Revisiting Generative vs. Dense Retrieval: A Static Value Prior Reverses the Gap

Renbo Tu[1,2*] Yang Liu[1] Shuying Sun[1†]

[1] Shopify [2] University of Toronto

Abstract

Generative retrieval (GR) has emerged as a strong candidate generator for sequential recommendation as language models and semantic-item identifiers have improved. However, GR is not a uniformly stronger replacement of dense retrieval (DR) in online retail candidate generation. We examine their differences on two Amazon Toys datasets under different metrics. GR leads on best relevant item value among users both methods cover, while DR leads on exact-target recall and relevant value across the ranked list. With deep analysis, we find that popularity contributes the most to GR's measured value lead. Adding the same static item value to DR in one reranking step reverses this lead and keeps recall above GR. Finally, on an industrial-scale internal e-commerce dataset with LLM-judged relevance, we show that a 14B SID-LLM still underperforms a 47× smaller dense encoder. Our code and artifacts are open-sourced at https://anonymous.4open.science/r/gr-dr-release-BCE4.

Keywords

generative retrieval, dense retrieval, sequential recommendation, semantic IDs, popularity bias, recommendation evaluation

ACM Reference Format:

Renbo Tu, Yang Liu, and Shuying Sun. 2026. Revisiting Generative vs. Dense Retrieval: A Static Value Prior Reverses the Gap. In *Proceedings of The 35th ACM International Conference on Information and Knowledge Management (CIKM '26)*. ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Dense retrieval (DR) has long been a standard candidate-generation approach for recommendation. It encodes a user's interaction history and catalog items in a shared embedding space, then retrieves nearby items [3, 36]. Advances in language models and semantic IDs have made generative retrieval (GR) an increasingly viable alternative. GR represents items with semantic IDs and generates the next item's ID one token at a time [6, 21, 25, 31, 32]. Recent work has demonstrated GR at industrial scale [26, 41–43], but its use does not establish that it consistently outperforms or can replace DR in candidate generation. As shown in Fig. 1, normally DR retrieves the held out next item more often, while GR's best relevant item has higher value on average for users both methods cover.

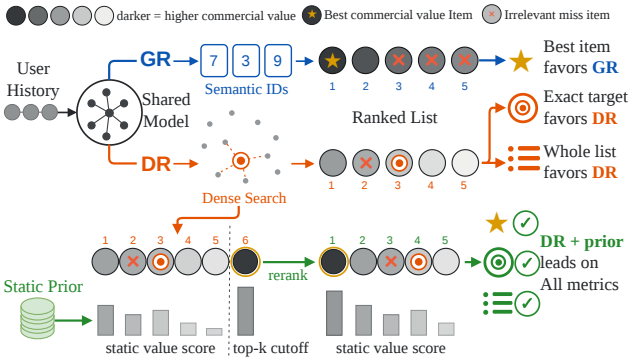


Figure 1: GR and DR use the same user history but generate candidates through semantic ID decoding and dense search. GR leads on best item value, while DR leads on exact target retrieval and full list value. A static prior reranks DR candidates and moves the potential valuable item into the top k.

The apparent winner depends on how retrieval is evaluated. This metric choice is particularly crucial in e-commerce candidate generation, where retrieved products differ not only in relevance to the user but also its commercial value. To understand why, we need to know whether GR's value lead comes from the user's history or from generally valuable items. Recall and target NDCG show whether the held out item is retrieved and where it ranks. They do not account for other relevant items, their ranks, or their value.

To understand the disagreement, in this paper, we measure both the best relevant item and relevant value across the full list. We apply these metrics in controlled GR and DR comparisons on Amazon Toys 2014 and 2023 [11, 14, 23, 25, 35]. The metric disagreement persists on both public e-commerce datasets. From analysis, popularity is the largest component of GR's measured value lead. Among users where GR wins, popularity contributes 49.8% of its value lead on 2014 and 65.4% on 2023. Reranking DR with the static item value score reverses the best item value gap while preserving a recall lead over GR. Further checks with cold targets, substitute credit, a random control, and graded industrial relevance do not reveal a consistent GR advantage. Our main contributions are:

- We define complementary metrics for sequential recommendation in e-commerce scenario and show that they favor different methods in the GR and DR comparison.
- We decompose GR's measured value lead and find that popularity is its largest component on both public datasets.
- We propose a training-free value prior, applied to DR as a single rerank of its retrieval results. This reverses GR's lead in best relevant item value while preserving DR's recall lead.

*This work was completed at Shopify.

†Corresponding author: shuying.sun@shopify.com.

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

CIKM '26, Rome, Italy

© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/XXXXXXX.XXXXXXX

2 Related Work

The Generative Turn for Retrieval Systems. Prior to decoder-only approaches, sequential recommendation models for retrieval have been prevalent in production [3, 36]. The canonical dense retrieval (DR) is therefore defined as bidirectional sequential models for next-item prediction [13, 18, 30]. Generative retrieval (GR) enters the foreground as a constrained autoregressive decoding mechanism to generative identifier tokens, semantic IDs (SIDs), mappable to an underlying item, whose codebook construction follows RQ-VAE [6, 21, 31, 32]. SID generative retrieval benefits from the rise of LLMs and demonstrates generalization, establishing itself for industry applications [25, 26, 41–43]. Note that GR umbrellas non-SID approaches such as P5 and HSTU [9, 22, 38]. Our controlled DR-GR comparison is grounded on LIGER, while research on GR's scaling difficulties and DR's capacity limits bounds our conclusions without determining them [24, 33, 35].

Difficulty of Retrieval Evaluation. Offline recommendation evaluation suffers from sensitivity to missing feedback, candidate sampling and temporal splitting decisions [2, 4, 5, 17, 20, 27]. Audits on these decisions reveal that they can change recommendation rankings, while recent studies identify GR artifacts from SID collision and benchmark shortcuts[8, 10, 12, 29, 40]. Our contribution identifies substitute non-identifiability under exact-target metrics and show how it worsens with catalog scale.

Popularity Bias is a Static Prior. In recommendation, popularity of certain products are deeply associated with benchmark accuracy [15, 28]. However, popularity also presents itself as a bias that degrades exposure and fairness for non-popular items [1, 19]. Popularity bias in GR becomes prominent due to inherent bias within LLMs [7, 37, 39]. While these works mitigate popularity bias, our result show that under a same model GR-DR comparison, the advantage from popularity is replaceable by a user-independent prior that can be transferred to DR without decoding.

3 Controlled Comparison Setup

Problem Definition. In the e-commerce setting, for each user u, let $h_u = (i_1, \ldots, i_t)$ denote the observed user product interaction history and $y_u = i_{t+1}$ the held out next item. A candidate generator x uses h_u to produce the ranked list

$$L_x^k(u) = \left(i_{x,u}^{(1)}, \ldots, i_{x,u}^{(k)}\right),$$

where $i_{x,u}^{(j)}$ is the item at rank j.

Models. We compare GR and DR using the TIGER and LIGER framework [25, 35]. Both arms use the same interaction data, item catalog, and jointly trained T5 encoder decoder. GR decodes valid semantic IDs through prefix constrained beam search, while DR ranks items by cosine similarity using the encoder representation of the same history. Thus, the arms differ mainly in their retrieval readout.

Datasets. Our public evaluation uses the 2014 and 2023 releases of Amazon Toys. The 2014 dataset contains 11,924 items and 19,412 user histories under a leave last out split. Among their held out targets, 19,116 were seen during training and 296 were absent from training. Our results reproduce the LIGER Recall@10 values, while our GR recall exceeds the published TIGER result. The 2023 catalog is 13.6× larger, which tests whether the observed pattern and the proposed intervention survive a substantial increase in catalog scale [14]. We transfer the value weights and reranking coefficient from 2014 to 2023 without change unless stated otherwise.

4 Evaluation Metrics

We compare GR and DR by asking *which retrieved items are relevant* and *how much value they carry.* A relevance labeling rule $q_u(i) \in [0, 1]$ assigns evaluation credit after ranking and is not a model score. It may credit only the held-out item, include substitute candidates, or assign partial credit through graded judgments. In contrast, $v_i \in [0, 1]$ measures a static property of item i. For any rule q, the relevant set with k candidates is defined as

$$R_x^k(u; q) = \{i \in L_x^k(u) \mid q_u(i) > 0\}.$$

This definition keeps relevance and item value separate across evaluation protocols. In this section, We first define exact-target and category-based relevance, then introduce static catalog value, and finally combine them at the best-item and whole-list levels.

4.1 Exact-Target Retrieval

The conventional next item protocol assigns credit only to the held out item, as defined by $q_u^{\text{exact}}(i) = \mathbb{1}[i = y_u]$. Recall@$k$ measures whether this item appears in the top-k results.

$$\text{Recall}_x@k = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \mathbb{1}[y_u \in L_x^k(u)].$$

NDCG@k evaluates the same target with a rank discount [16]. Both metrics measure recovery of the observed next interaction and give no credit to other potentially useful items.

4.2 Beyond Exact Match: Substitute Blindness

Exact target evaluation cannot distinguish a substitute from an irrelevant item because both receive zero credit. We refer to this limitation as *substitute blindness.* Therefore we define a coarse proxy for substitute-aware relevance using the shared leaf category as

$$q_u^{\text{cat}}(i) = \mathbb{1}[\text{leaf}(i) = \text{leaf}(y_u)], \quad S(u) = \{i \neq y_u : q_u^{\text{cat}}(i) = 1\}.$$

where $\text{leaf}(i)$ and $\text{leaf}(y_u)$ identify the leaf categories of candidate item i and the held-out target. An item receives $q_u^{\text{cat}}(i) = 1$ when their categories match. A larger substitute set $|S(u)|$ makes an exact miss less informative, but category overlap alone does not establish that two items are substitutes. We therefore stress-test this proxy using co-purchase credit, a mass-matched random-credit control, and LLM-judged graded relevance in Section 6.

4.3 Static Item Value

Relevance determines whether an item is useful for a given user. In e-commerce, products also differ in static catalog attributes that may matter to the platform. We summarize these attributes with a transparent item-value proxy

$$v_i(\mathbf{w}) = w_p \, \text{pop}_i + w_r \, \text{rate}_i + w_b \, \text{brand}_i, \quad w_p + w_r + w_b = 1. \quad (1)$$

Here, pop_i, rate_i, and brand_i are the catalog-percentile ranks of log popularity, the Wilson lower bound on star rating, and brand footprint, respectively [34]. All weights are nonnegative. Unless

Table 1: Amazon Toys results. Positive ΔBV@50 values favor the listed method over GR. Best values are bold.

	Recall@50		NDCG@50		ΔBV@50		vNDCG@50	
Method	2014	2023	2014	2023	2014	2023	2014	2023
GR	0.167	0.052	0.056	0.017	—	—	0.215	0.160
DR	**0.228**	**0.084**	**0.077**	**0.029**	−0.020	−0.016	0.222	0.168
DR+prior	0.210	0.083	0.072	0.028	**+0.023**	**+0.006**	**0.240**	**0.175**

otherwise stated, we use the default weights $\mathbf{w} = (0.25, 0.25, 0.50)$. This score is a transparent proxy for item value, not a universal measure of recommendation utility. In the sensitivity analysis, we vary $\mathbf{w}$ over nonnegative mixtures whose components sum to one. Unlike $q_u(i)$, which varies across users and relevance labeling rules, $v_i(\mathbf{w})$ is computed once for each catalog item.

4.4 Best-Item Value vs. Whole-List Value

Having defined relevance q and item value v separately, we evaluate each ranked list using two complementary metrics.

Conditional best-item value (BV). BV asks how valuable the best relevant item returned by method x is for user u.

$$\mathrm{BV}_x^k(u; q, \mathbf{w}) = \max_{i \in R_x^k(u;q)} v_i(\mathbf{w}).$$

In our main analysis, BV uses the binary category rule q^{cat} from Sec 4.2 to select relevant items but does not weight them by relevance magnitude. Let $C_{x,a}^k$ denote the users for whom both method x and comparator a retrieve at least one relevant item. We report their average gap as

$$\Delta\mathrm{BV}_{x-a}@k = \frac{1}{|C_{x,a}^k|} \sum_{u \in C_{x,a}^k} \left[\mathrm{BV}_x^k(u) - \mathrm{BV}_a^k(u)\right],$$

where q and $\mathbf{w}$ are omitted for readability. We compare the BV scores of methods x and a for each user u. A difference above $\tau = 0.05$ is a win for x, a difference below $-\tau$ is a win for a, and all other differences are ties.

Whole list Value. BV focuses on the best relevant item. Its comparison includes only users with a relevant item from both methods. To score the full top k list for every user, we use NDCG with item value weights (vNDCG). Each item contributes its relevance times its value, with less credit at lower ranks. For method x and user u, let $i_j = i_{x,u}^{(j)}$ be the item at rank j. The list score is

$$\mathrm{vDCG}_x^k(u) = \sum_{j=1}^{k} \frac{q_u(i_j) v_{i_j}(\mathbf{w})}{\log_2(j+1)}.$$

For each user, we pool the items returned by all methods and rank them by relevance times value, $q_u(i)v_i(\mathbf{w})$. The discounted score of the top k items is the ideal score $\mathrm{IDCG}^k(u)$. We divide each method's score by this shared score, then average across users

$$\mathrm{vNDCG}_x@k = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{\mathrm{vDCG}_x^k(u)}{\mathrm{IDCG}^k(u)}.$$

If the ideal score is zero for a user, their vNDCG score is zero. All methods use the same ideal score for each user, so their scores can be compared within the fixed pool. Absolute scores from different method pools should not be compared directly. To check whether coverage alone explains the metric gap, we also compare BV over all users, performing imputation by assigning zero when a method retrieves no relevant item.

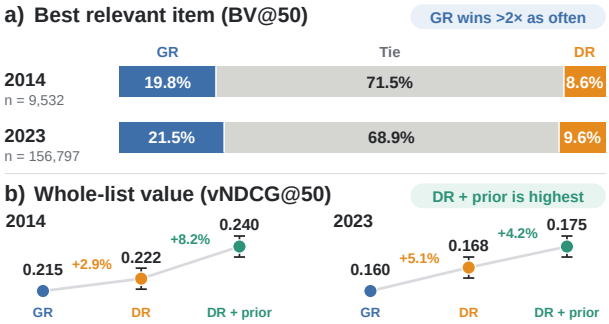


Figure 2: Metric choice reverses the result. (a) GR wins more often under BV@50 in both datasets. (b) Under vNDCG@50, DR scores higher than GR and DR+prior scores highest. Error bars show standard errors for adjacent differences.

5 From Metric Dependency to a Static Prior

We now apply these complementary metrics in a three-step analysis. We first show that GR leads on BV while DR leads on vNDCG. We then measure how popularity, rating, and brand contribute to GR's value advantage. Finally, we rerank DR with the same static item value and test whether the gap reverses.

5.1 The Metric-Dependent GR–DR Gap

We begin with the empirical disagreement. GR leads on BV, while raw DR leads on vNDCG@50, Recall@50, and target NDCG@50 on both datasets as shown in Table 1. GR wins the paired BV comparison more than twice as often as raw DR (Figure 2(a)), and its lead remains when users without a relevant retrieved item receive zero. DR's vNDCG@50 lead is 2.9% in 2014 and 5.1% in 2023, with confidence intervals above zero (Figure 2(b)). Value weighting compresses DR's relevance advantage by an order of magnitude. We conduct a full sweep on value weights to demonstrate robustness. Across the 231-point weight simplex shown in Fig. 3, the sign of the vNDCG gap holds on 223 points on 2014, failing only in popularity-dominated edge cases, and on all 231 points on 2023.

5.2 Diagnosing GR's Value Advantage

Among users where GR wins BV, we use Equation 1 to break its value lead over DR into popularity, rating, and brand. Popularity contributes for 49.8% of the lead on 2014 and 65.4% on 2023. It is the largest component on both datasets, and its share is higher on the larger 2023 catalog. Rating and brand contribute the remainder, but neither provides the same cross-dataset dominance.

This pattern narrows the interpretation of GR's value edge. Each value component is computed once per item and does not depend on the current history. The measured lift is therefore largely captured by static item-level properties, especially popularity. Repeated exposure in training offers a possible explanation for GR's value lead. The released 2014 popularity score correlates with how often held out targets appear in training histories (Spearman $\rho = 0.582$). At least ten training histories contain 50.3% of held out targets in 2014

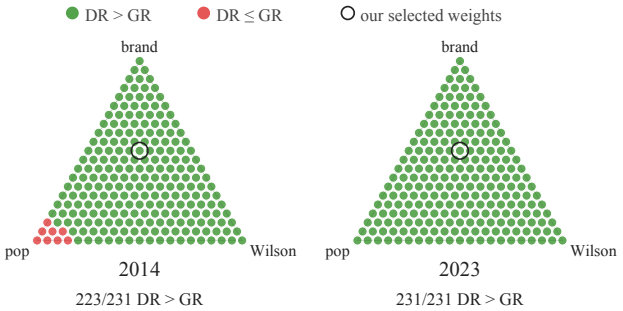


Figure 3: Sign of the raw DR-vs-GR vNDCG@50 gap over the static value on Amazon Toys datasets. Grid step is 0.05.

and 75.1% in 2023. GR may learn a preference for frequently seen items while predicting their semantic IDs. This could contribute to its lead because popularity is part of the item value score. That statement yields a falsifiable implication. If static item properties account for the operational advantage, then explicitly adding the same information to DR should reproduce it without autoregressive decoding.

5.3 Transferring the Static Prior to DR

To test whether static item value can close GR's lead in BV, we rerank DR's top 200 candidates for each user. We call this method DR+prior. DR+prior scales the DR similarity scores to 0, 1 within each pool, then average each score with the item's value $v_i(\mathbf{w})$ using the default weights from BV in Sec 4.3. Because BV uses the same item value, this test checks whether an explicit catalog prior can reproduce GR's measured lead. This requires no retraining or decoder at serving time. We score each candidate by

$$\sigma_u(i) \;=\; (1-\alpha)\,\tilde{s}_u(i) \;+\; \alpha\, v_i, \qquad \alpha = 0.5, \quad i \in P_u$$

As results, adding the prior turns DR's BV@50 deficit against GR into a lead. On 2014, DR moves from trailing by 0.020 to leading by 0.023. On 2023, it moves from trailing by 0.016 to leading by 0.006. As shown in Table 1, DR+prior also has the highest vNDCG@50 on both datasets, at 0.240 and 0.175. Its Recall@50 declines from raw DR but stays above GR.

6 Robustness Beyond Exact-Match Evaluation

The public results show that the item prior closes GR's value gap. We now test whether GR does better on cold targets or when useful alternatives to the held out target receive credit. For the latter, we use public substitute evidence and graded judgments on an industrial recommendation dataset.

6.1 Cold Items

We test whether generation helps retrieve targets absent from interaction training. GR finds none of 296 cold targets on 2014 and 5 of 1,899 on 2023. DR finds 47 and 27, while DR+prior finds 36 and 21, respectively. Both DR methods find more cold targets than GR on each dataset.

6.2 Substitute Credit and Graded Relevance

An exact target miss does not tell us whether a list contains a useful alternative. For the median user, the held out target has 157 same-leaf alternatives on 2014 and 1,046 on 2023, a 6.7× increase. Among

Table 2: Substitute credit on 2014. The last column shows the extra shift toward GR from co purchase credit compared with random credit of equal total weight.

	DR–GR	Real–random credit
Recall@50, exact only	+0.0447	—
+ co-purchase credit	−0.0183	−0.0190
+ graded NDCG@50	+0.0153	+0.0111

Table 3: Industrial dataset evaluation under an LLM-judge.

Population (NDCG@50)	GR	DR	DR+prior
Full evaluation set (n = 5,998)	0.453	0.474	**0.530**

DR's 8,635 exact target misses on 2014, 58.2% include an item from that category in the top 50 while the target ranks beyond 200. In another 25.7%, the target ranks between 50 and 200. GR finds the exact target in only 748 of these DR misses, so the misses alone do not show an advantage for GR.

We give binary credit to items linked to the target in Amazon co purchase data, which is reported in Table 2. In this test, a 0.0447 lead for DR under exact Recall@50 becomes a 0.0183 lead for GR. For the binary test, randomly chosen items given the same amount of credit move the gap at least as far toward GR. With graded NDCG, DR leads again by 0.0153. The binary reversal therefore does not establish that GR retrieves better alternatives. Leaf category overlap and co-purchase links are imperfect evidence of usefulness. We next test graded relevance on the industrial dataset.

6.3 Industrial LLM-Judged Relevance

The industrial evaluation uses real e-commerce interaction histories and a product catalog. GR uses a Qwen3 14B model that generates semantic IDs, while DR uses a GTE encoder with about 300 million parameters. GR has roughly 47× as many parameters. A substitute-aware LLM judge calibrated with 250 human labels scores each candidate's relevance from zero to one, allowing partial credit for useful alternatives. Because the models differ in backbone and size, this evaluation tests whether the public result also holds in this setting. It cannot isolate the effect of retrieval architecture. Table 3 shows that giving useful alternatives graded credit preserves the order GR < DR < DR+prior on the industrial dataset. The prior raises DR's NDCG@50 by 11.8%.

7 Conclusion

In this paper, we revisited the GR and DR in a controlled e-commerce comparison and we find two distinct parts to GR's measured advantage. First, metric choice changes the apparent winner. GR leads on the value of the best relevant item, while DR leads on exact target recall and relevant value across the ranked list on both public datasets. Second, popularity makes the largest contribution to GR's value lead. Adding the same static item value to DR in one reranking step reverses this gap while preserving DR's recall lead. Cold target and substitute credit checks do not reveal a broader GR advantage. Graded judgments on an industrial dataset place DR above a larger GR model and DR with the prior above both. These results make DR with a static value prior a useful baseline for future GR systems.

References

[1] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. The unfairness of popularity bias in recommendation. *arXiv preprint arXiv:1907.13286*, 2019. URL https://arxiv.org/abs/1907.13286.

[2] Rocío Cañamares and Pablo Castells. On target item sampling in offline recommender system evaluation. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 259–268, 2020. doi: 10.1145/3383313.3412259.

[3] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for YouTube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 191–198, 2016. doi: 10.1145/2959100.2959190.

[4] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. Performance of recommender algorithms on top-N recommendation tasks. In *Proceedings of the 4th ACM Conference on Recommender Systems*, pages 39–46, 2010. doi: 10.1145/1864708.1864721.

[5] Alexander Dallmann, Daniel Zoller, and Andreas Hotho. A case study on sampling strategies for evaluating neural sequential item recommendation models. In *Proceedings of the 15th ACM Conference on Recommender Systems*, pages 505–514, 2021. doi: 10.1145/3460231.3475943.

[6] Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. Autoregressive entity retrieval. In *9th International Conference on Learning Representations (ICLR)*, 2021. URL https://arxiv.org/abs/2010.00904.

[7] Zezhong Fan, Ziheng Chen, Luyi Ma, Jin Huang, Lalitesh Morishetti, Kaushiki Nag, Sushant Kumar, and Kannan Achan. CRAB: Codebook rebalancing for bias mitigation in generative recommendation. *arXiv preprint arXiv:2604.05113*, 2026. URL https://arxiv.org/abs/2604.05113.

[8] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. Are we really making much progress? a worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 101–109, 2019. doi: 10.1145/3298689.3347058.

[9] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. Recommendation as language processing (RLP): A unified pretrain, personalized prompt & predict paradigm (P5). In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 299–315, 2022. doi: 10.1145/3523227.3546767.

[10] Haoyu Han, Li Ma, Hanbing Wang, Bingheng Li, Daochen Zha, Chun How Tan, Huiji Gao, Xin Liu, Stephanie Moyerman, Sanjeev Katariya, Hui Liu, and Jiliang Tang. An embarrassingly simple graph heuristic reveals shortcut-solvable benchmarks for sequential recommendation. *arXiv preprint arXiv:2605.07125*, 2026. URL https://arxiv.org/abs/2605.07125.

[11] Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*, pages 507–517, 2016. doi: 10.1145/2872427.2883037.

[12] Balázs Hidasi and Ádám Tibor Czapp. Widespread flaws in offline evaluation of recommender systems. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 848–855, 2023. doi: 10.1145/3604915.3608839.

[13] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In *4th International Conference on Learning Representations (ICLR)*, 2016. URL https://arxiv.org/abs/1511.06939.

[14] Yupeng Hou, Jiacheng Li, Xiangjun Fu, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation: Benchmarking LLMs as semantic encoders. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3251–3265, 2024. doi: 10.18653/v1/2026.acl-long.147. URL https://arxiv.org/abs/2403.03952.

[15] Dietmar Jannach, Lukas Lerche, Iman Kamehkhosh, and Michael Jugovac. What recommenders recommend: An analysis of recommendation biases and possible countermeasures. *User Modeling and User-Adapted Interaction*, 25(5):427–491, 2015. doi: 10.1007/s11257-015-9165-3.

[16] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002. doi: 10.1145/582415.582418.

[17] Yitong Ji, Aixin Sun, Jie Zhang, and Chenliang Li. A critical study on data leakage in recommender system offline evaluation. *ACM Transactions on Information Systems*, 41(3):1–27, 2023. doi: 10.1145/3569930.

[18] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 197–206, 2018. doi: 10.1109/ICDM.2018.00035.

[19] Anastasiia Klimashevskaia, Dietmar Jannach, Mehdi Elahi, and Christoph Trattner. A survey on popularity bias in recommender systems. *User Modeling and User-Adapted Interaction*, 34(5):1777–1834, 2024. doi: 10.1007/s11257-024-09406-0.

[20] Walid Krichene and Steffen Rendle. On sampled metrics for item recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1748–1757, 2020. doi: 10.1145/3394486.3403226.

[21] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11513–11522, 2022. doi: 10.1109/CVPR52688.2022.01123.

[22] Yang Liu, Noel Loo, Ali Khanafer, Shuying Sun, Akshay Soni, Zhong Wu, and Linjun Yang. T-RoPE: Time-aware rotary position embedding for sequential recommendation. *arXiv preprint arXiv:2609.30576*, 2026. URL https://arxiv.org/abs/2609.30576.

[23] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–52, 2015. doi: 10.1145/2766462.2767755.

[24] Ronak Pradeep, Kai Hui, Jai Gupta, Adam D. Lelkes, Honglei Zhuang, Jimmy Lin, Donald Metzler, and Vinh Q. Tran. How does generative retrieval scale to millions of passages? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1305–1321, 2023. doi: 10.18653/v1/2023.emnlp-main.83. URL https://arxiv.org/abs/2305.11841.

[25] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Maheswaran Sathiamoorthy. Recommender systems with generative retrieval. In *Advances in Neural Information Processing Systems*, volume 36, pages 10299–10315, 2023. doi: 10.52202/075280-0452. URL https://arxiv.org/abs/2305.05065.

[26] Anima Singh, Trung Vu, Nikhil Mehta, Raghunandan Keshavan, Maheswaran Sathiamoorthy, Yilin Zheng, Lichan Hong, Lukasz Heldt, Li Wei, Devansh Tandon, Ed Chi, and Xinyang Yi. Better generalization with semantic IDs: A case study in ranking for recommendations. In *Proceedings of the 18th ACM Conference on Recommender Systems*, pages 1039–1044, 2024. doi: 10.1145/3640457.3688190. URL https://arxiv.org/abs/2306.08121.

[27] Harald Steck. Training and testing of recommender systems on data missing not at random. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 713–722, 2010. doi: 10.1145/1835804.1835895.

[28] Harald Steck. Item popularity and recommendation accuracy. In *Proceedings of the 5th ACM Conference on Recommender Systems*, pages 125–132, 2011. doi: 10.1145/2043932.2043957.

[29] Aixin Sun. Take a fresh look at recommender systems from an evaluation standpoint. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2629–2638, 2023. doi: 10.1145/3539618.3591931.

[30] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1441–1450, 2019. doi: 10.1145/3357384.3357895.

[31] Yi Tay, Vinh Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, Tal Schuster, William Cohen, and Donald Metzler. Transformer memory as a differentiable search index. In *Advances in Neural Information Processing Systems*, volume 35, pages 21831–21843, 2022. doi: 10.52202/068431-1587.

[32] Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao, Shibin Wu, Qi Chen, Yuqing Xia, Chengmin Chi, Guoshuai Zhao, Zheng Liu, Xing Xie, Hao Sun, Weiwei Deng, Qi Zhang, and Mao Yang. A neural corpus indexer for document retrieval. In *Advances in Neural Information Processing Systems*, volume 35, pages 25600–25614, 2022. doi: 10.52202/068431-1856.

[33] Orion Weller, Michael Boratko, Iftekhar Naim, and Jinhyuk Lee. On the theoretical limitations of embedding-based retrieval. In *14th International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2508.21038.

[34] Edwin B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927. doi: 10.1080/01621459.1927.10502953.

[35] Liu Yang, Fabian Paischer, Kaveh Hassani, Jiacheng Li, Shuai Shao, Zhang Gabriel Li, Yun He, Xue Feng, Nima Noorshams, Sem Park, Bo Long, Robert D. Nowak, Xiaoli Gao, and Hamid Eghbalzadeh. Unifying generative and dense retrieval for sequential recommendation. *arXiv preprint arXiv:2411.18814*, 2024. URL https://arxiv.org/abs/2411.18814.

[36] Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Lukasz Heldt, Aditee Kumthekar, Zhe Zhao, Li Wei, and Ed Chi. Sampling-bias-corrected neural modeling for large corpus item recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 269–277, 2019. doi: 10.1145/3298689.3346996.

[37] Jun Yin, Bangguo Zhu, Peng Huo, Ruochen Liu, Hao Chen, Senzhang Wang, Shirui Pan, and Chengqi Zhang. Echoes in filter bubble: Diagnosing and curing popularity bias in generative recommenders. *arXiv preprint arXiv:2605.16825*, 2026. URL https://arxiv.org/abs/2605.16825.

[38] Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Jiayuan He, Yinghai Lu, and Yu Shi. Actions speak louder than words: Trillion-parameter sequential transducers for generative recommendations. In *Proceedings of the 41st International Conference on Machine*

Learning, volume 235 of *Proceedings of Machine Learning Research*, pages 58484–58509. PMLR, 2024. URL https://proceedings.mlr.press/v235/zhai24a.html.

[39] Jizhi Zhang, Keqin Bao, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. Is ChatGPT fair for recommendation? evaluating fairness in large language model recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 993–999, 2023. doi: 10.1145/3604915.3608860.

[40] Qian Zhang, Lech Szymanski, Haibo Zhang, and Jeremiah D. Deng. How reliable are semantic-ID tokenizer comparisons in generative recommendation? *arXiv preprint arXiv:2605.25330*, 2026. URL https://arxiv.org/abs/2605.25330.

[41] Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. Adapting large language models by integrating collaborative semantics for recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 1435–1448, 2024. doi: 10.1109/ICDE60146.2024.00118.

[42] Guorui Zhou et al. OneRec technical report. *arXiv preprint arXiv:2506.13695*, 2025. URL https://arxiv.org/abs/2506.13695.

[43] Guorui Zhou et al. OneRec-V2 technical report. *arXiv preprint arXiv:2508.20900*, 2025. URL https://arxiv.org/abs/2508.20900.