Preprints.org

Article

Not peer-reviewed version

Generative Recommendation: A Survey of Models, Systems, and Industrial Advances

Siqi Liang * and Yudi Zhang *

Posted Date: 8 December 2025

doi: 10.20944/preprints202512.0741.v1

Keywords: generative recommendation system; large language model

Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1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

Generative Recommendation: A Survey of Models, Systems, and Industrial Advances

Siqi Liang [1,*] and Yudi Zhang [2,*]

[1] Purdue University
[2] Iowa State University
* Correspondence: lsq950917@gmail.com (S.L.); yudiz@iastate.edu (Y.Z.)

Abstract

The rapid advancement of large language models (LLMs) has sparked a paradigm shift in recommender system design, transforming traditional discriminative ranking architectures into unified generative frameworks. Conventional multi-stage cascading architectures, comprising retrieval, ranking, and auction, have achieved remarkable industrial success but remain limited by semantic gaps, stage inconsistencies, and feature fragmentation. In contrast, emerging Generative Recommendation Systems (GRS) formulate recommendation as a sequence generation task, leveraging transformer-based architectures and tokenized item representations to unify retrieval, ranking, and reasoning within a single generative backbone. This survey provides the first comprehensive synthesis of recent industrial progress in generative recommendation. We categorize over twenty state-of-the-art systems along four complementary dimensions including modeling paradigm (encoder-only, decoder-only, encoder–decoder), functional scope (retrieval, ranking, end-to-end frameworks), representation space (semantic ID–based, dense and hybrid representation), and training and alignment objectives (no alignment, alignment via reinforcement learning and preference optimization). We further summarize emerging research on scaling laws for generative recommenders and analyze their implications for efficiency, generalization, and model–data scaling behavior. Our analysis reveals three converging trends: (1) the unification of retrieval and ranking under shared generative architectures; (2) the integration of preference-aligned, reward-driven learning objectives; and (3) the rapid adoption of multimodal and cross-domain foundation models. Finally, we identify open challenges, including latency–scalability trade-offs, robustness under distribution shifts, interpretability of generative reasoning, and multimodal integration. Then propose a forward looking roadmap to guide future research and industrial deployment of next-generation generative recommender systems.

Keywords: generative recommendation system; large language model

1. Introduction

Recommender systems have become foundational to digital ecosystems, driving personalized user experiences across e-commerce, content streaming, and social media platforms while serving as the primary revenue engine for online advertising systems [1,2]. Modern industrial systems typically employ a multi-stage cascading architecture (MCA) consisting of retrieval, ranking, and auction [3], progressively narrowing billions of candidate items to dozens of final recommendations. While traditional approaches like collaborative filtering [1] and matrix factorization [4] established early foundations, contemporary

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

systems predominantly rely on Deep Learning Recommendation Models (DLRMs) that follow the MLP & Embedding paradigm [5].

Current retrieval systems predominantly rely on two-tower architectures with approximate nearest neighbor search [6], which suffer from three critical shortcomings: (1) The separate encoding of queries and items prevents deep cross-feature interactions during retrieval, creating a semantic gap in matching; (2) The embedding-based approach struggles with fine-grained feature-level matching, particularly for cold-start items or long-tail queries [7]; (3) The static nature of pre-computed item embeddings cannot adapt to real-time context changes in user sessions.

While modern ranking models like DIN [2] and DCN [8] have advanced predictive accuracy, they inherit three fundamental constraints: (1) The independent Click Through Rate (CTR) assumption ignores crucial item-item interactions within candidate sets, artificially isolating prediction decisions; (2) Heavy reliance on manual feature engineering introduces information bottlenecks, where human-designed feature crosses and aggregations fail to capture complex behavioral patterns; (3) The computational complexity scales near-linearly with candidate set size, making quality improvements prohibitively expensive at industrial scales [9].

Such traditional multi-stage cascade architecture introduces systemic inefficiencies: (1) Stage inconsistency emerges when different modules employ conflicting optimization objectives or capacity budgets, causing prediction discrepancies that compound through the pipeline [3]; (2) The reranking paradox occurs when initial scores become invalid after permutation, as most systems fail to model the non-stationary nature of item relevance in different contexts [10]; (3) Information fragmentation arises when critical signals (e.g., real-time feedback) cannot propagate backward through the pipeline stages.

Recently, the emergence of generative models has catalyzed a paradigm shift in the design of recommender systems. Inspired by advances in natural language processing (NLP), particularly large language models (LLMs), recent studies have demonstrated the effectiveness of sequence generation for retrieval [11,12], reranking [13,14], and even full-stack systems unifying search, recommendation, and query understanding [15–19]. Three key innovations characterize this transition:

- Dynamic Representation Learning: Unlike static two-tower retrievers, generative approaches, for example, TIGER [11] and LIGER [20] employ adaptive tokenization strategies that learn cascaded sparse-dense representations in a unified space. This enables simultaneous handling of semantic matching (through dense embeddings) and exact matching (via sparse lexical features), as shown in COBRA [21].
- Industrial-Scale Optimization: Frameworks such as HSTU [22], EGA-V1/V2 [3,23] and Meituan's MTGRBoost [**?**] demonstrate how generative models achieve superior computational efficiency through distributed parallelism, mixed precision, and model sharding—scaling to hundreds of GPUs while maintaining low inference latency.
- Architectural Unification: Modern generative recommenders like GRAM [12] and COBRA[21] overcome the semantic gap between retrieval and ranking through transformer-based architectures that jointly optimize for both breadth (recall) and precision (ranking). The MTGR framework [24] demonstrates how trillion-parameter sequential transducers [22] can replace multi-stage pipelines with single-model solutions.

In this survey, we present a comprehensive and structured overview of the emerging landscape of generative recommendation systems as deployed in large-scale industrial environments. Section 2 introduces the conceptual evolution from discriminative recommendation to generative paradigms, highlighting the motivations and foundational shifts that underpin this transition. Section 3 categorizes recent industrial generative recommender systems along four complementary dimensions: (A) *Modeling*

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

paradigm, (B) *Functional scope*, (C) *Representation space*, and (D) *Training and alignment objectives*. Each dimension captures a distinct facet of how generative modeling principles have been adapted, optimized, and scaled within real-world recommender ecosystems. Section 4 summarizes recent progress in understanding the *scaling laws* that govern generative recommendation models, elucidating how model capacity, data scale, and performance interact in industrial settings. Finally, Section 7 discusses key open challenges and outlines promising research directions for the next generation of generative recommender systems.

Our survey reveals several converging trends: the unification of retrieval and ranking within end-to-end generative architectures; the growing emphasis on preference alignment and reward-driven optimization; and the rapid integration of multimodal understanding into recommender pipelines. We conclude by identifying persistent challenges, such as latency–scalability trade-offs, robustness under distribution shifts, nuanced user intent modeling, interpretability of generative reasoning, and seamless multimodal fusion, and by outlining a forward-looking roadmap to guide both future research and industrial deployment of next-generation generative recommendation systems.

2. From Discriminative to Generative Recommendation

2.1. Discriminative Recommendation (DR)

Traditional discriminative recommenders, such as DIN [2] and DCN [8] model user–item interactions through score estimation. These systems learn a scoring function:

$$f_\theta(u, i) : \mathcal{U} \times \mathcal{I} \to \mathbb{R} \tag{1}$$

where $u \in \mathcal{U}$ is a user representation, $i \in \mathcal{I}$ is an item representation, and $f_\theta(u, i)$ is the predicted relevance score. The recommendation list is obtained by sorting scores:

$$\hat{\mathcal{R}}_u = \text{TopK}_i f_\theta(u, i) \tag{2}$$

and the model parameters θ are optimized using a discriminative loss (e.g., logistic, hinge, or pairwise BPR):

$$\mathcal{L}_{DR} = -\sum_{(u,i^+,i^-)} \log \sigma(f_\theta(u, i^+) - f_\theta(u, i^-)) \tag{3}$$

While efficient, DR models face key limitations:

- **Contextual generalization.** Discriminative models rely on fixed feature spaces and limited contextual windows. Given a scoring function $f_\theta(u, i)$, the model often operates on static embeddings derived from short-term behavior, without the ability to reason over high-level intents or open-ended goals. For example, in a query such as *"gift ideas for a 10-year-old"*, a DR model trained on co-click patterns cannot generalize to unseen combinations of age, context, and semantic intent.
- **Decoupled stages of retrieval and ranking.** Industrial pipelines typically separate candidate retrieval (R_ϕ) and ranking (f_θ), where

  $$\hat{\mathcal{R}}_u = \text{TopK}_i f_\theta(u, i), \qquad \text{with candidates } i \sim R_\phi(u).$$

  This modularization, while scalable, breaks the end-to-end optimization of user satisfaction. The retrieval model is optimized for recall, whereas the ranking model focuses on precision, often leading to sub-optimal joint performance.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

- **Token compositionality and representation bottlenecks.** DR systems treat item IDs as atomic categorical variables $i \in \{1, \ldots, |\mathcal{I}|\}$. This prevents compositional reasoning over attributes such as color, brand, or visual style. Multimodal signals—text, image, or video embeddings—are often appended post-hoc and not integrated into the underlying generative process, resulting in limited cross-modal understanding.
- **Limited reasoning and language grounding.** Since DR models predict scalar scores rather than structured outputs, they cannot naturally explain or verbalize recommendations. In contrast, generative models can express reasoning paths, e.g.,

$$P_\theta(\text{“recommend item A because similar users liked B”} \mid u),$$

  enabling interpretability and human-aligned feedback. This gap restricts the deployment of DR systems in conversational or interactive settings.
- **Sparse supervision and cold-start sensitivity.** DR models depend on observed user–item interactions $\mathcal{D} = \{(u, i, y)\}$, where y denotes engagement labels. The learning signal is extremely sparse—only positive clicks or purchases are known—making cold-start or long-tail item generalization difficult. Without pretraining on large textual or multimodal corpora, embeddings for new users/items remain uninformative.
- **Inflexibility to multi-task or multi-objective learning.** In industrial applications, recommendation must optimize heterogeneous objectives (CTR, GMV, diversity, fairness, latency). Traditional DR models require explicit multi-tower or multi-head architectures to approximate joint optimization:

$$\mathcal{L} = \sum_k w_k \mathcal{L}_k,$$

  but gradients across tasks often conflict, leading to unstable or biased optimization. Generative frameworks, on the other hand, can encode objectives as instructions or natural-language prompts, achieving more flexible multi-intent modeling.

2.2. Generative Recommendation (GR)

Generative recommendation (GR) redefines the problem as conditional sequence generation. Given a user behavior sequence:

$$\mathbf{x}_u = [i_1, i_2, \ldots, i_T]$$

a generative model defines the joint probability

$$P_\theta(\mathbf{x}_u) = \prod_{t=1}^{T} P_\theta(i_t \mid i_{<t}, u, \mathbf{c}_u),$$

where $\mathbf{c}_u$ denotes auxiliary context (query, time, device, etc.). Training minimizes the negative log-likelihood:

$$\mathcal{L}_{\text{GR}} = -\sum_u \sum_{t=1}^{T} \log P_\theta(i_t \mid i_{<t}, u, \mathbf{c}_u).$$

At inference time, the model generates top-k future items via beam search or sampling.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

This formulation unifies retrieval, ranking, and reasoning, and allows LLM-based architectures to reuse language pretraining.

$$i_{t+1} \sim P_\theta(i|u, i_{1:t}) \tag{4}$$

where the model generates the next item as a token conditioned on the user's context or textual query. Industrial deployment since 2023 has demonstrated GR's superiority in adaptability, personalization, and explainability—bridging retrieval, ranking, and even search dialogue [11,16,22,24–27].

As summarized in Table 1, DLRM and generative recommendation (GR) systems differ substantially in how they model user–item interactions, how they structure the training pipeline, and how they execute inference in production. The generative paradigm is still in its infancy in recommender research, but early results are promising.

Table 1. Comparison between DLRM and GR in Modeling, Training, and Inference.

	DLRM	**GR**
Model	Uses pointwise samples. The model structure emphasizes a large cross-module to combine user and item features.	Uses listwise samples. The model focuses on the User module to learn long-term behavioral patterns.
Training	Each exposure corresponds to one sample; repeated user interactions lead to redundant user feature computation. Negative sampling is required to reduce training cost.	Each exposure is treated as a token; all exposures from a user are packed into a single sequence sample. User features are computed once. Negative sampling is not required, and the cost of full-data training is negligible.
Inference	Inference targets differ across services, and the large Cross module must often be recomputed per request, leading to low computational reusability.	Cross module is computed independently per candidate, while User module computation is shared across candidates. KV cache can be applied to reduce attention costs across requests, improving throughput and enabling large-scale deployment.

3. Generative Recommendation Models in Industry

While diverse in implementation, recent industrial systems can be systematically understood through four dimensions: (A) modeling paradigm, (B) functional scope, (C) representation space, and (D) training and alignment objective. Each dimension captures a distinct aspect of how generative modeling has been adopted and scaled within industrial recommender ecosystems. By organizing the literature along these axes, we chart the evolutionary trajectory of generative recommendation in Figure 1 and Table 2

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

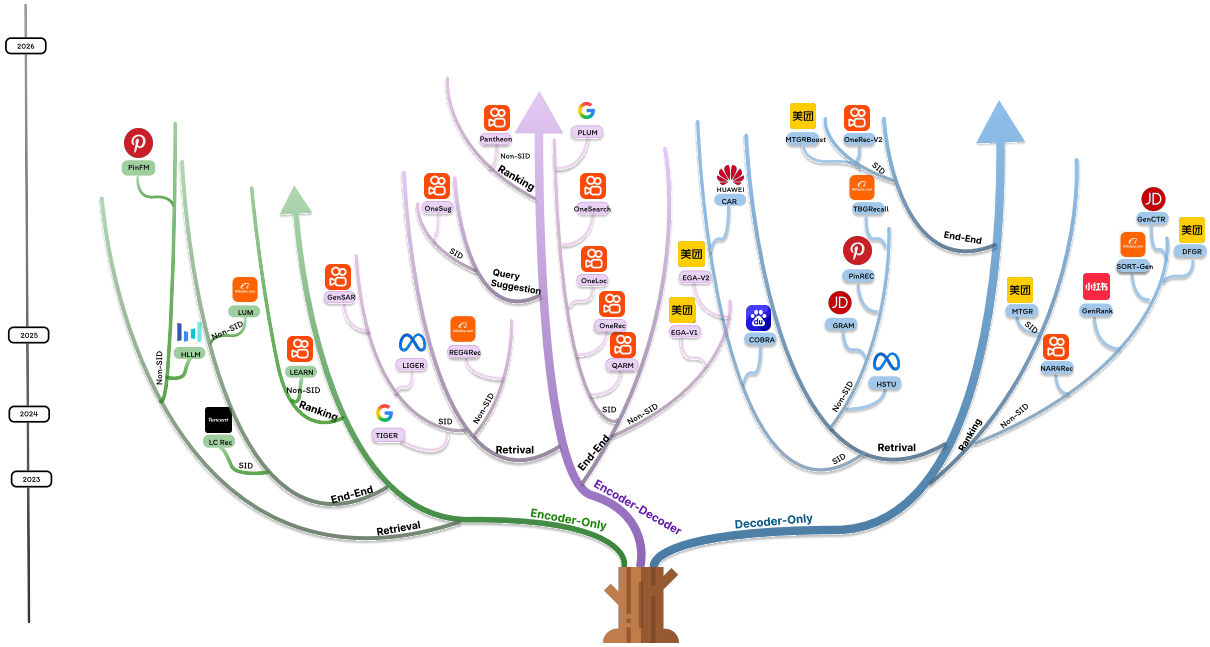

Figure 1. The evolutionary tree of modern generative recommender systems illustrates the progression of industrial RecSys research over recent years and highlights the most representative models. Models appearing on the same branch share closer architectural or conceptual lineage. Encoder-only frameworks are shown in green, encoder–decoder architectures in purple, and decoder-only models in blue. The vertical axis corresponds to release dates, reflecting the chronological evolution of the field. Each branch further subdivides according to functional scope, such as retrieval, ranking, end-to-end recommendation, or query suggestion, and is subsequently categorized by whether the model employs Semantic ID (SID)-based or non-SID representations.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

Table 2. Summary of Modern Generative Recommendation Model in Industries.

Paper	Venue	Model Type	Scope	Representation Space	Company
TIGER [11]	NIPS 2023	Encoder-decoder	Retrieval	SID	Google
LC Rec [25]	ICDE 2024	Encoder only	End-to-end	SID	Tencent
HSTU [22]	ICML 2024	Decoder only	Ranking/Retrieval	Non-SID	Meta
QARM [28]	Arxiv 2024	Encoder-decoder	End-to-end	SID	Kuaishou
GenSAR [15]	Arxiv 2025	Encoder–Decoder	Retrieval	SID	Kuaishou
LEARN [29]	AAAI 2025	Encoder only	Ranking	Non-SID	Kuaishou
HLLM [30]	RECSYS 2025	Encoder only	Ranking/Retrieval	Non-SID	ByteDance
LUM [31]	Arxiv 2025	Encoder only	End-to-End	Non-SID	Alibaba
LIGER [20]	TMLR 2025	Encoder-decoder	Retrieval	SID	Meta
GRAM [12]	WWW 2025	Decoder only	Retrieval	Non-SID	JD.com
COBRA [21]	Arxiv 2025	Decoder only	Retrieval	SID	Baidu
PinRec [32]	Arxiv 2025	Decoder only	Retrieval	Non-SID	Pinterest
Pantheon [33]	CoRR 2025	Encoder–Decoder	Reranking	Non-SID	Kuaishou
TBGRecall [34]	Arxiv 2025	Decoder only	Retrieval	Non-SID	Taobao
REG4Rec [35]	Arxiv 2025	Encoder-decoder	Retrieval	Non-SID	Alibaba
NAR4Rec [13]	KDD 2024	Decoder only	Ranking	Non-SID	Kuaishou
GenRank [36]	Arxiv 2025	Decoder only	Reranking	Non-SID	Xiaohongshu
MTGR [24]	CIKM 2025	Decoder only	Ranking	SID	Meituan
MTGRBoost [37]	Arxiv 2025	Decoder only	End-to-End	Non-SID	Meituan
SORT-Gen [38]	SIGIR 2025	Decoder only	(Re)Ranking	Non-SID	Taobao
GenCTR [39]	Arxiv 2025	Decoder only	Ranking	Non-SID	JD.com
DFGR [40]	Arxiv 2025	Decoder only	Ranking	Non-SID	Meituan
OneRec [16]	Arxiv 2025	Encoder-decoder	End-to-end	SID	Kuaishou
OneRec-V2 [41]	Arxiv 2025	Decoder only	End-to-end	SID	Kuaishou
EGA-V1 [3]	Arxiv 2025	Encoder-decoder	End-to-end	Non-SID	Meituan
EGA-V2 [23]	CIKM 2025	Encoder-decoder	End-to-end	Non-SID	Meituan
OneLoc [42]	Arxiv 2025	Encoder-decoder	End-to-end	SID	Kuaishou
OneSearch [27]	Arxiv 2025	Encoder-decoder	End-to-end	SID	Kuaishou
OneSug [43]	Arxiv 2025	Encoder-decoder	Query suggestion	SID	Kuaishou
CAR [44]	Arxiv 2025	Decoder only	Retrieval	SID	Huawei
PinFM [45]	RECSYS 2025	Encoder only	Retrieval	Non-SID	Pinterest
PLUM [46]	Arxiv 2025	Encoder-decoder	End-to-end	SID	Google

3.1. Modeling Paradigm

3.1.1. Encoder-Only Models

Encoder-only architectures represent the earliest industrial attempts to adapt large language models (LLMs) for recommendation without introducing autoregressive decoding. Instead of explicitly generating item sequences, these models focus on encoding user, item, and contextual information into a shared latent space that implicitly supports generative inference. Typically built upon bidirectional Transformer backbones (e.g., BERT-style encoders), they capture both semantic and collaborative dependencies through self-attention, producing dense contextual embeddings for downstream retrieval or ranking.

In practice, encoder-only models serve two roles in the generative ecosystem. First, they act as *representation backbones* that transfer pretrained language or multimodal knowledge to recommendation domains. For instance, LC Rec [25] from Tencent initiates this line of research by injecting collaborative adapters into frozen LLM backbones. Through lightweight parameter-efficient tuning, LC Rec [25] integrates user–item relational signals into textual token spaces, enabling high-quality semantic matching for recommendation and retrieval tasks. Building on this idea, LEARN [29] from Kuaishou formalizes a knowledge adaptation framework that transfers general LLM knowledge to recommendation domains through supervised contrastive learning, achieving strong practical gains in cold-start and long-tail settings. HLLM [30] from ByteDance further extends this paradigm to sequential modeling by organizing users and items hierarchically within a large-scale LLM encoder, improving session-level coherence and

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

personalization. LUM [31] from Alibaba explores scaling laws of encoder-only recommendation models, demonstrating power-law improvements with model and data size while identifying diminishing returns beyond trillion-parameter scales. PinFM [45] from Pinterest generalizes this paradigm to multimodal settings by jointly encoding visual, textual, and behavioral signals in a foundation-scale Transformer encoder, achieving strong cross-domain transfer across discovery, shopping, and ads applications.

Overall, encoder-only architectures focus on efficient representation transfer, knowledge distillation, and scaling analysis. They are computationally efficient and easily integrated into existing retrieval pipelines but lack the explicit generative capacity to model user intents or item sequences autoregressively.

3.1.2. Encoder–Decoder Models

Encoder–decoder architectures form one of the most influential paradigms in generative recommendation, providing a balanced trade-off between contextual understanding and generative flexibility. Analogous to sequence-to-sequence modeling in NLP, the encoder processes user–item interactions and contextual signals into a latent representation, while the decoder autoregressively generates target items, queries, or ranking decisions conditioned on that encoded context. This bidirectional encoding and unidirectional decoding design enables the model to simultaneously capture long-range dependencies and support controllable generation, making it well suited for both retrieval and re-ranking tasks.

TIGER [11] from Google initiates this line of research by framing recommendation as a Transformer-based encoder–decoder retrieval problem. It replaces dense vector indexing with text-to-text generation, demonstrating that generative retrieval can match or surpass dense retrieval when trained on multi-task signals across search, ads, and recommendation domains. LIGER [20] from Meta generalizes this framework to multilingual and multimodal retrieval, showing that a unified generative backbone can learn retrieval across heterogeneous sources including text, image, and behavioral modalities. These works establish encoder–decoder modeling as a universal interface for large-scale retrieval and cross-domain transfer.

Building on this foundation, OneRec [16] from Kuaishou exemplifies the transition toward fully end-to-end encoder–decoder recommendation. It integrates retrieval, ranking, and iterative preference alignment within a single generative backbone, fine-tuned via multi-stage preference optimization. The same architecture generalizes across multiple business verticals, including OneLoc [42] for local-life services, OneSearch [27] for e-commerce search, and OneSug [43] for query suggestion, collectively demonstrating the scalability and reusability of encoder–decoder foundations. Similarly, REG4Rec [35] and GenSAR [15] from Alibaba and Kuaishou employ instruction-style prompts to support cross-domain retrieval.

Encoder–decoder architectures naturally support multimodal integration, making them ideal for large foundation-scale models. QARM [28] from Kuaishou aligns visual, textual, and behavioral features through quantitative token alignment, bridging heterogeneous modalities within a shared latent space. PinFM [45] from Pinterest scales this principle to billion-parameter magnitude, jointly learning multimodal embeddings and sequential dependencies for visual discovery and ads recommendation. These works demonstrate that encoder–decoder frameworks can serve as unified backbones for cross-modal personalization at industrial scale.

Recent systems extend encoder–decoder modeling to support multi-objective learning and real-time efficiency. Pantheon [33] from Alibaba formulates generative ranking as Pareto-efficient policy optimization, balancing relevance, engagement, and monetization objectives within a single encoder–decoder architecture. EGA-V1 [3] and EGA-V2 [23] from Meituan unify recommendation and online advertising

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

in an end-to-end generative interface, where the encoder models user–ad context and the decoder jointly produces ad candidates and bidding outcomes. Both adopt hybrid autoregressive–non-autoregressive decoding to achieve real-time latency without sacrificing generation fidelity.

The most recent advance, PLUM [46] introduces a framework for adapting large pre-trained language models (LLMs) to industrial-scale generative recommendation. It replaces traditional ID embeddings with Semantic IDs (SID-v2) that discretize multimodal item features, applies continued pretraining (CPT) to align LLMs with user–item behavioral data, and fine-tunes the model for generative retrieval, directly generating item SIDs conditioned on user context. Deployed at YouTube scale, PLUM demonstrates superior retrieval performance, scaling efficiency, and sample efficiency compared to embedding-based production models, marking a major step toward LLM-based foundation recommenders.

Collectively, these systems illustrate the maturity of encoder–decoder modeling in industrial recommendation. They offer strong contextual understanding, flexible conditioning, and cross-domain generalization, making them well suited for complex, multi-objective tasks. However, their bidirectional encoding and cross-attention decoding incur higher latency and computational costs, motivating the ongoing shift toward decoder-only and hybrid foundation models in large-scale production environments.

3.1.3. Decoder-Only Models

Decoder-only architectures represent the mainstream paradigms of industrial generative recommendation systems. Inspired by autoregressive large language models (LLMs) such as GPT, these models treat recommendation as a conditional generation task, where the system predicts the next item (or sequence of items) given a user's historical interactions. Built upon unidirectional Transformer decoders, they model the sequential dependency among user behaviors and generate items token by token.

The fundamental idea of decoder-only modeling is to learn the sequential dynamics of user behavior via next-token prediction. HSTU [22] from Meta pioneers this direction with a trillion-parameter sequential transducer that introduces hierarchical temporal units for efficient long-range dependency modeling. MTGR [24] from Meituan extends this autoregressive paradigm to industrial-scale generative retrieval using SID tokenization, enabling the model to directly generate semantic item identifiers from user sequences. Its enhanced variant, MTGRBoost [37], adopts pipeline and tensor parallelism for large-cluster training, achieving 1.6–2.4$\times$ throughput gains while preserving linear scalability.

As generative models scale to billion-item catalogs, several works introduce architectural refinements to balance efficiency and precision. COBRA [21] from Baidu proposes a cascaded sparse–dense framework that first performs coarse-grained token generation over a sparse vocabulary and subsequently refines the results through dense re-ranking, significantly improving recall–precision trade-offs. PinRec [32] from Pinterest extends this idea by conditioning generation on downstream outcomes, such as engagement and conversion signals, thereby aligning token-level generation with long-term business metrics. TBGRecall [34] from Alibaba further demonstrates the scalability of SID-based generation in e-commerce recall, incorporating incremental vocabulary maintenance and parallel decoding for high-throughput online serving.

Beyond retrieval, decoder-only models also cast *ranking* as conditional sequence generation to capture inter-item dependencies. GenRank [38] generates list orderings conditioned on user context, employing lightweight ranking heads, hybrid sparse–dense representations, and parallel decoding for low latency at scale. SORT-Gen [38] formulates listwise sorting as generative permutation modeling, leveraging an autoregressive decoder to produce exposure-aware item orders under business or policy constraints. GenCTR [39] treats click-through estimation as a generative task, using the decoder to condi-

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

tion CTR prediction on previously placed items and context tokens, thereby modeling mutual influences within the slate. DFGR [40] introduces a dual-flow transformer that decouples user and item streams and then reconnects them through cross-flow attention, improving robustness and disentanglement for generative ranking.

Recent efforts explore more cognitively inspired generative processes to improve both reasoning depth and inference efficiency. CAR [44] generalizes token-level autoregression to chunk-level generation, unifying semantic understanding and behavioral prediction within a single Transformer. Inspired by the "slow thinking" mechanism in large language models (Test-Time Scaling), CAR introduces intermediate reasoning steps during generation, enhancing interpretability and reducing latency by 30–50% compared to standard autoregressive decoding. At Kuaishou, OneRec-V2 [41] and OneRec-Think [47] simplify this design via a "lazy decoder-only" approach that removes explicit encoders and scales to billions of parameters with 10× fewer FLOPs. OneRec-Think integrates in-text reasoning into decoding, enabling explainable and adaptive decision-making.

Compared with encoder–decoder frameworks, decoder-only architectures offer superior throughput and are well suited to production-grade serving due to KV-caching, parallelized decoding, and compatibility with mixture-of-experts (MoE) scaling. Their unidirectional dependency, however, can limit bidirectional context understanding, particularly in multimodal or multi-turn scenarios.

3.2. Functional Scope

The second dimension concerns what the generative model outputs and which stage of the recommendation pipeline it replaces or unifies. We identify four major functional categories: (a) Generative Retrieval (b) Generative Ranking (c) Unified End-to-End Recommendation (d) Generative Augmentation and Auxiliary Tasks.

3.2.1. Generative Retrieval

Generative retrieval reframes candidate generation as a text-to-item generation problem, where the model directly produces item identifiers (e.g., semantic IDs or tokenized representations) given a user's recent interactions or query context. Instead of searching a dense embedding index via nearest-neighbor lookups, the model autoregressively generates a sequence of item tokens that correspond to relevant candidates. This paradigm eliminates the need for complex vector indexing structures and enables flexible retrieval across textual, visual, and behavioral modalities. Leveraging large-scale Transformer architectures, generative retrieval models capture long-range dependencies and compositional preferences in user behavior, offering a unified and fully differentiable retrieval process.

Early work such as TIGER [11] pioneered this paradigm by introducing semantic item tokenization and demonstrating that generative retrieval can rival dense vector retrieval when trained on multi-task behavioral signals. LIGER [20] extended this framework to multilingual and multimodal retrieval, unifying text, image, and behavioral domains under a single encoder–decoder backbone. PinRec [32] further introduced multi-token outcome-conditioned decoding, improving result diversity and controllability in large-scale visual discovery.

Recent industrial deployments have scaled this approach to trillion-parameter regimes and multi-domain scenarios. HSTU [22] demonstrated that large sequential transducers could serve as unified autoregressive recommenders for both retrieval and ranking. GenSAR [15] integrated balanced search and recommendation within a generative retrieval framework, while REG4Rec [35] introduced reasoning-enhanced decoding for improved interpretability and cross-domain generalization. TBGRecall [34] and CAR [44] advanced chunk-level and reasoning-augmented autoregressive modeling to improve efficiency

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

and stability in long sequences. Meanwhile, GRAM [12] and COBRA [21]) explored hybrid sparse–dense decoding and multi-granular semantic fusion, combining generative and retrieval-based priors for scalability. Finally, PinFM [45] established a billion-scale foundation model for user activity sequences, unifying retrieval across multimodal interaction spaces.

Overall, generative retrieval offers a conceptually elegant and architecturally unified solution to large-scale recommendation: it consolidates candidate generation and early ranking into a single sequence model and simplifies training via end-to-end objectives. However, practical challenges remain, including token instability due to evolving vocabularies, efficiency constraints for real-time decoding, and coverage limitations for tail or unseen items. Ongoing research addresses these issues through hierarchical token quantization, dynamic vocabulary adaptation, and hybrid generative–dense retrieval frameworks.

3.2.2. Generative Reranking

Generative reranking redefines the ranking stage as a sequence generation problem, where the model outputs an ordered list of items or rank tokens conditioned on user, query, and contextual signals. Unlike traditional pointwise or pairwise scoring approaches that independently assign relevance scores to fixed candidates, generative reranking models directly produce a coherent ranked sequence. This formulation allows the model to capture inter-item dependencies and optimize listwise objectives in an end-to-end manner, aligning generation order with overall engagement or revenue goals.

Early explorations in this direction—such as HSTU [22], HLLM [30], and LEARN [29], leveraged large-scale transformer architectures and hierarchical LLM adaptation to enhance user and item representation learning. These works demonstrated that integrating generative reasoning into the ranking stage improves context modeling and sequential consistency compared to independent scoring mechanisms.

Recent industrial frameworks have scaled these ideas to production-level reranking and advertising systems. MTGR [24] proposed a dual-flow Transformer that disentangles user and item information streams for efficient autoregressive decoding. Similarly, DFGR [40] introduced a dual-flow generative ranking network, showing that autoregressive decoding can implicitly optimize both relevance and downstream business metrics.

At Alibaba and JD, SORT-Gen [38] and GenCTR [39] applied sequence-to-sequence generation to listwise ranking and click-through rate prediction, respectively, enabling multi-objective optimization across engagement, conversion, and advertising efficiency. Pantheon [33] extended this idea to multi-objective ensemble sorting using Pareto-efficient policy optimization, balancing user satisfaction, revenue, and diversity. In parallel, NAR4Rec [13] explored a non-autoregressive generative alternative to accelerate reranking without beam search, while GenBank [36] and other large-scale systems have focused on optimizing generative ranking architectures for industrial scalability.

Overall, generative reranking advances the goal of listwise, globally optimized recommendation by treating the ranking process as structured sequence generation rather than independent scoring. These models provide stronger modeling of inter-item relations and contextual coherence, bridging retrieval and ranking under a unified generation framework. However, practical deployment faces challenges such as inference latency, beam-search complexity, and maintaining diversity and fairness in generated lists, spurring ongoing research into efficient decoding, non-autoregressive alternatives, and hybrid generation–ranking pipelines.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

3.2.3. Unified End-to-End Recommendation

Unified end-to-end recommendation aim to consolidate the traditionally disjointed stages of retrieval, ranking, and post-ranking into a single generative backbone. Instead of training multiple specialized models, these systems formulate all subtasks, such as candidate generation, ordering, and re-scoring—as generative subtasks within a shared token space. Through multi-task pretraining and iterative refinement, they achieve parameter efficiency, cross-task knowledge transfer, and consistent optimization across the entire recommendation pipeline.

A major milestone in this direction is OneRec [16] (Kuaishou, Feb 2025), which introduced a unified encoder–decoder architecture with Iterative Preference Alignment (IPA), an RL-based mechanism that refines generated recommendations using user feedback and reward models. Its successor, OneRec-V2 [41] (Kuaishou, Jun 2025), transitioned to a fully autoregressive decoder and improved multi-domain alignment, enabling joint optimization for recommendation, search, and advertising tasks. Similarly, EGA-V1/V2 [3,23] (Meituan, 2025) unified candidate retrieval, ranking, and auction modeling into a single generative interface for online advertising, achieving token-level bidding and real-time end-to-end inference. LUM [31] (Alibaba, Feb 2025) extended this paradigm to study scaling laws in large user models, revealing predictable power-law improvements in performance as model and data size increase.

Other frameworks have advanced domain adaptation and specialization under this unified generative paradigm. LC Rec [25] adapted large language models for recommendation by integrating collaborative semantics through fine-tuned generative decoding. QARM [28] aligned multimodal features via quantitative token alignment, bridging textual, visual, and behavioral signals in a unified token space. MTGRBoost [37] built on the MTGR family to boost generative training efficiency through adaptive data sampling and multi-stage distillation. Extending the unified framework to new domains, OneSearch [27] applied generative modeling to e-commerce search, while OneLoc [42] introduced a geo-aware generative recommender for local-life services.

Overall, unified end-to-end recommendation represent a natural evolution of generative recommendation, merging retrieval, ranking, and decision optimization into a single, multi-task generative system. These models exhibit strong parameter sharing, multi-domain transferability, and improved consistency across objectives. However, they also introduce challenges in training stability, decoding latency, and balancing heterogeneous objectives, motivating ongoing research into scalable multi-objective optimization and efficient generation for industrial deployment.

3.2.4. Generative Augmentation and Auxiliary Tasks

Beyond core retrieval and ranking, generative modeling has increasingly been applied to auxiliary tasks that enhance the contextual understanding of users, items, and queries within recommendation ecosystems. These auxiliary generative modules operate alongside or upstream of the main recommendation backbone, enriching input representations, providing complementary training signals, and facilitating more adaptive user interaction flows.

Representative works include OneSug [43], which extends the OneRec family to query suggestion. It employs an encoder–decoder architecture to generate natural-language reformulations and intent expansions, effectively bridging search and recommendation behaviors. Similarly, OneSearch [27] and OneLoc [42] generalize this generative paradigm to search personalization and location-based recommendations, respectively, demonstrating that the same token-based generative modeling principles can be applied across vertical domains.

Other auxiliary generative tasks include user profiling, review summarization, and data augmentation for long-tail items, where large language or multimodal models synthesize missing information

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

or enhance semantic alignment between user intent and content representation. These approaches use the generative model not as a replacement for traditional ranking, but as a complementary knowledge source that strengthens representation learning and personalization.

Overall, generative augmentation and auxiliary tasks broaden the role of generative recommendation beyond direct item prediction. By enriching contextual understanding and enabling cross-domain transfer, these models contribute to a more unified and semantically aware ecosystem. Future research in this area focuses on tight integration between auxiliary generation and core retrieval/ranking pipelines, efficient multi-task training, and ensuring factuality and consistency in generated signals.

3.3. Representation Space

The third dimension focuses on how users, items, and contexts are represented within the generative process.

3.3.1. Semantic ID and Codebook Learning

The concept of the *Semantic ID (SID)* serves as a cornerstone of modern generative recommender systems. It replaces raw, arbitrary item identifiers with *learned discrete codes* that capture semantic similarity among items, enabling models to operate in a symbolic token space akin to natural language modeling. This design bridges the gap between high-cardinality categorical features and token-based sequence generation, allowing recommendation systems to leverage the same modeling principles as large language models (LLMs).

This idea is inspired by discrete latent representation learning in vision and language modeling, most notably, the *Vector Quantized Variational Autoencoder* (VQ-VAE) [48] and its successors such as *VQ-GAN* [49] and *AdaCode* [50]. In recommender systems, these codebook-based representations act as *semantic compressors* that map continuous item embeddings into compact symbolic tokens, preserving relational structure while enabling generative modeling of item sequences.

Hierarchical and Residual Quantization

To improve representational granularity without exponentially increasing vocabulary size, modern generative recommenders adopt hierarchical or residual quantization techniques. *Residual Quantization (RQ-VAE)* [51] and *RQ-KMeans* [16] recursively quantize residuals from prior levels, forming a multi-level code sequence $c = (c_1, c_2, \ldots, c_L)$. This design substantially expands the effective code space while keeping each sub-codebook small, allowing fine-grained reconstruction of semantic signals. In parallel, *Product Quantization (PQ)* [52] and its optimized variant *RQ-Optimized Product Quantization (RQ-OPQ)* [27] partition the embedding space into orthogonal subspaces and learn an adaptive rotation to align quantization axes with the data's variance structure. This rotation reduces quantization distortion and improves codebook utilization.

These principles have been widely adopted in large-scale generative recommenders. TIGER [11] (Google) first introduced SIDs to represent items as discrete tokens for generative retrieval. Building on this idea, LIGER [20] (Meta) unified dense and generative retrieval across multilingual and multimodal domains, while COBRA [21] (Baidu) employed a cascaded sparse–dense quantization scheme for scalable retrieval. GenSAR [15] (Kuaishou) incorporated SID-based quantization into a unified retrieval–recommendation framework, integrating symbolic and behavioral signals. Beyond retrieval, QARM [28] extended SID learning to multimodal alignment through quantitative token mapping across visual, textual, and behavioral modalities, while LC Rec [25] integrated collaborative adapters into frozen LLM backbones to inject SID-informed user–item semantics.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

Codebook-based Discretization

A codebook $\mathcal{E} = \{e_k\}_{k=1}^{K}$ defines a finite vocabulary of embedding vectors (codewords). During quantization, a continuous representation z_e is mapped to its nearest codeword:

$$e = \arg\min_{e_k \in \mathcal{E}} \|z_e - e_k\|_2^2, \tag{5}$$

producing a discrete SID token that can be modeled by decoder-only or encoder–decoder architectures as if it were a word token. This tokenization paradigm enables a unified modeling interface across modalities and tasks, allowing recommendation, search, and advertising to share pretrained generative backbones.

Systems such as OneRec [16] and OneRec-V2 [41] (Kuaishou) adopt SID-based vocabularies as fundamental representations for unified end-to-end recommendation, while OneSearch [27], OneLoc [42], and OneSug [43] extend this design to domain-specific applications such as search, local-life recommendation, and query suggestion. CAR (Act-With-Think) [44] further demonstrates the adaptability of SID-based representations to chunk-level autoregressive modeling, showing that discrete semantic tokens can effectively capture both behavioral and semantic regularities during inference.

Training Stability and Codebook Utilization

Despite their advantages, SIDs face practical challenges in training—most notably *codebook underutilization*, where only a small fraction of codes are activated. Coverage is typically quantified as:

$$\text{Coverage} = \frac{|\{i : n_i > 0\}|}{K}, \tag{6}$$

where n_i denotes the frequency of codeword i. Empirical findings from OneRec [16,41], GRID [53], and GenSAR [15] indicate that random initialization often yields poor coverage (< 0.3), while Gaussian or K-means initialization achieves robust utilization (> 0.9). Stabilization strategies include (1) initializing codebooks with the first batch via K-means, (2) progressively scaling embedding dimensions (e.g., from 32 to 128) and codebook sizes (e.g., 8K to 32K), and (3) adaptively balancing reconstruction and commitment losses during training.

As reported by OneRec-V2 [41], enlarging codebook size consistently improves retrieval and ranking accuracy with limited computational overhead. Similar trends are observed in TIGER and LIGER, where hierarchical quantization scales vocabulary expressiveness while maintaining tractable inference.

In summary, Semantic ID and codebook learning form the foundation of modern generative recommenders by providing compact yet semantically rich discrete representations for large item corpora. Through hierarchical quantization, adaptive codebook optimization, and multimodal alignment, these approaches enable efficient symbolic modeling across retrieval, ranking, and end-to-end frameworks. Future directions include hybrid quantization (combining residual and product schemes), dynamic codebook expansion for new items, and tighter coupling between SID learning and LLM pretraining to further unify representation learning across vision, language, and recommendation domains.

3.3.2. Dense and Hybrid Representations

While Semantic IDs provide discrete, tokenized representations suitable for generative modeling, a large body of recent work demonstrates that incorporating *dense* or *hybrid* representations—combining both dense and discrete features—significantly enhances the expressiveness and robustness of generative recommender systems. Dense representations preserve continuous semantic and contextual nuances

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

that may be lost during quantization, whereas hybrid approaches leverage the complementary strengths of symbolic and continuous encodings.

Early large-scale generative recommenders such as HSTU [22] (Meta) and LEARN [29] (Kuaishou) maintain dense embeddings for users and items within sequential transducer architectures. HSTU introduces hierarchical temporal units to efficiently model long-range dependencies, while LEARN focuses on knowledge adaptation from LLMs through dense representation transfer and contrastive learning. HLLM [30] (ByteDance) and LUM [31] (Alibaba) extend this idea to hierarchical user modeling and large-scale scaling analysis, showing that dense representations are particularly effective in capturing fine-grained personalization and scaling behavior.

Moreover, recent industrial systems increasingly adopt hybrid architectures that integrate continuous embeddings with discrete token representations. GRAM [12], PinRec [32], and GenRank [36] exemplify this paradigm in visual and e-commerce recommendation, where dense semantic features are fused with discrete item tokens for improved generalization and interpretability. TBGRecall [34] (Taobao) and REG4Rec [35] employ dense–discrete fusion under multi-task learning objectives, jointly optimizing retrieval and ranking signals. Meanwhile, NAR4Rec [13] (Kuaishou) explores non-autoregressive dense decoding to accelerate inference while maintaining hybrid representational fidelity.

In the ranking domain, GenCTR [39] and SORT-Gen [38] formulate listwise ranking as conditional generation, fusing dense contextual embeddings with discrete ranking tokens. DFGR [40] employs a dual-flow transformer that separates user and item streams, allowing hybrid attention between dense features and discrete behavioral sequences. Pantheon [33] extends hybrid modeling to multi-objective optimization, balancing relevance, engagement, and monetization via Pareto-efficient ensemble learning.

In domain-specific applications such as local services and advertising, hybrid encodings are crucial for integrating diverse feature types. MTGR [24] and its enhanced variant MTGRBoost [37] leverage dense representations for geographic, temporal, and contextual signals while retaining discrete tokens for user–item semantics. EGA-V1 [3] and EGA-V2 [23] incorporate hybrid representations into end-to-end advertising frameworks, unifying retrieval, ranking, and auction prediction within a single generative interface. PinFM [45] generalizes hybrid modeling to multimodal foundation models, integrating visual, textual, and sequential representations at billion-scale to support diverse content discovery scenarios.

Overall, dense and hybrid representations play a vital role in bridging the precision of continuous embeddings with the generative flexibility of discrete tokens. They enable finer semantic understanding, richer context modeling, and greater adaptability across domains and modalities. As hybrid modeling becomes the dominant design in industrial generative recommendation, future directions include dynamic modality balancing, adaptive fusion mechanisms, and scaling hybrid representations under multi-domain pretraining regimes.

3.4. Training and Alignment Objectives

3.4.1. No Alignment (Next-Token Prediction)

Next-token prediction (NTP) serves as the foundational training paradigm for most generative recommendation systems. Under this objective, the model learns to maximize the likelihood of the next item token conditioned on the user's historical interaction sequence:

$$\mathcal{L}_{\text{NTP}} = -\mathbf{E}_{(x_1,\ldots,x_T)\sim\mathcal{D}} \sum_{t=1}^{T} \log P(x_t \mid x_{<t}),$$

where $x_{<t}$ denotes the user's previous items. This maximum-likelihood objective allows the model to capture sequential dependency and temporal dynamics, learning directly from behavioral data without

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

the need for explicit supervision. It has become the de facto pretraining stage in generative recommender frameworks, providing semantic fluency and strong item compositionality analogous to next-token modeling in language models.

In recent large-scale recommender systems, the majority of models, including TIGER [11], LIGER [20], GenSAR [15], HSTU [22], GPSD [**?**], and MTGR [24], adopt next-token prediction as their primary pretraining objective. These models treat recommendation as a sequential generation problem, where each next item is predicted given the entire behavioral context. For instance, TIGER [11] formulates semantic retrieval as sequence generation over discrete item tokens; LIGER [20] integrates dense retrieval supervision while maintaining a generative next-item loss; and HSTU [22] demonstrates that autoregressive training can outperform traditional discriminative ranking models under similar parameter budgets. Similarly, GPSD [**?**] and MTGR [24] report consistent improvements in precision and recall as model scale and training corpus grow, confirming that NTP-based pretraining captures transferable user–item semantics and scaling behavior analogous to NLP models.

However, maximizing log-likelihood over historical sequences does not directly optimize for downstream business metrics such as engagement rate, dwell time, or revenue. Pure NTP pretraining often produces semantically coherent but behaviorally suboptimal recommendations. Consequently, recent research, including OneRec [16], OneRec-V2 [41], and EGA-V2 [23], introduces reinforcement learning or preference optimization in post-training alignment stages to align model generations with true user preferences and platform objectives, as discussed in the next section.

3.4.2. Alignment via Reinforcement Learning and Preference Optimization

Autoregressive pretraining through next-token prediction endows generative recommendation models with strong semantic fluency and compositional capabilities. However, maximizing likelihood on historical interaction sequences does not necessarily optimize for real user engagement or business objectives. To bridge this gap, modern generative recommenders increasingly adopt a post-training alignment stage that steers the model toward outputs yielding higher downstream rewards. This section surveys reinforcement learning (RL) and preference optimization techniques adapted to the generative recommendation setting.

We formulate the generative recommender as a policy

$$\pi_\theta(\mathbf{o} \mid u) \tag{7}$$

that, conditioned on user context u, produces a sequence of item tokens or semantic ID tokens $\mathbf{o} = (o_1, o_2, \ldots, o_T)$. The alignment objective is to maximize expected reward:

$$\max_\theta \ \mathbf{E}_{u\sim\mathcal{U},\,\mathbf{o}\sim\pi_\theta(\cdot|u)}\big[r(u,\mathbf{o})\big], \tag{8}$$

where $r(u,\mathbf{o})$ is a reward function reflecting engagement metrics (clicks, dwell time, conversions) or business KPIs (revenue, user satisfaction, diversity). Unlike supervised pretraining, which optimizes token-level likelihood, alignment directly targets the end-to-end utility of generated recommendations, enabling the model to learn complex trade-offs between relevance, diversity, and business objectives.

Applying RL or preference optimization to generative recommenders presents several domain-specific challenges. **Sparse and noisy rewards**: User feedback (clicks, purchases) is often sparse and delayed, with implicit signals being noisy and subject to selection biases. **Combinatorial action space**: The model generates sequences or slates of items, creating an exponentially large action space where credit assignment, determining which items in a sequence contributed to the reward becomes nontrivial.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

Exposure and position bias: Logged data reflects the behavior policy's biases (e.g., popular items are over-represented, position effects confound true preferences), which naive RL can amplify. **Off-policy learning**: Training data comes from a behavior policy π_β different from the current policy π_θ, and direct optimization without correction can lead to distribution shift and poor generalization. **Invalid token sequences**: In semantic ID systems, the model may generate token sequences that do not decode to valid items, requiring alignment to detect and penalize such outputs.

Examples from Generative Recommender Systems

OneRec [16] is an end-to-end generative recommendation system deployed at Kuaishou that adopts session-wise generation with a Mixture-of-Experts (MoE) architecture. To align recommendations with user preferences, OneRec incorporates Direct Preference Optimization (DPO) [54] enhanced with Iterative Preference Alignment (IPA) [55]: it trains a reward model on engagement signals (watch time, likes), generates multiple candidate sequences via beam search, and constructs preference pairs (o_w, o_l) where high-reward sequences o_w are preferred over low-reward sequences o_l. The process iterates, with the updated policy generating progressively harder preference pairs as the model improves. This addresses the challenge of obtaining informative preference data at scale and mitigates distribution shift. With only 1% DPO training ratio, OneRec+IPA achieves around 5% improvements in session watch time and like-through rate over the base model. OneRec-V2 [41] advances this framework by introducing Gradient-Bounded Policy Optimization (GBPO), a PPO variant that incorporates gradient clipping and adaptive KL penalties to prevent reward hacking and maintain multi-objective balance. It employs duration-aware reward shaping that explicitly accounts for watch time, likes, comments, and follows with careful normalization.

Practical Guidelines

Based on recent literature and industrial deployments, we distill the following best practices. **Reward design**: Construct reward models that combine multiple signals (clicks, conversions, dwell time, diversity) rather than relying on single metrics; model inter-item interactions and sequence-level effects (e.g., slate diversity, complementarity); validate reward models offline against held-out engagement data before RL deployment. **Training strategies**: (i) *Sample and rerank*, generate multiple candidate sequences (via sampling or beam search), score them with the reward model, and use the top-k vs. bottom-k as preference pairs for DPO; (ii) *Curriculum learning*, gradually increase the weight λ of the RL loss or the difficulty of preference pairs to stabilize training; (iii) *KL regularization*, maintain a KL penalty or reference policy constraint to prevent distribution shift and preserve generation fluency; (iv) *Hybrid objectives*, retain supervised likelihood loss throughout alignment to avoid degenerate outputs.

4. Scaling Law in Generative Recommendation Systems

Scaling laws, which is the observation that model performance improves predictably with increased model size, data volume, and compute, were first formalized in large-scale language modeling [56]. These laws typically describe smooth, power-law relationships between resource investment and performance gains, enabling researchers to forecast improvements as systems scale. While scaling laws have become foundational in NLP, where architectures have largely converged on the Transformer, their application to recommendation systems (RecSys) remains underexplored due to several fundamental challenges:

- **Architectural Heterogeneity**: Recommendation models exhibit far greater structural diversity (e.g., Wide & Deep, DIN, graph neural networks) than the Transformer-dominated NLP landscape, making it unclear how to systematically scale capacity across different paradigms.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

- **Extreme Data Scale**: User–item interaction logs span hundreds of millions of users and billions of items, yielding token counts up to 10^{14}—two orders of magnitude larger than typical LLM training corpora—which complicates both storage and computation.
- **Strict Latency Requirements**: Online recommendation services demand end-to-end response times under ∼30 ms, imposing severe constraints on model complexity and making deployment of large-scale models significantly more challenging than in NLP applications.

Despite these obstacles, recent work has begun to reveal systematic scaling behaviors in RecSys, emerging along two complementary lines: traditional discriminative architectures and newer generative frameworks.

4.1. Scaling Traditional Discriminative Models

Wukong [26] provides early empirical evidence that classical discriminative recommendation architectures can exhibit scaling-law–like behavior under certain conditions. Through the introduction of the Factorization Machine Block (FMB) for modeling high-order feature interactions and the Linear Compression Block (LCB) for reducing embedding dimensionality, Wukong enables deep residual stacking of interaction layers—a capability historically unstable in DLRM-style models due to gradient explosion and feature sparsity. With the addition of LayerNorm-based stabilization, the framework supports substantially deeper and wider networks than prior DLRM variants. Offline experiments show that both log-loss and MRR decrease predictably as model depth and embedding size grow, revealing an approximate power-law relationship between compute and performance. However, these traditional architectures face inherent scalability constraints: their reliance on extensive feature engineering and behavior-sequence sampling to manage large-scale interactions limits both the upper performance bound and training efficiency as resources scale.

GPSD [**?**] addresses some of these scalability bottlenecks by incorporating generative pretraining to regularize discriminative models. Instead of scaling purely supervised architectures, GPSD pretrains a Transformer on large-scale user behavior sequences in a generative manner, then fine-tunes it for discriminative downstream tasks (e.g., CTR prediction). By doing so, GPSD mitigates overfitting and provides stronger initialization for large discriminative backbones. The model is scaled from 13K to 327M dense parameters, and exhibits consistent power-law gains—a rare property among discriminative RecSys models. This suggests that injecting generative priors into discriminative systems can partially unlock scaling behaviors otherwise inaccessible to classical DLRMs.

4.2. Scaling Generative Recommendation Models

As generative recommendation frameworks mature, a central question has emerged: do these models exhibit scaling behaviors analogous to those observed in large language models (LLMs)? Recent works increasingly suggest that they do. Collectively, they reveal that generative recommenders not only benefit from increased model capacity, but also demand architectural, semantic, and computational innovations that enable scaling under real-world constraints.

HSTU [22] establishes that generative architectures can achieve superior scaling compared to traditional discriminative models. By processing complete user histories directly through Transformer-based sequence modeling, rather than relying on feature engineering and sampling, HSTU demonstrates higher performance ceilings and better training efficiency as resources increase, providing foundational evidence for scaling laws in generative recommendation.

Subsequent research systematically examines how generative recommenders behave as model capacity increases. OneRec-V2 [41] represents a significant step toward large-scale generative training in

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

practice. Its "Lazy Decoder-Only" architecture reduces computation by 94% and training cost by 90%, making it feasible to train models up to 8B parameters under realistic compute budgets. Experiments across 0.1B–8B models reveal steady convergence-loss improvements (from ∼3.57 to ∼3.19), albeit with diminishing returns beyond ∼2B parameters. This line of work underscores that scaling in generative recommendation is not solely about increasing parameters, but about co-optimizing model size and computational efficiency.

Complementary studies—including EGA-V1 [3], MTGR [24], DFGR [40], and TBGRecall [34]—further characterize the performance–compute landscape by varying model depth, FLOPs, and architectural configurations. These works consistently show that accuracy improves approximately linearly with the logarithm of parameter count, while computational cost grows exponentially. Their findings mirror classical LLM scaling laws and provide empirical guidelines for designing scalable generative recommenders under resource constraints.

While these models evaluate scaling in controlled settings, LUM [31] addresses the more challenging question of whether LLM-style scaling laws can manifest in industrial environments, where strict training-time and latency budgets typically suppress scaling benefits. LUM introduces a three-step generative-to-discriminative paradigm that decouples massive generative pretraining from real-time serving. This decomposition allows the generative component to scale to billions of parameters offline, while preserving DLRM-level inference cost online. Empirically, LUM exhibits clear power-law scaling: retrieval performance grows linearly with log-model-size (19M → 7B) and with log-sequence-length (256 → 8192). By demonstrating that architectural decomposition can unlock scaling otherwise infeasible in production, LUM bridges the gap between theoretical scaling laws and deployable systems.

In parallel, PinFM [45] examines scaling from a systems perspective, focusing on how to deploy extremely large user-sequence models (20B+ parameters) in production. Although not framed explicitly as a scaling-law study, PinFM shows that larger model capacity, larger embedding vocabularies (20M → 160M), and longer pretraining all yield consistent performance gains. To support this scaling, the authors introduce the Deduplicated Cross-Attention Transformer (DCAT), along with 4–8× embedding quantization and aggressive caching strategies. This demonstrates that infrastructure and system-level optimizations are essential enablers of scaling in industrial generative recommenders.

CAR [44] broadens the scaling discussion by shifting attention from model size to semantic resolution. Instead of increasing parameters, CAR scales the bit-width of Semantic IDs (SIDs) from 1 to 4 bits using hierarchical residual K-means, thereby enriching the semantic granularity of item representations. This yields monotonic improvements in Recall@5—25.8% on Beauty and 30.5% on Toys—as semantic precision increases. The authors relate this trend to slow-thinking mechanisms in LLMs, arguing that deeper semantic decomposition provides more refined intermediate reasoning steps. CAR thus reveals that scaling in generative recommendation can emerge not only from larger models, but also from richer tokenizations that enhance the model's cognitive depth.

A recent addition to this landscape is PLUM [46], which provides one of the most comprehensive scaling-law analyses for industrial generative retrieval to date. Using four Mixture-of-Experts models (110M → 3B activated parameters) trained on YouTube's production traffic, PLUM demonstrates clear power-law relationships between compute (Iso-FLOPs) and both training loss and evaluation loss. The performance frontier consistently shifts toward larger models as compute increases, and evaluation loss improves sooner than training loss—indicating stronger generalization for larger models. Retrieval accuracy (Recall@10) follows the same trend with limited saturation, even after multiple epochs for smaller models, showing no overfitting. Moreover, PLUM highlights an important constraint of industrial scaling: compute-optimal performance requires jointly scaling model size, data exposure, and batch size.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

Larger models can underperform when constrained by hardware-limited batch sizes, as seen with the 3B MoE model processing fewer than 0.6 epochs. PLUM therefore reinforces that scaling laws do hold for industrial generative recommenders, but only when compute, batch size, and data availability co-scale appropriately.

Together, these works paint a coherent picture: generative recommendation models exhibit scaling behaviors that strongly parallel those of LLMs, but require innovations across architecture, semantics, and systems to unlock these gains in practice.

4.3. Synthesis and Open Challenges

Collectively, these studies establish that recommendation systems do exhibit predictable scaling behaviors, though with important distinctions from language modeling. Successful industrial-scale deployment appears to require advances along multiple dimensions: (i) *architectural capacity* through deeper networks, mixture-of-experts, and efficient sequence modeling; (ii) *system efficiency* via training infrastructure, inference optimization, and serving-latency management; and (iii) *hybrid designs* that preserve domain-specific feature engineering while adopting generative modeling paradigms. The heterogeneous nature of user-item graphs, multi-modal features, and strict latency constraints fundamentally shape the scaling regime, suggesting that RecSys scaling laws will continue to diverge from pure language-modeling formulations. Key open questions include identifying optimal scaling strategies across different recommendation tasks, understanding the interplay between embedding-table scaling and neural-network scaling, and developing principled methods for trading off model capacity against inference latency in production environments.

Table 3. Deployment , Dataset, and Domain Characteristics. Percentage means how much online exposure was used for the developed model.

Paper	Online A/B Test	Offline Data	Scaling Law	Domain
TIGER [11]	N	Amazon reviews	N	Web-scale search
LC Rec [25]	N	Amazon reviews	N	social media
HSTU [22]	Y	Amazon reviews & MovieLens	Y	Social media
QARM [28]	Y	Kuaishou internal data	N	Short video / multimodal
GenSAR [15]	N	Amazon reviews & Chinese app	N	Short video / social feed
LEARN [29]	Y	Amazon reviews & Kuaishou internal data	N	Short video / social feed
HLLM [30]	Y	PixelRec & Amazon reviews	N	Short video / social feed
LUM [31]	Y	MovieLens & Amazon Booka	Y	E-commerce / search
LIGER [20]	N	Amazon & Steam [57]	N	Social media / cross-domain
GRAM [12]	Y (5%)	JD internal data	N	E-commerce
COBRA [21]	Y (10%)	Amazon	N	Web-scale search
PinRec [32]	Y	Pinterest internal data	N	Visual content discovery
Pantheon [33]	Y	Kuaishou internal data	N	Multi-objective ranking
TBGRecall [34]	Y (5%)	RecFlow & Taobao internal data	Y	E-commerce
REG4Rec [35]	Y	Amazon & Internal	N	E-commerce
NAR4Rec [13]	Y	Avito & Internal	N	Short video
GenRank [36]	Y (10%)	Xiaohongshu	N	Visual content discovery
MTGR [24]	Y (2%)	Meituan internal data	Y	Food delivery, local services
MTGRBoost [37]	N	Meituan internal data	N	Food delivery, local services
SORT-Gen [38]	Y	Taobao internal data	N	E-commerce
GenCTR [39]	Y	JD internal data	N	E-commerce advertising
DFGR [40]	N	RecFlow & KuaiSAR & Trec	Y	Food delivery, local services
OneRec [16]	Y (1%)	Kuaishou internal data	Y	Short video
OneRec-V2 [41]	Y (5%)	Kuaishou internal data	Y	Short video
EGA-V1 [3]	Y	Meituan internal data	Y	Food delivery, local services
EGA-V2 [23]	N	Meituan internal data	N	Food delivery, local services
OneLoc [42]	Y (10%)	Kuaishou internal data	Y	Local-life services
OneSearch [27]	Y	Kuaishou internal data	N	Search / e-commerce
OneSug [43]	Y	Kuaishou internal data	N	Query suggestion
CAR [44]	Y (5%)	Amazon reviews	Y	E-commerce / app recommendation
PinFM [45]	Y	Pinterest internal data	Y	Visual content discovery
PLUM [46]	Y	YouTube internal data	Y	Video recommendation

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

5. Scalability

5.1. Model-Architecture Scalability

Generative recommendation systems must scale their backbone architectures to support vast user histories, large item vocabularies, and long sequence modelling, all while maintaining inference and memory efficiency. Key concerns include handling very large parameter counts; accommodating long sequences of user actions; managing high-cardinality, evolving item/user spaces; and meeting serving constraints on latency and throughput.

For example, the HSTU [22] model uses a hierarchical sequential-transduction architecture with an extremely large parameter footprint, and an attention mechanism re-designed to keep computation roughly linear in sequence length. The OneRec-V2 [41] models simplify the OneRec [16] architecture by removing the encoder and using a decoder-only generative pipeline; this cut computation significantly while allowing increased model size and capacity.

5.2. System-Level Scalability

System-level scalability covers how the training and serving infrastructure supports large-scale generative recommenders: distributed training, tokenisation strategies for huge item sets, and efficient deployment under latency/throughput constraints.

Distributed Training and Parallelism

The deployment of generative recommendation models in industrial settings presents significant challenges in training efficiency and scalability, which the MTGRBoost [37] system directly addresses. Existing frameworks like TorchRec [58] are limited by their use of static embedding tables, which cannot accommodate real-time changes in user and item IDs and lead to inefficient memory use, and by communication bottlenecks caused by duplicate embedding lookups across distributed devices.

To overcome these limitations, MTGRBoost introduces a suite of optimizations centered around a dynamic, hash-based embedding table. This core innovation decouples the storage of keys and values, allowing for efficient capacity expansion by primarily migrating the smaller key structure rather than the large embedding vectors. This is complemented by automated embedding table merging, which reduces memory fragmentation and lookup latency, and a two-stage ID deduplication process that minimizes redundant communication during distributed embedding fetches. For the dense Transformer component, MTGRBoost implements a dynamic sequence batching mechanism that balances the computational load across GPUs by grouping sequences based on token count rather than a fixed number of samples, thereby mitigating the synchronization delays caused by variable-length user sequences.

The system employs a hybrid parallel training strategy, using model parallelism for the massive sparse embeddings and data parallelism for the smaller dense models, all within a pipelined workflow that overlaps data I/O, communication, and computation. Further implementation optimizations, including flexible checkpoint resuming, mixed-precision training, and gradient accumulation, contribute to its robustness. Extensive evaluation on real-world data from Meituan demonstrates that MTGRBoost achieves substantial improvements, boosting training throughput by 1.6x to 2.4x over TorchRec while maintaining model accuracy and demonstrating strong, near-linear scalability across over 100 GPUs. This work establishes a practical and efficient framework for the large-scale training of generative models, highlighting the critical role of system-level design in enabling their industrial adoption.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

Low-Latency Inference and Efficient Serving

To serve at web scale (hundreds of millions to billions of queries daily), generative recommenders require optimized decoding (e.g., multi-token output, prefix caching, parallel decoding), quantized kernels (e.g., 8-bit, mixed precision), and efficient hardware use. Recent analysis of inference scaling laws further shows that retrieval quality in generative systems follows predictable power-law improvements with increased inference-time compute, e.g., larger beam sizes or decoding budgets yield consistently lower miss rates, highlighting inference computation as a complementary axis of scalability alongside model and data size [59].

Beyond system-level optimization, model-aware decoding acceleration has become crucial for practical deployment. [60] propose AtSpeed, a speculative decoding framework designed for LLM-based generative recommenders. Unlike traditional speculative decoding for text generation, which verifies a single token stream (N-to-1), recommendation tasks require generating top-K ranked items via beam search, resulting in a more stringent N-to-K verification. AtSpeed addresses this challenge through two complementary strategies: (*i*) **AtSpeed-S**, which enforces strict top-K sequence alignment by minimizing reverse KL divergence with density regularization, and (*ii*) **AtSpeed-R**, which introduces a relaxed sampling verification that accepts high-probability non-top-K sequences while maintaining output fidelity. A tree-based attention mechanism further eliminates redundant prefix computations during verification, improving GPU utilization. Experiments on LLaMA-7B–based LC Rec models demonstrate near-lossless acceleration with up to **2.5**$\times$ decoding speedup, underscoring speculative decoding as a promising direction for low-latency generative recommendation serving.

6. Evaluation

6.1. Generative Retrieval Models

In the retrieval scenario, the goal is to directly generate relevant item identifiers for a user context. We summarize the model online and offline performance below:

TIGER [11] employs RQ-VAE to quantize item embeddings into semantic IDs, which a transformer autoregressively decodes, achieving 17.3% higher recall@5 than SOTA models on Amazon datasets [61] while reducing memory usage by 20$\times$.

HLLM [30] shows strong improvements on large-scale benchmarks. On PixelRec (8M), HLLM-1B surpasses the best reproduced baselines with a +22.93% average gain On Amazon Books, it delivers substantial improvements, with HLLM-1B (+166.42%), and HLLM-7B (+169.58%) outperforming all ID-, text-, and hybrid-based baselines. Even with frozen item encoders, HLLM remains stronger than ID-based models, validating its scalable caching strategy. In online A/B testing, HLLM-1B achieves a +0.705% lift in ranking performance without adding inference latency.

QARM [28] enhances multimodal recommendation by aligning MLLM-derived item representations with real interaction distributions through VQ and RQ code embeddings.Online A/B tests show substantial business impact: in advertising, QARM increases revenue by +9.704%/+3.147% for cold-start and non–cold-start groups and by +9.555%/+1.950% in an additional scenario, while in shopping it boosts GMV by +2.296%/+1.568%.

COBRA [21] proposes a hybrid framework synergizing sparse semantic IDs with dense vectors through cascaded generation and BeamFusion. Evaluated on Amazon datasets and Baidu's industrial platform (200M+ users), COBRA achieved +18.3% Recall@5 versus TIGER offline and +3.6% online conversion.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

PinRec [32] introduces outcome-conditioned generation and windowed multi-token prediction for Pinterest's recommender systems, achieving +30% improved recall@10 vs. TIGER offline on Search, and online A/B tests showing +0.28% fulfilled sessions and +3.33% Homefeed clicks.

GRAM [12] replaces traditional vector retrieval with LLM-generated shared attribute codes from e-commerce ontologies, trained with Direct Preference Optimization. It achieves +8% Recall@100 offline, and when deployed at JD.com, delivered +1.27% CTR and +2.46% revenue.

DFGR [15] introduces a dual-flow architecture splitting user sequences into real and fake streams, eliminating manual feature engineering while reducing training overhead by 2× and inference costs by 4× versus HSTU. On public (RecFlow [62], KuaiSAR [63]) and industrial (TRec) datasets, DFGR outperformed HSTU by 0.31–1.2% AUC and DLRM baselines by 0.52% AUC.

LIGER [20] addresses cold-start limitations by integrating semantic IDs with item text representations, optimizing both dense and generative objectives. Evaluation on Amazon datasets showed LIGER consistently achieved first or second-best performance with strong cold-start results (Beauty: 0.1008, Sports: 0.0585, Toys: 0.1306 Recall@10) versus TIGER's near-zero cold-start performance.

TBGRecall [34] replaces token-level generation with Next Session Prediction, predicting session-level embeddings for ANN-based retrieval. On RecFlow and Taobao industrial datasets, it achieved +9.61 HR points (≈+137%) at HR@1000 with consistently over +100% relative gains across HR@K metrics versus HSTU. Online A/B testing on Taobao yielded +0.60% transactions and +2.16% transaction amount.

PinFM [45] is evaluated on Pinterest's two most critical ranking applications—Home Feed (HF) and Item2Item (I2I)—using HIT@3 as the primary metric. Offline, multiple PinFM variants show consistent improvements over the baseline Transformers despite baselines having <0.2% of PinFM's parameter size; the strongest variant (PinFM-GraphSAGE-LT) achieves the largest HIT@3 gains, while lighter variants trade accuracy for significantly improved serving efficiency. Online A/B tests demonstrate clear business impact: PinFM deployed in I2I yields positive lifts across engagement metrics without requiring cold-start remediation, while in HF, after applying cold-start techniques, PinFM achieves +10% Fresh Saves and improves overall HIT@3 despite the complexity of the feed.

CAR [44] is evaluated on three Amazon Review categories (Beauty, Sports, Toys) under standard sequential-recommendation protocols. Experiments show that our CAR significantly outperforms existing methods based on traditional AR, improving Recall@5 by 7.93% to 22.30%.

6.2. Generative (Re)ranking Models

In multi-stage systems, after an initial candidate list is retrieved, a generative model can serve as a (re)ranker, typically predicting Click Through Rate (CTR)/Conversion Rate (CVR). We have the model performances below:

LEARN [29] achieves substantial gains on the Amazon Review benchmark, outperforming RecFormer with +10.08% to +29.45% lift across different domains. In large-scale online A/B testing on a short-video e-commerce platform (400M+ DAU), LEARN improves UAUC by +0.84pp and WUAUC by +0.76pp, with especially strong gains for cold-start and long-tail segments (e.g., +1.56% revenue and +0.17pp AUC for cold-start users; +8.77% revenue and +0.29pp AUC for cold-start items). Over a 9-day deployment, LEARN drives steady CVR and revenue improvements, reaching 2–3% revenue lift, which is highly significant given the platform's multi-million-dollar daily scale.

Pantheon [33] demonstrates strong improvements in large-scale live-streaming recommendation at Kuaishou (400M+ users, billions of exposures). Offline, Pantheon outperforms the long-standing formulation-based ensemble sort across all major objectives—achieving an average +1.62% GAUC improvement. Online A/B tests across four scenarios show consistently positive gains: Pantheon

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

improves clicked-user rate by +1.010%, +1.671%, +1.039%, and +0.518%, alongside substantial increases in watch time, in-room watch time, gifts.

NAR4Rec [13], a non-autoregressive generative reranking model, addresses inefficiencies in autoregressive approaches through simultaneous sequence generation and unlikelihood training. Offline experiments on Avito [1] and Kuaishou datasets show AUC=0.7234 vs. 0.7109 with 5× faster training. Online A/B tests on Kuaishou (300M+ users) demonstrate +1.16% views, +2.45% complete views, and +1.71% likes.

Genrank [36] introduces action-oriented sequence organization and lightweight position/time bias modeling for Xiaohongshu Explore feed. Offline experiments on hundreds of billions of exposure logs show over +0.0020 AUC improvements with 25% reduction in P99 inference latency. Online A/B tests report +1.25% engagements and +0.63% reads.

HSTU [22] reformulates ranking as sequential transduction with pointwise aggregated attention and introduces M-FALCON for efficient inference. On public benchmarks, HSTU improves NDCG@10 by up to 30% and HitRate@10 by 22.8% on ML-20M, with 0.004–0.006 lower Normalized Entropy on industrial datasets. Online A/B testing achieved +12.4% engagement for ranking and +6.2% for retrieval, with 1.5–3× higher throughput despite 285× computational complexity versus DLRMs.

DFGR [40] introduces a dual-flow architecture splitting sequences into real and fake streams, reducing training overhead by 2× and inference costs by 4× versus HSTU. Offline evaluation shows ∼0.4% AUC improvement over HSTU and ∼0.5% over DLRM baselines.

GenCTR [39] combines generative pretraining with discriminative CTR prediction through parameter sharing and model inheritance. Offline evaluation shows AUC gains up to 1.87%, while online A/B tests achieve +1.32% CTR and +1.66% RPM in search advertising systems.

SORT-Gen [38] optimizes multiple objectives (CTR, CVR, Gross Merchandise Value(GMV)) using Sequential Ordered Regression Transformer with mask-driven fast generation. Online A/B testing on Taobao achieved +83% click, +82% conversion, and +107% GMV with 19ms latency.

MTGR [24] combines DLRM feature richness with generative scalability through user sample aggregation while retaining cross features. Evaluation on Meituan's industrial dataset shows 0.90–1.47% offline gains (AUC, Group AUC) and 1.02–1.90% online improvements (CVR, CTR), achieving 65× computational complexity with unchanged training costs and 12% reduced inference costs.

6.3. End-to-End Generative Recommendation

End-to-end generative models unify retrieval, ranking, and auction. The premise is to have one model directly output the complete recommendation list in one pass.

LC Rec [25] was evaluated on Amazon's review data across three domains, where LC-Rec delivers substantial and consistent lifts——achieving +16%–69% gains in HR@1, +16%–28% in HR@5, +7%–20% in HR@10, and +15%–39% in NDCG.

LUM [31] demonstrates strong gains across public and industrial datasets. On MovieLens-1M, MovieLens-20M, and Amazon Books, LUM achieves the best AUC among all baselines with absolute lifts of +0.0036–0.0149 AUC. In industrial evaluation, LUM surpasses the production online model with +0.0176 AUC, +0.0133 R@10, and +0.0134 R@50, outperforming both traditional DLRMs and E2E-GRs under identical feature settings. In online A/B testing on Taobao's sponsored search system, LUM yields +2.9% CTR and +1.2% RPM, confirming significant improvements in real-world engagement and monetization.

[1] https://www.kaggle.com/c/avito-context-ad-clicks/data

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

EGA-V1 [3] was evaluated on Meituan's industrial dataset, showing substantial gains over MCA with Recall@50 improving by 77.5%, eCTR by 10.7%, and Incentive Compatibility (IC) regret reduced by 75%. Online A/B tests demonstrate significant business impact, including +13.6% Revenue Per Mille (RPM) and +3.1% advertiser Return on Investment (ROI), with minimal latency overhead.

EGA-V2 [23] outperforms both traditional multi-stage cascading systems and generative recommendation baselines on large-scale Meituan data, improving RPM by 16.5%, CTR by up to 9.3%, and achieving the lowest IC regret (2.7%).

OneRec [16] achieves up to 1.78% and 3.36% improvements in session watch time and like-through rate over strong baselines such as TIGER on industrial-scale data. With only 1% DPO training ratio, OneRec+IPA surpasses the base OneRec by around 5% in session watch time and like-through rate. Deployed on Kuaishou's main scene, it achieves a 1.68% increase in total watch time and a 6.56% gain in average view duration.

OneRec-V2 [41] demonstrates that the lazy decoder achieves similar or better convergence loss compared to encoder–decoder and naive decoder-only models, but with 10× fewer FLOPs and over 90% reduction in training cost. When deployed on 400M+ DAUs across Kuaishou & Kuaishou Lite, it shows significant engagement gains of +0.467% and +0.741% App Stay Time, along with consistent improvements in likes, comments, and follows.

OneLoc [42] shows strong gains in Recall@5,@10,@20 and NDCG over baselines on a large in-house dataset with 60M users, 900K items, and 440M interactions. When deployed to serve ∼400 million daily users, it achieves improvements of ∼21.016% in GMV and ∼17.891% in order volume.

OneSearch [27] demonstrates superior performance in recall and ranking on large industrial datasets. In online A/B tests, it achieves +1.67% item-CTR, +2.40% in buyers, and +3.22% in order volume, while reducing operational expenditure by ∼75% and boosting model FLOPs utilization from ∼3.26% to ∼27.32%.

PLUM's [46] generative retrieval model delivers substantial gains over YouTube's highly-optimized Large Embedding Model (LEM), achieving markedly larger effective vocabulary coverage (up to 13×), higher CTR (up to 1.42×), and competitive or better engagement metrics. Live A/B tests further show that adding PLUM to production candidate pools yields consistent online engagement improvements over the production LEM+ system, increasing Engaged Users by +0.07% (LFV) and +0.28% (Shorts), Panel CTR by +0.76% and +4.96%, Views by +0.80% and +0.39%, and Satisfaction by +0.06% and +0.39%.

7. Challenges and Future Directions

Despite the remarkable progress in generative recommendation systems, significant challenges remain that limit their widespread adoption and effectiveness. These challenges span technical, computational, and methodological domains, while emerging research directions point toward promising solutions and new capabilities that could further transform the field.

7.1. Current Challenges in Generative Recommendation Systems

Current Technical Limitations and Challenges

The deployment of generative recommendation systems faces substantial computational efficiency bottlenecks, with embedding stages creating up to 3.2x performance slowdowns and memory latency issues leading to GPU underutilization [64]. These bottlenecks become particularly pronounced as model sizes scale to the thousands of billions of parameters demonstrated by systems like HSTU, creating fundamental constraints on real-time serving capabilities.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

Infrastructure limitations pose significant barriers to widespread adoption, with more than 60% of GenAI initiatives remaining stuck at the proof-of-concept stage due to high compute costs, integration challenges, and talent scarcity [**?**]. Energy constraints have emerged as a critical bottleneck, with AI GPUs often sitting idle due to energy infrastructure limitations rather than compute availability [**?**]. This structural imbalance between exponentially growing computational demands and linearly growing power infrastructure creates fundamental scaling constraints.

Evaluation and Benchmarking Challenges

The evaluation of generative recommendation systems requires new frameworks that go beyond traditional metrics like NDCG, recall, and precision. Recent research identifies four critical evaluation dimensions specific to LLM-based recommenders: history length sensitivity, candidate position bias, generation-involved performance, and hallucination detection [65]. These dimensions address unique challenges in generative systems that are not captured by conventional evaluation approaches.

Position bias represents a particularly significant challenge, as LLMs demonstrate preferences for items at certain positions in candidate lists, which can systematically skew recommendation quality [65]. Hallucination issues, where models recommend non-existent items, pose threats to user experience and system reliability that require specialized detection and mitigation strategies.

The need for comprehensive benchmarking methodologies has led to the development of robust evaluation frameworks using diverse datasets and multiple aggregation approaches [66]. However, standardization across the field remains limited, with new algorithms frequently claiming state-of-the-art performance based on evaluations over limited datasets [66].

Scalability and Resource Constraints

Storage demands for generative recommendation systems are projected to exceed 2 exabytes by 2029, up from less than 1 exabyte in 2024 [**?**]. Traditional storage systems struggle with the high throughput and low latency requirements of real-time AI workflows, necessitating specialized infrastructure investments that may be prohibitive for many organizations.

Training costs are growing at 2.4x per year, with the most advanced models now costing hundreds of millions of dollars to train [**?**]. This exponential cost growth creates barriers to entry and limits experimentation, potentially concentrating advanced recommendation capabilities among a few well-resourced organizations.

The inference economy is projected to exceed $250 billion by 2030, overtaking training as the dominant expense in enterprise AI [**?**]. This shift emphasizes the critical importance of inference efficiency, where the dramatic improvements demonstrated by systems like OneRec and MTGR become essential for economic viability.

7.2. Future Directions

Integration of Multimodal Large Language Models

The integration of multimodal large language models represents one of the most promising research directions, with Gartner predicting that 40% of GenAI solutions will be multimodal by 2027 [**?**]. Advanced multimodal architectures like Google Gemini 3.0 enable seamless processing of text, image, audio, and video inputs, opening new possibilities for richer user understanding and more contextually-aware recommendations [**?**].

Parameter-efficient fine-tuning methods such as LoRA (Low-Rank Adaptation) are emerging as crucial techniques for making large-scale generative models more accessible and adaptable [**?**]. These ap-

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

proaches enable efficient customization of foundation models for specific recommendation tasks without requiring extensive retraining, potentially democratizing access to advanced generative recommendation capabilities.

The development of agentic AI systems represents another significant opportunity, with effective AI agents capable of accelerating business processes by 30% to 50% [**?**]. These systems could transform recommendation platforms from static systems to dynamic ecosystems that optimize and adapt instantaneously based on user behavior and contextual changes.

Foundation Models and Cross-Domain Transfer

The emergence of foundation models for recommendation systems, exemplified by PinFM's 20+ billion parameter architecture, suggests a future where large pre-trained models can be adapted across multiple recommendation domains and applications. This approach could address data sparsity issues and enable more effective knowledge transfer between different recommendation scenarios.

Cross-domain transfer learning represents a particularly promising research direction, where models trained on one type of content or user behavior can be adapted to improve recommendations in related domains. This capability could be especially valuable for new platforms or content categories with limited historical data.

Ethical AI and Responsible Recommendation

The increasing sophistication of generative recommendation systems raises important questions about algorithmic transparency, fairness, and user privacy. Future research must address user-side fairness to ensure equal recommendation quality across demographic groups, item-side fairness to provide fair exposure for different content categories, and algorithmic transparency to enable explainable recommendation decisions [65].

The concentration of computational resources among major technology companies raises concerns about market competition and innovation in the recommendation systems space. Ensuring broader access to advanced recommendation capabilities while maintaining responsible AI practices represents a significant challenge for the field.

Technical Innovation Frontiers

Future technical developments are expected to focus on more sophisticated fusion mechanisms for multimodal data, improved cross-modal attention architectures, and more efficient processing of high-dimensional information. The development of specialized hardware optimizations and novel algorithmic approaches could address current computational bottlenecks while enabling new capabilities.

The integration of real-time learning and adaptation capabilities represents another important frontier, where recommendation systems could continuously evolve based on immediate user feedback and changing preferences. This capability would require advances in online learning algorithms and efficient model updating mechanisms.

Industry Adoption and Standardization

The transition from research to widespread industrial adoption requires continued development of standardized evaluation frameworks, best practices for deployment, and tools for managing the complexity of large-scale generative recommendation systems. The success of current deployments provides valuable insights, but broader adoption will require addressing the challenges that currently limit most projects to proof-of-concept stages.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

The development of open-source frameworks and collaborative approaches, as promoted by initiatives like the Linux Foundation AI & Data's GenAI Commons, could accelerate progress by enabling shared solutions to common challenges such as scalability, efficiency, fairness, and privacy [**?**].

Long-term Vision and Transformative Potential

The ultimate vision for generative recommendation systems involves fully generative, interpretable, and adaptive systems that meet real-world demands while providing transparent and fair recommendations [**?**]. Achieving this vision will require continued advances across multiple dimensions, from algorithmic innovations and infrastructure developments to evaluation methodologies and ethical frameworks.

The potential for generative recommendation systems to transform how users discover and interact with content represents one of the most significant opportunities in modern AI applications. Success in addressing current challenges while realizing emerging opportunities could establish generative approaches as the dominant paradigm for recommendation systems across diverse domains and applications.

8. Conclusions

The rise of generative models in natural language processing and computer vision has spurred a parallel evolution in recommender systems. Traditional approaches, centered on retrieval and scoring, operate by ranking items from a fixed catalog based on user history, collaborative signals, or learned representations. While effective, such systems often struggle to adapt to rapid shifts in user intent, offer limited output diversity, and rely on complex, feature-heavy pipelines.

In contrast, generative recommendation offers a fundamentally different paradigm. Rather than selecting items, models generate them, either as identifiers, textual descriptions, or structured content, conditioned on serialized user behavior. Early works like P5 [67] and M6-Rec [68] proposed prompt-based frameworks to unify tasks across recommendation and user modeling. More recent architectures such as OneRec [16], COBRA [21], and GenRank [36] push this paradigm further, demonstrating end-to-end generation, multi-objective alignment, and real-world deployability.

The key breakthrough of generative recommendation lies in shifting from retrieval based matching to content generation. Traditional systems face two core challenges: (1) difficulty in capturing rapid changes in user intent, and (2) limited diversity and creativity in recommendation results. Generative models offer a potential solution to both. By leveraging user profiles and historical behavior sequences, generative recommenders can directly synthesize personalized content descriptions, product suggestions, or even entire articles. Not merely choosing the most similar items, but actively creating content aligned with user interests. This enables not only satisfaction of explicit needs, but also the discovery and guidance of latent user interests.

Companies like Meta have already deployed trillion-parameter generative recommender systems in production, reimagining recommendation tasks as sequence generation problems. These systems abandon traditional pipelines filled with handcrafted features, opting instead for a fully serialized representation of user behavior, akin to large language models (LLMs). This end-to-end generative paradigm significantly simplifies the recommendation process, improves performance, and allows recommender systems to benefit from scaling laws similar to those observed in LLMs like GPT. The result is not just better recommendations, but also reduced engineering complexity.

Despite its theoretical promise, generative recommendation faces several real-world challenges. First, there is the problem of item representation: how to convert complex entities such as products or

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

content into token sequences that generative models can understand, which directly impacts output quality. Second, controllability and precision remain concerns, particularly in multiturn interactions or complex user scenarios, where the generated content must be both relevant and coherent. Finally, computational cost is a major hurdle. Ensuring inference efficiency and scalability while maintaining effectiveness is a critical engineering challenge for deploying generative recommenders at an industrial scale.

Currently, generative recommendation has already achieved initial success in several leading companies. Tech giants such as Meta, Alibaba, and Kuaishou have actively explored and deployed generative architectures in real-world systems, demonstrating the feasibility and effectiveness of this new paradigm. These implementations span various tasks, including personalized ranking, item generation, and query suggestion. While still in its early stages, generative recommendation is rapidly evolving, with ongoing research focused on improving model controllability, inference efficiency, scaling models across modalities (text, image, audio), and alignment with user preferences. As the ecosystem matures, it is expected that generative paradigms will become a core component of next-generation recommender systems.

References

1. Resnick, P.; Varian, H.R. Collaborative Filtering Recommender Systems. In *The Adaptive Web: Methods and Strategies of Web Personalization*; Springer: Berlin, Heidelberg, 2007; Vol. 4321, *Lecture Notes in Computer Science*, chapter 9, pp. 291–324.
2. 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 Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, 2018, pp. 1059–1068.
3. Qiu, J.; Wang, Z.; Zhang, F.; Zheng, Z.; Zhu, J.; Fan, J.; Zhang, T.; Wang, H.; Wang, Y.; Wang, X. EGA-V1: Unifying Online Advertising with End-to-End Learning, 2025, [arXiv:cs.IR/2505.19755].
4. Koren, Y.; Bell, R.; Volinsky, C. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer* **2009**, *42*, 30–37.
5. Naumov, M.; Mudigere, D.; Shi, H.J.M.; Huang, J.; Sundaraman, N.; Park, J.; Wang, X.; Gupta, U.; Wu, C.J.; Azzolini, A.G.; et al. Deep Learning Recommendation Model for Personalization and Recommendation Systems, 2019, [arXiv:cs.IR/1906.00091].
6. Bromley, J.; Guyon, I.; LeCun, Y.; Säckinger, E.; Shah, R. Signature verification using a "Siamese" time delay neural network. In Proceedings of the Proceedings of the 7th International Conference on Neural Information Processing Systems, San Francisco, CA, USA, 1993; NIPS'93, p. 737–744.
7. Kim, Y. Applications and Future of Dense Retrieval in Industry. In Proceedings of the Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, 2022; SIGIR '22, p. 3373–3374. https://doi.org/10.1145/3477495.3536324.
8. Wang, R.; Fu, B.; Fu, G.; Wang, M. Deep & Cross Network for Ad Click Predictions. In Proceedings of the Proceedings of the ADKDD'17. ACM, 2017.
9. Bottou, L.; Bousquet, O. The Tradeoffs of Large Scale Learning. In Proceedings of the Advances in Neural Information Processing Systems; Platt, J.; Koller, D.; Singer, Y.; Roweis, S., Eds. Curran Associates, Inc., 2007, Vol. 20.
10. Xi, Y.; Liu, W.; Dai, X.; Tang, R.; Zhang, W.; Liu, Q.; He, X.; Yu, Y. Context-aware Reranking with Utility Maximization for Recommendation, 2022, [arXiv:cs.IR/2110.09059].
11. Rajput, S.; Mehta, N.; Singh, A.; Keshavan, R.H.; Vu, T.; Heldt, L.; Hong, L.; Tay, Y.; Tran, V.Q.; Samost, J.; et al. Recommender Systems with Generative Retrieval, 2023, [arXiv:cs.IR/2305.05065].
12. 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, 2025, [arXiv:cs.IR/2504.01403].

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

13. Ren, Y.; Yang, Q.; Wu, Y.; Xu, W.; Wang, Y.; Zhang, Z. Non-autoregressive Generative Models for Reranking Recommendation, 2025, [arXiv:cs.IR/2402.06871].
14. Xu, Z.; Zhang, Y. LLM-Enhanced Reranking for Complementary Product Recommendation, 2025, [arXiv:cs.IR/2507.16237].
15. Shi, T.; Xu, J.; Zhang, X.; Zang, X.; Zheng, K.; Song, Y.; Yu, E. Unified Generative Search and Recommendation, 2025, [arXiv:cs.IR/2504.05730].
16. 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, 2025, [arXiv:cs.IR/2502.18965].
17. Liang, S.; Zhang, Y.; Wang, Y. C-TLSAN: Content-Enhanced Time-Aware Long- and Short-Term Attention Network for Personalized Recommendation, 2025, [arXiv:cs.LG/2506.13021].
18. Liang, S.; Zhang, Y.; Guo, Y. PersonaAgent with GraphRAG: Community-Aware Knowledge Graphs for Personalized LLM, 2025, [arXiv:cs.LG/2511.17467].
19. Wu, L.; Zheng, Z.; Qiu, Z.; Wang, H.; Gu, H.; Shen, T.; Qin, C.; Zhu, C.; Zhu, H.; Liu, Q.; et al. A Survey on Large Language Models for Recommendation, 2024, [arXiv:cs.IR/2305.19860].
20. Yang, L.; Paischer, F.; Hassani, K.; Li, J.; Shao, S.; Li, Z.G.; He, Y.; Feng, X.; Noorshams, N.; Park, S.; et al. Unifying Generative and Dense Retrieval for Sequential Recommendation, 2024, [arXiv:cs.IR/2411.18814].
21. 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, 2025, [arXiv:cs.IR/2503.02453].
22. Zhai, J.; Liao, L.; Liu, X.; Wang, Y.; Li, R.; Cao, X.; Gao, L.; Gong, Z.; Gu, F.; He, J.; et al. Actions Speak Louder than Words: Trillion-Parameter Sequential Transducers for Generative Recommendations. In Proceedings of the Proceedings of the 41st International Conference on Machine Learning; Salakhutdinov, R.; Kolter, Z.; Heller, K.; Weller, A.; Oliver, N.; Scarlett, J.; Berkenkamp, F., Eds. PMLR, 21–27 Jul 2024, Vol. 235, *Proceedings of Machine Learning Research*, pp. 58484–58509.
23. 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, 2025, [arXiv:cs.IR/2505.17549].
24. 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, 2025, [arXiv:cs.IR/2505.18654].
25. 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 2024 IEEE 40th International Conference on Data Engineering (ICDE), 2024, pp. 1435–1448. https://doi.org/10.1109/ICDE60146.2024.00118.
26. Zhang, B.; Luo, L.; Chen, Y.; Nie, J.; Liu, X.; Guo, D.; Zhao, Y.; Li, S.; Hao, Y.; Yao, Y.; et al. Wukong: Towards a Scaling Law for Large-Scale Recommendation, 2024, [arXiv:cs.LG/2403.02545].
27. 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, 2025, [arXiv:cs.IR/2509.03236].
28. 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. *arXiv preprint arXiv:2411.11739* **2024**.
29. Jia, J.; Wang, Y.; Li, Y.; Chen, H.; Bai, X.; Liu, Z.; Liang, J.; Chen, Q.; Li, H.; Jiang, P.; et al. LEARN: Knowledge Adaptation from Large Language Model to Recommendation for Practical Industrial Application. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2025, Vol. 39, pp. 11861–11869.
30. 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**.
31. 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**.
32. Badrinath, A.; Agarwal, P.; Bhasin, L.; Yang, J.; Xu, J.; Rosenberg, C. PinRec: Outcome-Conditioned, Multi-Token Generative Retrieval for Industry-Scale Recommendation Systems, 2025, [arXiv:cs.IR/2504.10507].

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

33. Cao, J.; Xu, P.; Cheng, Y.; Guo, K.; Tang, J.; Wang, S.; Leng, D.; Yang, S.; Liu, Z.; Niu, Y.; et al. Pantheon: Personalized Multi-objective Ensemble Sort via Iterative Pareto Policy Optimization. *arXiv preprint arXiv:2505.13894* **2025**.
34. 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 preprint arXiv:2508.11977* **2025**.
35. 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, 2025, [arXiv:cs.IR/2508.15308].
36. 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, 2025, [arXiv:cs.IR/2505.04180].
37. Wang, Y.; Yan, X.; Ma, C.; Huang, M.; Li, X.; Yu, L.; Liu, C.; Han, R.; Jiang, H.; Yin, B.; et al. MTGRBoost: Boosting Large-scale Generative Recommendation Models in Meituan, 2025, [arXiv:cs.DC/2505.12663].
38. Meng, Y.; Guo, C.; Cao, Y.; Liu, T.; Zheng, B. A Generative Re-ranking Model for List-level Multi-objective Optimization at Taobao, 2025, [arXiv:cs.IR/2505.07197].
39. Kong, L.; Wang, L.; Peng, C.; Lin, Z.; Law, C.; Shao, J. Generative Click-through Rate Prediction with Applications to Search Advertising, 2025, [arXiv:cs.LG/2507.11246].
40. 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, 2025, [arXiv:cs.IR/2505.16752].
41. 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, 2025, [arXiv:cs.IR/2508.20900].
42. 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, 2025, [arXiv:cs.IR/2508.14646].
43. 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, 2025, [arXiv:cs.IR/2506.06913].
44. 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, 2025, [arXiv:cs.IR/2506.23643].
45. Chen, X.; Rajesh, K.; Lawhon, M.; Wang, Z.; Li, H.; Li, H.; Joshi, S.V.; Eksombatchai, P.; Yang, J.; Hsu, Y.P.; et al. Pinfm: foundation model for user activity sequences at a billion-scale visual discovery platform. In Proceedings of the Proceedings of the Nineteenth ACM Conference on Recommender Systems, 2025, pp. 381–390.
46. 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**.
47. 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**.
48. Oord, A.v.d.; Vinyals, O.; Kavukcuoglu, K. Neural Discrete Representation Learning. In Proceedings of the NeurIPS, 2017.
49. Esser, P.; Rombach, R.; Ommer, B. Taming Transformers for High-Resolution Image Synthesis. In Proceedings of the CVPR, 2021.
50. Wang, X.; Chen, K.; Zha, Z.J.; Luo, J. Learning Image-Adaptive Codebooks for Class-Agnostic Image Restoration. In Proceedings of the CVPR, 2023.
51. Zeghidour, N.; Luebs, A.; Omran, A.; Skoglund, J.; Tagliasacchi, M. SoundStream: An End-to-End Neural Audio Codec. In Proceedings of the IEEE/ACM Transactions on Audio, Speech, and Language Processing. IEEE, 2021. https://doi.org/10.1109/TASLP.2021.3129994.
52. Jégou, H.; Douze, M.; Schmid, C. Product Quantization for Nearest Neighbor Search. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2011**, *33*, 117–128. https://doi.org/10.1109/TPAMI.2010.57.
53. Ju, C.M.; Collins, L.; Neves, L.; Kumar, B.; Wang, L.Y.; Zhao, T.; Shah, N. Generative Recommendation with Semantic IDs: A Practitioner's Handbook, 2025, [arXiv:cs.IR/2507.22224].

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 December 2025 doi:10.20944/preprints202512.0741.v1

54. Rafailov, E.; Ouyang, L.; Wang, J.; Zhu, S.; Amodei, D.; Christiano, P.; Leike, J.; Irving, G. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. *arXiv preprint arXiv:2305.18290* **2023**.
55. Xiong, W.; Dong, H.; Ye, C.; Wang, Z.; Zhong, H.; Ji, H.; Jiang, N.; Zhang, T. Iterative Preference Learning from Human Feedback: Bridging Theory and Practice for RLHF under KL-constraint. In Proceedings of the Proceedings of the 41st International Conference on Machine Learning (ICML/PMLR), 2024, pp. 54715–54754.
56. OpenAI. GPT-4 Technical Report. https://openai.com/research/gpt-4, 2023. Accessed: 2025-08-04.
57. Kang, W.C.; McAuley, J. Self-Attentive Sequential Recommendation, 2018, [arXiv:cs.IR/1808.09781].
58. Ivchenko, D.; Van Der Staay, D.; Taylor, C.; Liu, X.; Feng, W.; Kindi, R.; Sudarshan, A.; Sefati, S. TorchRec: a PyTorch Domain Library for Recommendation Systems. In Proceedings of the Proceedings of the 16th ACM Conference on Recommender Systems, New York, NY, USA, 2022; RecSys '22, p. 482–483. https://doi.org/10.1145/3523227.3547387.
59. Cai, H.; Li, Y.; Yuan, R.; Wang, W.; Zhang, Z.; Li, W.; Chua, T.S. Exploring Training and Inference Scaling Laws in Generative Retrieval. In Proceedings of the Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, 2025; SIGIR '25, p. 1339–1349. https://doi.org/10.1145/3726302.3729973.
60. 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. In Proceedings of the Proceedings of the 13th International Conference on Learning Representations (ICLR), 2025.
61. Hou, Y.; Li, J.; He, Z.; Yan, A.; Chen, X.; McAuley, J. Bridging Language and Items for Retrieval and Recommendation. *arXiv preprint arXiv:2403.03952* **2024**.
62. Liu, Q.; Zheng, K.; Huang, R.; Li, W.; Cai, K.; Chai, Y.; Niu, Y.; Hui, Y.; Han, B.; Mou, N.; et al. RecFlow: An Industrial Full Flow Recommendation Dataset. *arXiv preprint arXiv:2410.20868* **2024**.
63. Sun, Z.; Si, Z.; Zang, X.; Leng, D.; Niu, Y.; Song, Y.; Zhang, X.; Xu, J. KuaiSAR: A Unified Search And Recommendation Dataset. In Proceedings of the Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, New York, NY, USA, 2023; CIKM '23, p. 5407–5411. https://doi.org/10.1145/3583780.3615123.
64. Jain, R.; Bhasi, V.M.; Jog, A.; Sivasubramaniam, A.; Kandemir, M.T.; Das, C.R. Pushing the Performance Envelope of DNN-based Recommendation Systems Inference on GPUs. In Proceedings of the 2024 57th IEEE/ACM International Symposium on Microarchitecture (MICRO). IEEE, 2024, pp. 1217–1232.
65. Jiang, C.; Wang, J.; Ma, W.; Clarke, C.L.; Wang, S.; Wu, C.; Zhang, M. Beyond Utility: Evaluating LLM as Recommender. In Proceedings of the Proceedings of the ACM on Web Conference 2025, 2025, pp. 3850–3862.
66. Shevchenko, V.; Belousov, N.; Vasilev, A.; Zholobov, V.; Sosedka, A.; Semenova, N.; Volodkevich, A.; Savchenko, A.; Zaytsev, A. From Variability to Stability: Advancing RecSys Benchmarking Practices. arXiv. org **2024**.
67. Geng, X.; Li, J.; Zhao, W.X.; Wen, J.R. P5: Unified Text-to-Text Pretraining for User Modeling and Recommendation. *arXiv preprint arXiv:2203.13366* **2022**.
68. Cui, L.; Wang, X.; Zhang, Y.; Xiao, X.; Xie, X. M6-Rec: A Simple and Effective Pre-Trained Model for Recommendation with Prompt Tuning. *arXiv preprint arXiv:2205.02180* **2022**.

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.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.