Article

A Survey of Generative Recommendation from a Tri-Decoupled Perspective: Tokenization, Architecture, and Optimization

Xiaopeng Li [†] , Bo Chen [†] , Junda She , Shiteng Cao , You Wang , Qinlin Jia , Haiying He , Zheli Zhou , Zhao Liu , Ji Liu , Zhiyang Zhang , Yu Zhou , Guoping Tang , Yiqing Yang , Chengcheng Guo , Si Dong , Kuo Cai , Pengyue Jia , Maolin Wang , Wanyu Wang , Shiyao Wang , Xinchen Luo , Qigen Hu , Qiang Luo , Xiao Lv , Chaoyi Ma , Ruiming Tang [*] , Kun Gai , Guorui Zhou [*] , Xiangyu Zhao [*]

Posted Date: 4 December 2025

doi: 10.20944/preprints202512.0203.v1

Keywords: generative recommendation; generative models; tokenization; preference alignment

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 December 2025 doi:10.20944/preprints202512.0203.v1

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

A Survey of Generative Recommendation from a Tri-Decoupled Perspective: Tokenization, Architecture, and Optimization

Xiaopeng Li [1,*], Bo Chen [2,*], Junda She [2], Shiteng Cao [2], You Wang [2], Qinlin Jia [2], Haiying He [1], Zheli Zhou [2], Zhao Liu [2], Ji Liu [2], Zhiyang Zhang [2], Yu Zhou [2], Guoping Tang [2], Yiqing Yang [2], Chengcheng Guo [2], Si Dong [2], Kuo Cai [2], Pengyue Jia [1], Maolin Wang [1], Wanyu Wang [1], Shiyao Wang [2], Xinchen Luo [2], Qigen Hu [2], Qiang Luo [2], Xiao Lv [2], Chaoyi Ma [2], Ruiming Tang [2,†], Kun Gai [3], Guorui Zhou [2,†] and Xiangyu Zhao [1,†]

[1] City University of Hong Kong
[2] Kuaishou Technology
[2] Unaffiliated
[*] Correspondence: tangruiming@kuaishou.com; zhouguorui@kuaishou.com; xianzhao@cityu.edu.hk
[†] These authors contributed equally to this work.

Abstract

The recommender systems community is witnessing a rapid shift from multi-stage cascaded discriminative pipelines (retrieval, ranking, and re-ranking) toward unified generative frameworks that directly generate items. Compared with traditional discriminative models, generative recommender systems offer the potential to mitigate cascaded error propagation, improve hardware utilization through unified architectures, and optimize beyond local user behaviors. This emerging paradigm has been catalyzed by the rise of generative models and the demand for end-to-end architectures that significantly improve Model FLOPS Utilization (MFU). In this survey, we provide a comprehensive analysis of generative recommendation through tri-decoupled perspective of tokenization, architecture, and optimization, three foundational components that collectively define existing generative systems. We trace the evolution of tokenization from sparse ID- and text-based encodings to semantic identifiers that balance vocabulary efficiency with semantic expressiveness; analyze encoder–decoder, decoder-only, and diffusion-based architectures that increasingly adopt unified, scalable, and efficient backbones; and review the transition from supervised next-token prediction to reinforcement learning–based preference alignment enabling multi-dimensional preference optimization. We further summarize practical deployments across cascade stages and application scenarios, and examine key open challenges. Taken together, this survey is intended to serve as a foundational reference for the research community and as an actionable blueprint for industrial practitioners building next-generation generative recommender systems. To support ongoing research, we maintain a living repository https://github.com/Kuaishou-RecModel/Tri-Decoupled-GenRec that continuously tracks emerging literature and reference implementations.

Keywords: generative recommendation; generative models; tokenization; preference alignment

1. Introduction

Recommender systems have become fundamental infrastructure in modern digital ecosystems, serving billions of users and managing catalogs containing tens of millions of items across diverse domains such as e-commerce [1], streaming media [2], music [3], social networks [4], etc. By analyzing user behavior patterns, item characteristics, and contextual signals, recommender systems provide rich personalization services that enhance user engagement, satisfaction, and platform value. The predominant paradigm in both academia and industry has long been discriminative models, which have demonstrated remarkable success in learning complex user-item interaction patterns and have become the de facto standard for large-scale recommendation deployment.

Modern discriminative recommendation models employ a scoring-based framework that processes item, user, and context features to predict engagement probabilities such as clicks, likes, or purchases. These models follow an "embedding & MLP" paradigm [6], where input features are first encoded into dense representations and then processed through neural architectures designed to capture complex feature interactions and the evolution of user behaviors [7,8]. The resulting prediction scores are optimized to discriminate between positive and negative samples while maintaining ranking quality. In practice, industrial recommender systems evolved into a cascaded discriminative framework to handle millions of items under strict latency constraints, which operates through multiple complementary stages: recall, pre-ranking, ranking, and re-ranking. This hierarchical approach refines millions of items to thousands, then dozens [9], producing nuanced rankings aligned with individual user preferences.

Despite their widespread adoption and success, discriminative models face several fundamental limitations that constrain their effectiveness. First, these models encounter significant challenges at the embedding level. By treating items as atomic units within embedding tables, they create semantic isolation [10] that exacerbates cold-start problems [11,12] while simultaneously introducing computational inefficiencies. These embedding tables consume approximately more than 90% of parameters, which are inherently sparse and storage-intensive [13,14]. Second, the discriminative architecture also presents significant challenges. A variety of specialized, small-scale operators can incur considerable communication and data transfer overhead [15–17]. Existing models suffer from severely limited hardware utilization efficiency, with Model FLOPS Utilization (MFU) typically less than 5% [15], thus missing out on the benefits of hardware computing power growth. This represents a stark contrast to Large Language Models, which achieve >40% MFU during training [18], highlighting a fundamental inefficiency in current recommendation architectures. Additionally, production recommender systems typically employ modest-sized models (dense MLP<0.1B) [7], which fundamentally constrains their capacity for scaling up and prevents them from exhibiting the emergent capabilities [19] observed in LLMs. Third, current approaches predominantly rely on discriminative training strategies [20,21] to optimize local decision boundaries with respect to users' posterior behaviors, lacking explicit characterization of full probability distribution over items and multi-dimensional preferences modeling (e.g., platform-level). Moreover, the multi-stage cascade caused by discriminative paradigm inevitably introduces cumulative errors, with progressive information loss degrading recommendation quality [22,23].

Recent advancements in Large Language Models (LLMs) have demonstrated exceptional semantic understanding and reasoning capabilities [24,25], prompting researchers to explore their application in recommender systems [26,27]. However, despite these advances, current LLM-enhanced approaches remain fundamentally constrained by the discriminative paradigm. Generative Recommendation (GR) represents a fundamental paradigm shift and gains attention rapidly in recent years, which is shown in Figure 1. Rather than scoring and ranking items within candidate sets, GR directly generates item identifiers through generative models, eliminating the need for multi-stage cascaded processing. As seen in Figure 2, compared to discriminative models, the generative recommendation paradigm demonstrates advantages three multiple dimensions: tokenization, architecture, and optimization.

First, generative recommendations revolutionize tokenization by operating at the semantic level rather than relying on traditional embedding approaches. Instead of embedding input features into dense vectors for downstream processing, these models utilize textual [28] or semantic identifiers [29] for feature extraction, enabling rich semantic modeling through unified vocabulary representations, which effectively addresses cold-start and cross-domain challenges that have long plagued discriminative systems [30,31]. Additionally, semantic modeling enables a more compact vocabulary design [29], achieving remarkable parameter efficiency compared to the redundant embedding table architectures of discriminative models.

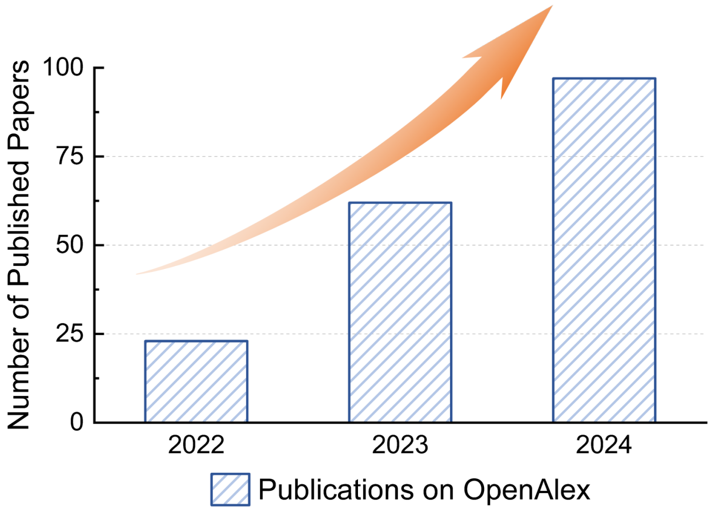
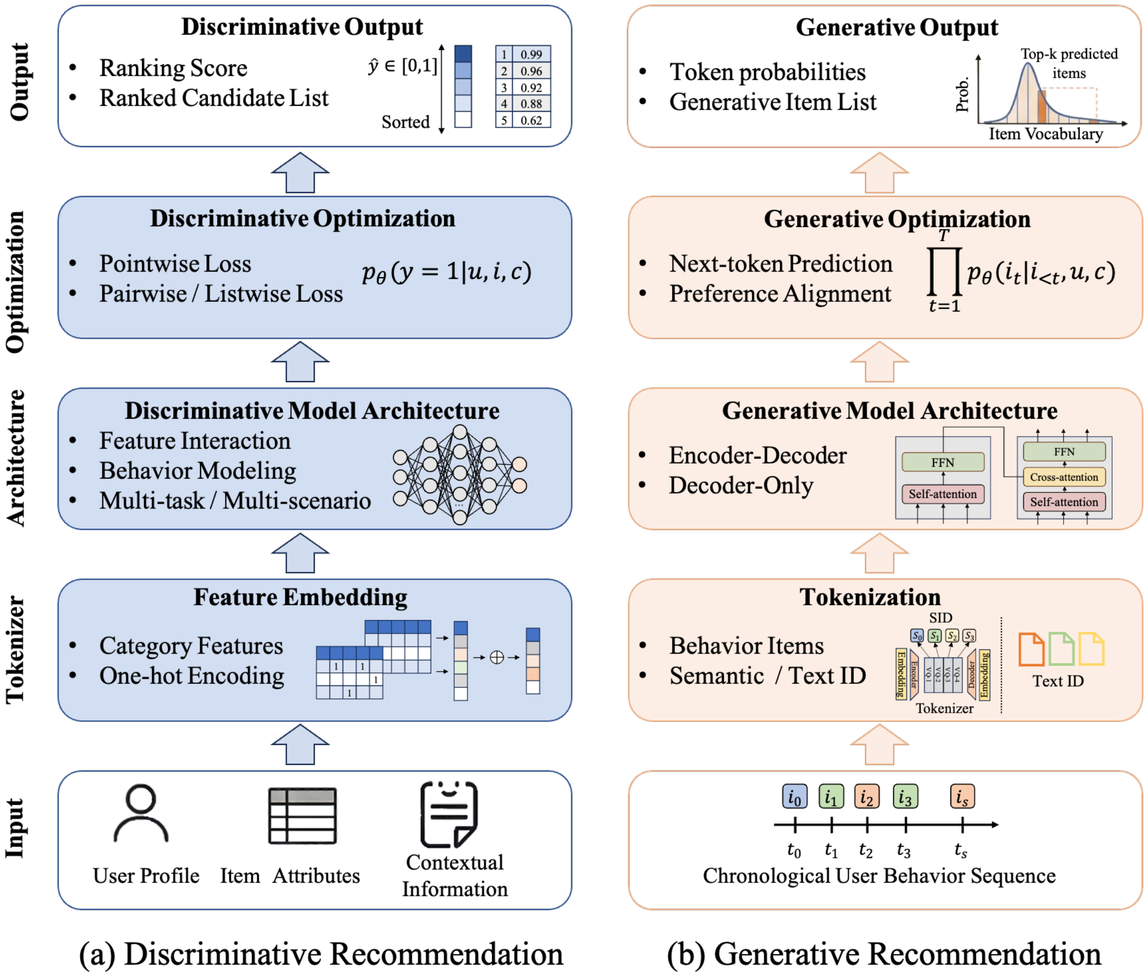


Figure 1. Number of publications on generative recommendation indexed in OpenAlex [5]. Results obtained through keyword search for “Generative Recommendation” within the topic area “Recommender Systems and Techniques”.



(a) Discriminative Recommendation (b) Generative Recommendation

Figure 2. Comparison of discriminative and generative recommendation paradigms.

Second, from an architectural design perspective, generative recommender systems typically employ generative model structures (e.g., encoder-decoder [29,32] and decoder-only architectures [33,34]). Furthermore, these generative architectures possess inherent model size scalability advantages, demonstrating enhanced capability to resolve complex recommendation scenarios as model size increases [35,36]. This architectural choice also provides promising potential for achieving higher MFU [33,36], thereby maximizing hardware computational efficiency. More importantly, by adopting generative paradigms aligned with mainstream NLP developments, recommender systems can seamlessly leverage the full spectrum of methodological innovations from the NLP community, enabling them to evolve synchronously with the rapidly advancing NLP field.

Third, the training methodology represents a fundamental departure from discriminative approaches. Generative models trained with Next-Token Prediction (NTP) [29,37] naturally capture the full probability distribution over items and model the entire user behavior generation process, rather than merely learning local decision boundaries. Besides, preference alignment strategies are also introduced, particularly reinforcement learning based techniques [15,38], enabling direct alignment with users' multidimensional preferences and platform-level objectives, creating a more holistic optimization framework that effectively balances the satisfaction of multiple stakeholders. Moreover, the generative training paradigm further enables end-to-end optimization [15,33], thereby preventing the cumulative information loss inherent in discriminative cascaded systems.

Despite significant advances in generative recommender systems, the research community lacks a comprehensive survey that systematically examines this field from a generative perspective, nor does existing literature deeply discuss the technological advancements and hardware constraints. We address this gap through a systematic survey and identify three fundamental dimensions: tokenization toward more conflict-free, efficient, and multimodal content representation, architectures that scales in accordance with complex scenario resolution requirements, and optimization that comprehensively balances multi-dimensional user preferences, platform performance metrics, and content provider interests.

Our contributions are threefold:

1. We present the first comprehensive survey that analyzes generative recommender systems through a tri-dimensional decomposition encompassing tokenization, architectural design, and optimization strategies, within which we organize existing work and trace the evolution of recommender systems from discriminative approaches toward the generative paradigm.
2. Through a systematic overview and analysis, we identify key trends toward efficient representation with semantic identifiers that balance vocabulary compactness and semantic expressiveness, advances in model architecture that facilitate improved scalability and resource-efficient computation, and multi-dimensional preference alignment aimed at balancing the objectives of users, the platform, and additional stakeholders.
3. We provide an in-depth discussion of its applications across different stages and scenarios, examine the current challenges, and outline promising future directions. We hope this survey will serve as a practical reference and blueprint for researchers and practitioners in both academia and industry.

We distinguish our survey from prior works in this area. Several surveys [39–41] are LLM-centered, reviewing LLM usage and enhancement across various recommendation tasks with coverage extending through 2024, and [42] extends coverage to 2025, but similarly adopts an LLM-centric perspective. Other surveys examine the field from alternative dimensions, such as diverse architectures and modalities [43], unified search and recommendation systems [44], and diffusion-based models [45]. The central focus of these works remains the LLM itself, encompassing both the LLM4Rec and LLMasRec paradigms. More recently,[46] discusses generative recommendation with an emphasis on industrial application stages, and [47] organize their analysis at the pipeline level, tracing the flow from data, feature representation, architectures, information fusion, and evaluation. In contrast to existing works, our survey adopts a fundamentally different paradigm by conceptualizing generative recommendation as an independent framework rather than an LLM-enhanced approach. We emphasize foundational elements from a tri-decoupled perspective—input tokenization, architecture modeling, and optimization—treating these as essential building blocks of a unified generative framework. This intrinsic generative modeling perspective enables comprehensive methodological coverage and illuminates future development trajectories.

The remainder of this survey is organized as follows. Section 2 introduces the background of evolutionary trajectory from discriminative to generative recommendation. Section 3 through 5 systematically examines the three core components of the generative recommendation paradigm: input tokenization, architecture design, and optimization strategies. Section 6 explores practical applications across various industrial stages and scenarios. Section 7 identifies current challenges and outlines promising future research directions. Finally, Section 8 concludes the survey with a synthesis of key insights and implications for the field.

2. Background and Preliminary

In this section, we begin by comparing and presenting an overview of discriminative and generative recommendation models. We then proceed to discuss generative recommendation models from multiple dimensions and analyze the advantages they offer.

2.1. Discriminative Recommendation

Discriminative recommendation models address the problem of distinguishing which items are more likely to be selected by users, given a candidate item set. For a user u, context c, and a set of candidate items $\mathcal{I}$, discriminative recommendation models learn a conditional model that scores items or estimates interaction probabilities:

$$f_\theta(u,i,c) \sim p_\theta(y=1 \mid u,i,c) \quad i \in \mathcal{I} \tag{1}$$

where y indicates whether the user u interacts with item i (e.g., clicks, purchases), and $f_\theta(\cdot)$ is a prediction model parameterized by θ.

According to the evolution of model complexity and representation capability, discriminative models can be largely divided into Machine Learning (ML)-based models and Deep Learning (DL)-based models. ML-based methods include similarity-based collaborative filtering methods, which estimate target users' preferences for uninteracted items by mining similarities users or items [48,49] (such as User-CF and Item-CF) and matrix factorization methods that decompose the interaction matrix into user vectors and item vectors to characterize the implicit relationships between users and items [50,51]. DL-based methods follow the "Embedding & MLP" paradigm. Features are first encoded into dense vectors, followed by a feature interaction module using an MLP backbone, which includes several representative types: 1) Feature interaction for high-order feature interactions modeling [7]. 2) Behavior modeling for short and long-term behavior patterns mining from users' temporal behavior sequence [52]. 3) Multi-task/scenario modeling for jointly optimizing multiple objectives [53] (e.g., CTR, CVR, dwell time, etc.) in different scenarios [54] (e.g., Ad slots, service platforms, etc.).

Moreover, LLM-enhanced recommendation models can further exploit the rich semantic knowledge and reasoning capabilities acquired by LLMs to better serve recommendation tasks, which can be categorized into three major classes: 1). Semantic Enhancement [26,27] , 2). Data Enhancement [55,56], and 3). Alignment Enhancement [57,58]. However, the discriminative paradigm struggles to fundamentally overcome inherent limitations, such as inefficient and semantically deficient tokenization, highly customized architectures with low MFU, and inherently local and constrained optimization objectives.

2.2. Generative Recommendation

Generative Recommendation (GR) can be viewed as an attempt to reformulate the recommendation task as a sequence generation problem: instead of merely scoring a predefined candidate set, GR explicitly models the generative process based on user-item interaction sequences, thereby directly generating recommended results. Formally, given a user u, context c, and their historical interaction sequence $i_{1:T} = (i_1, \ldots, i_T)$, the objective of generative recommendation can be defined as:

$$p_\theta(i_{1:T} \mid u, c) = \prod_{t=1}^{T} p_\theta(i_t \mid i_{<t}, u, c) \quad i \in \mathcal{I} \tag{2}$$

where i_t denotes the item with which the user interacts at timestep t, $i_{<t} = (i_1, \ldots, i_{t-1})$ represents the historical interaction prefix sequence. The model learns the generative distribution of user behavior sequences by maximizing the above conditional probability.

At the tokenenization level, unlike LLMs that use subword-based vocabularies, generative recommendation primarily treats a user's behavior sequence as tokenized corpus and generates the token sequence during prediction for grounding the next item in the real item space. Due to the dependency inherent in autoregressive architectures, GR models typically require tokenizers with semantic coherence and inter-token relationships, such as text-based tokenizers [59,60] used in LLMs or SID-based tokenization methods [15,29].

In terms of model architecture, generative recommendation models are primarily based on encoder–decoder and decoder-only structures. Beyond directly inheriting the backbone and pre-trained parameters from existing LLMs [34,61], several studies also design tailored architectures specifically for recommendation scenarios [35,37]. Given that recommendation inputs predominantly include user behavior sequences, GR models typically employ transformers' causal self-attention and feed-forward mechanisms to capture sequential dependencies. Moreover, to effectively integrate user profiles and contextual information as well as capture interaction signals, cross-attention and customized task-specific attention are commonly utilized to boost personalization and real-time preference modeling [15,35].

Regarding optimization strategies, the supervised training approach of next-token prediction in recommendation tasks primarily focuses on predicting the next item or next action based on a user's historical behavior [37,62]. The model approximates the posterior behavioral distribution of users by maximizing $\prod_{t=1}^{T} p_\theta(i_t \mid i_{<t}, u, c)$, thereby characterizing the evolution of user preferences. Moreover, Reinforcement Learning (RL) is also incorporated to optimize the GR based on multidimensional personalized rewards and platform-side metrics (e.g., diversity, fairness, etc.) [63,64].

Through the meticulous design of these three components, generative models have become a promising direction with significant development potential in the recommendation system field. We will present these modules of generative models in detail in Sections 3–5.

3. Tokenizer

Tokenization in LLMs is the process of splitting text into smaller units called tokens, (e.g., words, subwords, or characters) that the model can understand and process. Each token is mapped to a corresponding embedding, enabling the model to perform both comprehension and generation tasks. In generative recommendation (GR), tokenization process refers to how features, particularly items, are represented as discrete fundamental units. Similarly to language models, an effective tokenizer much balance semantic expressiveness with vocabulary size. Additionally, since the GR models must ultimately generate specific items, a central consideration in tokenization is how to accurately and efficiently ground the generated token sequence back to actual items.

Recent research can be categorized into three classes based on how items are represented: sparse ID-based, text-based, and semantic ID (SID)-based approaches. Sparse ID-based approaches follow conventional recommendation methods, where each item is represented by a randomly assigned sparse ID. These IDs carry no semantic information and result in an extremely large vocabulary, often on the order of hundreds of millions of tokens. Text-based methods represent items through their textual descriptions and directly leverage the existing vocabulary of LLMs, typically tens of thousands, by reframing recommendation as a question-answering task. This method enables more effective use of the world knowledge and reasoning capabilities of LLMs. However, a key challenge is that textual descriptions may not uniquely ground a specific item. Semantic ID-based approaches effectively address the limitations of both aforementioned methods. They provide compact representations with a controllable vocabulary size, which typically tens of thousands, while retaining substantial semantic information, thus enabling accurate grounding to specific items. Figure 3 illustrates the evolution of these three tokenization approaches in recent years. As shown, SID-based methods have gradually emerged as the dominant paradigm for generative recommendation since 2025.

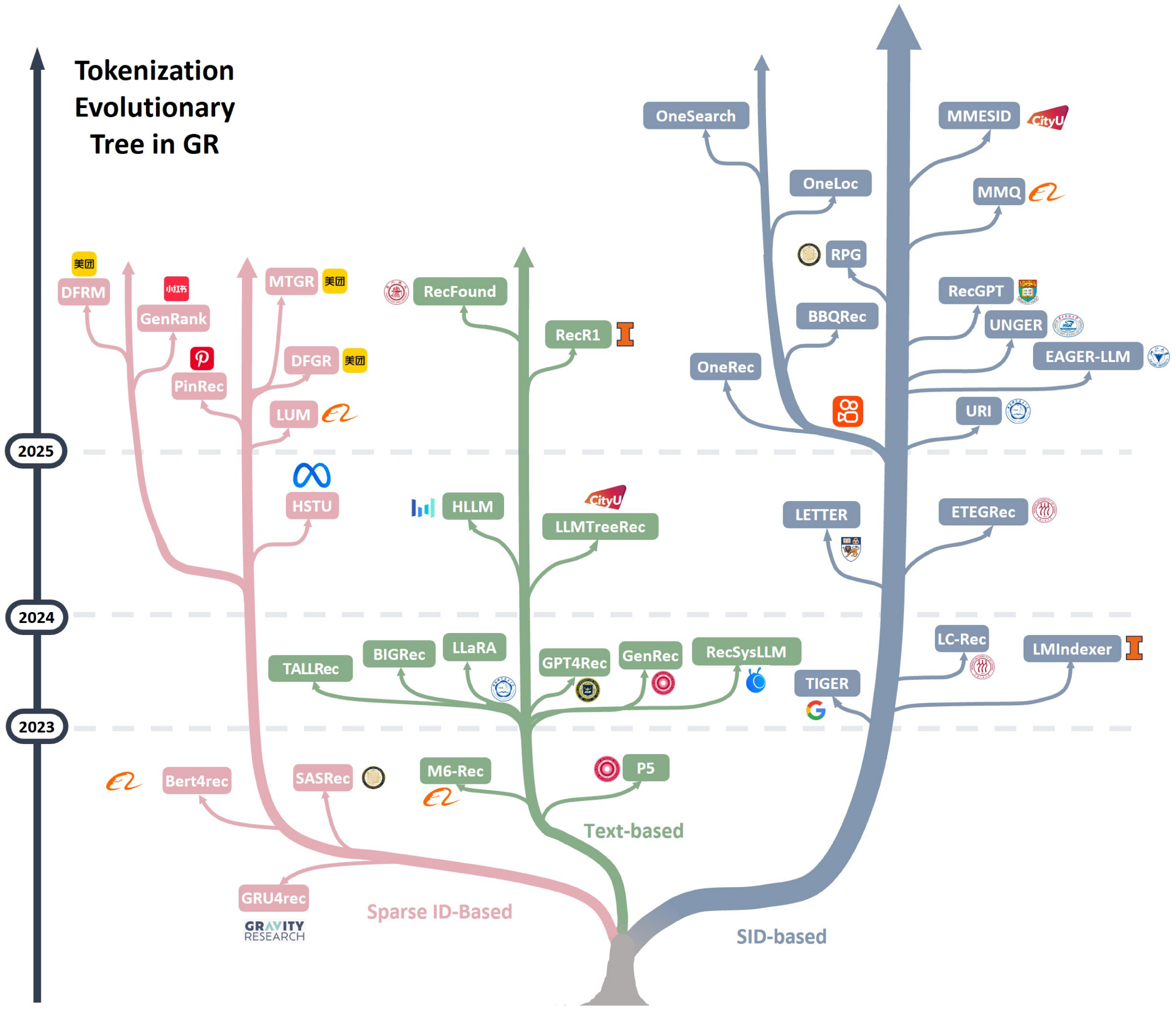


Figure 3. The evolution of tokenizer paradigms, showing the transition from sparse ID-based and text-based models toward semantic ID approaches, with affiliations noted beside each work.

3.1. *Sparse ID-Based IDENTIFIERS*

Sparse IDs form the foundation of traditional discriminative recommendation methods, which follow the "embedding & MLP" modeling paradigm [6]. Specifically, the embedding layer assign an independent parameterized vector to each sparse ID (e.g., user ID, item ID). The MLP layers then operate on these feature embeddings, employing various feature interaction networks (e.g., DCN [8]) and behavior modeling networks (e.g., DIN [52]) to capture feature co-occurrence relationships and generate personalized recommendations.

Sparse ID-based identifiers offer two key advantages: 1) Avoiding ID collision with a unique ID for each item. 2) Facilitation of direct representation of diverse features and feature interaction networks to learn co-occurrence relationships. Consequently, several generative methods directly adopt sparse ID-based tokenization [35,65,66]. To fully exploit these advantages, recent generative models convert user behaviors into chronologically ordered token sequences and reformulate recommendation as a sequential transduction task using causal autoregressive modeling.

HSTU [35] abandons traditional numerical features, introduces extra "action" tokens, and builds sequences by interleaving sparse IDs of items and actions in the form $[\text{item}_1, \text{act}_1, \ldots, \text{item}_n, \text{act}_n]$. This token sequence design unifies retrieval and ranking tasks into a single sequence modeling framework, enabling the generative model to predict either the next item or the next action based on the contextualized sequence. Furthermore, HSTU enhances the transformer architecture to improve its capacity for modeling sequential interactions, achieving substantial performance gains in industrial applications. Based on the foundation of HSTU, several works have explored the sparse ID-based generative recommendation. MTGR [65] further introduces rich interaction features and cross features of users and items as sparse IDs, thus improving ranking performance. LUM [67] incorporates "conditions" as special tokens to build sequences, i.e., $[\text{cond}_1, \text{item}_1, \ldots, \text{cond}_n, \text{item}_n]$. By assigning conditional tokens as search queries, scenarios, or categories, LUM can obtain various user interests. Similarly, PinRec [66] also integrates "condition" as sparse tokens, with the distinction is that the these conditions are user outcomes or behavior. (e.g., click or repin.) The outcome conditions are able to control the generation of outcome-specific and personalized item representations, aligning with business objectives.

Besides, GenRank [62] argues that HSTU introduces substantial overhead for ranking because the sequence with item IDs and action IDs interleaved is twice the length. To address the efficiency limitation, GenRank combines item tokens with action tokens by treating items as positional information and focuses on iteratively predicting the actions associated with each item, which is referred to as the action-oriented organization. In this paradigm, actions become the fundamental units in sequence generation, while items serve as contextual information, thus reducing the sequence length remarkably. Similarly, DFRM [68] also treats the item and action as a single token, concatenating the item ID embedding and action ID embedding to form a unified token. For items undergoing training, the action is replaced with a fake action to prevent information leakage.

However, although the above generative models have achieved great success, the sparse ID-based tokenizer has several limitations: 1). Lack of multimodal semantic information: IDs are assigned randomly and thus devoid of any inherent semantic meaning. 2). Cold-start problem: Since sparse ID embeddings are learned from interaction data, long-tail and cold-start items with insufficient interactions suffer from inadequate feature learning. 3). Vocabulary explosion: The enormous item vocabulary leads to an excessively large output space for next-item prediction, making it particularly challenging for generative modeling methods to adapt effectively.

3.2. Text-Based Identifiers

The sparse ID-based on collaborative information assigns random, discrete identifiers to items without any semantic information. Consequently, the model must learn representations solely from collaborative data, posing challenges for long-tail and cold-start items. To overcome this limitation, text-based tokenizers utilize pre-trained vocabularies of LLMs to represent items through their natural language descriptions, which allows recommendation tasks to be seamlessly integrated within the text understanding and generation paradigms of LLMs.

LLMs are pretrained on extensive text corpora, establishing robust performance within the dense and continuous semantic space formed by natural language. By representing items via textual attributes (e.g., title "iPhone 17 Pro") or structured templates (e.g., "Product: iPhone; Brand: Apple; Category: Electronics"), LLMs leverage eheir extensive world knowledge and reasoning capabilities to infer user preferences. This approach significantly alleviate cold-start and long-tail issues prevalent in traditional recommender systems while enabling cross-domain generalizability and few-shot or zero-shot recommendation capabilities. Additionally, it provides enhanced interpretability and conversational interaction capabilities, making it a promising direction for next-generation interactive recommendation systems.

M6-Rec [69] investigated controllable text generation schemes from the product perspective. In e-commerce scenarios, M6-Rec uses product attributes and descriptions to populate natural language templates, thereby tokenizing product from user interaction history. In contrast, LLMTreeRec [70] places greater emphasis on the hierarchical structure of product attributes. It organizes attribute information in a tree structure, which constrains generation results to maintain consistency within the same hierarchy and avoids the excessive text length that template filling can produce. TallRec [71] explores the Text ID scheme by representing items in an "attribute name: attribute value", facilitating textual item representation. BIGRec [72] employs a two-step approach. It first uses SFT to align the LLM's generation with the recommendation space, and then computes the L2 distance between actual and generated items for re-ranking, ensuring the output aligns with recommendable products. S-DPO [38] takes user interaction histories as text prompts and predicts the title of the target item. During S-DPO phrase, it optimizes the probability of positive samples given both positive and negative examples. Moreover, with the advancement of Multimodal Large Language Models (MLLMs) capabilities, leveraging MLLMs to summarize textual representations of items, rather than simply concatenating raw item attributes, has emerged as another promising direction for Text ID tokenization [73].

However, purely text-based approaches presents several challenges. The same item may be parsed into multiple different tokens, complicating the construction of collaborative relationships such as "*Item_A-Item_B*" within the model's parameter space. Therefore, several works [28,74] introduce entity representations that span multiple original tokens, thereby preserving the completeness of attribute information. LLaRa [75] has recognized the absence of collaborative signals, thus incorporating item representations obtained from traditional recommendation models with the inherent textual attributes of items during training. The mixed features are then used to progressively fine-tune the LLMs using LoRA [76], thereby enabling the recommendation model to integrate collaborative and semantic information.

However, text-based tokenization methods also have the following limitations: 1). Text-based item descriptions require a large number of tokens, thereby reducing computational efficiency. 2). Generated text tokens may not be grounded to actual items effectively, leading to ambiguity and inaccuracies in recommendations.

3.3. SID-Based Identifiers

Sparse ID-based methods suffer from limited semantics and a sparse vocabulary, while text-based methods face challenges of inefficient representation and difficulty in item grounding. To address these limitations, semantic ID (SID)-based methods are proposed, which represent an item using a fixed-length sequence of correlated semantic IDs, avoiding vocabulary explosion and enabling more efficient representation. In the following, we will present how SID are constructed and realeated challenges.

3.3.1. Semantic ID Construction

The construction of semantic IDs proceeds through a two-step process. In the first step, items' semantic information (e.g., textual information or image) is transformed into a semantic embedding through pre-trained embedding models, such as BERT [77] for text and CLIP [78] for multi-modal. In the second step, these semantic embeddings are quantized into semantic ID sequences via quantization methods [79–81], a tuple of codewords (c^0, c^1, c^2), where each codeword originates from a distinct codebook. To illustrate the quantization process, we examine RQ-VAE [80], the most widely adopted method as an example. For each item, its semantic information is first encoded into a semantic representation $\boldsymbol{z}$. The quantizer then performs multi-level quantization. At each level l, the algorithm identifies the closest code vector from the codebook $\{\boldsymbol{v}_k^l\}_{k=1}^K$ to the current latent representation input $\boldsymbol{z}$:

$$c^l = \arg\min_k ||\boldsymbol{z} - \boldsymbol{v}_k^l||_2^2, \tag{3}$$

and then the residual $\boldsymbol{r}^{l+1} = \boldsymbol{z} - \boldsymbol{v}^l_{c^l}$ is used as input for the subsequent level of quantization, and this process continues iteratively until all L levels are completed. In the following sections, we describe in detail how different embedding and quantization methods are employed in current research.

Embedding Extraction. As the initial step, embedding extraction determines what types of information should be incorporated into the process. TIGER [29] and LC-Rec [82] generate embeddings solely from static item-side content features (e.g., title, description, images), which ignore the collaborative signals between users and items that are crucial in recommendation scenarios. To solve the problem, several models such as LETTER [37], EAGER [83], OneRec [15,32], and UNGER [84] further inject collaborative signals and jointly learn collaborative and semantic cross-modality embeddings. In particular, for some location-based recommendation scenarios, the characteristics of the location are vital. Therefore, methods like OneLoc [85] and GNPR-SID [86] further inject geographical information into the embeddings. Additionally, TokenRec [87] explores the construction and quantization of embeddings using purely collaborative signals, which employs a GNN [88] to capture user–item interactions and produces collaborative-based embeddings for subsequent quantization.

Quantization. Various quantization methods have been proposed to address different scenarios in the quantization stage. The most widely adopted approach is residual-based quantization, which constructs a coarse-to-fine representation by quantizing the residual between the latent embedding and the cluster centroid. RQ-VAE is the most popular quantization approach and is widely used by TIGER [29], LC-Rec [82], COBRA [89], GFlowGR [90], STREAM-Rec [91], AtSpeed [92], SpecGR [93], RecBase [94]. Residual quantization naturally aligns with the auto-regressive decoding paradigm of LLMs because the generation of coarse-to-fine semantic IDs effectively shrinks the search space at each decoding step. To prevent codebook collapse during RQ-VAE training, OneRec [32] and OneLoc [85] adopt ResKmeans [95]. When clustering the residual vectors, they limit the maximum number of items that can be assigned to any codeword, thereby boosting both codebook utilization and stability. However, progressive residual can also produce the hourglass effect [96], where codebook tokens in intermediate layers become excessively concentrated, potentially introducing bias into downstream models. Additionally, residual SID generation in LLMs exhibits prefix dependency during inference. Specifically, each subsequent token can only be decoded after the previous SID token has been generated, which limits decoding efficiency.

Consequently, some works turn to parallel quantization approaches, which predict multiple IDs simultaneously to enable improved semantic modeling and more efficient generation. This approach is well-suited to parallel training and generation paradigms. For example, RPG [97] builds ultra-long SIDs using Product Quantization (PQ) [81] for fine-grained semantic modeling and combines parallel decoding to boost inference efficiency. To enhance fine-grained modeling, RecGPT [60] integrates finite scalar quantization (FSQ) [79] with a hybrid attention mechanism, i.e., cross-attention for two SID tokens within one item while causal-attention for those between two different items.

To extend generative recommendation across domains, some studies investigate the construction of semantic IDs in cross-domain settings. GMC [98] employs contrastive learning to enhance the representational consistency of items within the same domain, while RecBase [94] adopts curriculum learning to enhance the cross-domain representation capability from coarse to fine at the domain level.

3.3.2. Challenges for Semantic ID

Although different quantization techniques exhibit distinct advantages, they also share several fundamental challenges.

SID Collision. The first challenge is the SID collision [29,37,54,63,99], where multiple distinct items are mapped to identical SID sequences. This collision introduces ambiguity during item grounding, as the SID sequence no longer uniquely identifies a single item, thereby necessitating additional disambiguation strategies. Consequently, high collision rates lead to significant degradation in generative recommendation model performance. Collisions arise from inherent limitations of quantization methods: in RQ-VAE or ResKmeans, the learned centroids often become unevenly distributed or collapse, with most items clustering around a few dominant centroids while tail centroids receive little to no assignments, resulting in low codebook utilization. To mitigate the collision problem, recent work introduces additional optimization objectives during codebook training to encourage more balanced centroid distributions. SaviorRec [100] incorporates the Sinkhorn algorithm [101] during quantization, employing an entropy-regularized loss to enforce a more uniform assignment of items to cluster centroids. OneRec [32] and LETTER [37] leverage the constrained k-means for each residual, which limits the maximum number of items that can be assigned to each centroid. Beyond training-time de-collision strategies, another line of work alleviates collisions by adding additional token positions at the final layer. TIGER [29] adds a random token at the end of SID, while CAR [99] adds the item sparse ID. Furthermore, OneSearch [63] uses ResKmeans to encode the shared characteristics of items, and further uses optimized product quantization (OPQ) to encode the unique characteristics in the final SID token, thus improving the distinctiveness of SID.

Objective Inconsistency. The second issue is the objective inconsistency that arises from multi-stage training in generative recommendation, which typically comprises three stages: embedding extraction, SID quantization, and generative model training. Insufficient inter-stage interaction and perceptual alignment hinder the tokenization process from being optimized toward the ultimate objective, consequently affecting recommendation effectiveness. To avoid objective inconsistency between the embedding extraction and SID quantization stages, LMIndexer [102] proposes a self-supervised SID training framework. Specifically, it utilizes a generative language model to directly encode item text into several semantic identifiers, which is used to reconstruct the original item text for supervision. Moreover, more studies have been dedicated to resolving the inconsistency between SID quantization and generative model training. URI [103] utilizes one model to function both as an indexer (for SIDs construction) and a retriever (for item generation), which are trained through an EM algorithm. ETEGRec [104] proposes an end-to-end joint optimization framework for the tokenizer and the GR model, which are aligned with the proposed sequence-item and preference-semantic alignment loss. Similarly, MMQ [105] proposes a behavior-aware fine-tuning to enable the co-training of the GR model and tokenizer. It formulates item representations as a weighted combination of codebook vectors through a soft indexing mechanism, thus enabling continuous gradient propagation.

Multi-modal Integration. The third challenge lies in how to precisely model multi-modal information during the tokenization process. Beyond an item's intrinsic semantic content, collaborative signals play a crucial role in recommendations. Leveraging multi-modal information effectively can enhance quantization and improve recommendation performance consequently. Existing methods achieve multi-modal fusion during either the embedding extraction stage or the quantization phase. QARM [106] and OneRec [15,32,33] fine-tune a pretrained multi-modal model supervised by real user-item behavior. Instead, UNGER [84] performs contrastive alignment between multi-modal embedding and collaborative embedding. In addition to collaborative information, scenario-specific information also plays a vital role in practical industry application, such as geographic information in point of interest (POI) recommendation [85,86]. Besides fusion in the original embeddings, several research focuses on integrating multi-modal information in the SID quantization phase [37,107,108]. Among them, an intuitive method is to quantize each modality independently [83,107]. To further enhance consistency across different modalities, MME-SID [108] and LETTER [37] employ contrastive learning to strengthen inter-modal alignment and fusion. MMQ designs a multi-modal shared-specific tokenizer, which incorporates a Mixture-of-Experts (MoE) architecture during quantization. In particular, this framework maintains both modality-specific codebooks and a shared codebook, with a router performing weighted aggregation of outputs from different codebooks. BBQRec [109] proposes a behavior-aligned multi-modal quantization method to extract behavior-relevant information from multi-modal data. Furthermore, some research explores assigning different positions in the semantic ID sequence to model distinct modalities. TALKPLAY [110] encodes each modality as a separate position through K-means clustering, while EAGER-LLM [111] further allocates the first two codebook layers for multi-modal information and the subsequent two layers for collaborative signals.

Interpretability and Reasoning. Compared with the LLM-backbone models, SID-based GR models lack interpretability and cannot leverage the world knowledge and reasoning capabilities inherent in LLMs. Thus, recent research has explored incorporating SID tokens as new vocabulary entries into language models, enabling LLMs to better capture semantic meanings and exploit reasoning abilities to achieve further improvements. PLUM [112] employs pretraining tasks such as SID-to-Title and SID-to-Topic to equip LLMs with the ability to acquire the semantic correspondence between SID tokens and natural language descriptions. Furthermore, OneRec-Think [34] proposes a unified framework that bridges the semantic gap between discrete recommendation items and continuous reasoning spaces. To enhance the reasoning capabilities in recommendation tasks, it designs a retrieval-based reasoning paradigm that orchestrates multi-step deliberation with recommendation optimization.

Table 1. Comparison of different tokenizers

	Universality	**Semantics**	**Vocabulary**	**Item Grounding**
Sparse ID	×	×	Large	✓
Text	✓	✓	Moderate	×
Semantic ID	×	✓	Moderate	✓

3.4. Summary

This chapter introduces the fundamental concepts and evolving trends of sparse ID-based, Text-based, and Semantic ID (SID)-based approaches. Table. 1 compares these approaches from the perspective of universality, semantics, vocabulary size, and item grounding. Sparse ID-based methods lack universality and semantics, while text-based approaches struggle with reliable item grounding. Despite lacking universality and interpretability, SID-based techniques remain the most promising paradigm for generative recommendation at the current stage. Moreover, recent research [34,112] attempts to combine SID and text-based ID to improve the interpretability.

In the future, designing a more adaptive tokenization scheme remains an important research problem. Such a scheme should be capable of dynamically adjusting to factors such as item grounding collision rates, the emergence of new items, and cross-scenario generalization requirements. Moreover, evaluation of tokenizers is also an under-explored research direction. Although existing methods typically assess metrics such as collision rate, information entropy, and codebook utilization, these surface-level indicators fail to provide direct guidance on how to design tokenizers that are truly optimized for downstream generative recommendation performance.

4. Model Architecture

Traditional discriminative recommendation models typically follow the feature “embedding & MLP” paradigm. Specifically, sparse features are first mapped to dense vectors via an embedding layer [113]. Subsequently, MLPs or more sophisticated interaction modules [7,52,114] are employed to capture feature interactions, ultimately predict the probability of user clicks or purchases on candidate items within a multi-stage cascaded framework [9]. This approach reliance on a variety of specialized, heterogeneous, small-scale operators, which severely constrains hardware computational efficiency. The resulting irregular and fragmented computation patterns yield extremely low MFU [34], preventing recommender systems from benefiting from the rapid advances in computational hardware.

Generative architectures have the potential to avoid the aforementioned drawbacks due to the sequential token organization scheme and auto-regressive structure. Moreover, these generative structures unify the architecture, significantly enhancing computational regularity and dramatically improving hardware MFU. The generative structures also enable the scaling of model parameters, achieving substantial performance gains in a more hardware-friendly manner.

The architecture can be broadly categorized into three types: encoder-decoder [15], decoder-only [33,35], and diffusion-based structure [115]. As shown in Figure 4, generative recommendation models are rapidly growing in scale, with parameters rising from millions in 2023 to billions, and even tens of billions, by 2025. Benefiting from the widespread application of TIGER in generative retrieval, encoder-decoder structures have received much attention in recent years. In contrast, benefiting from advances in LLMs, decoder-only structures have risen sharply since 2024, especially for models exceeding 1 billion parameters. They exhibit superior scaling potential and are increasingly emerging as the dominant paradigm in generative recommender systems.

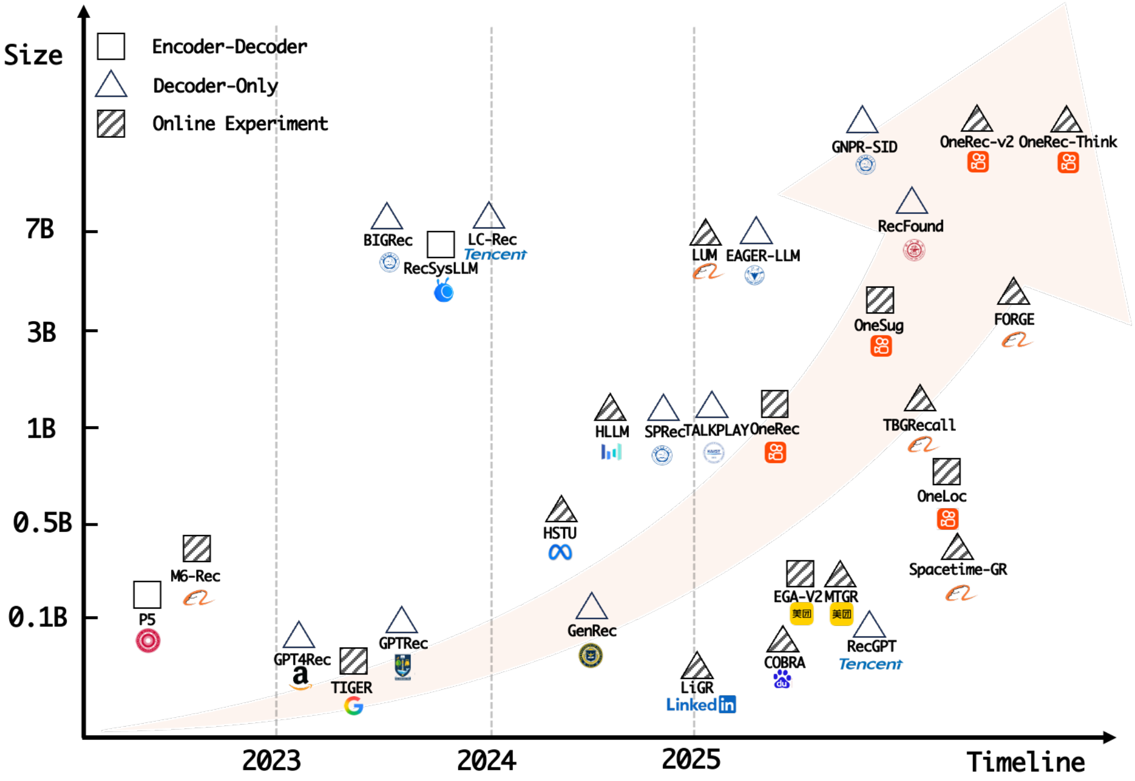


Figure 4. Trends in model architecture and scale indicate a gradual shift toward decoder-only architectures, accompanied by continued growth in model scale. **Online Experiment** indicates that the model has been deployed and is being used in production/industrial environments.

4.1. Encoder-Decoder Architecture

As recommender systems evolve toward a generative paradigm, the encoder-decoder structure has gradually emerged as a core technical pathway for building generative recommender systems, owing to its ability to effectively balance user preference understanding and next-item generation.

Early explorations primarily focused on directly transferring pre-trained encoder-decoder language models to recommendation tasks. For instance, P5 [28], built upon the T5 architecture [116], unifies five recommendation tasks with specific prompts. Similarly, M6-Rec [69], based on the M6 model [117], serializes user-item interactions into text, enabling the model to generate textual descriptions of the next recommended item. RecSysLLM [74] structures user behaviors and item attributes into textual inputs using a structured prompt format fed into the GLM pre-trained language model [118], and leverages a multi-task masked token prediction mechanism to recommend. However, these approaches face multiple challenges: general-purpose LLMs lack collaborative signals intrinsic to recommendation domains and their language modeling objective is fundamentally misaligned with recommendation goals. Besides, the high overhead of inference also hinders industrial deployment.

Consequently, an increasing number of studies have shifted toward designing dedicated encoder-decoder architectures tailored specifically for recommendation tasks. TIGER pioneered the application of generative retrieval for recommendation, which adopts a standard transformer encoder-decoder (T5) structure over pretrained semantic IDs and formalizes recommendation as a ”semantic ID sequence generation” problem. Building upon this, OneRec [32] further extends the approach by constructing an end-to-end generative architecture that completely eliminates the traditional multi-stage pipeline. The encoder aims to capture distinct scales of user interaction patterns through a unified transformer-based network, including the user static pathway, short-term pathway, positive-feedback pathway, and lifelong pathway. Subsequently, the decoder processes the user sequence through several transformer layers:

$$\mathbf{x}_m^{(i+1)} = \mathbf{x}_m^{(i)} + \text{CausalSelfAttn}(\mathbf{x}_m^{(i)}), \tag{4}$$

$$\mathbf{x}_m^{(i+1)} = \mathbf{x}_m^{(i+1)} + \text{CrossAttn}(\mathbf{x}_m^{(i+1)}, \mathbf{Z}_{\text{enc}}, \mathbf{Z}_{\text{enc}}), \tag{5}$$

$$\mathbf{x}_m^{(i+1)} = \mathbf{x}_m^{(i+1)} + \text{MoE}(\text{RMSNorm}(\mathbf{x}_m^{(i+1)})), \tag{6}$$

where $\mathbf{Z}_{\text{enc}}$ is the encoded information from the encoder layer. Each decoder layer incorporates a Mixture of Experts (MoE) feed-forward network with a top-k routing strategy to enhance model capacity while maintaining computational efficiency.

To better align with specific application scenarios, researchers have further optimized the architectures based on the characteristics of the scenario [63,85,107,119]. OneSug [119], designed for e-commerce query auto-completion, employs a dedicated encoder for modeling historical interactions and a specialized decoder for query generation. Specifically, the encoder processes historical queries, user profiles, and augmented prefixes through stacked multi-head self-attention and feed-forward layers to produce contextualized interaction features. The decoder then takes target query tokens as input and autoregressively generates query candidates. OneSearch [63], tailored for search scenarios, introduces Multi-view Behavior Sequence Injection to capture users' short-term and long-term preferences from multiple perspectives, and leverages a unified encoder-decoder structure to generate semantic ID sequences for recommendation. For local life service scenarios, OneLoc [85] incorporates a geolocation-aware self-attention module in the encoder to model location-based historical interactions. In the decoder, it designs neighbor-aware attention by utilizing the user's surrounding locations, thus integrating neighborhood context into the user's spatial representation to guide POI generation. For the advertising scenarios, EGA-V2 [107] encodes user interaction sequences via stacked self-attention and feed-forward networks, and autoregressively generates sequences of both next POI tokens and creative tokens through two dependent decoders in a multi-task schema, thus enabling efficient training and consistent generation quality.

Overall, the evolution of encoder-decoder architectures in generative recommendation has progressed from direct adaptation of general-purpose LLMs to a systematic engineering paradigm characterized by scenario-specific customization. Besides, the emergence of the One-series [15,63,85,119] has also driven the development of multi-stage cascaded frameworks towards unified end-to-end generation.

4.2. Decoder-Only Architecture

In traditional encoder-decoder architectures, computational resources are disproportionately allocated to encoding the input sequence, while the actual generation component remains relatively lightweight [33]. The imbalance between understanding and generation not only reduces overall efficiency but also hinders model scalability, particularly for modeling long-sequence behavior contexts. Therefore, decoder-only architectures have emerged as a more scalable and computationally effective alternative for generative recommendation.

With the rapid advancement of decoder-only LLMs [120,121], researchers have begun exploring the direct use of pre-trained decoder-only LLMs as backbones for recommendation. The first line of approaches leverages the natural language understanding and generation of LLMs by prompting LLMs to generate textual descriptions of target items with various SFT tasks. A subsequent "item grounding" step then maps the generated text back to concrete item candidates. Representative works include GenRec [122], BIGRec [72], Rec-R1 [123], GPT4Rec [59], RecFound [124], and Llama4Rec [125]. The key advantage of this paradigm is that it inherits the model architecture and tokenization vocabulary, enabling direct reuse of open-source LLMs with low deployment overhead. However, using natural language to represent items suffers from inherent limitations, as there exists a semantic gap between generated text and the discrete item space during the item grounding step.

To address these issues, a second line of approaches introduces dedicated semantic IDs into the generative pipeline, enabling direct modeling and generation of items, such as MME-SID [108], EAGER-LLM [111], RecGPT [61], as well as SpaceTime-GR [126], TALKPLAY [110], and GNPR-SID [86]. Recently, OneRec-think [34] further unifies dialogue understanding, chain-of-thought reasoning, and personalized recommendation within a single architecture.

Although the LLM-based recommendation approaches demonstrate strong potential, their generic linguistic priors may not fully align with the intrinsic characteristics of recommendation scenarios. Therefore, rather than relying directly on off-the-shelf LLMs, several work builds dedicated generative architectures from scratch, explicitly tailored for recommendation tasks. These models adopt a decoder-only transformer as the backbone and incorporate targeted designs that account for the unique properties of recommendation scenarios, such as user behavior sequences, interaction sparsity, and temporal dynamics, thereby preserving the advantages of the generative paradigm while significantly improving task-specific adaptability. RecGPT [60] and FORGE [127] employ pure decoder-only architectures and perform autoregressive training on large-scale user interaction sequences to achieve direct generation of the next item based on the pre-trained semantic IDs identifier.

To unify search and recommendation tasks and precisely control information flow among heterogeneous tokens, SynerGen [128] introduces task-specific masking matrices that simultaneously enforce temporal causality, session isolation, and cross-task alignment. To bridge the performance gap between generative recommenders and dense retrieval models, COBRA [89] fuses semantic IDs with denasse embeddings in the decoder, enabling joint prediction of both the next item's sparse semantic ID and its dense vector representation. During inference, it employs a coarse-to-fine strategy, starting with semantic ID generation and refining them into dense vectors. To improve the decoding efficiency, RPG [129] proposes a multi-token prediction mechanism based on the parallel SID encoding. Combined with a graph-based decoding strategy, RPG efficiently maps discrete ID sequences back to final candidate items. Similarly, CAR [99] groups semantic IDs into concept blocks and designs an autoregressive transformer decoder that supports parallel block-wise prediction, effectively avoiding the latency accumulation caused by token-by-token autoregressive dependencies.

OneRec-V2 [33] extends the efficiency optimization to the underlying computational architecture by introducing a Lazy Decoder structure. Specifically, OneRec-V2 first employs a context processor to integrate heterogeneous, multimodal user behavior signals. To improve parameter and computational efficiency, OneRec-V2 omits the standard key and value projection operations in attention computation. Instead, multiple Lazy Decoder blocks share the same set of key-value pairs generated by the context processor, and grouped query attention is further employed to reduce computational overhead.

Rather than generating item tokens autoregressively, some generative recommendation also focused on modeling user actions as structured behavioral sequences. HSTU [35] frames recommendation as structured sequence prediction over behavioral time series, emphasizing intent modeling over token-by-token generation. The HSTU architecture consists of a stack of layers connected via residual connections:

$$\mathbf{u}^{(i)}, \mathbf{v}^{(i)}, \mathbf{q}^{(i)}, \mathbf{k}^{(i)} = \mathrm{Split}\big(\mathrm{SiLU}(f_1(\mathbf{x}^{(i)}))\big), \tag{7}$$

$$\mathbf{x}^{(i+1)} = f_2\big(\mathrm{Norm}(\mathrm{SiLU}\left({\mathbf{q}^{(i)}}^T \mathbf{k}^{(i)} + \mathrm{rab}^{p,t}\right) \mathbf{v}^{(i)}) \odot \mathbf{u}^{(i)}\big). \tag{8}$$

HSTU significantly outperforms the standard transformer in ranking tasks, primarily due to its structural innovations. Specifically, as shown in Equation (8), HSTU replaces the conventional softmax with pointwise aggregated attention to preserve preference intensity signals, enabling it to capture the strength of user preferences better and making it more suitable for streaming scenarios where the item vocabulary is non-stationary.

Based on the HSTU architecture, numerous works have been done to optimize the structure for various scenarios application. For instance, MTGR [65] designs dedicated attention patterns for different feature types in ranking tasks: full attention is applied to static information, dynamic autoregressive masking is used for real-time behaviors, and a diagonal mask is imposed on candidate items to prevent information leakage. Additionally, several approaches focus on session-level modeling. Specifically, INTSR [130] introduces a session-level masking strategy. Moreover, to unify query-agnostic recommendation and query-based search tasks, INTSR introduces Query-Driven Block, which separates the processing of the query placeholders from the ordinary historical behavior sequence and incorporates a generic query placeholder Q capable of representing search queries. LiGR [131], targeting the re-ranking scenario, further proposes an in-session set-wise attention mechanism, where items in a recommendation list are presented simultaneously and ignore inherent causal ordering.

4.3. Diffusion-Based Architecture

In contrast to the sequential prediction mechanism of transformer-based models, diffusion-based models generate recommendations through parallel iterative denoising of the full target sequence. This approach enables bidirectional attention across all tokens, breaks causal dependencies, and allows flexible control over generation steps for efficient decoding. These characteristics make diffusion-based generation a promising research direction.

Diffusion-based recommendation models enable parallel generation and provide richer supervision through denoising objectives. Diff4Rec [115] employs VAE to map discrete interactions into 64-dimensional latent vectors where diffusion with curriculum scheduling generates semantically consistent augmentations, while CaDiRec [132] further enhances this with context-aware weighting and transformer-based UNet for temporal dependencies. RecDiff [133] combines GCN with diffusion in latent space to refine user representations for embedding enhancement, and DDRM [134] directly denoises continuous embeddings via MLPs for robust ranking. For multimodal scenarios, DiffCL [135] and DimeRec [136] leverage diffusion as a feature augmenter. DiffCL generates hard positives from GCN-aggregated features for contrastive alignment, while DimeRec conditions on multi-interest vectors for joint optimization. While prior works operate in continuous spaces, DiffGRM [137] pioneers discrete diffusion for Semantic ID generation.

4.4. Summary

Driven by unified efficient generative architectures, generative recommendations are rapidly converging with advances from LLMs. Key innovations, such as MTP, MoE and GQA, have been effectively adapted to recommendation tasks, enabling scalable foundations that replace fragmented legacy designs. This standardization enables the entire community to concentrate collective efforts on improving a unified, scalable architectural foundation, accelerating innovation, and reducing redundant engineering overhead. Additionally, these generative models exhibit clear scaling laws, especially for decoder-only architectures [33,35], where increased capacity consistently yields predictable performance gains. Building on this standardized foundation, recent work on generative structure mainly emphasizes task-aware refinements. Techniques include task-specific attention masking [128], MoE-based feature routing [15], geolocation-aware attention [85], query-aware encoders [119], and hybrid semantic-dense representation fusion [89]. These innovations enable fine-grained, context-sensitive modeling across diverse applications.

While generative recommender systems have made remarkable progress, several challenges remain unresolved. First, despite efficiency techniques like MoE and GQA, real-time deployment under latency constraints remains challenging. Second, existing methods [67,128,130] share backbones but rely on decoupled decoding phases or task-specific prompts. A holistic generative framework capable of supporting diverse scenarios and tasks remains an open research direction, requiring further investigation.

5. Optimization Strategy

Unlike discriminative recommendation models, which are trained with a binary classification objective to predict whether a candidate item should be recommended, generative recommendation models are trained to generate the next item directly. During supervised learning, the optimization target is typically the next-token prediction [29,37,62], and various auxiliary objectives, such as modality alignment, spatial constraints, and ranking-enhanced objectives, are introduced to further strengthen the model's capabilities. Moreover, given the end-to-end modeling capability of generative models, reinforcement learning-based preference alignment has also been introduced to better align users' multidimensional preferences and diverse business objectives [32,63,85,123,138] of different platforms. Figure 4 illustrates the optimization strategies of several representative generative recommendation models. As shown, in addition to the basic supervised training objectives, an increasing number of recent works adopt preference alignment methods to directly optimize the multidimensional preferences of users and platforms, thus further enhancing the end-to-end modeling capabilities of generative recommendation models.

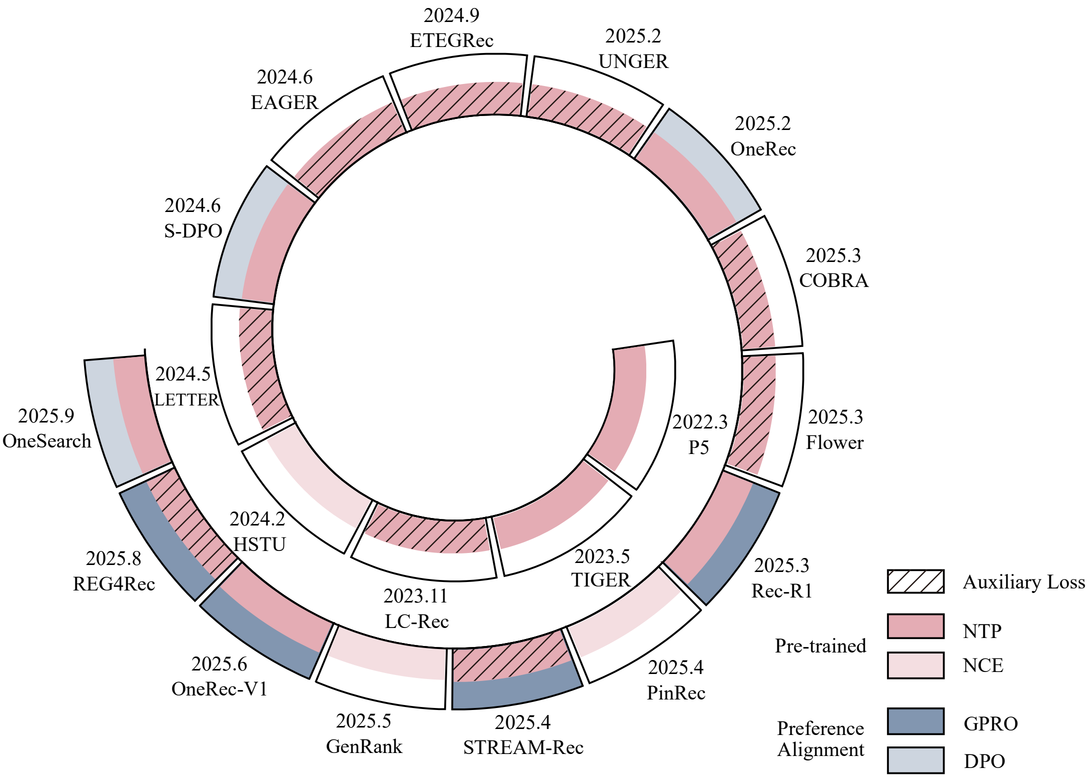


Figure 5. Landscape of optimization strategy taxonomy. Inner circle: pre-training supervised learning strategies; outer circle: preference alignment. Shaded cells indicate the use of additional auxiliary losses. Recent work shows a trend toward jointly employing supervised learning and preference alignment.

5.1. Supervised Learning

NTP Modeling. For generative recommendation, given a user's historical behavior tokens, the model learns user preferences and predicts the next item through an autoregressive training objective. TIGER [29] introduces the generative paradigm into the retrieval stage and trains an encoder–decoder in a sequence-to-sequence manner with next-token prediction objective, which can be referred to Equation (2). Similarly, many generative recommendation studies adopt NTP as their primary supervised training objective [15,60].

Besides, several work further modifies the training objective to better match practical goals. LETTER [37] modifies the NTP loss into a ranking-guided generation loss by altering the temperature to emphasize the penalty for hard-negative samples, thus improving the ranking performance. To

doi:10.20944/preprints202512.0203.v1

optimize both sparse and dense representation prediction jointly, COBRA [89] designs a composite loss function that combines losses for sparse ID prediction and dense vector prediction. REG4Rec [139] adds an auxiliary category-prediction task to support reliability assessment and encourage consistency. To compensate for the information loss caused by tokenization, UNGER [84] introduces an intra-modality knowledge distillation task that transfers item knowledge from the tokenization step through contrastive learning.

Beyond training models from scratch, several studies also inherit parameters from LLMs and design additional training tasks to adapt these general-purpose LLMs for item generation in recommendation scenarios. By using LLMs as the backbone, user behaviors are serialized into textual prompts as inputs to the LLMs, which are then fine-tuned either with full-parameter updates or via parameter-efficient tuning methods [59,122] (e.g., LoRA). LC-Rec [82] further designs a series of carefully designed tuning tasks to better align language and collaborative semantics. RecFound [124] designs a broader set of recommendation-specific tasks (including generative and embedding tasks) and proposes a step-wise convergence-oriented sample strategy for stabilizing the convergence of multi-task training. EAGER-LLM [111] adopts an annealing adapter tuning schedule to gradually reduce adapter update strength during training, thus mitigating catastrophic forgetting and ensuring stable convergence on recommendation tasks.

NCE Modeling. However, for methods that adopt sparse IDs as the tokenization scheme, computing the NTP loss becomes challenging due to the extremely large vocabulary size, leading to computational instability and inefficiency. To address this issue, they primarily adopt NCE-style optimization [35] to approximate the softmax over the full vocabulary, thereby avoiding vanishing gradients and enabling efficient training. HSTU [35] and GenRank [62] approximate the full softmax with sampled softmax, treating the true next token as the positive and sampling negatives from the catalog to learn the conditional distribution on an approximated vocabulary. Motivated by the absence of strict ordering, PinRec [66] further employs an efficient multi-token objective to enable prediction beyond the next token at a timestep window. IntSR [130] leverages the InfoNCE optimization objective and designs a hard negative sampling strategy that only instances that exactly exist when user-item interaction occurs can be treated as negatives. SessionRec [140] trains at the session level under a next session prediction and further enhances the model's ability to distinguish hard negative samples by involving a ranking task. In addition, some works, like MTGR [65], adopt optimization strategies similar to traditional discriminative models, preserving user–item cross features and optimizing the model with discriminative loss.

Overall, current supervised training practice centers on autoregressive NTP loss, while sparse ID-based generative models adopt NCE-style optimization to ensure computational stability and efficiency. These objectives are often complemented by various auxiliary losses that optimize the model for specific tasks and requirements. However, supervised training alone can only learn from users' observed behaviors and cannot fully align with users' multidimensional preferences or optimize for platform-level objectives, thereby limiting the model's ability to support end-to-end deployment. These multi-objective, multi-dimensional preferences include long-term user preferences, business-oriented metrics such as GMV and retention, and broader platform-level considerations encompassing safety, diversity, and fairness [32].

5.2. Preference Alignment

To align with users' implicit multidimensional preferences beyond explicit behavioral imitation and optimize for platform-level objectives, reinforcement learning (RL)-based preference alignment optimization [141] is proposed. By optimizing cumulative reward over sequential user interactions, RL enables generative models to pursue long-term objectives such as retention or lifetime value [142], rather than immediate accuracy. Moreover, RL allows direct optimization of complex, non-differentiable metrics, including novelty [143] and diversity [139], thereby better aligning model behavior with real-world business and ecosystem goals.

In current generative recommendation research, reinforcement learning methods for preference alignment primarily focus on result supervision, providing feedback signals based on the model's final recommended outcomes. These approaches are profoundly inspired by alignment techniques in LLMs, such as Direct Preference Optimization (DPO) [144] and Group Relative Policy Optimization (GRPO) [145], which have evolved into two mainstream paradigms in this field.

DPO Modeling. The core idea of DPO-based methods is to directly optimize models using pairwise preference data, encouraging outputs closer to the chosen samples rather than the rejected ones. This approach bypasses explicit reward modeling in traditional RLHF and instead performs implicit policy optimization through a simple classification loss. In recommendation, the effectiveness of DPO-based methods largely depends on constructing high-quality preference pairs. Most works regard users' posterior behaviors (e.g., clicks or purchases) as positive examples, while strategies for generating negative ones vary widely. S-DPO [38] extends standard DPO by pairing one positive with multiple negatives in a single step, enabling more discriminative learning across candidates. RosePO [146] refines this process through selective rejection sampling, emphasizing negatives that improve both helpfulness and harmlessness. SPRec [147] introduces a self-evolving mechanism that dynamically selects hard negatives from the model's previous predictions, forming an iterative feedback loop that progressively sharpens preference boundaries.

More advanced approaches, such as OneLoc [85], OneSearch [63], and OneSug [119], combine heuristic rules with predictive models, and design feedback-aware weighting to improve preference construction. Specifically, OneLoc [85] adopts a generation–evaluation framework, using beam search to produce candidates and rank them with two reward signals: a GMV prediction model estimating commercial value and a rule-based geographic proximity score capturing spatial relevance. OneSug [119] constructs multi-level preference pairs across nine behavior types (e.g., exposure, click, order), assigning calibrated weights that reflect user intent strength. Building on this, OneSearch [63] further introduces a three-tower reward model to compute user–item relevance scores for pair selection, along with feedback-based weighting derived from CTR and CVR statistics, guiding the model to emphasize stronger preference signals. This graded weighting enables finer-grained alignment, allowing the model to capture continuous preference intensity rather than binary distinctions.

GRPO Modeling. GRPO extends preference optimization from a pairwise to a groupwise setting. Unlike implicit rewards constructed from pairwise samples, GRPO assigns an explicit reward signal to each candidate and updates the policy to shift probability mass toward higher-reward candidates. It enables the recommendation model to better capture ranking capability while requires a reliable and accurate reward function. Current research typically employs a hybrid reward system that integrates diverse sources of feedback, which can be roughly categorized into rule-based and model-based signals.

At the basic level, rule-based rewards ensure format compliance and alignment with observed user interactions. For example, VRAgent-R1 [148] provides a positive reward when the generated output conforms to the expected format and correctly reflects user behaviors. STREAM-Rec [91] extends this by introducing graded behavioral rewards and assigning high positive rewards for highly matched items, while negative rewards to those with low matching quality, thereby providing richer guidance within GRPO's groupwise updates.

Building on this, methods such as Rec-R1 [123], and RecLLM-R1 [138] incorporate posterior ranking metrics to evaluate the quality of entire candidate groups. Specifically, Rec-R1 leverages metrics like NDCG, and RecLLM-R1 adopts the Longest Common Subsequence algorithm, which assigns higher rewards to predictions with more correct items ranked at earlier positions. Moreover, some frameworks explicitly combine rule-based and predictive model signals. OneRec [15] integrates a point-wise P-Score model predicting user preference with rule-based components such as format compliance and ecosystem relevance. To stabilize optimization over this complex reward, OneRec employs Early Clipped Policy Optimization (ECPO), a GRPO variant that clips updates when negative advantages or policy-ratio shifts are too large.

Building on GRPO's reward framework, recent methods have incorporated explicit reasoning to guide generation. OneRec-Think [34] introduces chain-of-thought reasoning with item-text alignment and reasoning scaffolding, optimized via multi-path rewards to improve recommendation accuracy. REG4Rec [139] expands reasoning space by generating multiple semantic tokens per item, while pruning inconsistent paths through self-reflection. RecZero [64] adopts a "Think-before-Recommendation" RL paradigm, leveraging structured reasoning templates and rule-based rewards to elicit the model's reasoning abilities further, paving the way for reasoning-driven recommendations that rely on explicit thinking. Together, these methods illustrate the progression from heuristic reasoning to autonomous RL-based reasoning, highlighting the growing role of explicit reasoning in generative recommendation and its close integration with GRPO reward design.

5.3. Summary

In summary, the optimization strategies for generative recommendation models are gradually evolving from simple behavior-based supervised learning augmented with various auxiliary objectives, to a hybrid framework that integrates supervised pretraining with RL–based preference alignment. Moreover, during the preference alignment stage, the reward signal has evolved from relying solely on users' posterior behaviors to a more comprehensive reward system that integrates multi-dimensional signals, including more generalizable user preferences and platform-level objectives. This shift enables generative recommendation models to better capture complex preferences and facilitates end-to-end recommendation.

Looking ahead, generative recommendation may further advance in optimization strategies along several key directions. The first important direction is to enhance the reasoning capability of generative recommendation models, enabling more accurate inference of user preferences, intentions, and evolving thought patterns from historical interactions [34,139]. The significance of this direction has been widely validated in both general LLMs and downstream applications. The second promising direction is the transition from rule-based or model-specific single-aspect reward scoring to a unified Reward Agent, particularly supported by the preference understanding and role-playing capabilities of LLMs. Such a unified Reward Agent would automatically comprehend and balance multi-dimensional preferences, providing a more generalizable and robust alignment signal while reducing manual intervention and improving adaptability. Moreover, providing list-wise reward signals rather than the currently dominant point-wise signals [32] is vital for enhancing the holistic quality of end-to-end recommendation, particularly for list-based recommendation scenarios such as short videos or e-commerce.

6. Application

Generative recommendations have achieved significant commercial success over the past year. We examine these developments along two dimensions: first, how generative models integrate into cascaded recommender systems, and second, their deployment across diverse scenarios. Our analysis synthesizes recent research highlighting key innovations and practical applications, as illustrated in Figure 6.

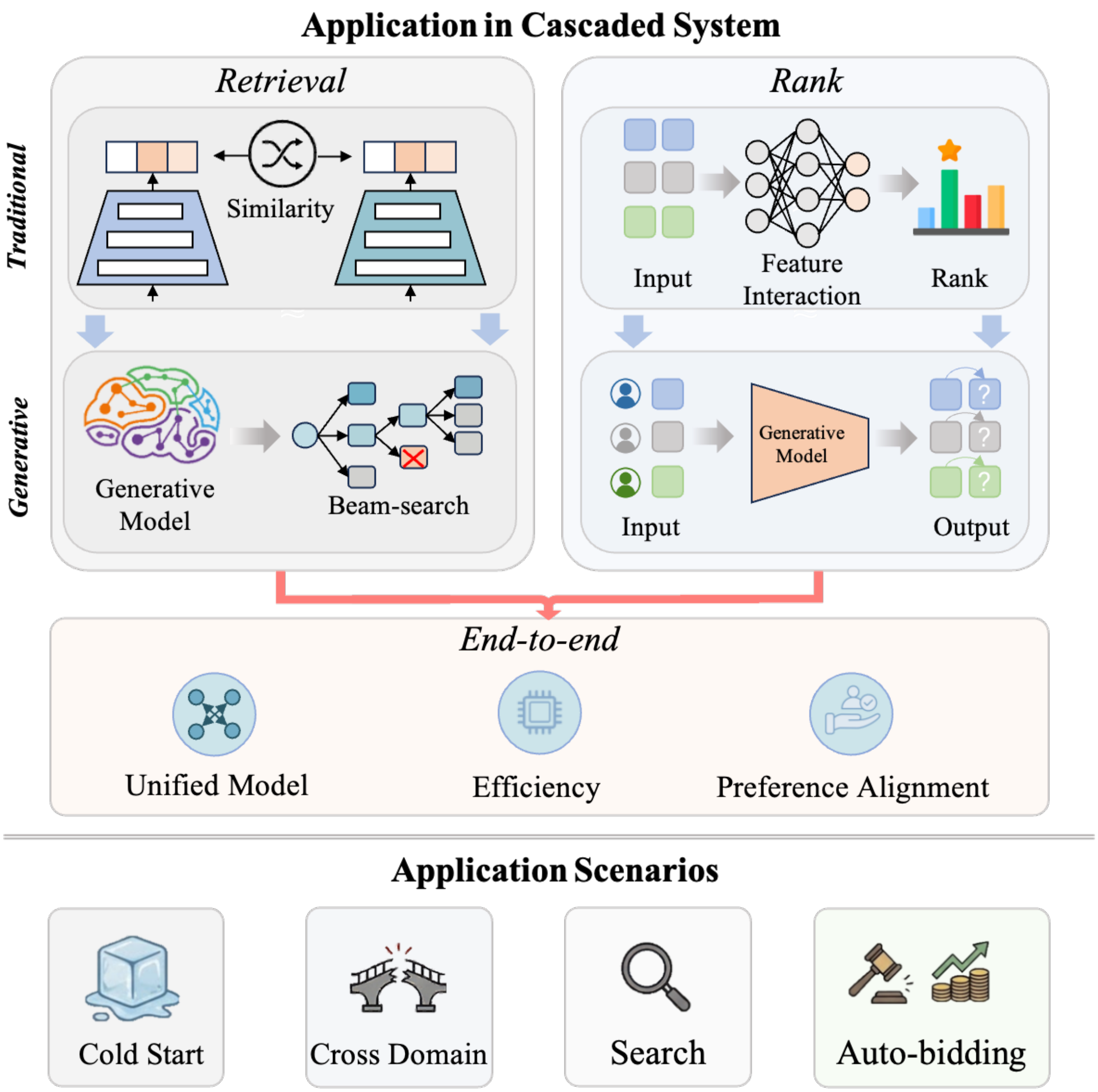


Figure 6. Applications of generative recommendation models. Top: application in cascaded systems. Bottom: application in different scenarios.

6.1. Generative Recommendation in Cascaded System

6.1.1. Retrieval

The retrieval is the initial stage of cascaded recommender systems, narrowing millions of items to thousands of candidates for subsequent ranking. Conventional retrieval systems (e.g., DSSM [149]), employ dual-tower architectures that separately encode users and items, using Maximum Inner-Product Search (MIPS) or Approximate Nearest Neighbor (ANN) for matching. While computationally efficient, it offers limited expressiveness for multiple business targets and constraints.

Generative models are fundamentally transforming retrieval methodology in three dimensions. First, from similarity-based to generative retrieval. Rather than relying on similarity-based search, generative retrieval reframes retrieval as a next-token prediction task via beam search [29]. Second, generative retrieval is advancing toward outcome-conditioned retrieval, where objectives, including user-level and provider-level, are incorporated into training via RL, enabling a better balance among multiple objectives [66,150]. Third, it enables multi-interest retrieval, such as Kuaiformer [151] that introduces multi-interest query tokens and adaptive sequence compression, enabling interest-specific retrieval.

6.1.2. Rank

Traditional ranking models relied on feature interaction and behavior modeling operators [7,114] to capture user interests and behavioral patterns. Generative methods, by contrast, introduce fundamental shifts in ranking model design and implementation that emerge across three key dimensions. First, generative ranking models have fundamentally transformed data organization by introducing user-centric arrangements that more effectively capture sequential behaviors and preferences. HSTU [35] organizes user data into a unified, user-centric token sequence, interleaving items and actions, and the following works[62,130] share a similar manner. Second, model architectures have shifted toward generative frameworks. For example, DFGR [68] introduces single-stream and dual-stream generative ranking networks, and HLLM [152] employs hierarchical LLMs to perform top-K ranking, which have all been integrated into architectures with a transformer-based backbone. Third, significant optimizations have been made to model architectures owing to the rapid growth of the LLMs research community. M-FALCON [35] implements batched target-aware inference with KV cache to share HSTU computations across multiple candidates, and hardware-specific optimizations through model-chip co-design, such as MTIA 2i [153], further enhance efficiency.

As for the re-ranking task, most existing neural models follow a two-stage generator–evaluator paradigm [154], which first generates candidates and then ranks them separately. Generative-based rerankers, employing generative models capable of directly editing and reorganizing the candidate list, offer two key advantages. Firstly, they remove the need to maintain separate generation and evaluation modules, simplifying the architecture and training pipeline, such as GoalRank [155], proposing a generator-only large ranking model trained with group-relative optimization. Secondly, they can optimize multi-objective trade-offs in a single integrated objective, like SORT-Gen [156], a generative re-ranking model for list-level multi-objective optimization. KC-GenRe [157], a knowledge-constrained generative re-ranker, for knowledge graph completion, showing that it can achieve consistent gains under complex, real-world constraints.

6.1.3. End-to-End

End-to-end modeling has emerged as a transformative paradigm that addresses the fundamental limitations of cascaded architectures, particularly objective misalignment and error accumulation throughout the pipeline. Recent research demonstrates a clear convergence toward this unified approach across the field.

This paradigm delivers three primary characteristics. First, end-to-end modeling significantly enhances performance by eliminating the inherent losses of multi-stage cascading systems. Through unified model architectures and multi-objective optimization, models such as OneRec [32], OneSug [119], and ETEGRec [104] have achieved substantial performance improvements compared to traditional cascaded approaches. Second, this approach yields remarkable improvements in computational efficiency, particularly in Model FLOPs Utilization (MFU). such as OneRec [15] achieve training and inference MFU rates of 23.7% and 28.8% respectively, compared to just 4.6% and 11.2% for cascaded systems. It also enables systems to fully capitalize on the benefits of model scaling, thereby unlocking greater potential from larger architectures [15,33,85,104]. Third, to meet the complex requirements of industrial-scale recommender systems and enable end-to-end optimization, RL-based preference alignment has become a necessary approach for aligning user and platform objectives across multiple dimensions [15,63]. Designing and integrating diverse reward strategies will therefore remain a critical direction for industrial generative recommendation.

6.2. Generative Recommendation in Various Application Scenarios

6.2.1. Cold Start

The cold-start problem is a fundamental challenge in recommender systems, which arises when predictions are needed for users, items, or scenarios that lack adequate interaction history, hindering the model's ability to learn reliable preference patterns. Generative paradigms offer transformative opportunities for easing cold-start challenges. First, the LLMs' intrinsic world knowledge can be utilized to depict item attributes and user preferences, like LLMTreeRec [70], using LLM to construct a tree structure, thereby alleviating the limitations caused by insufficient collaborative signals. Besides, generative models bridge the recommendation space with the semantic space via semantic identifiers, enabling more effective generalization to long-tail and cold-start items [82,158]. LC-REC [82] employs RQ-VAE to learn item semantic tokens that align collaborative signals and language semantics into a unified token space, demonstrating improvements for cold-start items. These diverse approaches illustrate that generative methods possess significant advantages in addressing cold-start problems that traditional methods struggle to solve.

6.2.2. Cross Domain

Cross-domain recommendation aims to transfer knowledge across domains so that preference patterns learned in one scenario can generalize to improve performance in others. Traditional approaches typically rely on multi-task learning [159], feature-level alignment [160], or domain adaptation [161] approaches to bridge domains. Recently, generative approaches have introduced new perspectives through cross-domain semantic construction. First, unified representation learning enables better knowledge transfer. RecGPT [60] employs an FSQ-based tokenizer that enables more effective knowledge transfer within the semantic space by creating a unified semantic representation across domains. Second, domain-sensitive structure design. Like GMC [98], adopting domain-specific fine-tuning for for targeted adaptation. Third, domain-adaptive decoding strategy. GenCDR [162] develops domain-aware routing strategies besides with domain-aware prefix-tree structures to support efficient and accurate multi-target generative modeling. These approaches collectively show how generative methods can effectively address the fundamental challenges inherent in cross-domain recommender systems.

6.2.3. Search

Unlike the recommendation task, which relies on implicit user interest modeling, the search task uses explicit user queries to infer user intent and provide personalized services. Recent work on generative search addresses the limitations of traditional cascade architectures by introducing a unified, generative paradigm. First, to handle short and ambiguous queries, OneSug [119] introduces Prefix2Query, which mines complete queries from interaction logs based on similarity with the current prefix. GRAM [163] enhances query-side semantic representation through joint training and alignment of queries and products. Second, end-to-end modeling has emerged as a critical approach. OneSearch [63] formulates search as a single generative task that directly maps user and query context to an ordered list of semantic item IDs, thereby unifying recall, ranking, and personalization within one model. Third, unified frameworks for search and recommendation have gained attention. GenSAR [164] presents a generative framework that treats both search and recommendation as generative tasks, employing dual-purpose IDs and task-aware training to balance the two kinds of tasks.

6.2.4. Auto-Bidding

In advertising auctions, auto-bidding plays a critical role by automatically optimizing bids at the impression level to achieve objectives like maximizing conversions, meeting target Cost Per Acquisition (CPA), and adhering to budget constraints. Traditional methods employ rule-based approaches (e.g., PID [165]) and offline RL-based methods [166] that leverage Markov Decision Process (MDP) modeling capabilities to enhance decision-making. Recent generative approaches have demonstrated promising improvements in two key areas. The first is long-range temporal modeling. DiffBid [167] proposes diffusion models to produce bidding trajectories that naturally capture long-range temporal structure and enforce temporal coherence across steps, enabling the model to learn the long-range dependencies crucial for sequential bid optimization. The second area is multi-objective balancing. GAS [168] combines transformers with Monte Carlo Tree Search (MCTS), employing a generator for bidding sequences and critics for multi-objective evaluation; this framework refines trajectories to balance multiple objectives, including conversions, CPA, and budget constraints. GAVE [169] uses Return-To-Go signals to encode target metrics directly into the generation process, with a value function guiding sampling toward high-return trajectories while improving offline-to-online performance transfer.

7. Challenges and Future Direction

In this section, we provide a comprehensive discussion and identify the pivotal directions for innovation that will shape the next generation of recommendation technologies.

7.1. End-to-End Modeling

The adoption of the generative paradigm facilitates end-to-end modeling, which demonstrates significant advantages over traditional multi-stage cascaded pipelines by mitigating error accumulation and addressing objective misalignment across stages [32]. Furthermore, utilizing a single unified model can substantially enhance both training and inference efficiency, substantially reduce labor and engineering overheads, enabling more resources to be devoted to model scaling.

Two critical challenges need to be explored. The first concerns **model scaling**. Existing research has demonstrated that scaling up model size yields substantial performance improvements [170]. Current end-to-end recommendation models predominantly adopt transformer-based architectures, with deployed models approximately one billion parameters [15] due to latency constraints. Looking ahead, it is worth exploring further scaling up models toward LLM-level capacity while maintaining acceptable inference latency. The second challenge involves **unified reward design** within preference alignment frameworks. Current systems primarily employ rule-based or model-specific single-aspect reward as signals, yet they remain focused on relatively narrow metrics. Looking ahead, developing a unified Reward Agent, particularly supported by the preference understanding and role-playing capabilities of LLMs, contributes to automatically comprehending and balancing multi-dimensional preferences (e.g., user-level and platform-level considerations).

7.2. Efficiency

Benefiting from a unified transformer-based architecture, generative recommendation models can rapidly inherit advances in both hardware and LLMs development, including specialized accelerator operators, KV-cache-friendly execution strategies, decoding acceleration, etc. [171]. At the algorithmic level, the rapidly evolving technological ecosystem of the language-model community, such as parameter-efficient tuning, pruning and distillation, quantization, and mixture-of-experts architectures, can be readily transferred to reduce online efficiency costs in generative recommender systems [172].

However, several key challenges remain. First, current generative recommendation models lack an integrated **algorithm–system co-design** framework [35,153] tailored for large-scale streaming training and low-latency, high-throughput inference in recommendation scenarios. Large-scale industrial recommender systems process hundreds of millions of streaming samples every day. Under such real-time, continually-updated training settings, it becomes crucial to design efficient training and preference-alignment frameworks that can keep pace with streaming data. Moreover, as model sizes continue to scale up, developing inference frameworks that satisfy strict requirements on low latency and high concurrency will also become a central research direction. Second, **ultra-long behavior modeling** is both essential and a major efficiency bottleneck for generative recommendations due to the computational complexity of attention mechanisms [173]. Therefore, future research is encouraged to explore more efficient sequence modeling paradigms, such as memory-augmented structure and RAG-augmented training paradigms.

7.3. Reasoning

Recent emerging Reasoning Language Language Models (e.g., Deepseek-R1 [24], OpenAI o3 [25]) have already demonstrated strong capabilities through internal reasoning trajectories during prediction. Generative recommendations leverage similar reasoning capabilities to infer users' short-/long-term interests and analyze their behavior patterns, thereby achieving improved recommendation performance [34]. Furthermore, the model can integrate contextual information (e.g., temporal and geographic locations) and external information (e.g., trending events and news) to achieve a more comprehensive understanding and inference of user intent. It also supports rule-aware reasoning to generate recommendations that satisfy various constraints like diversity and fairness.

However, several challenges remain to be addressed. First, constructing **reasoning chain** for recommendation tasks is challenging. Currently, the field lacks methodologies for the large-scale generation of chain-of-thought tailored to personalized recommendations. Given users' unique characteristics, human experts and LLMs struggle to produce effective reasoning chains. Second, **adaptive reasoning** under efficiency constraints is largely unexplored. Models need to learn adaptive reasoning strategies based on query difficulty in order to satisfy strict latency requirements and prevent overthinking. Finally, developing **self-evolving generative recommenders** that continually reflect on their decisions, revise their reasoning policies, and improve from online feedback demands new architectural designs that harness the benefits of self-improvement while mitigating risks such as bias amplification, and catastrophic forgetting.

7.4. Data Optimization

From a data perspective, generative recommendations leveraging their semantic understanding and data-efficient capabilities [174], substantially alleviate the performance degradation caused by sparse or missing data in scenarios involving long-tail items, cold-start users, and newly introduced items [29,158]. Moreover, user-centric data organization facilitates a shift from feature–driven modeling to model-centric approaches, fundamentally reducing the substantial human effort previously required for feature extraction and design [35].

However, several critical issues warrant careful attention. First, training **data bias** presents a substantial challenge. Generative recommendation models are typically trained on historical user interaction data, which predominantly consists of positive actions. This data inherently contains exposure bias, position bias, and other systematic biases that cannot be effectively addressed through traditional debiasing methods [175]. Therefore, designing debiasing strategies for generative recommendation models is a research direction worth exploring. Second, although recommendation scenarios provide abundant user–item interaction data for model training, the **construction of high-quality data** remains a fundamental bottleneck, such as reasoning-oriented chain-of-thought (CoT) data for enhancing model thinking abilities, user and platform multi-aspect preference data for alignment, and explicit intent-level annotations.

7.5. Interactive Agent

Recent agent-based and conversational systems have gained significant attention in recommender systems [176]. Built on LLM backbones, these systems provide a natural interface for users to express their needs explicitly. By enabling richer interactions and multi-turn dialogues, generative models can jointly conduct conversational generation and item generation more naturally and coherently. Besides, the inclusion of explanatory dialogue helps improve users' trust and acceptance of the recommended items. Moreover, it can further plan dialogue strategies, interleave clarification questions with recommendation actions, and invoke external tools such as web search, knowledge bases, or transaction APIs [177,178], thus unifying search, recommendation, and question answering within a unified generative decision-making framework.

Despite these advantages, several open challenges remain. First, current agent-based generative systems struggle to deliver **personalized dialogue recommendations** that align with user preferences and provide user-style dialogue and personalized recommendation rationales. Second, designing a **user-centric memory mechanism** tailored for conversational recommendation scenarios, thereby enhancing the model's multifaceted understanding of users and strengthening its personalization capability.

7.6. From Recommendation to Generation

Large-scale generative models such as Kling [179], and Sora [180] are driving a paradigm shift from edit-enhanced workflows to end-to-end content generation. These multimodal models can synthesize complete video, audio, and advertising materials in real time based on language instructions and user constraints. Enabled by powerful multimodal generative models, an important trend in short-video and music platforms is the transition from merely recommending existing content that users might enjoy to proactively generating new content for users, thereby achieving a deeper and more precise alignment with user interests. Furthermore, this paradigm can generate tailored content for niche topics and underserved segments, effectively addressing supply-demand mismatches.

However, the transition to fully generative recommender systems faces several formidable challenges. First, the strong personalization inherent in generated content leads to extremely **sparse feedback signals**, which introduces further complexity and difficulty for the development of effective recommendation algorithms. Moreover, given the substantial resource demands of existing generative models, an important open problem is how to **trade off** generation cost against the potential value it creates, thereby ensuring the sustainability and overall robustness of the ecosystem.

8. Conclusions

Generative recommendation represents a fundamentally novel modeling paradigm that shifts traditional discriminative architectures toward a unified generative framework, enabling end-to-end optimization and flexible adaptation across diverse recommendation scenarios. This survey extensively reviews generative recommender system from several perspectives, including background, tokenization schemes, model architectures, optimization strategies, real-world application, and future research directions, providing a clear picture of recent developments in this field, encompassing both methodological innovations and practical deployment considerations. We hope this survey offers a comprehensive conceptual framework and practical roadmap that will greatly benefit future research and industrial applications along this emerging yet highly promising direction.

References

1. Schafer, J.B.; Konstan, J.A.; Riedl, J. E-commerce recommendation applications. *DMKD* **2001**, pp. 115–153.
2. Gomez-Uribe, C.A.; Hunt, N. The netflix recommender system: Algorithms, business value, and innovation. *TMIS* **2015**, pp. 1–19.
3. Song, Y.; Dixon, S.; Pearce, M. A survey of music recommendation systems and future perspectives. In Proceedings of the CMMR, 2012, pp. 395–410.
4. Konstas, I.; Stathopoulos, V.; Jose, J.M. On social networks and collaborative recommendation. In Proceedings of the Proc. of SIGIR, 2009, pp. 195–202.
5. Priem, J.; Piwowar, H.; Orr, R. OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts. *arXiv preprint arXiv:2205.01833* **2022**.
6. Guo, H.; Chen, B.; Tang, R.; Li, Z.; He, X. Autodis: Automatic discretization for embedding numerical features in CTR prediction. *arXiv preprint arXiv:2012.08986* **2020**.
7. Guo, H.; Tang, R.; Ye, Y.; Li, Z.; He, X. DeepFM: A factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* **2017**.
8. Wang, R.; Fu, B.; Fu, G.; Wang, M. Deep & cross network for ad click predictions. In *Proc. of KDD*; 2017; pp. 1–7.
9. Qin, J.; Zhu, J.; Chen, B.; Liu, Z.; Liu, W.; Tang, R.; Zhang, R.; Yu, Y.; Zhang, W. Rankflow: Joint optimization of multi-stage cascade ranking systems as flows. In Proceedings of the Proc. of SIGIR, 2022, pp. 814–824.
10. He, R.; McAuley, J. VBPR: Visual bayesian personalized ranking from implicit feedback. In Proceedings of the Proc. of AAAI, 2016.
11. Zhu, Y.; Xie, R.; Zhuang, F.; Ge, K.; Sun, Y.; Zhang, X.; Lin, L.; Cao, J. Learning to warm up cold item embeddings for cold-start recommendation with meta scaling and shifting networks. In Proceedings of the Proc. of SIGIR, 2021, pp. 1167–1176.

12. Xu, X.; Dong, H.; Qi, L.; Zhang, X.; Xiang, H.; Xia, X.; Xu, Y.; Dou, W. Cmclrec: Cross-modal contrastive learning for user cold-start sequential recommendation. In Proceedings of the Proc. of SIGIR, 2024, pp. 1589–1598.
13. Zha, D.; Feng, L.; Bhushanam, B.; Choudhary, D.; Nie, J.; Tian, Y.; Chae, J.; Ma, Y.; Kejariwal, A.; Hu, X. Autoshard: Automated embedding table sharding for recommender systems. In Proceedings of the Proc. of KDD, 2022, pp. 4461–4471.
14. Jha, G.K.; Thomas, A.; Jain, N.; Gobriel, S.; Rosing, T.; Iyer, R. Mem-rec: Memory efficient recommendation system using alternative representation. In Proceedings of the Proc. of ACML, 2024, pp. 518–533.
15. Zhou, G.; Deng, J.; Zhang, J.; Cai, K.; Ren, L.; Luo, Q.; Wang, Q.; Hu, Q.; Huang, R.; Wang, S.; et al. OneRec Technical Report. *arXiv preprint arXiv:2506.13695* **2025**.
16. Gupta, U.; Wu, C.J.; Wang, X.; Naumov, M.; Reagen, B.; Brooks, D.; Cottel, B.; Hazelwood, K.; Hempstead, M.; Jia, B.; et al. The architectural implications of facebook's dnn-based personalized recommendation. In Proceedings of the Proc. of HPCA, 2020, pp. 488–501.
17. Yang, L.; Wang, Y.; Yu, Y.; Weng, Q.; Dong, J.; Liu, K.; Zhang, C.; Zi, Y.; Li, H.; Zhang, Z.; et al. {GPU-Disaggregated} Serving for Deep Learning Recommendation Models at Scale. In Proceedings of the Proc. of NSDI, 2025, pp. 847–863.
18. Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H.W.; Sutton, C.; Gehrmann, S.; et al. Palm: Scaling language modeling with pathways. *JMLR* **2023**, pp. 1–113.
19. Wei, J.; Tay, Y.; Bommasani, R.; Raffel, C.; Zoph, B.; Borgeaud, S.; Yogatama, D.; et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682* **2022**.
20. Burges, C.; Shaked, T.; Renshaw, E.; Lazier, A.; Deeds, M.; Hamilton, N.; Hullender, G. Learning to rank using gradient descent. In Proceedings of the Proc. of ICML, 2005, pp. 89–96.
21. Xu, J.; Li, H. Adarank: A boosting algorithm for information retrieval. In Proceedings of the Proc. of SIGIR, 2007, pp. 391–398.
22. Covington, P.; Adams, J.; Sargin, E. Deep neural networks for youtube recommendations. In Proceedings of the Proc. of RecSys, 2016, pp. 191–198.
23. Evnine, A.; Ioannidis, S.; Kalimeris, D.; Kalyanaraman, S.; Li, W.; Nir, I.; Sun, W.; Weinsberg, U. Achieving a better tradeoff in multi-stage recommender systems through personalization. In Proceedings of the Proc. of KDD, 2024, pp. 4939–4950.
24. Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948* **2025**.
25. OpenAI. OpenAI o3 and o4-mini System Card. Technical report, OpenAI, 2025.
26. Jiang, Y.; Yang, Y.; Xia, L.; Luo, D.; Lin, K.; Huang, C. RecLM: Recommendation Instruction Tuning. *arXiv preprint arXiv:2412.19302* **2024**.
27. Xi, Y.; Liu, W.; Lin, J.; Cai, X.; Zhu, H.; Zhu, J.; Chen, B.; Tang, R.; Zhang, W.; Yu, Y. Towards open-world recommendation with knowledge augmentation from large language models. In Proceedings of the Proc. of RecSys, 2024, pp. 12–22.
28. Geng, S.; Liu, S.; Fu, Z.; Ge, Y.; Zhang, Y. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In Proceedings of the Proc. of RecSys, 2022, pp. 299–315.
29. Rajput, S.; Mehta, N.; Singh, A.; Hulikal Keshavan, R.; Vu, T.; Heldt, L.; Hong, L.; Tay, Y.; Tran, V.; Samost, J.; et al. Recommender systems with generative retrieval. *Proc. of NeurIPS* **2023**, pp. 10299–10315.
30. Liu, R.; Chen, H.; Bei, Y.; Shen, Q.; Zhong, F.; Wang, S.; Wang, J. Fine Tuning Out-of-Vocabulary Item Recommendation with User Sequence Imagination. *Proc. of NeurIPS* **2024**, pp. 8930–8955.
31. Hou, Y.; Mu, S.; Zhao, W.X.; Li, Y.; Ding, B.; Wen, J.R. Towards universal sequence representation learning for recommender systems. In Proceedings of the Proc. of KDD, 2022, pp. 585–593.
32. Deng, J.; Wang, S.; Cai, K.; Ren, L.; Hu, Q.; Ding, W.; Luo, Q.; Zhou, G. Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment. *arXiv preprint arXiv:2502.18965* **2025**.
33. Zhou, G.; Hu, H.; Cheng, H.; Wang, H.; Deng, J.; Zhang, J.; Cai, K.; Ren, L.; Ren, L.; Yu, L.; et al. Onerec-v2 technical report. *arXiv preprint arXiv:2508.20900* **2025**.
34. Liu, Z.; Wang, S.; Wang, X.; Zhang, R.; Deng, J.; Bao, H.; Zhang, J.; Li, W.; Zheng, P.; Wu, X.; et al. OneRec-Think: In-Text Reasoning for Generative Recommendation. *arXiv preprint arXiv:2510.11639* **2025**.
35. Zhai, J.; Liao, L.; Liu, X.; Wang, Y.; Li, R.; Cao, X.; Gao, L.; Gong, Z.; Gu, F.; He, M.; et al. Actions speak louder than words: Trillion-parameter sequential transducers for generative recommendations. *arXiv preprint arXiv:2402.17152* **2024**.

36. Zhu, J.; Fan, Z.; Zhu, X.; Jiang, Y.; Wang, H.; Han, X.; Ding, H.; Wang, X.; Zhao, W.; Gong, Z.; et al. RankMixer: Scaling Up Ranking Models in Industrial Recommenders. *arXiv preprint arXiv:2507.15551* **2025**.
37. Wang, W.; Bao, H.; Lin, X.; Zhang, J.; Li, Y.; Feng, F.; Ng, S.K.; Chua, T.S. Learnable item tokenization for generative recommendation. In Proceedings of the Proc. of CIKM, 2024, pp. 2400–2409.
38. Chen, Y.; Tan, J.; Zhang, A.; Yang, Z.; Sheng, L.; Zhang, E.; Wang, X.; Chua, T.S. On softmax direct preference optimization for recommendation. *Proc. of NeurIPS* **2024**, pp. 27463–27489.
39. Lin, J.; Dai, X.; Xi, Y.; Liu, W.; Chen, B.; Zhang, H.; Liu, Y.; Wu, C.; Li, X.; Zhu, C.; et al. How can recommender systems benefit from large language models: A survey. *TOIS* **2025**, pp. 1–47.
40. Liu, Q.; Zhao, X.; Wang, Y.; Wang, Y.; Zhang, Z.; Sun, Y.; Li, X.; Wang, M.; Jia, P.; Chen, C.; et al. Large Language Model Enhanced Recommender Systems: A Survey. *arXiv preprint arXiv:2412.13432* **2024**.
41. Wang, Q.; Li, J.; Wang, S.; Xing, Q.; et al. Towards next-generation llm-based recommender systems: A survey and beyond. *arXiv preprint arXiv:2410.19744* **2024**.
42. Hou, M.; Wu, L.; Liao, Y.; Yang, Y.; Zhang, Z.; Zheng, C.; Wu, H.; Hong, R. A Survey on Generative Recommendation: Data, Model, and Tasks. *arXiv preprint arXiv:2510.27157* **2025**.
43. Deldjoo, Y.; He, Z.; McAuley, J.; Korikov, A.; Sanner, S.; Ramisa, A.; Vidal, R.; Sathiamoorthy, M.; Kasirzadeh, A.; Milano, S. A review of modern recommender systems using generative models (gen-recsys). In Proceedings of the Proc. of KDD, 2024, pp. 6448–6458.
44. Li, Y.; Lin, X.; Wang, W.; Feng, F.; Pang, L.; Li, W.; Nie, L.; He, X.; Chua, T.S. A survey of generative search and recommendation in the era of large language models. *arXiv preprint arXiv:2404.16924* **2024**.
45. Wei, T.R.; Fang, Y. Diffusion Models in Recommendation Systems: A Survey. *arXiv arXiv:2501.10548* **2025**.
46. Yang, Z.; Lin, H.; Zhang, Z.; et al. Gr-llms: Recent advances in generative recommendation based on large language models. *arXiv preprint arXiv:2507.06507* **2025**.
47. Zhao, Y.; Tan, C.; Shi, L.; Zhong, Y.; Kou, F.; Zhang, P.; Chen, W.; Ma, C. Generative Recommender Systems: A Comprehensive Survey on Model, Framework, and Application. *Information Fusion* **2025**, p. 103919.
48. Resnick, P.; Iacovou, N.; Suchak, M.; Bergstrom, P.; Riedl, J. Grouplens: An open architecture for collaborative filtering of netnews. In Proceedings of the CSCW, 1994, pp. 175–186.
49. Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. Item-based collaborative filtering recommendation algorithms. In Proceedings of the Proc. of WWW, 2001, pp. 285–295.
50. Koren, Y.; Bell, R.; Volinsky, C. Matrix factorization techniques for recommender systems. *Computer* **2009**, pp. 30–37.
51. Zhang, Y. An introduction to matrix factorization and factorization machines in recommendation system, and beyond. *arXiv preprint arXiv:2203.11026* **2022**.
52. Zhou, G.; Zhu, X.; Song, C.; Fan, Y.; Zhu, H.; Ma, X.; Yan, Y.; Jin, J.; Li, H.; Gai, K. Deep interest network for click-through rate prediction. In Proceedings of the Proc. of KDD, 2018, pp. 1059–1068.
53. Ma, J.; Zhao, Z.; Yi, X.; Chen, J.; Hong, L.; Chi, E.H. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In Proceedings of the Proc. of KDD, 2018, pp. 1930–1939.
54. Sheng, X.R.; Zhao, L.; Zhou, G.; Ding, X.; Dai, B.; Luo, Q.; Yang, S.; Lv, J.; Zhang, C.; Deng, H.; et al. One model to serve all: Star topology adaptive recommender for multi-domain ctr prediction. In Proceedings of the Proc. of CIKM, 2021, pp. 4104–4113.
55. Wang, B.; Liu, F.; Zhang, C.; Chen, J.; Wu, Y.; Zhou, S.; Lou, X.; Wang, J.; Feng, Y.; Chen, C.; et al. LLM4DSR: Leveraging Large Language Model for Denoising Sequential Recommendation. *TOIS* **2025**, pp. 1–32.
56. Huang, F.; Bei, Y.; Yang, Z.; Jiang, J.; Chen, H.; Shen, Q.; Wang, S.; Karray, F.; Yu, P.S. Large Language Model Simulator for Cold-Start Recommendation. In Proceedings of the Proc. of WSDM, 2025, pp. 261–270.
57. Li, X.; Chen, B.; Hou, L.; Tang, R. Ctrl: Connect tabular and language model for ctr prediction. *CoRR* **2023**.
58. Liu, Q.; Wu, X.; Wang, W.; Wang, Y.; Zhu, Y.; Zhao, X.; Tian, F.; Zheng, Y. Llmemb: Large language model can be a good embedding generator for sequential recommendation. In Proceedings of the Proc. of AAAI, 2025, pp. 12183–12191.
59. Li, J.; Zhang, W.; Wang, T.; Xiong, G.; Lu, A.; Medioni, G. GPT4Rec: A generative framework for personalized recommendation and user interests interpretation. *arXiv preprint arXiv:2304.03879* **2023**.
60. Jiang, Y.; Ren, X.; Xia, L.; Luo, D.; Lin, K.; Huang, C. RecGPT: A Foundation Model for Sequential Recommendation. *arXiv preprint arXiv:2506.06270* **2025**.
61. Ngo, H.; Nguyen, D.Q. Recgpt: Generative pre-training for text-based recommendation. *arXiv preprint arXiv:2405.12715* **2024**.
62. Huang, Y.; Chen, Y.; Cao, X.; Yang, R.; Qi, M.; Zhu, Y.; Han, Q.; Liu, Y.; Liu, Z.; Yao, X.; et al. Towards Large-scale Generative Ranking. *arXiv preprint arXiv:2505.04180* **2025**.

63. Chen, B.; Guo, X.; Wang, S.; Liang, Z.; Lv, Y.; Ma, Y.; Xiao, X.; Xue, B.; Zhang, X.; Yang, Y.; et al. Onesearch: A preliminary exploration of the unified end-to-end generative framework for e-commerce search. *arXiv preprint arXiv:2509.03236* **2025**.
64. Kong, X.; Jiang, J.; Liu, B.; Xu, Z.; Zhu, H.; Xu, J.; Zheng, B.; Wu, J.; Wang, X. Think before Recommendation: Autonomous Reasoning-enhanced Recommender. *arXiv preprint arXiv:2510.23077* **2025**.
65. Han, R.; Yin, B.; Chen, S.; Jiang, H.; Jiang, F.; Li, X.; Ma, C.; Huang, M.; Li, X.; Jing, C.; et al. MTGR: Industrial-Scale Generative Recommendation Framework in Meituan. *arXiv preprint arXiv:2505.18654* **2025**.
66. Badrinath, A.; Agarwal, P.; Bhasin, L.; Yang, J.; Xu, J.; Rosenberg, C. PinRec: Outcome-Conditioned, Multi-Token Generative Retrieval for Industry-Scale Recommendation Systems. *arXiv arXiv:2504.10507* **2025**.
67. Yan, B.; Liu, S.; Zeng, Z.; Wang, Z.; Zhang, Y.; Yuan, Y.; Liu, L.; Liu, J.; Wang, D.; Su, W.; et al. Unlocking Scaling Law in Industrial Recommendation Systems with a Three-step Paradigm based Large User Model. *arXiv preprint arXiv:2502.08309* **2025**.
68. Guo, H.; Xue, E.; Huang, L.; Wang, S.; Wang, X.; Wang, L.; Wang, J.; Chen, S. Action is All You Need: Dual-Flow Generative Ranking Network for Recommendation. *arXiv preprint arXiv:2505.16752* **2025**.
69. Cui, Z.; Ma, J.; Zhou, C.; Zhou, J.; Yang, H. M6-rec: Generative pretrained language models are open-ended recommender systems. *arXiv preprint arXiv:2205.08084* **2022**.
70. Zhang, W.; Wu, C.; Li, X.; Wang, Y.; Dong, K.; Wang, Y.; Dai, X.; Zhao, X.; Guo, H.; Tang, R. Llmtreerec: Unleashing the power of large language models for cold-start recommendations. In Proceedings of the Proc. of COLING, 2025, pp. 886–896.
71. Bao, K.; Zhang, J.; Zhang, Y.; Wang, W.; Feng, F.; He, X. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In Proceedings of the Proc. of RecSys, 2023, pp. 1007–1014.
72. Bao, K.; Zhang, J.; Wang, W.; Zhang, Y.; Yang, Z.; Luo, Y.; Chen, C.; Feng, F.; Tian, Q. A bi-step grounding paradigm for large language models in recommendation systems. *TORS* **2025**, pp. 1–27.
73. Karra, S.R.; Tulabandhula, T. Interarec: Interactive recommendations using multimodal large language models. In Proceedings of the Proc. of PAKDD, 2024, pp. 32–43.
74. Chu, Z.; Hao, H.; Ouyang, X.; Wang, S.; Wang, Y.; Shen, Y.; Gu, J.; Cui, Q.; Li, L.; Xue, S.; et al. Leveraging large language models for pre-trained recommender systems. *arXiv preprint arXiv:2308.10837* **2023**.
75. Liao, J.; Li, S.; Yang, Z.; Wu, J.; Yuan, Y.; Wang, X.; He, X. Llara: Large language-recommendation assistant. In Proceedings of the Proc. of SIGIR, 2024, pp. 1785–1795.
76. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. Lora: Low-rank adaptation of large language models. *ICLR* **2022**, p. 3.
77. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the Proc. of NAACL, 2019, pp. 4171–4186.
78. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning transferable visual models from natural language supervision. In Proceedings of the Proc. of ICML, 2021, pp. 8748–8763.
79. Mentzer, F.; Minnen, D.; Agustsson, E.; Tschannen, M. Finite scalar quantization: Vq-vae made simple. *arXiv preprint arXiv:2309.15505* **2023**.
80. Lee, D.; Kim, C.; Kim, S.; Cho, M.; Han, W.S. Autoregressive image generation using residual quantization. In Proceedings of the Proc. of CVPR, 2022, pp. 11523–11532.
81. Jegou, H.; Douze, M.; Schmid, C. Product quantization for nearest neighbor search. *IEEE PAMI* **2010**, pp. 117–128.
82. Zheng, B.; Hou, Y.; Lu, H.; Chen, Y.; Zhao, W.X.; Chen, M.; Wen, J.R. Adapting large language models by integrating collaborative semantics for recommendation. In Proceedings of the ICDE, 2024, pp. 1435–1448.
83. Wang, Y.; Xun, J.; Hong, M.; Zhu, J.; Jin, T.; Lin, W.; Li, H.; Li, L.; Xia, Y.; Zhao, Z.; et al. Eager: Two-stream generative recommender with behavior-semantic collaboration. In Proceedings of the Proc. of KDD, 2024, pp. 3245–3254.
84. Xiao, L.; Wang, H.; Wang, C.; Ji, L.; Wang, Y.; Zhu, J.; Dong, Z.; Zhang, R.; Li, R. UNGER: Generative Recommendation with A Unified Code via Semantic and Collaborative Integration. *TOIS* **2025**.
85. Wei, Z.; Cai, K.; She, J.; Chen, J.; Chen, M.; Zeng, Y.; Luo, Q.; Zeng, W.; Tang, R.; Gai, K.; et al. OneLoc: Geo-Aware Generative Recommender Systems for Local Life Service. *arXiv preprint arXiv:2508.14646* **2025**.
86. Wang, D.; Huang, Y.; Gao, S.; Wang, Y.; Huang, C.; Shang, S. Generative Next POI Recommendation with Semantic ID. In Proceedings of the Proc. of KDD, 2025, pp. 2904–2914.

87. Qu, H.; Fan, W.; Zhao, Z.; Li, Q. Tokenrec: Learning to tokenize id for llm-based generative recommendations. *IEEE TKDE* **2025**.
88. Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G. The graph neural network model. *IEEE Trans. Neural Netw.* **2008**.
89. Yang, Y.; Ji, Z.; Li, Z.; Li, Y.; Mo, Z.; Ding, Y.; Chen, K.; Zhang, Z.; Li, J.; Li, S.; et al. Sparse meets dense: Unified generative recommendations with cascaded sparse-dense representations. *arXiv preprint arXiv:2503.02453* **2025**.
90. Wang, Y.; Zhou, S.; Lu, J.; Liu, Q.; Li, X.; Zhang, W.; Li, F.; Wang, P.; Xu, J.; Zheng, B.; et al. GFlowGR: Fine-tuning Generative Recommendation Frameworks with Generative Flow Networks. *arXiv preprint arXiv:2506.16114* **2025**.
91. Zhang, J.; Zhang, B.; Sun, W.; Lu, H.; Zhao, W.X.; Chen, Y.; Wen, J.R. Slow Thinking for Sequential Recommendation. *arXiv preprint arXiv:2504.09627* **2025**.
92. Lin, X.; Yang, C.; Wang, W.; Li, Y.; Du, C.; Feng, F.; Ng, S.K.; Chua, T.S. Efficient inference for large language model-based generative recommendation. *arXiv preprint arXiv:2410.05165* **2024**.
93. Ding, Y.; Hou, Y.; Li, J.; McAuley, J. Inductive generative recommendation via retrieval-based speculation. *arXiv preprint arXiv:2410.02939* **2024**.
94. Zhou, S.; Gan, W.; Liu, Q.; Lei, K.; Zhu, J.; Huang, H.; Xia, Y.; Tang, R.; Dong, Z.; Zhao, Z. RecBase: Generative Foundation Model Pretraining for Zero-Shot Recommendation. In Proceedings of the Proc. of EMNLP, 2025, pp. 15598–15610.
95. Yin, X.; Chen, S.; Hu, E. Regularized soft K-means for discriminant analysis. *Neurocomputing* **2013**, pp. 29–42.
96. Xie, W.; Wang, H.; Zhang, L.; Zhou, R.; Lian, D.; Chen, E. Breaking determinism: Fuzzy modeling of sequential recommendation using discrete state space diffusion model. *Proc. of NeurIPS* **2024**, pp. 22720–22744.
97. Hou, Y.; Li, J.; Shin, A.; Jeon, J.; Santhanam, A.; Shao, W.; Hassani, K.; Yao, N.; McAuley, J. Generating long semantic ids in parallel for recommendation. In Proceedings of the Proc. of KDD, 2025, pp. 956–966.
98. Jin, J.; Zhang, Y.; Feng, F.; He, X. Generative Multi-Target Cross-Domain Recommendation. *arXiv preprint arXiv:2507.12871* **2025**.
99. Wang, Y.; Gan, W.; Xiao, L.; Zhu, J.; Chang, H.; Wang, H.; Zhang, R.; Dong, Z.; Tang, R.; Li, R. Act-With-Think: Chunk Auto-Regressive Modeling for Generative Recommendation. *arXiv preprint arXiv:2506.23643* **2025**.
100. Yao, Y.; Li, Z.; Xiao, S.; Du, B.; Zhu, J.; Zheng, J.; Kong, X.; Jiang, Y. SaviorRec: Semantic-Behavior Alignment for Cold-Start Recommendation. *arXiv preprint arXiv:2508.01375* **2025**.
101. Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. *Proc. of NeurIPS* **2013**.
102. Jin, B.; Zeng, H.; Wang, G.; Chen, X.; Wei, T.; Li, R.; Wang, Z.; Li, Z.; Li, Y.; Lu, H.; et al. Language models as semantic indexers. *arXiv preprint arXiv:2310.07815* **2023**.
103. Li, W.; Zheng, K.; Lian, D.; Liu, Q.; Bao, W.; Yu, Y.E.; Song, Y.; Li, H.; Gai, K. Making Transformer Decoders Better Differentiable Indexers. In Proceedings of the Proc. of ICLR, 2025.
104. Liu, E.; Zheng, B.; Ling, C.; Hu, L.; Li, H.; Zhao, W.X. Generative recommender with end-to-end learnable item tokenization. In Proceedings of the Proc. of SIGIR, 2025, pp. 729–739.
105. Xu, Y.; Zhang, M.; Li, C.; Liao, Z.; Xing, H.; Deng, H.; Hu, J.; Zhang, Y.; Zeng, X.; Zhang, J. MMQ: Multimodal Mixture-of-Quantization Tokenization for Semantic ID Generation and User Behavioral Adaptation. *arXiv preprint arXiv:2508.15281* **2025**.
106. Luo, X.; Cao, J.; Sun, T.; Yu, J.; Huang, R.; Yuan, W.; Lin, H.; Zheng, Y.; Wang, S.; Hu, Q.; et al. Qarm: Quantitative alignment multi-modal recommendation at kuaishou. In Proceedings of the Proc. of CIKM, 2025, pp. 5915–5922.
107. Zheng, Z.; Wang, Z.; Yang, F.; Fan, J.; Zhang, T.; Wang, Y.; Wang, X. Ega-v2: An end-to-end generative framework for industrial advertising. *arXiv preprint arXiv:2505.17549* **2025**.
108. Wang, Y.; Pan, J.; Li, X.; Wang, M.; Wang, Y.; Liu, Y.; Liu, D.; Jiang, J.; Zhao, X. Empowering Large Language Model for Sequential Recommendation via Multimodal Embeddings and Semantic IDs. In Proceedings of the Proc. of CIKM, 2025, pp. 3209–3219.
109. Li, K.; Xiang, R.; Bai, Y.; Tang, Y.; Cheng, Y.; Liu, X.; Jiang, P.; Gai, K. Bbqrec: Behavior-bind quantization for multi-modal sequential recommendation. *arXiv preprint arXiv:2504.06636* **2025**.
110. Doh, S.; Choi, K.; Nam, J. TALKPLAY: Multimodal Music Recommendation with Large Language Models. *arXiv preprint arXiv:2502.13713* **2025**.
111. Hong, M.; Xia, Y.; Wang, Z.; Zhu, J.; Wang, Y.; Cai, S.; Yang, X.; Dai, Q.; Dong, Z.; Zhang, Z.; et al. EAGER-LLM: Enhancing Large Language Models as Recommenders through Exogenous Behavior-Semantic Integration. In Proceedings of the WWW, 2025, pp. 2754–2762.

112. He, R.; Heldt, L.; Hong, L.; Keshavan, R.; Mao, S.; Mehta, N.; Su, Z.; Tsai, A.; Wang, Y.; Wang, S.C.; et al. PLUM: Adapting Pre-trained Language Models for Industrial-scale Generative Recommendations. *arXiv preprint arXiv:2510.07784* **2025**.
113. Guo, H.; Chen, B.; Tang, R.; Zhang, W.; Li, Z.; He, X. An embedding learning framework for numerical features in ctr prediction. In Proceedings of the Proc. of KDD, 2021, pp. 2910–2918.
114. Lian, J.; Zhou, X.; Zhang, F.; Chen, Z.; Xie, X.; Sun, G. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In Proceedings of the Proc. of KDD, 2018, pp. 1754–1763.
115. Wu, Z.; Wang, X.; Chen, H.; Li, K.; Han, Y.; Sun, L.; Zhu, W. Diff4rec: Sequential recommendation with curriculum-scheduled diffusion augmentation. In Proceedings of the Proc. of ACM MM, 2023, pp. 9329–9335.
116. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR* **2020**, pp. 1–67.
117. Lin, J.; Men, R.; Yang, A.; Zhou, C.; Ding, M.; Zhang, Y.; Wang, P.; Wang, A.; Jiang, L.; Jia, X.; et al. M6: A chinese multimodal pretrainer. *arXiv preprint arXiv:2103.00823* **2021**.
118. Du, Z.; Qian, Y.; Liu, X.; Ding, M.; Qiu, J.; Yang, Z.; Tang, J. Glm: General language model pretraining with autoregressive blank infilling. In Proceedings of the Proc. of ACL, 2022, pp. 320–335.
119. Guo, X.; Chen, B.; Wang, S.; Yang, Y.; Lei, C.; Ding, Y.; Li, H. OneSug: The Unified End-to-End Generative Framework for E-commerce Query Suggestion. *arXiv preprint arXiv:2506.06913* **2025**.
120. Liu, A.; Feng, B.; Xue, B.; Wang, B.; Wu, B.; Lu, C.; Zhao, C.; Deng, C.; Zhang, C.; Ruan, C.; et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437* **2024**.
121. Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; Fan, Y.; Ge, W.; Han, Y.; Huang, F.; et al. Qwen technical report. *arXiv preprint arXiv:2309.16609* **2023**.
122. Ji, J.; Li, Z.; Xu, S.; Hua, W.; Ge, Y.; Tan, J.; Zhang, Y. Genrec: Large language model for generative recommendation. In Proceedings of the ECIR, 2024, pp. 494–502.
123. Lin, J.; Wang, T.; Qian, K. Rec-r1: Bridging generative large language models and user-centric recommendation systems via reinforcement learning. *arXiv preprint arXiv:2503.24289* **2025**.
124. Zhou, Z.; Zhu, C.; Lin, J.; Chen, B.; Tang, R.; Zhang, W.; Yu, Y. Generative Representational Learning of Foundation Models for Recommendation. *arXiv preprint arXiv:2506.11999* **2025**.
125. Luo, S.; Yao, Y.; He, B.; Huang, Y.; Zhou, A.; Zhang, X.; Xiao, Y.; Zhan, M.; Song, L. Integrating large language models into recommendation via mutual augmentation and adaptive aggregation. *arXiv preprint arXiv:2401.13870* **2024**.
126. Lin, H.; Yang, Z.; Xue, J.; Zhang, Z.; Wang, L.; Gu, Y.; Xu, Y.; Li, X. Spacetime-GR: A Spacetime-Aware Generative Model for Large Scale Online POI Recommendation. *arXiv preprint arXiv:2508.16126* **2025**.
127. Fu, K.; Zhang, T.; Xiao, S.; Wang, Z.; Zhang, X.; Zhang, C.; Yan, Y.; Zheng, J.; Li, Y.; Chen, Z.; et al. FORGE: Forming Semantic Identifiers for Generative Retrieval in Industrial Datasets. *arXiv arXiv:2509.20904* **2025**.
128. Gao, V.R.; Xue, C.; Versage, M.; Zhou, X.; Wang, Z.; Li, C.; Seonwoo, Y.; Chen, N.; Ge, Z.; Kundu, G.; et al. SynerGen: Contextualized Generative Recommender for Unified Search and Recommendation. *arXiv preprint arXiv:2509.21777* **2025**.
129. Zheng, B.; Liu, E.; Chen, Z.; Ma, Z.; Wang, Y.; Zhao, W.X.; Wen, J.R. Pre-training Generative Recommender with Multi-Identifier Item Tokenization. *arXiv preprint arXiv:2504.04400* **2025**.
130. Yan, H.; Xu, L.; Sun, J.; Ou, N.; Luo, W.; Tan, X.; Cheng, R.; Liu, K.; Chu, X. IntSR: An Integrated Generative Framework for Search and Recommendation. *arXiv preprint arXiv:2509.21179* **2025**.
131. Borisyuk, F.; Hertel, L.; Parameswaran, G.; Srivastava, G.; Ramanujam, S.S.; Ocejo, B.; Du, P.; Akterskii, A.; Daftary, N.; Tang, S.; et al. From Features to Transformers: Redefining Ranking for Scalable Impact. *arXiv preprint arXiv:2502.03417* **2025**.
132. Cui, Z.; Wu, H.; He, B.; Cheng, J.; Ma, C. Diffusion-based Contrastive Learning for Sequential Recommendation. *arXiv preprint arXiv:2405.09369* **2024**.
133. Li, Z.; Xia, L.; Huang, C. Recdiff: Diffusion model for social recommendation. In Proceedings of the Proc. of CIKM, 2024, pp. 1346–1355.
134. Zhao, J.; Wenjie, W.; Xu, Y.; Sun, T.; Feng, F.; Chua, T.S. Denoising diffusion recommender model. In Proceedings of the Proc. of SIGIR, 2024, pp. 1370–1379.
135. Song, Q.; Hu, J.; Xiao, L.; Sun, B.; Gao, X.; Li, S. Diffcl: A diffusion-based contrastive learning framework with semantic alignment for multimodal recommendations. *IEEE TNNLS* **2025**.

136. Li, W.; Huang, R.; Zhao, H.; Liu, C.; Zheng, K.; Liu, Q.; et al. DimeRec: A unified framework for enhanced sequential recommendation via generative diffusion models. In Proceedings of the Proc. of WSDM, 2025, pp. 726–734.
137. Liu, Z.; Zhu, Y.; Yang, Y.; Tang, G.; Huang, R.; Luo, Q.; Lv, X.; Tang, R.; Gai, K.; Zhou, G. DiffGRM: Diffusion-based Generative Recommendation Model. *arXiv preprint arXiv:2510.21805* **2025**.
138. Xie, Y.; Ren, X.; Qi, Y.; Hu, Y.; Shan, L. RecLLM-R1: A Two-Stage Training Paradigm with Reinforcement Learning and Chain-of-Thought v1. *arXiv preprint arXiv:2506.19235* **2025**.
139. Xing, H.; Deng, H.; Mao, Y.; Hu, J.; Xu, Y.; Zhang, H.; Wang, J.; Wang, S.; Zhang, Y.; Zeng, X.; et al. REG4Rec: Reasoning-Enhanced Generative Model for Large-Scale Recommendation Systems. *arXiv preprint arXiv:2508.15308* **2025**.
140. Huang, L.; Guo, H.; Peng, L.; Zhang, L.; Wang, X.; Wang, D.; et al. SessionRec: Next Session Prediction Paradigm For Generative Sequential Recommendation. *arXiv preprint arXiv:2502.10157* **2025**.
141. Afsar, M.M.; Crump, T.; Far, B. Reinforcement learning based recommender systems: A survey. *ACM Computing Surveys* **2022**, pp. 1–38.
142. Zhang, C.; Chen, S.; Zhang, X.; Dai, S.; Yu, W.; Xu, J. Reinforcing Long-Term Performance in Recommender Systems with User-Oriented Exploration Policy. In Proceedings of the Proc. of SIGIR, 2024, pp. 1850–1860.
143. Sharma, A.; Li, H.; Li, X.; Jiao, J. Optimizing novelty of top-k recommendations using large language models and reinforcement learning. In Proceedings of the Proc. of KDD, 2024, pp. 5669–5679.
144. Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C.D.; Ermon, S.; Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Proc. of NeurIPS* **2023**, pp. 53728–53741.
145. Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300* **2024**.
146. Liao, J.; He, X.; Xie, R.; Wu, J.; Yuan, Y.; Sun, X.; Kang, Z.; Wang, X. Rosepo: Aligning llm-based recommenders with human values. *arXiv preprint arXiv:2410.12519* **2024**.
147. Gao, C.; Chen, R.; Yuan, S.; Huang, K.; Yu, Y.; He, X. SPRec: Leveraging self-play to debias preference alignment for large language model-based recommendations. *arXiv e-prints* **2024**, pp. arXiv–2412.
148. Chen, S.; Chen, B.; Yu, C.; Luo, Y.; Yi, O.; Cheng, L.; Zhuo, C.; Li, Z.; Wang, Y. VRAgent-R1: Boosting Video Recommendation with MLLM-based Agents via Reinforcement Learning. *arXiv arXiv:2507.02626* **2025**.
149. Huang, P.S.; He, X.; Gao, J.; Deng, L.; Acero, A.; Heck, L. Learning deep structured semantic models for web search using clickthrough data. In Proceedings of the Proc. of CIKM, 2013, pp. 2333–2338.
150. Liang, Z.; Wu, C.; Huang, D.; Sun, W.; Wang, Z.; Yan, Y.; Wu, J.; Jiang, Y.; Zheng, B.; Chen, K.; et al. Tbgrecall: A generative retrieval model for e-commerce recommendation scenarios. *arXiv arXiv:2508.11977* **2025**.
151. Liu, C.; Cao, J.; Huang, R.; Zheng, K.; Luo, Q.; Gai, K.; Zhou, G. KuaiFormer: Transformer-Based Retrieval at Kuaishou. *arXiv preprint arXiv:2411.10057* **2024**.
152. Chen, J.; Chi, L.; Peng, B.; Yuan, Z. Hllm: Enhancing sequential recommendations via hierarchical large language models for item and user modeling. *arXiv preprint arXiv:2409.12740* **2024**.
153. Coburn, J.; Tang, C.; Asal, S.A.; Agrawal, N.; Chinta, R.; Dixit, H.; Dodds, B.; Dwarakapuram, S.; Firoozshahian, A.; Gao, C.; et al. Meta's Second Generation AI Chip: Model-Chip Co-Design and Productionization Experiences. In Proceedings of the Proc. of ISCA, 2025, pp. 1689–1702.
154. Xi, Y.; Liu, W.; Dai, X.; Tang, R.; Zhang, W.; Liu, Q.; He, X.; Yu, Y. Context-aware reranking with utility maximization for recommendation. *arXiv preprint arXiv:2110.09059* **2021**.
155. Zhang, K.; Wang, X.; Liu, S.; Yang, H.; Li, X.; Hu, L.; Li, H.; Cao, Q.; Sun, F.; Gai, K. GoalRank: Group-Relative Optimization for a Large Ranking Model. *arXiv preprint arXiv:2509.22046* **2025**.
156. Meng, Y.; Guo, C.; Cao, Y.; Liu, T.; Zheng, B. A generative re-ranking model for list-level multi-objective optimization at taobao. In Proceedings of the Proc. of SIGIR, 2025, pp. 4213–4218.
157. Wang, Y.; Hu, M.; Huang, Z.; Li, D.; Yang, D.; Lu, X. Kc-genre: A knowledge-constrained generative re-ranking method based on large language models for knowledge graph completion. *arXiv preprint arXiv:2403.17532* **2024**.
158. Hou, Y.; Zhang, J.; Lin, Z.; Lu, H.; Xie, R.; McAuley, J.; Zhao, W.X. Large language models are zero-shot rankers for recommender systems. In Proceedings of the ECIR, 2024, pp. 364–381.
159. Zhang, Z.; Liu, S.; Yu, J.; Cai, Q.; Zhao, X.; Zhang, C.; Liu, Z.; Liu, Q.; Zhao, H.; Hu, L.; et al. M3oe: Multi-domain multi-task mixture-of experts recommendation framework. In Proceedings of the Proc. of SIGIR, 2024, pp. 893–902.
160. Chang, J.; Zhang, C.; Hui, Y.; Leng, D.; et al. Pepnet: Parameter and embedding personalized network for infusing with personalized prior information. In Proceedings of the Proc. of KDD, 2023, pp. 3795–3804.

161. Li, X.; Yan, F.; Zhao, X.; Wang, Y.; Chen, B.; Guo, H.; Tang, R. Hamur: Hyper adapter for multi-domain recommendation. In Proceedings of the Proc. of CIKM, 2023, pp. 1268–1277.
162. Hu, P.; Lu, W.; Wang, J. From IDs to Semantics: A Generative Framework for Cross-Domain Recommendation with Adaptive Semantic Tokenization. *arXiv preprint arXiv:2511.08006* **2025**.
163. Pang, M.; Yuan, C.; He, X.; Fang, Z.; Xie, D.; Qu, F.; Jiang, X.; Peng, C.; Lin, Z.; Luo, Z.; et al. Generative Retrieval and Alignment Model: A New Paradigm for E-commerce Retrieval. In Proceedings of the WWW, 2025, pp. 413–421.
164. Shi, T.; Xu, J.; Zhang, X.; Zang, X.; Zheng, K.; Song, Y.; Yu, E. Unified Generative Search and Recommendation. *arXiv preprint arXiv:2504.05730* **2025**.
165. Chen, Y.; Berkhin, P.; Anderson, B.; Devanur, N.R. Real-time bidding algorithms for performance-based display ad allocation. In Proceedings of the Proc. of KDD, 2011, pp. 1307–1315.
166. Fujimoto, S.; Meger, D.; Precup, D. Off-policy deep reinforcement learning without exploration. In Proceedings of the Proc. of ICML, 2019, pp. 2052–2062.
167. Guo, J.; Huo, Y.; Zhang, Z.; Wang, T.; Yu, C.; Xu, J.; Zheng, B.; Zhang, Y. Generative auto-bidding via conditional diffusion modeling. In Proceedings of the Proc. of KDD, 2024, pp. 5038–5049.
168. Li, Y.; Mao, S.; Gao, J.; Jiang, N.; Xu, Y.; Cai, Q.; Pan, F.; Jiang, P.; An, B. GAS: Generative Auto-bidding with Post-training Search. In Proceedings of the WWW, 2025, pp. 315–324.
169. Gao, J.; Li, Y.; Mao, S.; Jiang, P.; Jiang, N.; Wang, Y.; Cai, Q.; Pan, F.; Jiang, P.; Gai, K.; et al. Generative auto-bidding with value-guided explorations. In Proceedings of the Proc. of SIGIR, 2025, pp. 244–254.
170. Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T.B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; Amodei, D. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361* **2020**.
171. Li, J.; Xu, J.; Huang, S.; Chen, Y.; Li, W.; Liu, J.; Lian, Y.; Pan, J.; et al. Large language model inference acceleration: A comprehensive hardware perspective. *arXiv preprint arXiv:2410.04466* **2024**.
172. Miao, X.; Oliaro, G.; Zhang, Z.; Cheng, X.; Jin, H.; Chen, T.; Jia, Z. Towards efficient generative large language model serving: A survey from algorithms to systems. *ACM Computing Surveys* **2025**, pp. 1–37.
173. Catania, F.; Spitale, M.; Garzotto, F. Conversational agents in therapeutic interventions for neurodevelopmental disorders: A survey. *ACM Computing Surveys* **2023**, pp. 1–34.
174. Lin, J.; Shan, R.; Zhu, C.; Du, K.; Chen, B.; Quan, S.; Tang, R.; Yu, Y.; Zhang, W. Rella: Retrieval-enhanced large language models for lifelong sequential behavior comprehension in recommendation. In Proceedings of the WWW, 2024, pp. 3497–3508.
175. Liu, Q.; Zhu, J.; Lai, Y.; Dong, X.; Fan, L.; Bian, Z.; Dong, Z.; Wu, X.M. Evaluating recabilities of foundation models: A multi-domain, multi-dataset benchmark. *arXiv preprint arXiv:2508.21354* **2025**.
176. Zhang, Y.; Qiao, S.; Zhang, J.; Lin, T.H.; Gao, C.; Li, Y. A survey of large language model empowered agents for recommendation and search: Towards next-generation information retrieval. *arXiv preprint arXiv:2503.05659* **2025**.
177. Zhu, Y.; Steck, H.; Liang, D.; He, Y.; Kallus, N.; Li, J. LLM-based Conversational Recommendation Agents with Collaborative Verbalized Experience. In Proceedings of the Proc. of EMNLP Findings, 2025, pp. 2207–2220.
178. Zhao, Y.; Wu, J.; Wang, X.; Tang, W.; Wang, D.; De Rijke, M. Let me do it for you: Towards llm empowered recommendation via tool learning. In Proceedings of the Proc. of SIGIR, 2024, pp. 1796–1806.
179. Kuaishou Technology. Kuaishou Unveils Proprietary Video Generation Model "Kling", 2024.
180. OpenAI. Sora: Creating video from text, 2024.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.