particularly those relying on dynamic structural assumptions.

In contrast, K-means clustering offers a more suitable solution under these constraints. It enables efficient partitioning of the data space into a fixed number of clusters through minimization of the within-cluster variance:

$$\min_{\mathbf{C}^k} \sum_{\mathbf{h}^{k-1} \in \mathcal{H}^{k-1}} \min_i \|\mathbf{h} - \mathbf{c}_i^k\|_2^2 \tag{4}$$

where $\mathcal{H}^{k-1}$ represents the collection of residual embeddings corresponding to the target item at the $k-1$-th layer, and $\mathbf{h}^{k-1}$ denotes the $k-1$-th layer's hidden states. To meet the requirements of scalability and distributed processing, we employ the distributed K-means algorithm (Oliva, Setola, and Hadjicostis 2013) to leverage the multi-threading capabilities of computers. This allows us to compute a compact centroid matrix $\mathbf{C}^k$ that effectively embeds the structure of the entire semantic latent space. And the codebook $\mathbf{C}^k$ can be formulated as follows:

$$\mathbf{C}^k = [c_1^k; c_2^k; \ldots, c_l^k]^T \tag{5}$$

Residual ID Calculation For each item $\mathbf{i} \in \mathcal{I}$, the corresponding ID is determined by finding the nearest index to the residual embedding $\mathbf{h}^{k-1}$ at the current layer, which can be formulated as:

$$id^k = \underset{i}{\operatorname{argmin}} \|\mathbf{h}^{k-1} - \mathbf{c}_i^k\|_2 \tag{6}$$

The residual embedding $\mathbf{h}^k$ at the k-th layer is obtained by the following formula, where $-$ denotes element-wise subtraction of vectors:

$$\mathbf{h}^k = \mathbf{h}^{k-1} - \mathbf{c}_{id^k}^k \tag{7}$$

For the $(k+1)$-th layer, the aforementioned steps are repeated. As for the first layer, the input embedding $\mathbf{h}^1$ is derived from the Motivation Spreading module.

Experiments

In this section, we aim to explore three key questions:

- **RQ1**: Does the proposed framework achieve superior performance compared to state-of-the-art semantic model-based methods under more general and diverse experimental settings?
- **RQ2**: How do individual components of the framework contribute to the overall performance?
- **RQ3**: Can the proposed method improve inference efficiency while maintaining or even enhancing model effectiveness?

Experimental Settings

Datasets and Metrics We evaluate **MoMoREC** on three real-world datasets on Amazon (Hou et al. 2024): **Beauty** encompasses user reviews and corresponding metadata for beauty and personal care products, collected over the period from May 1996 to September 2023. **Video Game** and **Baby Product** contain user reviews and associated metadata for video games and Baby products. To more accurately reflect real-world recommendation settings, we use the complete three datasets, preserving all user-item interactions and item information. The statistics can be seen in Table 1, where Long-tail Item defined as items that have received 5 or fewer user interactions. We commonly employ recall, MRR, and precision at @10 and @20 as evaluation metrics in recommendation systems.

Dataset	Users	Items	Interactions	Long-tail
Beauty	12609	112565	18061	11651
Video Game	2766656	137270	4555500	80336
Baby Product	3386206	217726	5953891	139528

Table 1: The Statistics of Dataset

Backbones We adopt **SASRec** (Kang and McAuley 2018) as the base sequential recommendation architecture. SASRec employs a self-attentive mechanism to model long-range dependencies in user interaction sequences, effectively capturing complex behavioral patterns through learned attention weights.

Baselines We review recent advances in recommendation systems and identify representative models that effectively integrate semantic information to enhance recommendation performance. **LLMESR** (Liu et al. 2024b) is a sequential recommendation framework enhanced by LLM to address

Dataset	Method	@10			@20		
		Recall	MRR	Precision	Recall	MRR	Precision
Beauty	Base	0.0303	0.0095	0.003	0.045	0.0105	0.003
	LLMESR	0.0134	0.0035	0.0006	0.0209	0.0037	0.0005
	MoRec	0.0156	0.004	0.0008	0.0217	0.0042	0.0006
	whitenREC	0.0127	0.0036	0.0009	0.0199	0.0039	0.0007
	LLMInit	0.033	0.0094	0.003	0.0533	0.0096	0.003
	RLMRec	0.0423	0.0112	0.0021	0.0659	0.012	0.0016
	UniSRec	0.0124	0.0034	0.0006	0.0189	0.0036	0.0005
	AlphaFuse	0.0151	0.0042	0.0007	0.0192	0.0043	0.0006
	MoMoREC	0.1725	0.0951	0.0172	0.2142	0.0979	0.0107
	Best Impr. (×)	4.07	8.49	8.19	3.25	8.15	6.69
Video Game	Base	0.1425	0.0383	0.0071	0.1989	0.0402	0.005
	LLMESR	0.1041	0.0277	0.0052	0.153	0.0294	0.0038
	MoRec	0.1334	0.0365	0.0067	0.1893	0.0385	0.0047
	whitenREC	0.1117	0.0297	0.0056	0.162	0.0315	0.0041
	LLMInit	0.0371	0.0101	0.0018	0.0581	0.0108	0.0014
	RLMRec	0.1444	0.038	0.0072	0.2012	0.0404	0.0051
	UniSRec	0.1253	0.0341	0.0063	0.18	0.036	0.0045
	AlphaFuse	0.0369	0.01	0.0019	0.0582	0.0108	0.0014
	MoMoREC	0.6578	0.4485	0.0658	0.7352	0.4539	0.0368
	Best Impr. (×)	4.55	11.71	9.13	3.65	11.23	7.21
Baby Product	Base	0.0624	0.0163	0.0031	0.0923	0.0174	0.0023
	RLMRec	0.061	0.0167	0.0031	0.0907	0.0177	0.0023
	MoMoREC	0.7285	0.5177	0.0729	0.7982	0.5226	0.0399
	Best Impr. (×)	11.67	31.00	23.51	8.64	29.52	17.34

Best Impr. (×) denotes the improvement ratio over the best baseline, which is underlined.

Table 2: Performance of various models on different datasets in terms of MRR (Mean Reciprocal Rank), Precision, and Recall, Evaluated at Top-10 and Top-20 Rankings.

long-tail user and item challenges through semantic embeddings and retrieval-augmented self-distillation, without increasing inference cost. **RLMRec** (Ren et al. 2024) integrates representation learning to capture semantic user-item relationships and align them with collaborative signals. **MoRec** (Yuan et al. 2023) employs an architecture consisting of a modality encoder followed by a user behavior modeling network. **whitenREC** (Zhang et al. 2024a) applies whitening transformations to pre-trained text embeddings, balancing isotropy and semantic preservation in item representations. **LLMInit** (Qu et al. 2024) uses LLM embeddings to initialize ID embeddings without residual parts. **UniSRec** (Hou et al. 2022) adopts a universal sequence representation learning approach that learns transferable item and sequence representations using item description text. **AlphaFuse** (Hu et al. 2025) learns ID embeddings in the null space of language embeddings, preserving semantic integrity without additional trainable modules.

Integrated with Recommendation Model We retain the item_id as an ID feature and initialize its embeddings randomly. For the Motivation Semantic ID, we similarly initialize individual embedding matrices for each ID. The multi-dimensional Motivation Semantic ID embeddings are first projected to the hidden_dim dimension, then concatenated, and subsequently fed into a three-layer deep neural network (DNN). The resulting output is concatenated with the original item_id embedding, and the combined representation is passed through a final DNN to generate the ultimate recommendation representation. The input to our model consists of a 6-dimensional Motivation Semantic ID, with all other architectural and training configurations remaining consistent with the original SASRec framework.

Effectiveness of MoMoREC (RQ1)

Based on the experimental results presented in Table 2, our proposed **MoMoREC** model consistently achieves superior performance across all three datasets under both Top-10 and Top-20 evaluation settings. On the **Beauty** dataset, **MoMoREC** achieves a Recall@10 of 0.1725 and MRR@10 of 0.0951, outperforming the strongest baseline by 4.07× and 8.49×, respectively, with an equally significant 8.19× gain in Precision@10. Similarly, on the **Video Game** dataset, **MoMoREC** demonstrates remarkable gains, improving upon the best-performing baselines by 4.55× in Recall@10 and a substantial 11.71× in MRR@10. For the **Baby Products** dataset, **MoMoREC**'s superiority is even more pronounced, achieving an improvement of 11.67× in Recall@10 and an exceptional 31.00× in MRR@10, while the baselines expe-

rience notable performance degradation. In contrast, methods relying on static language model embeddings—such as LLMInit and LLMESR—show limited performance. These dramatic improvements highlight **MoMoREC**'s ability to effectively model user preferences, even in domains with sparse interactions and long-tail item distributions. Overall, **MoMoREC**'s consistent dominance across diverse domains and metrics confirms its effectiveness as a highly scalable and expressive recommendation framework.

Ablation Study (RQ2)

The influence of Spreading and Residual Part We first conduct ablation studies on the two key components — Motivation Spreading and Residual Semantic ID Retrieval — as shown in Table 3. "w/o Spe." denotes removing the Motivation Spreading module and not providing Residual Semantic IDs for long-tail items. "w/o Res." denotes removing Motivation Spreading entirely and using only cluster-based semantic IDs instead. The ablation study on the Beauty dataset clearly shows the necessity of both components: removing either Motivation Spreading (w/o Spe.) or Residual Semantic ID (w/o Res.) leads to significant performance drops in all metrics (R@10, M@10, P@10). This confirms that our full model—integrating both mechanisms—achieves superior recommendation accuracy by effectively modeling user motivation and refining semantic representations for long-tail items.

Dataset	Model	SASRec Backbone		
		R@10	M@10	P@10
Beauty	**MoMoREC**	0.1725	0.0951	0.0172
	-w/o Spe.	0.0791	0.0403	0.0052
	-w/o Res.	0.0942	0.0465	0.0094

Table 3: Ablation study results with SASRec Backbone

The Further study in Motivation Generation Module For the Motivation Generation module, we select two competitive models, Qwen2.5-7B-Instruct (Yang et al. 2024) and Qwen3-8B (Yang et al. 2025), employing an identical prompt for both. The outputs are evaluated by the Analysis Agent based on two sub-dimensions including Relevance and Content Depth. The evaluation results are then compared against human annotations to assess inter-rater consistency. A total of 12,000 data instances were annotated, and the detailed results are summarized in Table 4. It can be observed that agents with higher ELO scores exhibit greater consistency with human annotations, indicating a closer alignment with human judgment.

Efficiency of MoMoREC (RQ3)

Inference Cost As shown in Table 5, we evaluated the number of parameters and average inference time across different models. We found that, although the parameter count increased compared to the base model, the increase in inference time was not significant (+7.9%). This suggests that, despite the incorporation of multiple Motivation Semantic ID embeddings, the impact on inference latency is minimal relative to the performance gains, which indirectly demonstrates the efficiency of our proposed model.

Model	ELO	Relevance	Content Depth
Qwen2.5-7B	1044	97.72%	97.87%
Qwen3-8B	1164	97.94%	98.00%

Table 4: Comparison between human-annotated evaluation consistency and ELO scores of different Agents from two aspects

Method	Paras	Runtime
Base	8.89M	550.00
LLMESR	150.06M	861.00 (+56.5%)
MoRec	8.89M	582.50 (+5.9%)
whitenREC	140.73M	883.65 (+60.7%)
LLMInit	8.89M	663.45 (+20.6%)
RLMRec	150.05M	572.97 (+4.2%)
UniSRec	141.22M	862.55 (+56.8%)
AlphaFuse	8.89M	605.38 (+10.1%)
MoMoREC	41.63M	593.54 (+7.9%)

Table 5: Comparison of Model Parameters (paras) and Runtime (millisecond)

Potential for Full Removal of Trainable Parameters We compare our motivation-finetuned model with the unfine-tuned model gte-Qwen2-7B-instruct (Yang et al. 2024) using Pearson and Spearman correlation coefficients. As shown in Table 6, the unfine-tuned model achieves competitive performance, only 4% lower than the fine-tuned counterpart. Compared to the great improvement observed in the main experiments, this marginal 4% degradation in a sub-module's performance is negligible, supporting the potential feasibility of eliminating task-specific fine-tuning entirely.

Model	Pearson	Spearman
gte-Qwen2	0.8061	0.8027
Trained-model	0.8474	0.8450

Table 6: Comparison Between Unfine-tuned Model and Trained Embedding Model

Implementation Details

Motivation Generation

We use Qwen3-8B (Zhang et al. 2025a) as generation model. We select item pairs with a window size of 2 and appearing more than 5 times as input. The generation prompt is as follows:

Generation Module

The title of Item 1 is [ITEM_1_TITLE], and its subcategory is [ITEM_1_CATEGORY].
The title of Item 2 is [ITEM_2_TITLE], and its subcategory is [ITEM_2_CATEGORY].
If product images are provided: The images shown above are the product display pages for Item 1 and Item 2, respectively.
Based on the information above, provide the specific motivations for purchasing these two items. Only output the concrete motivations and reasons. Motivations should be from the consumer's perspective—for example, (consumer) addressing a specific need—rather than from the seller's perspective, such as (seller) providing better service to (consumer). Motivations should not be overly broad, such as "meeting needs" or "providing companionship." They should be highly specific—for example, satisfying a user's golf-playing needs, or providing in-home companionship for the elderly. You should carefully consider: Does the motivation you provide align with the fundamental purpose of the target products? Avoid excessive speculation. For instance, a wine storage jar is generally used for household or industrial wine preservation—it may appear in family gatherings, but that does not make "family gathering" the core motivation. The primary motivation should center on wine storage, regardless of the context.
The output must include the following: "Your judgment of the shared purchase motivation"; "Your reasoning behind this motivation": "Explain in detail why you made this judgment. Your explanation should thoroughly address how your answer satisfies relevance, timeliness, and insightfulness, justifying why this interpretation best meets all three criteria."

And the Summary Prompt can be seen as follows:

Summary Module

There is a piece of text summarizing consumer motivations for purchasing products. However, it may contain irrelevant content or be poorly structured. Please format and summarize the core motivation based on this text.
Below is the input text, which includes both the motivation and reasoning. Only focus on the stated motivation, and ignore any thought process or analysis:

<Text Start>
[output_text]
<Text End>

Based on the above text, provide a summary in the following strict format. Do not output anything else!
The motivation should be:

- As concise as possible — no more than twenty Chinese characters;
- Highly generalized, reflecting the essential nature of the product;
- Focused solely on the most central consumer motivation aligned with the product's actual use;
- Based strictly on the product's attributes — do not invent or generalize broadly (e.g., "meet needs", "provide companionship");
- Output only one single motivation — do not list multiple items.

Follow this exact output format:

Motivation: [Your summarized consumer purchase motivation — no more than twenty characters]

Motivation Spreading

Embedding Model For embedding model, we use Conan-embedding-v1 (Li et al. 2024) as our base model. We trained the model for 3 epochs using the AdamW optimizer with a learning rate of 5×10^{-5} and a linear learning rate scheduler with no warmup. The AdamW parameters were set with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1 \times 10^{-8}$, with no weight decay applied. We used a per-device training batch size of 128 and 4 gradient accumulation steps, resulting in an effective batch size of 512. For reproducibility, the random seed was fixed to 42. To accelerate training and save memory, we enabled FP16 mixed-precision and applied gradient clipping with a maximum norm of 1.0. The model was evaluated every 20 steps, using a per-device evaluation batch size of 2048. The best checkpoint was selected based on the spearman score.

Spreading Module For bucketing, we employ the LSH algorithm with a cosine similarity threshold of 0.8 and a maximum capacity of 25,000 elements per bucket.

For Similarity Module, we use same embedding model in section Embedding Model to get the embedding of items. For each long-tail item, we select the top 3 motivation-associated items and determine the diffusion motive via a majority voting mechanism.

Residual Semantic ID Retrieval We employ a 6-layer Residual Semantic ID module, where each layer has a codebook of shape [64, 1024], indicating that the semantic ID at each layer can take one of 64 possible values.

Further Explore of each Component

In this section, we define an intermediate metric for each sub-module to systematically investigate how to enhance its capability.

Motivation Generation

We compared the performance of various LLMs on our task.

Comparison between the Judgement Agent and Human Annotators We compared the capabilities of two models as judging agents and evaluated the final judging model by comparing its outputs with high-confidence human annotations (i.e., cases where all three annotators reached in agreement). The results show that the choice of the LLM plays

a critical role in the performance of the judging model. Selecting a stronger LLM leads to a higher agreement rate with human annotators.

Judging Model	Agreement Rate
gpt-4o-mini-0718	72.48%
Qwen2.5-7B-Instruct	49.85%

Table 7: Consistency results on annotation data

Comparison of Various Generation Models Here, we perform model inference using our judgement model and five candidate generation models. The results can be found in Table 8:

Candidate Model	ELO Score
chatglm3-6b (GLM et al. 2024)	1149.45
Qwen2.5-7B-Instruct (Yang et al. 2024)	1249.90
Qwen2.5-14B-Instruct	1231.57
DeepSeek-V2-Lite-Chat(Liu et al. 2024a)	1231.08
Ministral-8B-Instruct-2410(Jiang et al. 2023)	1244.31
DeepSeek-R1-Distill-qwen(Liu et al. 2024a)	1093.65
Qwen3-8B(Yang et al. 2025)	1390.93

Table 8: ELO Scores of Various Generation Models

Motivation Spreading

Embedding Model We test different base models and annotated their Spearman and Pearson correlation coefficients to compare the performance of each model.

Model	P.	Sp.
Conan-embedding-v1(Li et al. 2024)	0.853	0.862
chinese-roberta(Cui et al. 2019)	0.851	0.859
xiaobu-embedding-v2(Lier007 2024)	0.852	0.860
zpoint_large_embedding_zh	0.852	0.861
stella-mrl-large-zh-v3.5	0.849	0.863
gte-Qwen2-7B-instruct(Yang et al. 2024)	0.848	0.856

Table 9: Performance of Various Models, where P. means Pearson and Sp. means Spearman

Residual Semantic ID Retrieval

We test different dimension's impact on model performance. As shown in Figure 6, Model performance exhibits a steady improvement with increasing embedding dimensionality, peaking when the dimension is set to 5.

Online Experiment

In this section, we evaluate our motivation embedding on the Taobao 88VIP Best-seller List. The motivation semantic ID is incorporated as an ID feature into a traditional recommendation model. A gating neural network (Chang et al. 2023)

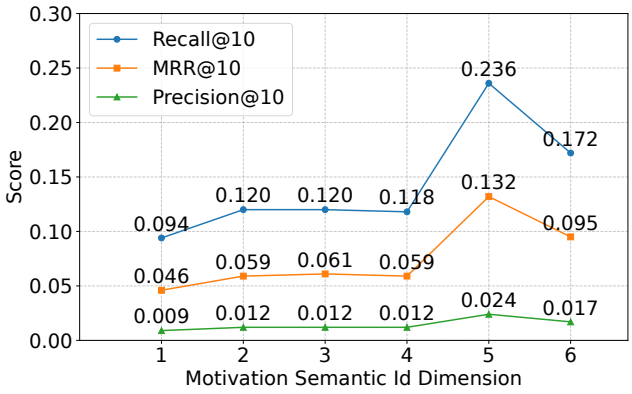

Figure 6: Impact of Motivation Semantic ID Embedding Dimensionality on Model Performance in the Beauty Dataset

is introduced, where the motivation semantic ID serves as the gating signal to dynamically modulate the parameters at each layer. We conduct a 14-day online A/B test. The results, including the Transaction Conversion Rate (TCR), Gross Merchandise Value (GMV), and User Click-Through Rate (UCTR), are summarized in Table 10. As shown in the statistics, although UCTR slightly decreases, both TCR and GMV increase, with GMV showing a more significant improvement.

UCTR	GMV	TCR
-1.3%	+6.3%	+1%

Table 10: Online A/B Test on the Taobao 88VIP Best-seller List: Relative Improvements from Introducing the Motivation Semantic ID Compared to the Online Model

Conclusion

In this work, we propose **MoMoREC**, a novel framework that fundamentally rethinks user purchase behavior from the deeper perspective of motivation. The framework follows a three-stage pipeline: First, the Motivation Generation module introduces, for the first time, a multi-agent paradigm to comprehensively understand and generate user motivations through collaborative agent reasoning. Second, for long-tail items, we design a cluster-based diffusion architecture to effectively attach motivational signals to underrepresented products. Finally, the Residual Semantic ID Retrieval module generates Motivation Semantic IDs that encapsulate high-level motivational representations. Extensive experiments demonstrate that the generated Motivation Semantic IDs significantly enhance the performance of recommendation models, validating their effectiveness and potential in sequential recommendation tasks. Meanwhile, online experiments show that the Motivation Semantic IDs can be seamlessly integrated with traditional recommendation models. It effectively uncovers users' latent interest patterns and improves key business metrics.

References

Alcan, D.; Ozdemir, K.; Ozkan, B.; Mucan, A. Y.; and Ozcan, T. 2022. A comparative analysis of apriori and fp-growth algorithms for market basket analysis using multi-level association rule mining. In *Global Joint Conference On Industrial Engineering And Its Application Areas*, 128–137. Springer.

Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; Fan, Y.; Ge, W.; Han, Y.; Huang, F.; et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Boubdir, M.; Kim, E.; Ermis, B.; Hooker, S.; and Fadaee, M. 2024. Elo uncovered: Robustness and best practices in language model evaluation. *Advances in Neural Information Processing Systems*, 37: 106135–106161.

Chang, J.; Zhang, C.; Hui, Y.; Leng, D.; Niu, Y.; Song, Y.; and Gai, K. 2023. Pepnet: Parameter and embedding personalized network for infusing with personalized prior information. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3795–3804.

Chen, Q.; Qin, L.; Liu, J.; Peng, D.; Guan, J.; Wang, P.; Hu, M.; Zhou, Y.; Gao, T.; and Che, W. 2025. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567*.

Chiang, W.-L.; Zheng, L.; Sheng, Y.; Angelopoulos, A. N.; Li, T.; Li, D.; Zhu, B.; Zhang, H.; Jordan, M.; Gonzalez, J. E.; et al. 2024. Chatbot arena: An open platform for evaluating llms by human preference. In *Forty-first International Conference on Machine Learning*.

Cui, Y.; Che, W.; Liu, T.; Qin, B.; Yang, Z.; Wang, S.; and Hu, G. 2019. Pre-Training with Whole Word Masking for Chinese BERT. *arXiv preprint arXiv:1906.08101*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 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)*, 4171–4186.

Feng, Y.; Lv, F.; Shen, W.; Wang, M.; Sun, F.; Zhu, Y.; and Yang, K. 2019. Deep session interest network for click-through rate prediction. *arXiv preprint arXiv:1905.06482*.

GLM, T.; Zeng, A.; Xu, B.; Wang, B.; Zhang, C.; Yin, D.; Zhang, D.; Rojas, D.; Feng, G.; Zhao, H.; et al. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*.

Guo, T.; Chen, X.; Wang, Y.; Chang, R.; Pei, S.; Chawla, N. V.; Wiest, O.; and Zhang, X. 2024. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*.

Hou, Y.; Li, J.; He, Z.; Yan, A.; Chen, X.; and McAuley, J. 2024. Bridging Language and Items for Retrieval and Recommendation. *arXiv preprint arXiv:2403.03952*.

Hou, Y.; Mu, S.; Zhao, W. X.; Li, Y.; Ding, B.; and Wen, J.-R. 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.

Hu, G.; Zhang, A.; Liu, S.; Cai, Z.; Yang, X.; and Wang, X. 2025. Alphafuse: Learn id embeddings for sequential recommendation in null space of language embeddings. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1614–1623.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Singh Chaplot, D.; de las Casas, D.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. 2023. Mistral 7B. *arXiv e-prints*, arXiv–2310.

Kang, W.-C.; and McAuley, J. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, 197–206. IEEE.

Li, S.; Tang, Y.; Chen, S.; and Chen, X. 2024. Conan-embedding: General text embedding with more and better negative samples. *arXiv preprint arXiv:2408.15710*.

Liao, J.; Xie, R.; Li, S.; Wang, X.; Sun, X.; Kang, Z.; and He, X. 2025. Multi-Grained Patch Training for Efficient LLM-based Recommendation. *arXiv preprint arXiv:2501.15087*.

Lier007. 2024. Xiaobu-Embedding-v2. https://huggingface.co/lier007/xiaobu-embedding-v2. Accessed: 2025-04-05.

Lin, X.; Luo, J.; Pan, J.; Pan, W.; Ming, Z.; Liu, X.; Huang, S.; and Jiang, J. 2024. Multi-sequence attentive user representation learning for side-information integrated sequential recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, 414–423.

Liu, A.; Feng, B.; Wang, B.; Wang, B.; Liu, B.; Zhao, C.; Dengr, C.; Ruan, C.; Dai, D.; Guo, D.; et al. 2024a. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*.

Liu, Q.; Wu, X.; Wang, Y.; Zhang, Z.; Tian, F.; Zheng, Y.; and Zhao, X. 2024b. Llm-esr: Large language models enhancement for long-tailed sequential recommendation. *Advances in Neural Information Processing Systems*, 37: 26701–26727.

Malzer, C.; and Baum, M. 2020. A hybrid approach to hierarchical density-based cluster selection. In *2020 IEEE international conference on multisensor fusion and integration for intelligent systems (MFI)*, 223–228. IEEE.

Moonshot-AI. 2024. moonshot-v1. https://platform.moonshot.cn/.

Newling, J.; and Fleuret, F. 2016. Nested mini-batch k-means. *Advances in neural information processing systems*, 29.

Oliva, G.; Setola, R.; and Hadjicostis, C. N. 2013. Distributed k-means algorithm. *arXiv preprint arXiv:1312.4176*.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744.

Ozaki, R.; and Sevastyanova, K. 2011. Going hybrid: An analysis of consumer purchase motivations. *Energy policy*, 39(5): 2217–2227.

Qu, Z.; Xie, R.; Xiao, C.; Kang, Z.; and Sun, X. 2024. The elephant in the room: rethinking the usage of pre-trained language model in sequential recommendation. In *Proceedings of the 18th ACM Conference on Recommender Systems*, 53–62.

Rajaraman, A.; and Ullman, J. D. 2011. *Mining of massive datasets*. Autoedicion.

Rajput, S.; Mehta, N.; Singh, A.; Hulikal Keshavan, R.; Vu, T.; Heldt, L.; Hong, L.; Tay, Y.; Tran, V.; Samost, J.; et al. 2023. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems*, 36: 10299–10315.

Ren, X.; Wei, W.; Xia, L.; Su, L.; Cheng, S.; Wang, J.; Yin, D.; and Huang, C. 2024. Representation learning with large language models for recommendation. In *Proceedings of the ACM web conference 2024*, 3464–3475.

Sheng, X.-R.; Zhao, L.; Zhou, G.; Ding, X.; Dai, B.; Luo, Q.; Yang, S.; Lv, J.; Zhang, C.; Deng, H.; et al. 2021. One model to serve all: Star topology adaptive recommender for multi-domain ctr prediction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 4104–4113.

Su, J. 2022. Cosent (1): A More Effective Sentence Vector Scheme than Sentence BERT. Blog post.

Wang, R.; Shivanna, R.; Cheng, D.; Jain, S.; Lin, D.; Hong, L.; and Chi, E. 2021. Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In *Proceedings of the web conference 2021*, 1785–1797.

Wu, L.; Zheng, Z.; Qiu, Z.; Wang, H.; Gu, H.; Shen, T.; Qin, C.; Zhu, C.; Zhu, H.; Liu, Q.; et al. 2024. A survey on large language models for recommendation. *World Wide Web*, 27(5): 60.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; et al. 2024. Qwen2. 5 Technical Report. *arXiv preprint arXiv:2412.15115*.

Yuan, Z.; Yuan, F.; Song, Y.; Li, Y.; Fu, J.; Yang, F.; Pan, Y.; and Ni, Y. 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.

Zhang, L.; Zhou, X.; Zeng, Z.; and Shen, Z. 2024a. Are id embeddings necessary? whitening pre-trained text embeddings for effective sequential recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, 530–543. IEEE.

Zhang, L.; Zhou, X.; Zeng, Z.; and Shen, Z. 2024b. Dual-view whitening on pre-trained text embeddings for sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 9332–9340.

Zhang, Y.; Li, M.; Long, D.; Zhang, X.; Lin, H.; Yang, B.; Xie, P.; Yang, A.; Liu, D.; Lin, J.; et al. 2025a. Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models. *arXiv preprint arXiv:2506.05176*.

Zhang, Y.; Qiao, S.; Zhang, J.; Lin, T.-H.; Gao, C.; and Li, Y. 2025b. A survey of large language model empowered agents for recommendation and search: Towards next-generation information retrieval. *arXiv preprint arXiv:2503.05659*.

Zhang, Y.; Wang, R.; Cheng, D. Z.; Yao, T.; Yi, X.; Hong, L.; Caverlee, J.; and Chi, E. H. 2023. Empowering long-tail item recommendation through cross decoupling network (cdn). In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 5608–5617.

Zhao, Z.; Zhou, K.; Wang, X.; Zhao, W. X.; Pan, F.; Cao, Z.; and Wen, J.-R. 2023. Alleviating the long-tail problem in conversational recommender systems. In *Proceedings of the 17th ACM conference on recommender systems*, 374–385.

Zhou, G.; Zhu, X.; Song, C.; Fan, Y.; Zhu, H.; Ma, X.; Yan, Y.; Jin, J.; Li, H.; and Gai, K. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 1059–1068.