Decoding Matters: Addressing Amplification Bias and Homogeneity Issue for LLM-based Recommendation

Keqin Bao[1,2*] **Jizhi Zhang**[1] **Yang Zhang**[3†]
Xinyue Huo[1] **Chong Chen**[2] **Fuli Feng**[1†]
[1]University of Science and Technology of China,
[2]Huawei Inc. [3]National University of Singapore
{baokq, cdzhangjizhi, xinyueh}@mail.ustc.edu.cn,
chenchong55@huawei.com, {zyang1580, fulifeng93}@gmail.com

Abstract

Adapting Large Language Models (LLMs) for recommendation requires careful consideration of the decoding process, given the inherent differences between generating items and natural language. Existing approaches often directly apply LLMs' original decoding methods. However, we find these methods encounter significant challenges: 1) amplification bias—where standard length normalization inflates scores for items containing tokens with generation probabilities close to 1 (termed ghost tokens), and 2) homogeneity issue—generating multiple similar or repetitive items for a user. To tackle these challenges, we introduce a new decoding approach named *Debiasing-Diversifying Decoding* (D^3). D^3 disables length normalization for ghost tokens to alleviate amplification bias, and it incorporates a text-free assistant model to encourage tokens less frequently generated by LLMs for counteracting recommendation homogeneity. Extensive experiments on real-world datasets demonstrate the method's effectiveness in enhancing accuracy and diversity. The code is available at https://github.com/SAI990323/DecodingMatters.

1 Introduction

Researchers have endeavored to adapt Large Language Models (LLMs) for recommender systems, seeking to harness the remarkable abilities of LLMs to improve recommendation performance (Bao et al., 2023b; Zhang et al., 2023a; Wei et al., 2024; Harte et al., 2023). Numerous adaptation solutions have been proposed, with some showcasing notable success. Among these, using LLMs to perform recommendations in a generative manner is particularly promising (Bao et al., 2023a; Tan et al., 2024; Zheng et al., 2024). This approach involves teaching LLMs to directly generate suitable item representations (*e.g.*, item titles) as recommendations based on given instructions. By doing so, this approach aligns well with the generative nature of LLMs, potentially enabling a more effective utilization of their capabilities for recommendation.

To generate an item, LLMs must produce a sequence of tokens representing the item, necessitating multi-step decoding during generation. Existing approaches typically adopt original decoding strategies of the utilized language models, such as beam search (Bao et al., 2023a; Zheng et al., 2024; Tan et al., 2024). However, there are distinct disparities between generating recommended items and natural language (Bao et al., 2023a; Zheng et al., 2024). For instance, in terms of the generation space, items only correspond to a specific, non-uniformly sampled subset of the entire language space; in terms of generation results, a set of items should be recommended, as opposed to only ond appriopriate language output required in typical NLP tasks. These distinctions potentially introduce new challenges for directly applying the heuristic decoding strategies.

This work delves into studying the decoding process of LLMs for generating recommendations. We identified two potential critical issues for applying the original decoding strategies of LLMs:

- Amplification Bias: Due to the non-uniform distribution of items in the language space, some items may contain tokens with generation probabilities close to 1 under certain conditions (referred to as ghost tokens). Existing decoding methods tend to enhance the scores for these items. Typically, these methods utilize length normalization to counteract the length bias during generation (Wu et al., 2016; Vaswani et al., 2017) — longer sequences tend to have lower probabilities than shorter sequences merely due to the multiplication of probabilities for each token. However, when ghost tokens appear, multiplying their generation probabilities (near 1)

[*] Work is done during internship at Huawei Inc.
[†] Corresponding Author

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 10540–10552
November 12-16, 2024

doesn't significantly decrease the final score, but length normalization is still applied, resulting in score amplification.

- Homogeneity Issue: With the original decoding methods, LLMs often produce items with similar structures and content when offering multiple recommendations to a user, as textually similar sequences typically receive similar scores. For example, when suggesting products, the model may recommend several items from the same series or category (e.g., "PlayStation 3" and "PlayStation 4"). Moreover, the model frequently repeats item features based on past user interactions, due to inheriting the match-and-copy mechanism of LLMs (Olsson et al., 2022; Wang et al., 2022).

Both amplification bias and homogeneity issues would undeniably exert substantial influences on recommendation quality, impacting factors such as accuracy and diversity. Addressing these issues is crucial. However, this endeavor is undoubtedly non-trivial. As discussed, the emergence of amplification bias is intricately linked with the process of addressing the length bias of LLMs' generation. Therefore, it's crucial to navigate addressing amplification bias without reintroducing the length bias. Meanwhile, to address the homogeneity issue, we must enhance the generation probability of tokens in LLM that receive less attention. However, there is a lack of clear guidance on how to balance the original predictions with enhancing less attention tokens and which tokens to select.

To this end, we introduce a novel strategy called *Debiasing-Diversifying Decoding* (D^3). To mitigate amplification bias while still addressing length bias, D^3 considers selectively applying length normalization to tokens while excluding it for ghost tokens. When implementing, we find that excluding ghost tokens results in relatively uniform item lengths, making the remaining length normalization unnecessary. Consequently, D^3 removes all length normalization during decoding, facilitating its implementation. To tackle the homogeneity issue, D^3 incorporates scores from a text-free assistant model at each decoding step to guide token generation. This helps avoid generating similar and repetitive items, as the text-free model is not influenced by repetitive text and can provide meaningful non-repeated/non-similar token suggestions based on its recommendation capabilities.

In total, our contribution is as follows:

- We highlight the importance of decoding strategies for LLMs' generation in recommendation scenarios and identify two critical challenges: amplification bias towards items with ghost tokens and the homogenous issues of generating similar and repeated items.
- We point out the ghost token and the repetition phenomenon and propose to mitigate this issue by removing the length penalty and utilizing a text-free assistant model.
- Extensive experiments have demonstrated that our methods can simultaneously enhance the accuracy and diversity of recommendations.

2 Related Work

In this section, we will briefly introduce the development and status of the related fields of this article, mainly including LLMs for recommender systems and decoding strategies for language models.

2.1 Large Language Models for Recommender Systems

The remarkable capabilities demonstrated by LLMs have sparked interest within the recommender systems community, leading to a surge in research aimed at integrating LLMs into these systems (Wu et al., 2023; Lin et al., 2023a; Bao et al., 2024). This research can be broadly divided into two streams. The first stream involves directly utilizing LLMs as encoders to represent items, which are then input into conventional recommendation models (Yuan et al., 2023; Xi et al., 2023). However, the prevalent LLMs are primarily decoder-only architectures, optimized for next-token-prediction, which is not conducive to information encoding. This limitation potentially constrains the full potential of LLMs in recommendation tasks.

The second stream of research focuses on harnessing the generative power of LLMs to produce recommendations directly (Lin et al., 2023b; Liao et al., 2024; Wang et al., 2023; Zhang et al., 2023b, 2024a,b). Despite this approach, it has been observed that LLMs have limited exposure to recommendation-specific data during pre-training, necessitating fine-tuning for effective application in this domain. While methods of this nature are on the rise, they often prioritize improving LLMs' recommendation performance through training enhancements, overlooking a crucial aspect: the detailed examination of the models' recommendation outputs and the considerations for the decoding

phase.

2.2 Decoding in Language Model

In the domain of open-ended text generation, the issue of diversity has garnered significant attention, primarily due to the inherent dependency of language models on maximizing probability during the generation process (Welleck et al., 2020; Holtzman et al., 2020). To address this, a common approach is to adjust the temperature parameter. This method modifies the sharpness of the probability distribution of tokens at each step of generation, reducing the likelihood of exceedingly similar outcomes due to overly high probabilities of specific tokens being chosen. Beyond this, Diverse Beam Search (DBS) (Vijayakumar et al., 2016) introduces the concept of beam groups, partitioning the generation process to manage similarity across different groups, thereby increasing the diversity of the generated results. Currently, due to the differences in the decoding space, using LLMs for recommendations presents unique challenges. There is a lack of comprehensive discussion on the decoding stage in this field. In this paper, we conduct the first exploratory analysis and optimization of the decoding process and the recommendations.

3 Preliminary

In this section, we will introduce the background knowledge of the decoding process of current LLM-based recommender (recLLM) methods.

Considering the decoding process, the model takes an instruction input and a sequence of user historical interactions to generate a suitable token sequence representing a recommended item. Our input, of length n, is denoted as $x = x_1 \cdots x_n$, where each x_i is a token from the model's vocabulary V. During decoding, an item is represented by generating a sequence of tokens having length m, denoted as $y = y_1 \cdots y_m$. At decoding time, tokens are generated iteratively, one at a time, conditioned on the previous context and expect to select the output with the highest probability:

$$p(y|x) = \prod_{i=1}^{m} p(y_i|x, y_{<i}), \tag{1}$$

where $p(y_i|x, y_{<i})$ represents the probability distribution of the next token. However, given the impracticality of traversing all possible scenarios to calculate this probability, the greedy search can apply to select the token x_i with the highest probability at each step.

Beam Search. Due to the limitations of greedy decoding methods, which can only generate a single item and often fail to find the optimal sequence of items (Holtzman et al., 2020), researchers currently using LLMs for recommendation typically employ beam search methods to generate multiple items simultaneously (Zheng et al., 2024; Tan et al., 2024). In detail, the formula for calculating the score for a hypothesis h at step t in beam search is:

$$\mathcal{S}(h_{\leq t}) = \mathcal{S}(h_{\leq t-1}) + \log(p(h_t|x, h_{\leq t-1})), \tag{2}$$

where $\mathcal{S}(h_{\leq t})$ is the score of the hypothesis up to step t used for selecting hypothesis, $p(h_t|x, h_{\leq t-1})$ is the conditional probability of the token h_t given the previous hypothesis $h_{\leq t-1}$ and the input x.

The beam search keeps track of the top B hypotheses at each step based on their scores, where B is the beam width. It then expands each of these hypotheses by considering the k most likely tokens. The new hypotheses are scored according to the above formula, and the top B hypotheses are retained for the next step. This process continues until an end-of-sequence token is predicted or the maximum sequence length is reached. Moreover, in natural language generation, to avoid repetitiveness and overly long outputs, a length normalization term is typically added after the generation is complete.

$$\mathcal{S}(h) = \mathcal{S}(h)/h_L^{\alpha}, \tag{3}$$

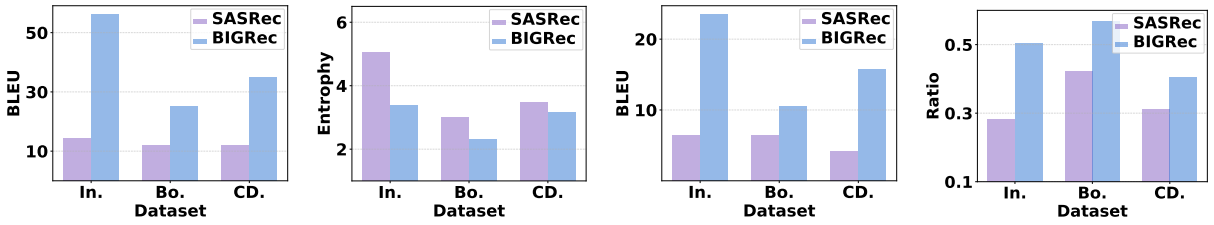
where h_L is the length of the h, α is the hyperparameters to control the length penalty. Once the beam search is completed, the top B hypotheses are selected as the final output $[y_1, y_2, ..., y_B]$.

4 Issues for Existing Decoding Method

In this section, we elaborate on the amplification bias and homogeneity issues of the existing decoding method, clarifying our motivation.

4.1 Amplification Bias

As discussed, due to the non-uniformity of the item generation space, there are ghost tokens whose generation becomes deterministic once their preceding tokens have appeared. The model easily learns this deterministic characteristic during training and subsequently assigns a near 1-generation probability



(a) (b) (c) (d)

Figure 1: Homogeneity comparison of recommendation results between recLLM method BIGRec and traditional method SASRec on three datasets: *Instruments* (*In.*), *Books* (*Bo.*), and *CDs* (*CD.*). (a) and (b) show text similarity and category diversity (measured by entropy) for the first 5 tokens within the top 10 recommendations, where higher similarity and lower entropy indicate greater homogeneity. (c) and (d) display text similarity and category repetition in top-10 recommendations compared to historical interactions.

to these tokens during the generation phase. For Eq. (2) and Eq. (3), it is evident that these tokens minimally impact the value of $\mathcal{S}$ but increase the length h_L. This means they act like ghosts, occupying a sequence position without changing the score $\mathcal{S}$. Consequently, if the length normalization in Eq. (3) is still applied, the final score would be improperly amplified.

4.2 Homogeneity Issue

Next, we illustrate the homogeneity issue in generated results by analyzing the recommendation similarity within a recommended list and their relative repetition compared to historical data.

Recommendation Similarity. We note that the top-10 recommendations from recLLM often begin with similar tokens, reducing diversity. To compare, we examine the first 5 tokens in these recommendations against those from conventional models[1]. Furthermore, we also assess the category diversity of the recommended items. As depicted in Figure 1a, we utilized BLEU to measure the textual similarity among recommendation results, and Entropy to assess the category diversity. It is evident that, compared to traditional models, the results from recLLMs exhibit higher degrees of similarities. This suggests that, despite their impressive performance, LLMs tend to generate multiple homogeneous recommendations, given a user. This homogeneity might be attributed to the excessive scores given to the first few similar tokens by the beam search, which makes it challenging for more diverse outcomes to be included in the final candidate generations (Vijayakumar et al., 2016).

Repetition Phenomenon. Looking deeper, we observe that certain tokens repetitively appear both in the recommended items and historically interacted items. Thus, we continue our analysis to determine the extent of similarity between the recommended items and the user's historical interactions. Similarly, we computed the textual and categorical similarities between the recommendations generated by LLMs/traditional models and users' historical interaction records. As illustrated in Figure 1c, we observe that similar to the preceding findings, the recommendations based on recLLM also exhibit higher similarity to users' historical interaction sequences. This suggests that while incorporating textual information offers certain benefits, it also introduces a new bias: the model tends to copy from the preceding text (Olsson et al., 2022), leading to a prevalence of similar items in the recommendations.

5 Debiasing-Divesifying Decoding

Next, we introduce Debiasing-Diversifying Decoding, which incorporates two new designs into the existing method to address the two issues.

Remove Amplification Bias. To address amplification bias, according to its source, an intuitive approach is to exclude ghost tokens when calculating the sequence length for normalization in Eq. (3). In essence, apply length normalization only to normal tokens. However, our analysis reveals that removing these tokens results in item token sequences with a much uniform length distribution, making length normalization for the remaining tokens unnecessary[2]. Therefore, we opt to directly eliminate the length normalization to neutralize the impact of

[1]For traditional models, we replaced the item IDs they produced with their corresponding textual representations.

[2]The detailed analysis can be seen in Appendix §C

the ghost token and remove the amplification bias for the decoding.

Address Homogeneity Issue. In light of the Homogeneity analysis, we conclude that during beam search, numerous potential recommendation outcomes are prematurely pruned due to the dominance of certain tokens with exceptionally high scores (Vijayakumar et al., 2016). Consequently, this not only impairs the recommendation quality but also the recommendation diversity. To address this issue, it is imperative to refine the model's scoring mechanism to ensure a more soft distribution of scores, thereby broadening the range of choices available during the generation phase. Hence, we aim to adjust the score of each candidate token at each step to help recLLM better generate the item to be recommended.

To achieve this objective, the key is to boost the scores of tokens that are meaningful but underestimated by recLLM, while avoiding excessive disregard of tokens with higher scores from recLLM to maintain recommendation performance. Relying solely on recLLM's predictions to accomplish this is challenging. Therefore, we propose leveraging an additional text-free model to assist. Although this model has poor recommendation abilities, it can still provide meaningful recommendation proposals, which are less prone to textual similarity issues, for recLLM reference.

In detail, we leverage the text-free model to generate additional scores for tokens at each decoding step, using these scores to refine the token scores of recLLM. Let $\mathcal{I}$ denote the set of all items, and $\mathcal{I}_i$ denote the i-th item. We denote the probability of the text-free model to recommending $\mathcal{I}_i$ as $p_{TF}(\mathcal{I}_i)$. At a decoding step, given the hypothesis $h_{<t}$ that has already been produced, we need to generate a new token h_t based on it. For which, only a subset of items is eligible for the generation, that is, the items matching with $h_{<t}$, denoted as $\mathcal{I}_{h<t}$. Based on this set, we define the token score $\mathcal{S}_{TF}(h_{\leq t})$ of the text-free model for a potential token h_t as follows:

$$\mathcal{L}_{TF}(h_t|h_{\leq t-1}) = log\left(\frac{\sum_{i\in\mathcal{I}_{h_{<t},h_t}} p_{TF}(\mathcal{I}_i)}{\sum_{i\in\mathcal{I}_{h_{<t}}} p_{TF}(\mathcal{I}_i)}\right), \quad (4)$$

$$\mathcal{S}_{TF}(h_{\leq t}) = \mathcal{S}_{TF}(h_{\leq t-1}) + \mathcal{L}_{TF}(h_t|h_{\leq t-1}), \quad (5)$$

where $\mathcal{I}_{h_{<t},h_t}$ denote the items in $\mathcal{I}_{h_{<t}}$ with h_t coming after $h_{<t}$, $\mathcal{L}_{TF}(h_t|h_{\leq t-1})$ denotes the log probability for the token by the text-free assistant model, similar to Eq. (2). Then the overall score function we used to guide the recLLM to generate step by step is:

$$\tilde{\mathcal{S}}(h_{\leq t}) = \alpha * \mathcal{S}(h_{\leq t}) + (1-\alpha) * \mathcal{S}_{TF}(h_{\leq t}), \quad (6)$$

where α is a hyper-parameter to control the assistance level of the text-free model. Notably, $\mathcal{S}(h_{\leq t})$ is computed without the length normalization.

This formulation indicates that during each step of generation, we do not completely depend on the knowledge encoded within the recLLM. Instead, we harness logits from a text-free model, which is detached from linguistic context, to guide the recLLM in generating. By infusing text-free model inferences at every stage, we mitigate the homogeneity and redundancy that stem from the model's overdependence on language-based attributes.

6 Experiment

6.1 Experimental Settings

6.1.1 Dataset

We conduct experiments on six real-world datasets from Amazon review data[3], including *Instruments*, *CDs*, *Games*, *Toys*, *Sports*, and *Books*. All datasets contain user review data from May 1996 to October 2018. To preprocess the data, we follow the strategy outlined in the BIGRec paper to truncate the dataset based on time information, considering the high cost of training LLMS. Moreover, we filter out unpopular users and items with fewer than five interactions and set the maximum item sequence length to 10 to meet the baseline requirements. Details of the preprocessing steps and statistics of the processed datasets can be found in Appendix §A.

6.1.2 Evaluation Protocal

To evaluate the model's top-K recommendation performance (accuracy), we use two commonly used metrics: Hit Ratio (HR@K) and Normalized Discounted Cumulative Gain (NDCG@K) (Bao et al., 2023a; Rajput et al., 2024), which we compute using the all-ranking protocol (Krichene and Rendle, 2020). In this paper, we set K as 5 and 10. To better align with real-world scenarios, we follow previous work (Bao et al., 2023a) when dividing the dataset, that is, splitting a dataset into training, validation, and test sets according to timestamps. This ensures that there will be no data leakage issues (Ji et al., 2023) during the training and testing of the model.

[3] https://cseweb.ucsd.edu/~jmcauley/datasets.html#amazon_reviews.

Table 1: Recommendation accuracy of the compared methods on different-domain datasets."+Temp" denotes adjusting the temperature coefficient for BIGRec's decoding. "+D^3" denotes applying our decoding method to TIGER/BIGRec. The best results are bolded.

Datasets	Metrics	Caser	GRU4Rec	SASRec	GRU4Rec*	SASRec*	TIGER	+D^3	BIGRec	+Temp	+D^3
Instruments	NDCG@5	0.0550	0.0562	0.0643	0.0712	0.0672	0.0764	0.0785	0.0813	0.0795	**0.0816**
	HR@5	0.0678	0.0681	0.0715	0.0843	0.0798	0.0853	0.0893	0.0929	0.0897	**0.0938**
	NDCG@10	0.0595	0.0614	0.0676	0.0772	0.0731	0.0797	0.0830	0.0856	0.0841	**0.0871**
	HR@10	0.0817	0.0843	0.0817	0.1030	0.0980	0.0958	0.1032	0.1062	0.1040	**0.1110**
CDs	NDCG@5	0.0161	0.0248	0.0477	0.0435	0.0418	0.0484	0.0620	0.0640	0.0513	**0.0823**
	HR@5	0.0224	0.0342	0.0647	0.0566	0.0561	0.0559	0.0748	0.0786	0.0674	**0.1025**
	NDCG@10	0.0193	0.0288	0.0535	0.0482	0.0478	0.0512	0.0659	0.0694	0.0579	**0.0871**
	HR@10	0.0485	0.0467	0.0824	0.0715	0.0745	0.0646	0.0868	0.0956	0.0879	**0.1090**
Games	NDCG@5	0.0122	0.0169	0.0237	0.0282	0.0263	0.0367	0.0394	0.0318	0.0279	**0.0415**
	HR@5	0.0187	0.0261	0.0338	0.0407	0.0383	0.0495	0.0521	0.0420	0.0379	**0.0565**
	NDCG@10	0.0164	0.0221	0.0290	0.0352	0.0328	0.0417	0.0453	0.0372	0.0355	**0.0480**
	HR@10	0.0321	0.0423	0.0502	0.0624	0.0586	0.0651	0.0706	0.0610	0.0553	**0.0767**
Toys	NDCG@5	0.0228	0.0200	0.0356	0.0367	0.0371	0.0421	0.0487	0.0570	0.0497	**0.0652**
	HR@5	0.0316	0.0275	0.0473	0.0518	0.0531	0.0534	0.0610	0.0739	0.0660	**0.0841**
	NDCG@10	0.0276	0.0238	0.0398	0.0422	0.0433	0.0465	0.0535	0.0643	0.0565	**0.0713**
	HR@10	0.0465	0.0392	0.0745	0.0688	0.0721	0.0668	0.7600	0.0965	0.0871	**0.1025**
Sports	NDCG@5	0.0439	0.0586	0.0695	0.0577	0.0627	0.0846	0.0876	0.0932	0.0870	**0.0970**
	HR@5	0.0541	0.0663	0.0770	0.0760	0.0785	0.0915	0.0964	0.1040	0.0977	**0.1083**
	NDCG@10	0.0469	0.0618	0.0725	0.0624	0.0677	0.0869	0.0901	0.0975	0.0918	**0.1013**
	HR@10	0.0633	0.0761	0.0866	0.0906	0.0939	0.0984	0.1040	0.1171	0.1125	**0.1215**
Books	NDCG@5	0.0042	0.0060	0.0097	0.0076	0.0103	0.0145	0.0181	0.0182	0.0197	**0.0217**
	HR@5	0.0069	0.0094	0.0146	0.0119	0.0128	0.0183	0.0233	0.0239	0.0253	**0.0280**
	NDCG@10	0.0060	0.0078	0.0123	0.0099	0.0151	0.0157	0.0200	0.0204	0.0218	**0.0240**
	HR@10	0.0123	0.0149	0.0226	0.0189	0.0229	0.0220	0.0290	0.0308	0.0317	**0.0353**

6.1.3 Baselines

We adopt the following representative sequential recommendation models as baselines for our experimental comparison:

- Caser (Tang and Wang, 2018) is a method using CNN that models user behaviors through horizontal and vertical convolutional.
- GRU4Rec (Hidasi et al., 2016) is an RNN-based method that uses GRU to model the user behavior via encoding the item sequence.
- SASRec (Kang and McAuley, 2018) is a method using Transformer to model the item sequences.
- GRU4Rec* (Hidasi et al., 2016) is similar to GRU4Rec but uses the LLM-based embedding to initialize the item embedding.
- SASRec* (Kang and McAuley, 2018) is similar to SASRec but uses the LLM-based embedding to initialize the item embedding.
- TIGER (Rajput et al., 2024) is a generative retrieval paradigm for sequential recommendation and introduces a semantic ID to uniquely identify items. We extend to LLMs, using the instruction combined with semantic ID, and generate multiple semantic ID sequences representing items during the recommendation process.
- BIGRec (Bao et al., 2023a) is a method that uses instruction-tuning to directly generate items. We have made changes to the decoding method of this approach, ensuring that the produced results are always within the item list of the dataset.

6.1.4 Implementation Details

For the traditional recommendation models, we optimize them using binary cross-entropy loss and the Adam optimizer with a learning rate searched in $[1e\text{-}2, 1e\text{-}3, 1e\text{-}4]$. We process the data in batches of size 1024, and we adjust the weight decay within the range of $[1e\text{-}2, 1e\text{-}3, 1e\text{-}4, 1e\text{-}5, 1e\text{-}6]$. For LLM-based methods, for efficiency, we apply Qwen1.5-1.8B (Bai et al., 2023) as the backbone LLM. we use the AdamW (Loshchilov and Hutter, 2019) optimizer and adjust the learning rate within the range of $[1e\text{-}3, 1e\text{-}4, 5e\text{-}5]$. During the training, we applied the cosine learning scheduler for 3 epochs and set the early stop patience as one epoch[4]. In our experiments related to the temperature coefficient, we adjusted it within the range of

[4]In our preliminary experiments, we found that for LLMs trained on the entire dataset, the model generally converges after only 1 to 2 epochs of training. When we opt to use a language-agnostic model to assist the recommendation model based on LLM in inference, we simply employ the SASRec (Kang and McAuley, 2018)

$[1.0, 1.5, 2.0]$. All experiments were conducted on an Ascend 910B with 32GB VRAM.

6.2 Main Results

In this subsection, we study the recommendation performance of our proposed D^3 method over six open-world datasets. As shown in Table 1, it showcases our primary experimental results. Notably, the results can be categorized into the following groups: traditional models (Caser, GRU4Rec, SASRec), traditional models enhanced by LLM embeddings (GRU4Rec*, SASRec*), TIGER with different decoding methods, recLLM (BIGRec) with different decoding strategies. Regarding recLLM decoding, we investigate not only its default strategy and our D^3 approach but also a temperature scaling coefficient strategy (denoted as "+temp") commonly used to adjust generation scores. As for TIGER, despite not being fully LLM-based, we still apply our method for it to verify our method's wide applicability. From the table, we have the following observations:

- When compared to the baseline, our decoding approach with recLLM often gets better performance, surpassing all baselines to achieve the best recommendation performance on all datasets. This underscores the significance of studying the decoding strategy for recLLM and demonstrates the superiority of our decoding method which removes the amplification bias and alleviates the homogeneity issue.
- Considering that TIGER also employs beam search for generation during decoding, it may face similar challenges to recLLM. We applied our D^3 approach to TIGER. Comparing the results of the original TIGER and the variant with D^3 (columns 8 and 9), we observe improvements when using our strategy. This demonstrates the generality of our method, indicating that it is also suitable for non-linguistic, generation-based recommendation systems.
- In comparing TIGER (using LLMs but presenting items with semantic IDs) with the recLLM method BIGrec, we find that to leverage LLMs, it is more suitable to directly represent items with text. We conjecture that this may be due to the fact that, while TIGER incorporates information from LLMs during the training of its encoder, using LLMs as encoders can result in the loss of some information. Additionally, using the complete textual representation of items allows for the full utilization of the knowledge learned by the LLM.
- We found that simply adjusting the decoding temperature does not improve recommendation performance, although it may enhance recommendation diversity (discussed later in Figure 2).

7 Analysis

This section further explores our decoding approach in detail and discusses potential extensions. First, we conduct an ablation analysis to validate the influence of our two key strategies on recommendation accuracy. We then examine the impact of the method's designs on recommendation diversity and investigate an extension of our methods which can enable easy adjustment of the recommendation type distribution. Finally, we validate the generalizability of the proposed method using additional backbones, approaches, and datasets.

7.1 Ablation

To evaluate the influence of different components of the proposed method D^3 on accuracy, we conducted ablation studies here. The core elements of D^3 include 1) removing length normalization to address amplification bias, denoted by "RLN", and 2) integrating a text-free assistant model for augmenting diversity, denoted as "TFA". We systematically deactivated the components one by one to analyze their effects. The results are summarized in Table 2, where we draw the following conclusions:

- Disabling the removal of length normalization ("-RLN") or the text-free assistant ("-TFA") model could both lead to decreased performance compared to the complete version of D^3. These results emphasize the importance of both designs in addressing their respective issues. Notably, homogeneous recommendations could also limit recommendation accuracy, so the "-TFA" model also exhibits decreased performance.
- When applying "-RLN" or "-TFA" to D^3, only one of the D^3 designs is effective, yet it still outperforms the Baseline. This indicates the presence of both the amplification bias and homogeneity issue and demonstrates that addressing them individually can lead to performance improvements.

7.2 Diversity

As discussed in our previous work, existing generation strategies often result in homogeneous recommendations. In D^3, we integrate a text-free

Table 2: Ablation results for "D^3" in terms of accuracy, reporting HR@10. "Baseline" denotes recLLM's original decoding, serving as a reference here. "- RLN" denotes deactivating the operation of D^3 on length normalization, i.e., deactivating addressing amplification bias. "- TFA" denotes deactivating the use of the text-free model.

	Instruments	Books	CDs	Sports	Toys	Games
Baseline	0.1062	0.0308	0.0956	0.1171	0.0965	0.0610
D^3	**0.1111**	**0.0354**	**0.1190**	**0.1215**	**0.1025**	**0.0767**
- RLN	0.1093	0.0353	0.1000	0.1200	0.0975	0.0659
- TFA	0.1086	0.0309	0.1115	0.1192	0.1006	0.0732

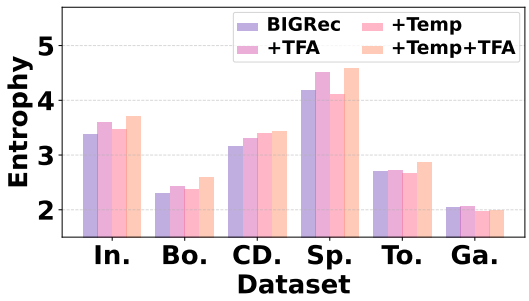


Figure 2: Recommendation diversity (measured by entropy) of the original BIGRec and the variants with other decoding strategies. "+TFA" denotes the variant applying our text-free model assistant decoding, "+Temp" denotes the variant using the widely-used temperature scaling to increase diversity, and "+Temp+TFA" denotes the variant combining "+TFA" and "+Temp". Smaller entropy denotes less diversity.

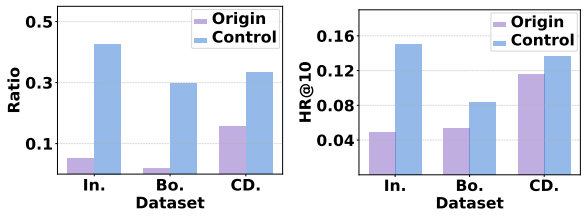


Figure 3: Effectiveness of employing the proposed TFA to enhance recommendation ratio (left) and accuracy (right) for a specified target category of items.

Datasets	CID	+D^3	BIGRec	+D^3
CDs	0.0610	**0.0921**	0.0869	**0.1179**
Games	0.0447	**0.0598**	0.0727	**0.0856**

Table 3: We report the HR@10 results of our methods on another approach – CID (Hua et al., 2023) and another backbone –Llama3.1-8B (Dubey et al., 2024). "+D^3" denotes applying our decoding method. The best results are bolded.

assistant to address the issue. Here, we further investigate this design by exploring two research questions: (1) Can the use of a text-free assistant (TFA) model improve recommendation diversity? (2) How does the effect of TFA combined with diversity-enhancing decoding strategy (temperature adjustment)? As shown in the Figure 2, we have the following findings:

- Adjusting the temperature improves diversity, but as Table 1 shows, it results in lower performance.
- Using a text-free model for assisted decoding can not only improve the recommendation accuracy but also enhance recommendation diversity.
- Utilizing a text-free model for assisted decoding is a plug-and-play approach. It can be combined with temperature adjustment to further improve the diversity of recommendation results.

7.3 Distribution Adjustment

Taking it one step further, since we can use a text-free model to assist LLMs in decoding, we can also use a similar approach to control the distribution of recommendation results when we have specific recommendation needs. For instance, we could recommend more computer-related books to users in the computer science field. Specifically, this involves strengthening the logits of certain items during the assisted decoding process.

To validate this extension attribute, we simulated a scenario where we need to directionally enhance the recommendation of a specific group of items and improve recommendation accuracy for that group. Therefore, when using the text-free model, we only assist items of that particular group. As shown in Figure 3, we plotted[5] the proportion of recommendations for specific categories before and after the control, and the recommendation performance. We can see that through this approach, we significantly improved the recommendation proportion and accuracy for specific categories of items. This verifies the feasibility of the extension of our method to adjust the recommendation distribution

[5]Resuls on other three datasets are shown in Figure 5.

Metrics	BIGRec	$+D^3$
NDCG@10	0.0277	**0.0398**
HR@10	0.0563	**0.0814**

Table 4: We report the NDCG@10 and HR@10 results of our methods on steam datasets (Kang and McAuley, 2018). "$+D^3$" denotes applying our decoding method. The best results are bolded.

during the decoding stage.

7.4 Generalizability

To show the generalizability of our methods, we conduct further experiments using additional backbones, approaches, and datasets. In detail, as shown in Table 3, we present the results of applying our decoding method to a new backbone model, Llama3.1-8B (Dubey et al., 2024), and a new item representation approach, CID (Hua et al., 2023), on the CDs and Games datasets.

The findings demonstrate consistent improvements in performance with both the new backbone and the new method, highlighting the generalizability and effectiveness of our decoding approach. Besides, we have also applied our approach to the Steam (Kang and McAuley, 2018) dataset. Given the substantial size of this dataset, we randomly selected 10% of the users to expedite the validation process of our method. The results of our experiments are presented in Table 4. These findings demonstrate that our decoding method is not only applicable to other platforms and datasets but also further validates the generalizability of our approach.

8 Conclusion

In this paper, we begin by conducting a comprehensive analysis of the decoding strategies currently used in LLMs for recommendation, highlighting the critical role of the decoding process. During the analysis, we identify two major issues in existing methodologies: (1) amplification bias, *i.e.,* amplifying scores for items with ghost tokens, and (2) homogeneity issue, *i.e.,* generating highly similar recommendations and easy-to-produce recommendations with textual features similar to historical interactions This work introduces a novel approach, D^3, to address the issues by eliminating length normalization and incorporating a text-free model to assist in decoding. Extensive experimental results demonstrate that D^3 could significantly enhance recommendation accuracy and diversity. Furthermore, the method can be generalized to non-text generative recommendation frameworks, such as TIGER, indicating its versatility and effectiveness in improving recommendation systems.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (62272437), and the advanced computing resources provided by the Supercomputing Center of the USTC.

Limitations

This paper primarily focuses on the issues and challenges that arise when deploying LLMs in the domain of recommendation systems, particularly at the decoding phase, and offers a viable and effective solution. However, our study has several limitations: 1) The experiments conducted in this study are solely based on the Qwen1.5-1.8B model. 2) The current investigation emphasizes performance enhancement in sequential recommendation tasks. Although the method introduced in this paper does not significantly increase the inference time compared to existing recLLM methods, it encounters efficiency issues in inference when applied to real-world recommendation scenarios. A potential solution to address the inference efficiency problem is to eliminate the generation of ghost tokens. By integrating existing generation technologies like vLLM (Kwon et al., 2023), it is possible to significantly reduce the number of inference tokens and, consequently, the inference time. However, due to the limitations of our experimental setup and the fact that our primary focus was not on this topic, it was not experimentally tested in this paper.

Ethical Considerations

In this paper, we introduce D^3 to mitigate amplification bias and the problem of homogeneity in existing LLM-based recommendation decoding strategies, without introducing new ethical dilemmas. We employ publicly available data while carefully avoiding sensitive information. Nevertheless, as recommendations depend on user behavior data, privacy issues may arise. These concerns can be managed by securing user consent. Moreover, the usage of LLMs might perpetuate unseen societal biases. We recommend comprehensive risk evaluations and caution users about the potential risks associated with model deployment.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. arXiv preprint arXiv:2309.16609.

Keqin Bao, Jizhi Zhang, Xinyu Lin, Yang Zhang, Wenjie Wang, and Fuli Feng. 2024. Large language models for recommendation: Past, present, and future. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 2993–2996.

Keqin Bao, Jizhi Zhang, Wenjie Wang, Yang Zhang, Zhengyi Yang, Yancheng Luo, Chong Chen, Fuli Feng, and Qi Tian. 2023a. A bi-step grounding paradigm for large language models in recommendation systems. arXiv preprint arXiv:2308.08434.

Keqin Bao et al. 2023b. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In RecSys, pages 1007–1014. ACM.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. arXiv preprint arXiv:2407.21783.

Jesse Harte, Wouter Zorgdrager, Panos Louridas, Asterios Katsifodimos, Dietmar Jannach, and Marios Fragkoulis. 2023. Leveraging large language models for sequential recommendation. In Proceedings of the 17th ACM Conference on Recommender Systems, pages 1096–1102.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. 4th International Conference on Learning Representations, ICLR 2016.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. 8th International Conference on Learning Representations, ICLR 2020.

Wenyue Hua, Shuyuan Xu, Yingqiang Ge, and Yongfeng Zhang. 2023. How to index item ids for recommendation foundation models. In Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region, pages 195–204.

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

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

Walid Krichene and Steffen Rendle. 2020. On sampled metrics for item recommendation. In Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, pages 1748–1757.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles.

Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. 2024. Llara: Large language-recommendation assistant. SIGIR 2024.

Jianghao Lin et al. 2023a. How can recommender systems benefit from large language models: A survey. arXiv preprint arXiv:2306.05817.

Xinyu Lin et al. 2023b. A multi-facet paradigm to bridge large language model and recommendation. arXiv preprint arXiv:2310.06491.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. 7th International Conference on Learning Representations, ICLR 2019.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. In-context learning and induction heads. Transformer Circuits Thread. Https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. 2024. Recommender systems with generative retrieval. Advances in Neural Information Processing Systems, 36.

Juntao Tan, Shuyuan Xu, Wenyue Hua, Yingqiang Ge, Zelong Li, and Yongfeng Zhang. 2024. Towards llm-recsys alignment with textual id learning. SIGIR 2024.

Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In Proceedings of the eleventh ACM international conference on web search and data mining, pages 565–573.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. Advances in neural information processing systems, 30.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. arXiv preprint arXiv:1610.02424.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2022. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. ICLR 2023.

Yancheng Wang et al. 2023. Recmind: Large language model powered agent for recommendation. arXiv preprint arXiv:2308.14296.

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 WSDM 2024.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text generation with unlikelihood training. 8th International Conference on Learning Representations, ICLR 2020.

Likang Wu et al. 2023. A survey on large language models for recommendation. arXiv preprint arXiv:2305.19860.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144.

Yunjia Xi et al. 2023. Towards open-world recommendation with knowledge augmentation from large language models. CoRR, abs/2306.10933.

Zheng Yuan et al. 2023. Where to go next for recommender systems? ID- vs. modality-based recommender models revisited. In SIGIR 2023, pages 2639–2649. ACM.

Jizhi Zhang, Keqin Bao, Wenjie Wang, Yang Zhang, Wentao Shi, Wanhong Xu, Fuli Feng, and Tat-Seng Chua. 2024a. Prospect personalized recommendation on large language model-based agent platform. arXiv preprint arXiv:2402.18240.

Junjie Zhang et al. 2023a. Recommendation as instruction following: A large language model empowered recommendation approach. arXiv preprint arXiv:2305.07001.

Yang Zhang, Keqin Bao, Ming Yan, Wenjie Wang, Fuli Feng, and Xiangnan He. 2024b. Text-like encoding of collaborative information in large language models for recommendation. ACL 2024.

Yang Zhang, Fuli Feng, Jizhi Zhang, Keqin Bao, Qifan Wang, and Xiangnan He. 2023b. Collm: Integrating collaborative embeddings into large language models for recommendation. arXiv preprint arXiv:2310.19488.

Bowen Zheng et al. 2024. Adapting large language models by integrating collaborative semantics for recommendation. ICDE 2024.

A Data Statistic

As shown in Table 5, we outline the condition of our dataset. During the preprocessing phase, we took into consideration both the resource implications of training LLMs and the sparsity issues associated with randomly sampled datasets. Consequently, starting from October 2017, we processed the dataset to ensure that each user/item had a minimum of 5-core interactions post-processing. Should the number of items post-processing fall below a threshold (set to 10,000 in our case), we would extend the timeframe by an additional month backward and repeat the procedure, ultimately resulting in six distinct datasets. (Note: Even utilizing the full *Instruments* data, there are only 9,239 items available).

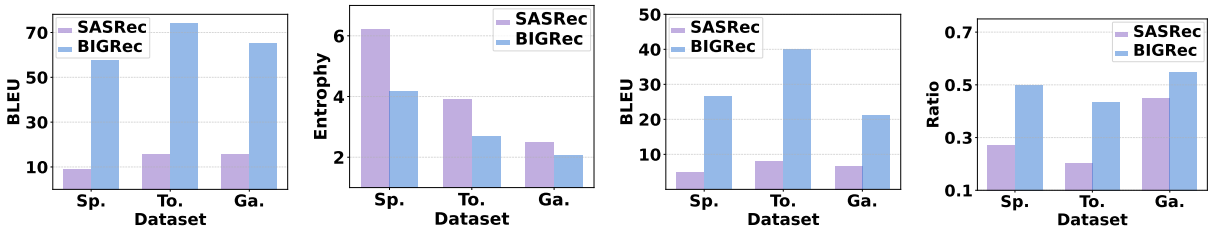
B Analysis of Amplification Bias

As we have discussed in the main text, due to the specific, non-uniform of the item space in recommendations, there exists a substantial number of ghost tokens during the generation of items. These tokens become determinate following the generation of preceding tokens. Therefore, their generation probability is approximately 1, which, under the default decoding strategy, inadvertently inflates the score of the item, thereby introducing bias. In order to mitigate this issue, we begin by analyzing the length of item representations in terms of tokens within each dataset, as well as the length after removing these tokens, as shown in Table 6. Our findings indicate that:

- The proportion of ghost tokens significantly impacts the total length, playing a decisive role.
- There is considerable variance in the original token lengths, suggesting that length significantly influences the scores of items during the generation process.
- After the removal of ghost tokens, the variance is reduced, and the lengths of items become relatively uniform. Therefore, it becomes feasible to eliminate the length normalization factor directly.

	Instruments	CDs	Games	Toys	Sports	Books
Item	9239	14239	11037	11252	16003	41722
Train	140482	148685	201613	112755	181477	682998
Valid	17561	18586	25202	14095	22685	85376
Test	17562	18587	25203	14096	22686	85376

Table 5: The table presents statistical information for six datasets. The first row shows the number of items in our dataset, and the second, third, and fourth rows display the number of sequences in the training, validation, and test sets, respectively.

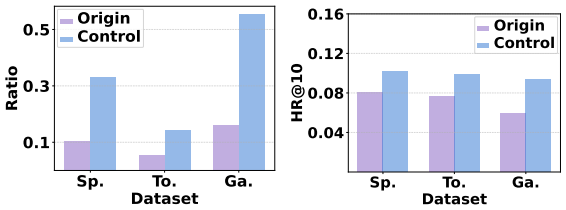


(a) (b) (c) (d)

Figure 4: The analysis of recommendation results using LLMs on the remaining three datasets (abbreviated as *Sp.* for *Sports*, *To.* for *Toys*, and *Ga.* for *Games*) is presented in these four figures. In Figures (a) and (b), the text-similarity of the top 5 tokens within the top 10 recommendations and the entropy of the overall recommendation categories are illustrated, respectively. Higher text similarity and lower entropy indicate a higher level of homogeneity in recommendations. Figures (c) and (d) depict display text similarity and category repetition in top-10 recommendations versus historical interactions.

Dataset	Origin		Remove	
	Avg	Var	Avg	Var
Instruments	19.34	55.7	4.31	1.67
CDs	7.64	27.83	3.34	1.15
Games	12.95	35.16	4.6	2.65
Toys	15.8	45.41	4.5	2.19
Sports	18.17	68.54	4.12	1.72
Books	12.27	36.74	3.47	0.84

Table 6: This table presents the average length and its associated variance of the token lists representing items in each dataset before and after removing ghost tokens.



(a) (b)

Figure 5: These Figures showcase the impact of our proposed TFA method on modifying recommendation distributions. In particular, Figure (a) shows the percentage of recommended items for a particular category after adjustments, whereas Figure (b) depicts the performance of recommendation within that category.

C Analysis of Homogeneity Issue

Figure 4 illustrates the homogeneity issue observed in the *Sports*, *Toys*, and *Games* datasets, a phenomenon that is consistent with findings in the other three datasets. LLM-based recommender systems demonstrate a significant level of similarity both in terms of text and category. This confirms that across all six datasets, while the integration of textual information enhances performance, it simultaneously introduces a text-level bias. This bias diminishes the diversity of the recommendations, causing them to converge on specific characteristics. It's important to clarify that the apparent issue identified within our study is not inherently detrimental. On one side, we note that even text-free models also recommend items with textual similarities, pointing to a fundamental tendency within recommender systems. On the other side, as evidenced by Figure 6, where we have mapped out the proportion of repetitive item categories within the training set, it's clear that the practice of suggesting items within similar categories inherently fulfills the preferences of certain user demographics. However, when deploying LLM-based recom-

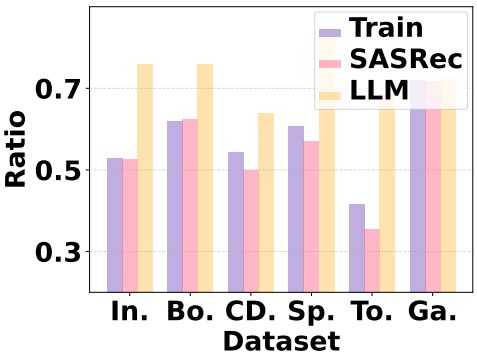


Figure 6: This figure displays the ratio of the training set's ground truth category, item categories of top-1 recommendations by traditional text-free models, and LLM-based recommender systems appear in the user historical interactions.

mender systems, their inherent copy mechanisms tend to exaggerate this feature, which might detract from the experience of a broader user base. Consequently, our findings advocate for a strategic adjustment to this issue, aiming for moderation rather than elimination, to balance user satisfaction across the spectrum optimally.