Exploring the Escalation of Source Bias in User, Data, and Recommender System Feedback Loop

Yuqi Zhou
Sunhao Dai
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
{yuqizhou,sunhaodai}@ruc.edu.cn

Liang Pang
CAS Key Laboratory of AI Safety
Institute of Computing Technology
Chinese Academy of Sciences
Beijing, China
pangliang@ict.ac.cn

Gang Wang
Zhenhua Dong
Huawei Noah's Ark Lab
Shenzhen, China
wanggang110@huawei.com
dongzhenhua@huawei.com

Jun Xu*
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
junxu@ruc.edu.cn

Ji-Rong Wen
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
jrwen@ruc.edu.cn

Abstract

Recommender systems are essential for information access, allowing users to present their content for recommendation. With the rise of large language models (LLMs), AI-generated content (AIGC), primarily in the form of text, has become a central part of the content ecosystem. As AIGC becomes increasingly prevalent, it is important to understand how it affects the performance and dynamics of recommender systems. To this end, we construct an environment that incorporates AIGC to explore its short-term impact. The results from popular sequential recommendation models reveal that **AIGC are ranked higher in the recommender system**, reflecting the phenomenon of source bias [13, 41]. To further explore the long-term impact of AIGC, we introduce a feedback loop with realistic simulators. The results show that the model's preference for AIGC increases as the user clicks on AIGC rises and the model trains on simulated click data. This leads to two issues: In the short term, bias toward AIGC encourages LLM-based content creation, increasing AIGC content, and causing unfair traffic distribution. From a long-term perspective, our experiments also show that when AIGC dominates the content ecosystem after a feedback loop, it can lead to a decline in recommendation performance. To address these issues, we propose a debiasing method based on L1-loss optimization to maintain long-term content ecosystem balance. In a real-world environment with AIGC generated by mainstream LLMs, our method ensures a balance between AIGC and human-generated content in the ecosystem. The code and dataset are available at https://github.com/Yuqi-Zhou/Rec_SourceBias.

CCS Concepts

• **Information systems → Recommender systems**.

Keywords

Source Bias, AI-Generated Content, Large Language Model

ACM Reference Format:
Yuqi Zhou, Sunhao Dai, Liang Pang, Gang Wang, Zhenhua Dong, Jun Xu, Ji-Rong Wen. 2025. Exploring the Escalation of Source Bias in User, Data, and Recommender System Feedback Loop. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25), July 13–18, 2025, Padua, Italy.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3726302.3729972

1 Introduction

The development of Large Language Models (LLMs) is advancing rapidly [46], demonstrating strong capabilities and performing excellently in many text generation tasks, such as machine translation [22], summarization [44], and complex question answering [3, 42]. Due to the cost-effectiveness, high quality, and speed of generating AIGC compared to Human Generated Content (HGC), an increasing volume of online content is being produced by various LLMs and Schick [29] suggests that the synthetic content could dominate up to 90% of the Internet. This shift is altering the existing content creation paradigm and resulting in a prevalence of AIGC on the internet [10, 11].

When AIGC floods into the internet, these contents will be disseminated by the current information retrieval systems, especially recommender systems, which play a central role in shaping users' online experiences. However, the impact of this rapidly growing AIGC content on current and future recommender systems has yet to be explored. Therefore, an important research question emerges: **RQ1: What short-term impacts will the influx of AIGC have on recommender systems?** This research primarily focuses on AIGC in the form of high-quality **text** generated by LLMs, which are increasingly prevalent on the internet. Unlike other modalities such as images, LLM-generated text is harder to distinguish, potentially introducing more subtle biases. In the recommender system, feedback data from user interactions with LLM-generated text is

[*] Corresponding author

SIGIR '25, Padua, Italy

ACM ISBN 979-8-4007-1592-1/2025/07
https://doi.org/10.1145/3726302.3729972

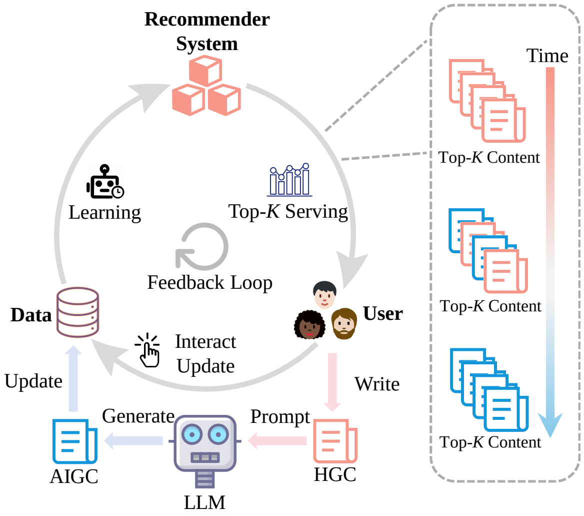


Figure 1: Model preference grows over time with the feedback loop of humans, data, and the recommender system. The red color is used for HGC icon and the blue color is used for AIGC icon. The subsequent figures use the same color scheme.

reused to update recommendation models, forming a feedback loop involving users, data, and the system as shown in Figure 1. With the continuous increase in AIGC, they will gradually influence various stages of the feedback loop, raising another research question: **RQ2: What long-term impacts will arise when AIGC further participates in the feedback loop?**

To explore the two research questions, we first examine AIGC's impact on recommender systems across three phases: HGC dominate phase, HGC-AIGC coexist phase, and AIGC dominate phase, as shown in Figure 2. These phases respectively correspond to the past, present, and future, representing the influence of different stages of AIGC flooding on recommender systems. In the HGC-dominated phase, AIGC starts influencing the candidate set and Top-K results within the feedback loop. During the HGC-AIGC coexist phase, AIGC further affects users' histories via interaction, potentially amplifying earlier effects. In the AIGC dominate phase, AIGC prevails in training, likely intensifying these influences.

To answer RQ1, we construct evaluation datasets from three domains in Amazon's product dataset by prompting LLMs to rewrite product descriptions into AIGC copies [13]. We then evaluate popular recommendation models using mixed candidate sets of HGC and AIGC items during the **HGC dominate phase**. Results show that models often rank AIGC copies higher, even when their semantics match the original HGC. This suggests product traffic can be boosted by rewriting descriptions with LLMs, while also highlighting risks—such as enabling malicious users to spread fake news [48] via LLM-generated content.

For RQ2, we conduct experiments by injecting AIGC into users' interaction histories during the **HGC-AIGC coexist phase**, and into model training data during the **AIGC-dominated phase**, within a feedback loop scenario. Results from four widely used click models show that both user behaviors and model updates on polluted data increasingly favor AIGC items. Consequently, AIGC content is ranked progressively higher, eventually reaching top positions. Moreover, as the loop progresses and AIGC proportion rises, we observe a corresponding decline in the recommendation model's performance. This further underscores the detrimental impact of excessive AIGC on recommender systems.

Based on the above findings, content creators may rewrite all descriptions to gain higher rankings, creating unfairness for other providers. Moreover, due to hallucinations [18], LLM-generated texts may contain inaccuracies, harming user experience. Prior studies and our results further suggest that training on AIGC clicked by users can cause model collapse [2, 5, 31], degrading recommendation performance. In sum, AIGC's dominance in recommender systems poses long-term risks to content fairness, user experience, and model quality. This calls for disrupting AIGC propagation in the feedback loop, leading to a new research question: **RQ3: How can the model maintain consistent preferences for both HGC and AIGC in the feedback loop?**

To answer RQ3, we first examine prior debiasing methods [8, 13, 41] and find they fail in feedback loop scenarios, unable to maintain long-term system balance. To address this, we propose a black-box debiasing method that preserves model neutrality toward both HGC and AIGC. Our approach prompts LLMs to uniformly rewrite all training data to get AIGC copies, avoiding the need to distinguish between sources. We then apply an L1 loss to constrain outputs of the item and history encoders, ensuring semantically similar HGC and AIGC are mapped to aligned embeddings. Experiments show our method effectively reduces bias and maintains prediction neutrality across varying AIGC proportions.

The major contributions of this paper are summarized as follows:

(1) We find that LLM-generated text descriptions can be ranked higher in recommender systems.

(2) We uncover that the recommendation model's preference for AIGC is gradually amplified in the feedback loop, with AIGC sequentially affecting data, users, and recommender systems.

(3) We propose a debiasing method that can effectively alleviate preference during the feedback loop by aligning the item and user embedding spaces, thereby balancing the content ecosystem.

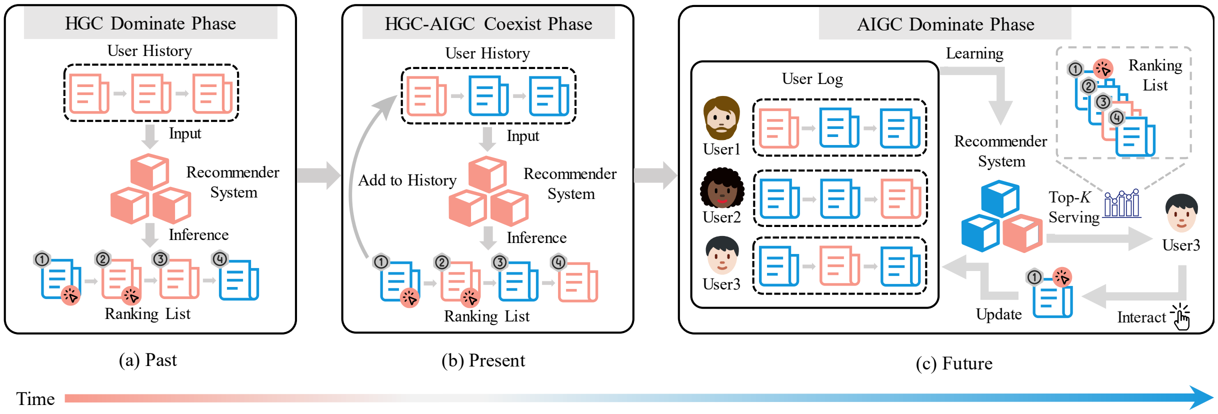
2 Preliminaries

In this section, we formulate the recommendation problem, introduce three stages of AIGC flooding into recommender systems, and explore the role of the feedback loop in propagating source bias.

2.1 Recommendation Problem Formulation

Assume that we have a set of items $\mathcal{I}$ and a set of user interaction sequences $\mathcal{S}$, where $i \in \mathcal{I}$ denotes an item and $s \in \mathcal{S}$ denotes an interaction sequence. The numbers of items and sequences are denoted as $|\mathcal{I}|$ and $|\mathcal{S}|$, respectively. Generally, the interaction sequence s is chronologically ordered with items: $\{i_1, \cdots, i_n\}$, where n is the number of interactions and i_t is the t-th item with which the user has interacted. For convenience, we use s_t to denote the subsequence, *i.e.*, $s_t = \{i_1, \cdots, i_t\}$, where $1 \leq t < n$.

Based on the above notations, we now define the task of recommendation. Formally, given the history interaction sequence of a user $s_t = \{i_1, \cdots, i_t\}$, the goal of recommendation is to train a recommendation model f_θ parameterized by θ. The model f_θ is



(a) Past

(b) Present

(c) Future

Figure 2: Three phases occur during the integration of AIGC into the recommendation content ecosystem: HGC dominate phase, HGC-AIGC coexist phase, and AIGC dominate phase. (1) The HGC dominate phase is a past period when AIGC has just flooded into the recommender systems and only influence the candidate list. (2) The HGC-AIGC coexist phase is a present period where the recommendation model's inputs s contain an increasing number of AIGC. (3) The AIGC dominate phase is a future period during which AIGC influences each stage of the feedback loop.

used to predict the next item i_{t+1} the user is likely to interact with at the $(t + 1)$-th step.

2.2 Three Phases Involving AIGC Content

After AIGC integrates into the content ecosystem of recommender systems, it will gradually impact the three processes of the feedback loop over time: Top-K serving, users' interaction, and model training. We divide the impact of AIGC on recommender systems by feedback loop over time into three phases: HGC Dominate, HGC-AIGC Coexist, and AIGC Dominate. Each phase corresponds to a real-world scenario representing the past, present, and future.

HGC Dominate Phase: With the widespread use of LLMs and the popularization of AIGC on the internet, it is easy for HGC to have corresponding AIGC copies or even be directly generated by LLMs. Thus, the items selected for the recommendation model's Top-K ranking are a combination of HGC and AIGC. In the HGC dominate phase, the research question aims to validate whether the recommendation models will rank AIGC at a higher position, a phenomenon known as source bias [8, 13, 41].

HGC-AIGC Coexist Phase: With the increasing proliferation of LLMs and AIGC on the Internet, the presence of AIGC in users' recommendation candidate lists will rapidly grow. These contents will be interacted with users and added to their interaction sequences, which will be used as input for recommendation models later. In the HGC-AIGC coexist phase, the research question is whether the model's preference for AIGC will be amplified when AIGC interacted with users is added to users' interaction sequence.

AIGC Dominate Phase: In the future, with the decreasing cost and increasing accessibility of LLMs, AIGC will dominate the ecosystem of recommender systems. Furthermore, AIGC will influence any stage of the feedback loop, namely Top-K serving, users' interaction, and model training in Figure 1. In other words, AIGC will pollute candidate list $\mathcal{I}$, users' interaction history sequence s, and the model's training data $\mathcal{S}$. Furthermore, within the iterative feedback loop, recommendation models undergo training on data $\mathcal{S}$ containing AIGC. In the AIGC dominate phase, the research question is whether the preference will be amplified when recommendation models are further trained on polluted data.

In conclusion, the integration of AIGC into the recommender system will impact various aspects, such as the candidate item set, users' interactions, and data used for model training. Based on the affected aspects, the evolution of the recommender system will progressively exhibit three phases: HGC Dominate, HGC-AIGC Coexist, and AIGC Dominate. We will explore the changes in preference across these three phases to answer RQ1 and RQ2.

3 Source Bias in Recommender Systems

In this section, we first introduce the experimental settings in Section 3.1 and then provide the data construction process and verify the AIGC quality through human evaluation in Section 3.2. In Section 3.3, we validate the existence of source bias in recommender systems during the HGC dominate phase. In Section 3.4 and Section 3.5, we verify that source bias is amplified in the feedback loop due to users' interaction behavior and the model training process.

3.1 Experimental Settings

3.1.1 Datasets. Our training and evaluation are conducted on a series of real-world datasets (Amazon [26]), comprising large corpora of product reviews and descriptions obtained from Amazon.com. Top-level product categories are treated as separate datasets, and we focus on three categories: "Health", "Beauty", and "Sports". We use the textual descriptions of products that users have commented on as input to predict which product the user might review next. Due to the low quality of short text rewriting[1], we exclude items with descriptions containing fewer than 20 words from the training set to maintain training stability. We sort the data based on the review time of the target item and split it into training and testing sets in a 7:3 ratio. In the training dataset, we exclude users and items

[1] LLM frequently expands short texts during the rewriting process, leading to semantic

Table 1: Statistics of the experimental datasets.

Dataset	Health	Beauty	Sports
# Users	18,036	11,391	16,639
# Items	13,972	11,897	13,089
# Click Behaviors	346,355	198,502	296,337

with fewer than five interactions and randomly select 4 negative items from the entire set for each product reviewed by users. The statistics of datasets after processing are shown in Table1.

3.1.2 Recommendation Models. For our main experiments, we select four representative models: BERT4Rec [32], SASRec [21], GRU4Rec [16], LRURec [43]. These models are enhanced by various pre-trained language models, including BERT [14] and RoBERTa [25]. Our focus on sequential recommendation models is motivated by their widespread use in current industrial recommender systems. We input the product's textual description into these pretrained models and use the average pooled embedding of the outputs from the pretrained models as the item embedding. Item embedding is then used as the input for the four sequence recommendation models mentioned above. We use the `bert-base-uncased` checkpoint for BERT and the `roberta-base` checkpoint for RoBERTa.

3.1.3 Evaluation Metrics. To evaluate the ranking performance of the recommendation models, we compute the Top-K Normalized Discounted Cumulative Gain (NDCG@K) and Mean Average Precision (MAP@K) separately for HGC and AIGC items, where $K \in \{3, 5\}$. To further measure the recommendation models' preferences for different source texts, the candidates during testing are divided into two parts: one part consists of original HGC, and the other part consists of copies of AIGC. To get a simple and efficient measuring way, we utilize the relative percentage difference [8, 13, 41]:

$$\text{Relative } \Delta = \frac{\texttt{Metric}_{\text{HGC}} - \texttt{Metric}_{\text{AIGC}}}{(\texttt{Metric}_{\text{HGC}} + \texttt{Metric}_{\text{AIGC}})/2} \times 100\%, \tag{1}$$

where $\texttt{Metric}_{\text{HGC}}$ and $\texttt{Metric}_{\text{AIGC}}$ are calculated on the same candidate set comprising both HGC and AIGC. For a given metric (either NDCG@K or MAP@K), when measuring the metric for one data source, we set the labels of the other data source to 0. Relative $\Delta > 0$ indicates a preference of the recommendation models towards HGC, while Relative $\Delta < 0$ indicates a preference towards AIGC. The greater the absolute value of Relative Δ, the stronger the preference recommendation model for AIGC or HGC.

3.1.4 Experimental Details. To ensure computational efficiency, all pre-trained language models are frozen. All recommendation models are trained for 5 epochs, and the best-performing model on the development set is selected for testing on the test set. The batch size is set to 128, the learning rate is set to 1e-3, and the Adam optimizer is used for training. The dimension of item vectors is set to 768, and all score calculations utilize the dot function. The text input to the model is truncated to 512 tokens, and the user's historical sequence is limited to 10 interactions. To ensure reproducibility, we run each experiment with five different seeds and report the averaged results.

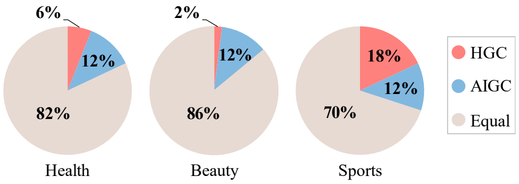


Figure 3: Quality verification of the constructed datasets through human evaluation.

3.2 AIGC Data Construction and Verification

3.2.1 Data Construction. Following the setting in previous works [8, 13, 41], we reconstruct the dataset from Amazon to evaluate source bias in recommender systems. For each item $i \in \mathcal{I}$, we utilize the same rewriting prompt "*Please rewrite the following text: {{human-written text}}*" to empower LLMs to produce text without extra constraints, all the while upholding semantic equivalence to the initial HGC. Specifically, we chose some popular LLMs ChatGPT (*i.e.,* `gpt-3.5-turbo-0613`), Llama (*i.e.,* `llama-2-7b-chat`) [34], Mistral (*i.e.,* `Mistral-7B-Instruct-v0.2`) [20], and Gemini-pro (*i.e.,* `Gemini 1.5 Pro`) [33] to rewrite each seed HGC, as these LLMs are the most widely used. The temperature of all LLMs for generation is set at 0.2 and the maximum generation length is 256.[2]

After rewriting, we can obtain HGC data and the corresponding AIGC data for each dataset. Formally, we have two sets of items denoted by $\mathcal{I}^H$ and $\mathcal{I}^G$, respectively. Here, $i^H \in \mathcal{I}^H$ represents an item written by a human, while $i^G \in \mathcal{I}^G$ represents an item generated by LLMs. Each item i^H has its corresponding AIGC copy $i^G \in \mathcal{I}^G$. In the LLMs era, the task of recommendation is to predict the next item i_{t+1} the user is likely to interact with from a mixed set of items $\mathcal{I} = \mathcal{I}^H \cup \mathcal{I}^G$, rather than just $\mathcal{I}^H$.

3.2.2 Human Evaluation. To validate that the rewritten data does not affect users' interaction behaviors, we conduct a human evaluation study by sampling 50 triples from the Health, Beauty, and Sports, respectively. For each domain, we recruit three colleagues for data annotation. Each human annotator is asked to indicate which item they would be more inclined to purchase based on the textual description of products in the browsing purchase history, with options being "Human items", "LLM items" and "Equal". Each triple is annotated by at least three annotators, and the votes determine the final label. The evaluation results in Figure 3 demonstrate the consistency of humans' interaction behaviors on HGC and AIGC, providing reliable assurance for the evaluation and analysis of source bias.

3.3 Preference in HGC Dominate Phase

In this subsection, we examine the recommendation models during the HGC dominate phase, aiming to explore whether AIGC will be ranked higher. We train recommendation models on each dataset with items from $\mathcal{I}^H$ and test the model's performance on candidate items from $\mathcal{I}^H \cup \mathcal{I}^G$. As shown in Table 2, it can be observed that most recommendation models exhibit preference for AIGC in terms of metrics such as NDCG@K and MAP@K. An important point

[2] The examples of rewritten text can be found in the supporting materials.

Table 2: Performance comparison of recommendation models based on BERT and RoBERTa for mixed HGC and AIGC item sets on the Health, Beauty, and Sports dataset. Relative $\Delta < 0$ indicates that the recommendation models rank AIGC higher than HGC, while Relative $\Delta > 0$ indicates that the models rank HGC higher than AIGC. The absolute value of Relative Δ indicates the degree of bias, with a larger value representing a stronger bias. Unless otherwise stated, AIGC will be generated using ChatGPT with BERT as the encoder model, and Relative Δ is calculated based on NDCG@5.

PLM	Model	Corpus	Health				Beauty				Sports			
			NDCG@3	NDCG@5	MAP@3	MAP@5	NDCG@3	NDCG@5	MAP@3	MAP@5	NDCG@3	NDCG@5	MAP@3	MAP@5
BERT	GRU4Rec	Human-Written	32.77	41.18	28.39	33.06	27.92	36.88	23.80	28.79	31.80	41.12	26.75	31.97
		LLM-Generated	41.28	48.74	36.92	41.03	44.21	52.41	39.54	44.09	51.73	58.92	47.03	51.00
		Relative Δ	-22.99	-16.80	-26.13	-21.51	-45.18	-34.78	-49.71	-42.01	-47.74	-35.60	-54.98	-45.87
	SASRec	Human-Written	24.47	32.74	20.54	25.11	23.80	32.27	20.32	25.00	25.45	34.91	21.39	26.62
		LLM-Generated	39.82	47.60	35.80	40.10	34.51	43.38	30.28	35.18	44.56	52.73	39.95	44.48
		Relative Δ	-47.76	-36.99	-54.16	-45.96	-36.73	-29.38	-39.34	-33.85	-54.58	-40.68	-60.50	-50.24
	BERT4Rec	Human-Written	26.49	35.22	22.88	27.70	21.53	30.31	18.42	23.27	26.00	35.54	22.06	27.37
		LLM-Generated	32.89	40.97	28.67	33.12	35.51	43.52	31.25	35.67	40.46	49.12	36.17	40.95
		Relative Δ	-21.57	-15.09	-22.47	-17.85	-49.00	-35.81	-51.66	-42.09	-43.49	-32.06	-48.46	-39.77
	LRURec	Human-Written	34.22	42.13	30.13	34.51	31.30	39.26	27.26	31.66	34.12	42.31	29.55	34.09
		LLM-Generated	32.29	40.22	28.33	32.69	38.30	46.24	33.84	38.24	40.84	49.40	36.10	40.86
		Relative Δ	-20.13	5.80	4.65	6.16	5.43	-16.35	-21.54	-18.82	-17.92	-15.46	-19.93	-18.05
RoBERTa	GRU4Rec	Human-Written	30.96	39.01	26.52	31.01	36.64	45.07	32.35	36.98	40.54	48.72	35.88	40.43
		LLM-Generated	44.10	50.98	39.48	43.26	34.61	43.58	30.00	34.98	43.43	50.82	38.61	42.72
		Relative Δ	-35.01	-26.60	-39.27	-32.97	5.69	3.36	7.55	5.55	-6.88	-4.20	-7.33	-5.51
	SASRec	Human-Written	24.54	32.53	20.81	25.25	20.74	29.56	17.74	22.59	17.79	24.70	15.31	19.12
		LLM-Generated	38.63	46.76	34.32	38.78	27.87	36.93	24.24	29.26	26.57	35.65	23.00	28.00
		Relative Δ	-44.62	-35.91	-49.02	-42.26	-29.32	-22.17	-30.94	-25.70	-39.57	-36.30	-40.15	-37.69
	BERT4Rec	Human-Written	29.37	37.60	25.40	29.97	26.29	34.86	22.84	27.59	31.14	39.98	27.15	32.05
		LLM-Generated	40.02	47.20	35.55	39.52	29.95	38.28	25.81	30.40	39.26	47.96	34.60	39.45
		Relative Δ	-30.68	-22.65	-33.30	-27.48	-13.02	-9.35	-12.22	-9.68	-23.06	-18.14	-24.16	-20.68
	LRURec	Human-Written	25.35	34.33	21.19	26.15	33.06	40.23	29.40	33.36	30.24	39.49	26.30	31.42
		LLM-Generated	44.21	51.24	39.71	43.59	30.37	39.95	25.95	31.26	39.32	48.00	34.55	39.36
		Relative Δ	-54.21	-39.51	-60.80	-50.01	8.46	0.70	12.46	6.50	-26.11	-19.47	-27.12	-22.44

is that the higher metric on LLM-Generated compared to Human-Written does not imply better ranking performance on AIGC. When measuring, the candidate set is $I^H \cup I^G$, with each HGC having a AIGC-copy. For Human-Written, AIGC's positive samples are treated as negative, focusing only on HGC's positive item. Thus, the higher score on LLM-Generated only reflects HGC have a lower ranking score than its AIGC-copy.

To verify the widespread presence of preference in recommender systems, we test recommendation models on AIGC generated by more popular LLMs such as Llama, Mistral, and Gemini-Pro. The results in Table 3 indicate the varying degrees of preference on AIGC generated by different LLMs, confirming the prevalence and significance of preference. Furthermore, ChatGPT demonstrates a smaller preference compared to other LLMs, likely due to its better alignment with human behavior during pre-training.

Finding 1: During the HGC dominate phase, various recommendation models based on different PLMs tend to show a preference for AIGC generated by various LLMs across three datasets from diverse domains.

3.4 Source Bias in HGC-AIGC Coexist Phase

In this subsection, we validate the recommendation models during the HGC-AIGC coexist phase, which aims to explore whether preference will be amplified with the number of users' interaction on AIGC. When AIGC is further integrated into the recommender systems, users will interact with both HGC and AIGC. These items will be added to users' interaction history sequences, influencing the output of the recommendation models. In order to simulate this process, we train recommendation models on each dataset using items from $\mathcal{I}^H$. When testing, we vary the proportion of AIGC in users' interaction history sequence $s = \{i_1, \cdots, i_n\}$. For $i_t \in s$, it originates from $\mathcal{I}^H$ with probability p and from $\mathcal{I}^G$ with probability $1 - p$ where p ranges from 0 to 1 in intervals of 0.1. This allows us to simulate the impact of users' interactions on AIGC on the preference at different levels of AIGC propagation.

The results, as shown in Figure 4, indicate that the preference for AIGC of all sequential recommendation models increases as the proportion of AIGC in the historical sequence increases across the three datasets. While the extent of preference exhibited by the same model varies across different datasets, they all show the same trend: the more AIGC the user interact with, the more pronounced the preference phenomenon becomes in recommender systems.

Finding 2: In the feedback loop, the more users interact with AIGC, the model will recommend more AIGC in Top-K serving, thereby amplifying the preference.

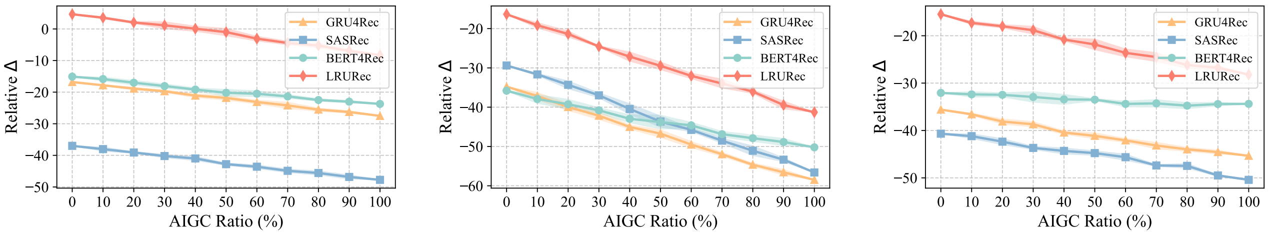
3.5 Preference in AIGC Dominate Phase

In this subsection, we validate the recommendation models during the AIGC dominate phase, which aims to explore whether preference will be further amplified with AIGC items participating in model training with the feedback loop. When AIGC dominates the recommender ecosystem in the future, it will influence any stage of the feedback loop, namely Top-K serving, interaction, and training as shown in Figure 1, corresponding to the candidate list $\mathcal{I}$, users' interaction history sequence s, and the model's training data $\mathcal{S}$. To investigate the changing trend of preference during the AIGC dominate phase, we will construct a realistic scenario involving users' interactions. In this scenario, users are more inclined to interact

Table 3: Relative Δ of recommendation models with AIGC copies generated by ChatGPT, Llama2, Mistral, and Gemini-Pro.

PLM	Model	Health				Beauty				Sports			
		ChatGPT	Llama2	Mistral	Gemini-Pro	ChatGPT	Llama2	Mistral	Gemini-Pro	ChatGPT	Llama2	Mistral	Gemini-Pro
BERT	GRU4Rec	-16.80	-	-18.3	-23.7	-34.78	-37.91	-33.83	-47.62	-35.60	-24.14	-49.58	-43.23
	SASRec	-36.99	-	-46.07	-50.66	-29.38	-36.57	-30.82	-62.06	-40.68	-31.7	-53.36	-56.17
	BERT4Rec	-15.09	-	-9.093	-22.69	-35.81	-23.94	-18.75	-35.93	-32.06	-2.134	-38.23	-46.35
	LRURec	4.65	-	-10.83	-11.70	-16.35	-27.11	-13.32	-31.57	-15.46	-44.10	-35.7	-36.12
RoBERTa	GRU4Rec	-26.60	-	-20.21	-27.27	3.36	-28.39	-16.50	-35.77	-4.20	-2.685	-19.71	-19.32
	SASRec	-35.91	-	-41.5	-52.13	-22.17	-42.13	-12.58	-25.95	-36.30	-81.06	-58.85	-54.86
	BERT4Rec	-22.65	-	-27.44	-40.10	-9.35	-44.29	-20.79	-52.88	-18.14	-18.79	-30.30	-41.18
	LRURec	-39.51	-	-41.66	-49.38	0.70	-13.73	-7.91	-23.23	-19.47	-32.55	-39.02	-37.94

Note: We omit the result for Health dataset as Llama2 refuses to rewrite 97.7% of the product description due to that Health contains sensitive information.



(a) Health (b) Beauty (c) Sports

Figure 4: Relative Δ of recommendation models is depicted along with its 95% confidence interval, shown with error bars. X-axis represents the ratio of AIGC in users' interaction sequence.

with items positioned higher for model training and testing rather than merely mixing the data proportions.

Specifically, we will first train recommendation models with items from $\mathcal{I}^H$ and use the trained models to simulate users' interactions on item sets $\mathcal{I}^H \cup \mathcal{I}^G$. To simulate users' behavior, the position-based click model (PBM) [1, 28] is used, where a interaction is registered only when the item is viewed and is relevant. Here, $E = 1$ indicates that an item is examined by a user. For each impression, the likelihood of examination is determined by the position in the list of candidate items [45]:

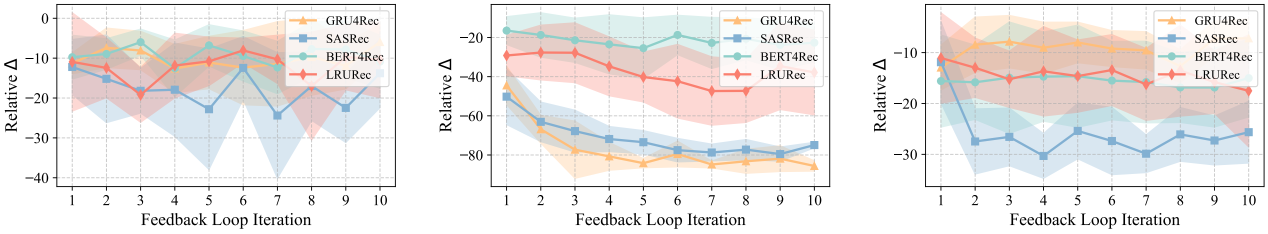
$$P\big(E = 1|\text{rank}(i) = k\big) = k^{-\eta}, \tag{2}$$

where η represents the hyper-parameter that controls the severity of position bias, and rank(i) is the rank position of item i in the candidate item list. After obtaining users' interaction results, we use the proportion of users' interactions with AIGC to adjust the proportion of AIGC items in the interaction sequence s, as mentioned in Section 3.4, and then retrain the model using the aforementioned simulated interaction results and the mixed historical sequence. In the above training process, we iterate 10 times, assessing the preference level in each model at each iteration. During testing, the proportion of AIGC in users' interaction history matches the proportion used in training for that iteration. The complete training process is provided in Algorithm 1.

Under the condition of $\eta = +\infty$, we test the Relative Δ at different iterations of the feedback loop. The results in Figure 6 show that the absolute Relative Δ of all models increases with each iteration until it converges to a value near the end. This suggests that without intervening in the model's preference for AIGC, this preference will amplify with each feedback loop, ultimately leading to an AIGC-dominated content ecosystem. It is worth noting that on Beauty dataset, the models do not initially exhibit preference with Relative $\Delta > 0$. However, it still emerges and amplifies as the feedback loop progresses, further indicating the ubiquity of preference even if the model initially does not show a preference for AIGC. Additionally, we also record the performance changes of the model after the first loop and after the 20th loop. Results in Table 4 show that an excessively high proportion of AIGC not only disrupts the content

Algorithm 1: Feedback Loop for Model Training

```
Input: Interaction dataset $\mathcal{S}$; number of feedback loop
       iterations $E$; parameters $\eta$, $p$
Output: Trained models $f_\theta^1, f_\theta^2, \cdots f_\theta^E$
$p \leftarrow 0$
$\mathcal{S}^e \leftarrow \mathcal{S}$
for $e = 1, \cdots, E$ do
    Train model $f_\theta^e$ on dataset $\mathcal{S}^e$
    $\mathcal{S}^e \leftarrow \{\}$
    for $(s_t, i_{t+1})$ in $\mathcal{S}$ do
        $s_t \leftarrow \{i^L \mathbf{1}(\text{Bernoulli}(p^{e-1}) = 1) + i^H \mathbf{1}(\text{Bernoulli}(p^{e-1}) \neq 1) : i \in s_t\}$
        Get users' interaction probabilities $\mathcal{Y}$ with Eq. (2)
        Sample users' interaction item $i_{t+1}$ from $\mathcal{I}$ with $\mathcal{Y}$
        $\mathcal{S}^e \leftarrow \mathcal{S}^e \cup (s_t, i_{t+1})$
    Update $p$ with probability of $i_{t+1}$ from $\mathcal{I}^G$
return $f_\theta^1, f_\theta^2, \cdots, f_\theta^E$
```



(a) Health with LR

(b) Health with NCM

(c) Health with TNCM

Figure 5: Comparison of Relative Δ for recommendation models under different click model settings across feedback loop iterations (X-axis), with 95% confidence intervals represented by error bars.

ecosystem but also leads to a decline in model performance.

Although PBM is widely used and more realistic, we also implement three other training-based click models—LR, NCM, and TNCM [30]—for a more comprehensive evaluation. The results in Figure 5 show similar phenomena as with PBM. Additionally, the model bias amplification through the feedback loop is more pronounced in these realistic neural models, emphasizing the inevitability of the issue. For simplicity and ease of comparison, we select PBM for the subsequent experiments.

Finding 3: Finally, by introducing AIGC pollute the feedback loop, including Top-K serving, users' interactions, and model training, the preference will be pushed to the top.

4 Debias During the Feedback Loop

In previous sections, we validate the presence of bias in recommender systems. Furthermore, with the proliferation of AIGC on the internet, bias amplifies throughout the feedback loop, thereby causing long-term impacts on the content ecosystem. Therefore, we need to eliminate the model's preference for AIGC. Although previous work [13, 41] has attempted to address this bias, they do not account for the feedback loop inherent in real-world scenarios. In these settings, the margin loss used in their methods causes the model to ultimately favor HGC, leading to the collapse of the ecosystem with a dominance of HGC content. In this paper, we propose a new approach relying on L1 loss that effectively tackles this issue, ensuring a more stable and balanced content ecosystem.

Specifically, for each i in $\mathcal{I}$, regardless of whether it originates from $\mathcal{I}^H$ or $\mathcal{I}^G$, its corresponding rewriting copy i' in $\mathcal{I}'$ is derived from the rewriting process of LLM as described in Section 3.2. In this way, we obtain the original training data triple (s_t, i_{t+1}, i'_{t+1}) for feedback loop training. We utilize the L1 loss function to calculate the difference in scores between i_{t+1} and i'_{t+1} as:

$$\mathcal{L}_{\text{Debias-I}} = \sum_{s \in \mathcal{S}} \sum_{t=1}^{n-1} \left| f_\theta(s_t, i'_{t+1}) - f_\theta(s_t, i_{t+1}) \right|, \quad (3)$$

which can eliminate the additional score introduced by the LLM rewriting process compared to the user interaction sequence s. Hence, it can be incorporated as a component of the loss function to alleviate the bias. What's more, for each item i in the user interaction sequence s, we can obtain its rewritten copy s' by replacing each item i with corresponding i'. Again, we utilize the L1 loss function to calculate the difference in scores between s and s' in comparison to candidate item i. Furthermore, in addition to aligning the embedding representations of user interaction sequences s and s' before and after rewriting, we aim to minimize the entropy $\mathbb{H}$ of the embedding representation for each interaction sequence s and s'. This ensures that the embedding representations $\text{Emb}(s)$ generated from different s composed of different items i move farther away from each other. The debiasing loss for the history encoder side can be expressed as follows:

Table 4: Performance (NDCG@3) of recommendation models on different iteration of feedback loop. "Iter=1" and "Iter=20" indicate the 1st and 20th iterations of the feedback loop.

Model	Health		Beauty		Sports	
	Iter=1	Iter=20	Iter=1	Iter=20	Iter=1	Iter=20
GRU4Rec	56.60	42.50 -14.10	60.18	43.73 -16.45	58.33	47.67 -10.66
SASRec	40.26	37.08 -3.18	50.53	36.44 -14.09	44.19	38.00 -6.19
BERT4Rec	42.88	35.78 -7.10	42.92	35.76 -7.16	39.51	35.98 -3.53
LRURec	43.00	37.85 -5.15	52.34	39.58 -12.76	50.64	44.28 -6.34

$$\mathcal{L}_{\text{Debias-U}} = \sum_{s \in \mathcal{S}} \sum_{t=1}^{n-1} \left| f_\theta(s'_t, i_{t+1}) - f_\theta(s_t, i_{t+1}) \right| + \mathbb{H}(\text{Softmax}(\text{Emb}(s'_t))) + \mathbb{H}(\text{Softmax}(\text{Emb}(s_t))), \quad (4)$$

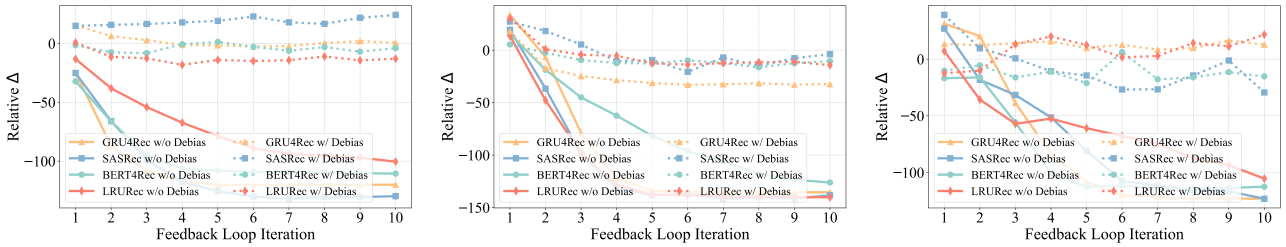
which can measure the additional score resulting from the history encoder's preference for user interaction sequence s' combined with AIGC item, in comparison to item i. Therefore, this can also be used as part of the loss function to mitigate the bias caused by the history encoder. Based on the additional constraints defined in Eq. (3) and Eq. (4), we can define the final loss for model training:

$$\mathcal{L} = \mathcal{L}_{\text{ranking}} + \alpha \mathcal{L}_{\text{Debias-I}} + \beta \mathcal{L}_{\text{Debias-U}}, \quad (5)$$

where $\mathcal{L}_{\text{ranking}}$ can be either contrastive loss or regression loss. α and β are the debiasing coefficients that can balance the recommendation performance and the level of the bias. The larger coefficient indicates a greater penalty on the biased samples, which may result in a decrease in the recommendation performance.

4.1 Experimental Results

Figure 6 illustrates the Relative Δ of the model and the debiasing model at different iterations of the feedback loop. The dashed line represents the model with our proposed method. Compared to previous methods, our approach focuses solely on the differences before and after rewriting, which enables us to continuously achieve



(a) Health (b) Beauty (c) Sports

Figure 6: Comparison of Relative Δ for recommendation models across feedback loop iterations (X-axis) on different datasets.

Table 5: Performance of recommendation models after feedback loop. “*w/o* Debias” refers to a model without our debiasing method, while “*w/* Debias” refers to one with it.

Model	Health		Beauty		Sports	
	w/o Debias	Debias	*w/o* Debias	Debias	*w/o* Debias	Debias
GRU4Rec	68.59	72.71 +4.12	68.44	71.61 +3.17	68.79	73.10 +4.32
SASRec	64.72	65.76 +1.04	65.62	65.75 +0.13	64.37	63.34 -1.03
BERT4Rec	63.82	63.28 -0.54	64.15	63.60 -0.55	62.20	60.55 -1.65
LRURec	65.37	66.73 +1.36	66.59	68.31 +1.72	64.60	64.63 +0.03

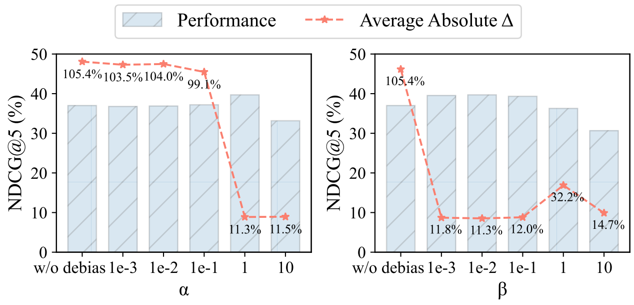


Figure 7: Performance and Average Absolute Δ of recommendation models with different coefficients α and β in our proposed debiasing method.

debiasing effects in iterative feedback, resulting in a smaller absolute Relative Δ. Furthermore, the dashed line remaining within a stable range also demonstrates that L1 loss can prevent the model from biasing towards either HGC or AIGC during the dynamic process.

Table 5 presents the ranking performance of models on all datasets . Our debiasing method not only eliminates biases but also enhances model performance in most cases. Suggested by these findings it can be found that introducing AIGC samples during the debiasing process of AIGC appears to enhance the model's capacity to differentiate between similar items.

4.2 Further Analysis

4.2.1 Performance w.r.t. the Coefficients α and β. As shown in Eq.(5), our debiasing method uses coefficients α and β to balance the ranking loss and debiasing loss, achieving a trade-off between model performance and bias reduction. In the experiment, we vary α and β within the range $\{1e\text{-}3, 1e\text{-}2, 1e\text{-}1, 1, 10\}$, while fixing the other coefficient at the value that yields the best recommendation performance. The Average Absolute Δ represents the average absolute value of the Relative Δ during the feedback loop. Models trained without debiasing constraints are labeled as “*w/o* debias”.

Table 6: Relative Δ on Health with varying AIGC settings: "ChatGPT" refers to AIGC generated solely by ChatGPT. "Mixed" refers to AIGC from multiple LLMs, with Llama3 used for debiased texts in both "Mixed" and "ChatGPT."

Model	NDCG@5			MAP@5		
	w/o Debias	Mixed	ChatGPT	*w/o* Debias	Mixed	ChatGPT
GRU4Rec	-121.93	-4.66	2.10	-138.23	-6.44	3.38
SASRec	-122.11	5.19	-29.32	-136.92	7.74	-35.57
BERT4Rec	-109.10	12.84	15.17	-123.43	15.08	18.23
LRURec	-120.44	-7.80	-25.93	-135.31	-7.22	-30.93

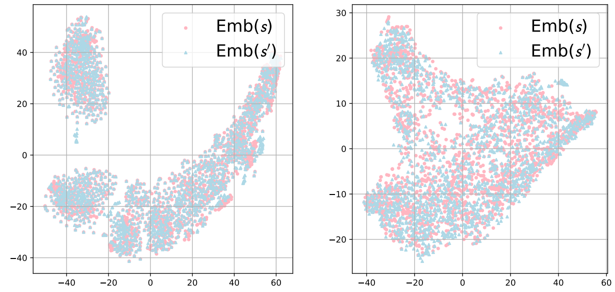
The results on Figure 7 show that as α increases, the Average Absolute Δ decreases, indicating improved bias mitigation. The model also maintains ranking performance and outperforms the model without debiasing constraints. This improvement is likely due to the inclusion of AIGC samples, which may enhance the model's ability to distinguish relevance. However, when α becomes too large, performance declines, possibly because $\mathcal{L}_{\text{Debias-I}}$ shifts the focus too much on distinguishing HGC from AIGC, neglecting ranking. A similar trend is observed with β: increasing β leads to performance degradation, likely because forcing interaction sequences to be closer disrupts the model's ranking capability.

4.2.2 Ablation Study. In this experiment, we investigate whether the two proposed components of loss, $\mathcal{L}_{\text{Debias-U}}$ and $\mathcal{L}_{\text{Debias-I}}$, can effectively eliminate the source bias. We conduct experiments to evaluate the Average Absolute Δ on the models trained only on our debiasing method without $\mathcal{L}_{\text{Debias-U}}$ and $\mathcal{L}_{\text{Debias-I}}$, denoted as “*w/o* Debias-U” and “*w/o* Debias-I”, respectively. The results in Figure 7 show that the Average Absolute Δ of the model improves across all models. After removing all debiasing constraints except for the “*w/o* Debias-U model” with SASRec implementation, the Average Absolute Δ increases. This observation confirms the effectiveness of constraining the item encoder and user encoder in our proposed loss function. Meanwhile, the $\mathcal{L}_{\text{Debais-I}}$ loss is more effective compared to the $\mathcal{L}_{\text{Debais-U}}$ loss may result from the fact that debiasing directly on the items used for evaluation is more straightforward.

4.2.3 More Realistic Debiasing Setting. In the previous setting, the AIGC used in the experiments is generated by ChatGPT. However, in real-world scenarios, there are many different types of LLMs. To better validate the effectiveness of our debiasing method in a real-world setting, we will use ChatGPT, Llama, Mistral, and Gemini to

Table 7: Average Absolute Δ of recommendation models with various debiasing method variants. "*w/o* Debias-U" refers to the model trained without $\mathcal{L}_{\text{Debias-U}}$, while "*w/o* Deias-I" refers to the model trained without $\mathcal{L}_{\text{Debias-I}}$.

Model	NDCG@5			MAP@5		
	Debias	*w/o* Debias-U	*w/o* Debias-I	Debias	*w/o* Debias-U	*w/o* Debias-I
GRU4Rec	6.45	8.07 +1.62	130.00 +123.55	7.12	9.65 +2.53	142.89 +135.77
SASRec	21.69	18.43 -3.26	124.80 +103.11	24.39	20.84 -3.55	135.84 +115.00
BERT4Rec	8.18	8.65 +0.47	25.86 +17.68	8.28	9.84 +1.56	29.40 +19.56
LRURec	8.97	29.72 +20.75	122.30 +113.33	11.34	32.74 +21.4	131.93 +120.59



(a) *w/o* $\mathbb{H}(\text{Emb}(s))$ Maximized (b) *w/* $\mathbb{H}(\text{Emb}(s))$ Maximized

Figure 8: User history embedding visualization of GRU4Rec trained without and with $\mathbb{H}$(Emb(s)) on Health dataset.

generate AIGC, and Llama3 to rewrite the text for debiasing ("Mixed" setting). We also use AIGC texts all generated by ChatGPT and use Llama3 for debiasing to simulate an extreme scenario ("ChatGPT" setting). In Table 6, whether AIGC is generated using a mix of LLMs or solely ChatGPT, after applying our debiasing method, the model achieves a smaller absolute value of Relative Δ compared to over 100 before debiasing. This means that a single LLM can correct biases in complex environments with AIGC generated by multiple LLMs. Moreover, our method only requires a one-time rewrite of the corpus by LLM and computation of AIGC-copy embeddings by PLM during training, with no additional cost during inference, making it practical for real-world use.

4.2.4 Visualize of Interaction Sequence Representation. As shown in Figure 8, we visualize the interaction sequence representation **Emb**(s) of models with various debiasing constraints using T-SNE [35], in which the models are denoted as *w/o* $\mathbb{H}(\text{Emb}(s))$ and *w/* $\mathbb{H}(\text{Emb}(s))$ to indicate whether the term maximizing user embedding entropy is included in $\mathcal{L}_{\text{Debias-U}}$. Both types of debiasing constraints on Emb(s) can maintain the mapping representation of historical sequences before and after rewriting. However, our proposed debiasing constraints—minimizing both entropy $\mathbb{H}$ and the distance between s and s'—encourage a more uniform distribution of user history embeddings. This prevents different histories from collapsing into the same representation, preserving ranking performance while aligning AIGC and HGC sequences.

5 Related Work

Large Language Models for Recommender Systems. Recent advancements in LLMs have attracted considerable interest among researchers to leverage these models [23, 24, 39, 47] to develop an enhanced recommender system. Some works utilize LLMs to generate knowledge-rich texts or use LLM-derived embeddings to enhance recommender systems, known as LLM-enhanced recommender systems [27, 37, 40]. Another line of work leverages LLMs that act as the ranking model to approach recommendation tasks, known as LLM-as-recommenders [4, 9, 17]. In addition to exploring how recommender systems can benefit from LLMs, we also need to consider the potential challenges that the development of LLMs may pose to recommender systems [10, 11]. Distinguished from these works, our study primarily investigates the impact of AIGC content on recommender systems, specifically focusing on the changes and influences of source bias in the feedback loop of recommender systems.

Effects of Artificial Intelligence Generated Content. The rise of large language models (LLMs) has accelerated the spread of AIGC (AI-generated content), bringing broad societal and technological impacts [6, 10, 11, 38]. AIGC raises concerns such as misinformation [7], harmful content [19], and even performance degradation in future models when used for training [2, 5, 31]. Recent studies also reveal that neural retrieval models tend to favor AIGC, ranking it higher in text [8, 13], image [41], and video retrieval [15]—a phenomenon known as source bias. Wang et al. [36] attributes this to the lower perplexity of AIGC, which aligns better with PLM-based retrieval models. While most work addresses this bias on the retrieval side, Dai et al. [12] instead leverages retriever feedback to construct preference data for LLM debiasing. In contrast, our study explores source bias in recommender systems, where AIGC affects not only model outputs and user behavior, but also future training data. This forms a feedback loop that can reinforce and amplify the bias over time.

6 Conclusion

In this paper, we delve into exploring the effect of AIGC in recommender systems. Through extensive experiments with several representative recommendation models across three datasets from different domains, we uncover the prevalence of preference for AIGC in recommender systems. Furthermore, we validate that the preference is gradually amplified in the feedback loop, where AIGC will be incorporated into users' interaction histories and the training data as time progresses. To mitigate preference and prevent its further amplification in the feedback loop, we propose a black-box debiasing solution that ensures the impartiality of the model prediction towards both HGC and AIGC in the feedback loop.

Acknowledgments

This work was funded by the National Key R&D Program of China (2023YFA1008704), the National Natural Science Foundation of China (No. 62472426, 62276248), Engineering Research Center of Next-Generation Intelligent Search and Recommendation, Ministry of Education, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, fund for building world-class universities (disciplines) of Renmin University of China and the Youth Innovation Promotion Association CAS under Grants No.2023111. This work was supported by the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China (RUC24QSDL013).

References

[1] Qingyao Ai, Tao Yang, Huazheng Wang, and Jiaxin Mao. 2021. Unbiased learning to rank: online or offline? *ACM Transactions on Information Systems (TOIS)* 39, 2 (2021), 1–29.

[2] Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard G Baraniuk. 2023. Self-consuming generative models go mad. *arXiv preprint arXiv:2307.01850* 4 (2023), 14.

[3] Hossein Bahak, Farzaneh Taheri, Zahra Zojaji, and Arefeh Kazemi. 2023. Evaluating chatgpt as a question answering system: A comprehensive analysis and comparison with existing models. *arXiv preprint arXiv:2312.07592* (2023).

[4] Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 1007–1014.

[5] Martin Briesch, Dominik Sobania, and Franz Rothlauf. 2023. Large language models suffer from their own output: An analysis of the self-consuming training loop. *arXiv preprint arXiv:2311.16822* (2023).

[6] Yihan Cao, Siyu Li, Yixin Liu, Zhiling Yan, Yutong Dai, Philip S Yu, and Lichao Sun. 2023. A comprehensive survey of ai-generated content (aigc): A history of generative ai from gan to chatgpt. *arXiv preprint arXiv:2303.04226* (2023).

[7] Canyu Chen and Kai Shu. 2024. Combating misinformation in the age of llms: Opportunities and challenges. *AI Magazine* 45, 3 (2024), 354–368.

[8] Sunhao Dai, Weihao Liu, Yuqi Zhou, Liang Pang, Rongju Ruan, Gang Wang, Zhenhua Dong, Jun Xu, and Ji-Rong Wen. 2024. Cocktail: A Comprehensive Information Retrieval Benchmark with LLM-Generated Documents Integration. *Findings of the Association for Computational Linguistics: ACL 2024* (2024).

[9] Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. 2023. Uncovering chatgpt's capabilities in recommender systems. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 1126–1132.

[10] Sunhao Dai, Chen Xu, Shicheng Xu, Liang Pang, Zhenhua Dong, and Jun Xu. 2024. Bias and Unfairness in Information Retrieval Systems: New Challenges in the LLM Era. (2024), 6437–6447.

[11] Sunhao Dai, Chen Xu, Shicheng Xu, Liang Pang, Zhenhua Dong, and Jun Xu. 2025. Unifying Bias and Unfairness in Information Retrieval: New Challenges in the LLM Era. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*. 998–1001.

[12] Sunhao Dai, Yuqi Zhou, Liang Pang, Zhuoyang Li, Zhaocheng Du, Gang Wang, and Jun Xu. 2025. Mitigating Source Bias with LLM Alignment. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[13] Sunhao Dai, Yuqi Zhou, Liang Pang, Weihao Liu, Xiaolin Hu, Yong Liu, Xiao Zhang, Gang Wang, and Jun Xu. 2024. Neural Retrievers are Biased Towards LLM-Generated Content. *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2024).

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*. 4171–4186.

[15] Haowen Gao, Liang Pang, Shicheng Xu, Leigang Qu, Tat-Seng Chua, Huawei Shen, and Xueqi Cheng. 2025. Generative Ghost: Investigating Ranking Bias Hidden in AI-Generated Videos. *arXiv preprint arXiv:2502.07327* (2025).

[16] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[17] Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2024. Large language models are zero-shot rankers for recommender systems. In *European Conference on Information Retrieval*. Springer, 364–381.

[18] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems* 43, 2 (2025), 1–55.

[19] Yangsibo Huang, Samyak Gupta, Mengzhou Xia, Kai Li, and Danqi Chen. 2024. Catastrophic Jailbreak of Open-source LLMs via Exploiting Generation. In *12th International Conference on Learning Representations, ICLR 2024*.

[20] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7B. *arXiv* (2023).

[21] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.

[22] Viet Lai, Nghia Ngo, Amir Pouran Ben Veyseh, Hiêu Mân, Franck Dernoncourt, Trung Bui, and Thien Nguyen. 2023. ChatGPT Beyond English: Towards a Comprehensive Evaluation of Large Language Models in Multilingual Learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. 13171–13189.

[23] Yongqi Li, Xinyu Lin, Wenjie Wang, Fuli Feng, Liang Pang, Wenjie Li, Liqiang Nie, Xiangnan He, and Tat-Seng Chua. 2024. A survey of generative search and recommendation in the era of large language models. *arXiv preprint arXiv:2404.16924* (2024).

[24] Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Hao Zhang, Yong Liu, Chuhan Wu, Xiangyang Li, Chenxu Zhu, et al. 2025. How can recommender systems benefit from large language models: A survey. *ACM Transactions on Information Systems* 43, 2 (2025), 1–47.

[25] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[26] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 43–52.

[27] Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. Representation learning with large language models for recommendation. In *Proceedings of the ACM Web Conference 2024*. 3464–3475.

[28] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web*. 521–530.

[29] Nina Schick. 2020. *Deep Fakes and the Infocalypse: What You Urgently Need To Know*. Monoray.

[30] Mikhail Shirokikh, Ilya Shenbin, Anton Alekseev, Anna Volodkevich, Alexey Vasilev, Andrey V Savchenko, and Sergey Nikolenko. 2024. Neural Click Models for Recommender Systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2553–2558.

[31] Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. 2023. Model dementia: Generated data makes models forget. *arXiv e-prints* (2023), arXiv–2305.

[32] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.

[33] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805* (2023).

[34] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[35] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[36] Haoyu Wang, Sunhao Dai, Haiyuan Zhao, Liang Pang, Xiao Zhang, Gang Wang, Zhenhua Dong, Jun Xu, and Ji-Rong Wen. 2025. Perplexity Trap: PLM-Based Retrievers Overrate Low Perplexity Documents. In *13th International Conference on Learning Representations, ICLR 2025*.

[37] Wei Wei, Xubin Ren, Jiabin Tang, Qinyong Wang, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. Llmrec: Large language models with graph augmentation for recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. 806–815.

[38] Jiayang Wu, Wensheng Gan, Zefeng Chen, Shicheng Wan, and Hong Lin. 2023. Ai-generated content (aigc): A survey. *arXiv preprint arXiv:2304.06632* (2023).

[39] Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, et al. 2024. A survey on large language models for recommendation. *World Wide Web* 27, 5 (2024), 60.

[40] Yunjia Xi, Weiwen Liu, Jianghao Lin, Xiaoling Cai, Hong Zhu, Jieming Zhu, Bo Chen, Ruiming Tang, Weinan Zhang, and Yong Yu. 2024. Towards open-world recommendation with knowledge augmentation from large language models. In *Proceedings of the 18th ACM Conference on Recommender Systems*. 12–22.

[41] Shicheng Xu, Danyang Hou, Liang Pang, Jingcheng Deng, Jun Xu, Huawei Shen, and Xueqi Cheng. 2024. Invisible relevance bias: Text-image retrieval models prefer ai-generated images. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*. 208–217.

[42] Shicheng Xu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. 2024. Search-in-the-Chain: Interactively Enhancing Large Language Models with Search for Knowledge-intensive Tasks. In *Proceedings of the ACM Web Conference 2024*. 1362–1373.

[43] Zhenrui Yue, Yueqi Wang, Zhankui He, Huimin Zeng, Julian McAuley, and Dong Wang. 2024. Linear recurrent units for sequential recommendation. In *Proceedings of the 17th ACM international conference on web search and data mining*. 930–938.

[44] Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023. Extractive Summarization via ChatGPT for Faithful Summary Generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. 3270–3278.

[45] Haiyuan Zhao, Jun Xu, Xiao Zhang, Guohao Cai, Zhenhua Dong, and Ji-Rong Wen. 2023. Unbiased Top-k Learning to Rank with Causal Likelihood Decomposition. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*. 129–138.

[46] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223* 1, 2 (2023).

[47] Zihuai Zhao, Wenqi Fan, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, et al. 2024. Recommender systems in the era of large language models (llms). *IEEE Transactions on Knowledge and Data Engineering* (2024).

[48] Xinyi Zhou and Reza Zafarani. 2020. A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Computing Surveys (CSUR)* 53, 5 (2020), 1–40.