Revisiting CF-Integrated LLM Recommenders through Dimensional Collapse

Seo Jimin
jimin.seo@kaist.ac.kr
KAIST
Daejeon, Republic of Korea

Hongseok Kang
ghdtjr0311@kaist.ac.kr
KAIST
Daejeon, Republic of Korea

Sein Kim
rlatpdlsgns@kaist.ac.kr
KAIST
Daejeon, Republic of Korea

Jiwan Kim
kim.jiwan@kaist.ac.kr
KAIST
Daejeon, Republic of Korea

Chanyoung Park*
cy.park@kaist.ac.kr
KAIST
Daejeon, Republic of Korea

Abstract

Although recent work integrates large language models (LLMs) with collaborative filtering (CF) recommenders by projecting pre-trained CF item embeddings into the LLM token space, whether such CF-Integrated LLM recommenders faithfully preserve and exploit the underlying collaborative signals remains unclear. In this work, we systematically revisit three representative CF-Integrated LLM recommenders and show that their recommendation behavior diverges substantially from that of the base CF model, despite sharing the same pre-trained item embeddings. Empirically, their gains are highly dataset-dependent: they improve recommendation performance on popularity-skewed datasets but often underperform on datasets with more balanced item distributions. Further analysis reveals that these gains are accompanied by a severe loss of recommendation diversity and stronger concentration on popular items, rather than improved recommendations of unpopular ones. We attribute this behavior to dimensional collapse in the learned item representation space, where effective rank drops sharply during training. To mitigate this issue, we propose a simple orthogonal regularization that alleviates dimensional collapse, restores recommendation diversity, and generally improves recommendation performance across multiple base CF models. Our findings suggest that the apparent benefits of CF-Integrated LLM recommenders often stem from amplified popularity bias rather than faithful utilization of collaborative knowledge.

CCS Concepts

• **Information systems → Recommender systems**.

Keywords

Recommender System, LLM-based Recommender System, Popularity Bias, Dimensional Collapse

*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
GALOP Workshop @ KDD 2026, Seoul, Republic of Korea
© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

In the era of information explosion, recommender systems play a pivotal role in delivering items tailored to specific user preferences across diverse e-commerce platforms [4, 32]. As recommender systems have become widely adopted across e-commerce and content platforms, a broad range of machine learning-based recommendation algorithms has been studied [8, 9, 40].

Traditionally, ID-based recommender systems [15, 24, 28, 37] have achieved significant success in providing personalized recommendations through Collaborative Filtering (CF) [29, 35], which identifies similar users and items based on their interaction patterns. These methods represent each item with a randomly assigned ID and learn its embedding using interaction data, capturing collaborative knowledge. However, this reliance on interaction-driven learning mechanisms leads to poor performance for tail users and items with limited interactions, a challenge commonly referred to as the cold-start problem [1, 13, 25].

To address this limitation, recent studies have leveraged Large Language Models (LLMs). By harnessing their extensive world knowledge and ability to understand textual item descriptions, these models can effectively infer user preferences even for items with sparse or no interaction data [5, 12, 20, 38]. While these LLM-based approaches have demonstrated efficacy in cold-start and cross-domain scenarios, their heavy reliance on textual semantics often results in suboptimal performance compared to traditional CF models in scenarios where sufficient interaction data is available (i.e., warm scenarios) [26]. To bridge this gap, recent frameworks such as A-LLMRec [26], LLaRA [30], and LLM-SRec [27] aim to combine collaborative knowledge with textual understanding within a unified architecture. These methods project item embeddings from a pre-trained CF model (base CF model), such as SASRec [24], into the LLM token space through an alignment adapter, enabling the LLM to leverage collaborative signals alongside textual semantics. We refer to this line of work as **CF-Integrated LLMRecs** throughout this paper.

Existing CF-Integrated LLMRecs methods assume that LLMs can leverage the collaborative knowledge encoded in pre-trained CF embeddings—behavioral patterns from user-item interactions that textual information alone cannot capture. We first examine whether CF-Integrated LLMRecs and the base CF model (i.e., SASRec) actually produce similar recommendations, given that they share the same item embeddings. Specifically, for each model we identify the set of users whose ground-truth item appears in the top-10

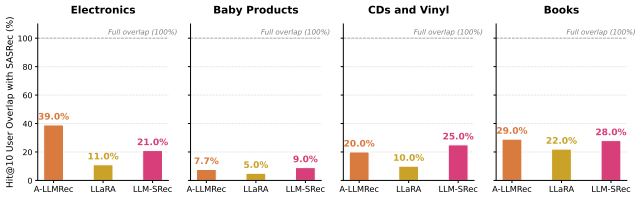

Figure 1: Hit@10 user overlap with SASRec across CF-Integrated LLMRecs. For each model, we report the proportion of its Hit@10 users that also appear in SASRec's Hit@10 user set. Despite sharing the same pre-trained CF item embeddings, CF-Integrated LLMRecs succeed on a largely disjoint set of users from SASRec across all datasets.

recommendations (Hit@10), and measure the overlap between the two sets. As illustrated in Figure 1, the overlap is surprisingly small: the hit users of CF-Integrated LLMRecs barely coincide with those of SASRec. This raises a question of whether performance gains of CF-Integrated LLMRecs truly stem from the use of collaborative knowledge encoded in CF embeddings.

Motivated by this observation, we evaluate the recommendation performance of CF-Integrated LLMRecs across six datasets and find that their performance gains are not consistent — they outperform the base CF model on some datasets but substantially underperform on others. To examine this inconsistency, we analyze the recommendation results in detail and find that *CF-Integrated LLMRecs universally suffer from a significant reduction in recommendation diversity*. We trace this to a geometric cause: item embeddings undergo dimensional collapse during training — the effective rank drops sharply, concentrating representation along a few dominant axes. This diversity loss further explains the inconsistent gains: on datasets where the training data already exhibits strong popularity bias, reduced diversity coincides with the data bias and concentrates recommendations on popular items, producing apparent performance gains that mask the underlying collapse rather than reflecting effective utilization of collaborative knowledge.

To mitigate this issue, we propose a simple regularization strategy that encourages the projection layer to preserve the geometric structure of item embeddings, effectively mitigating dimensional collapse and restoring recommendation diversity, which in turn alleviates the excessive popularity bias on popularity-skewed datasets. The main contributions of this paper are as follows:

- We revisit CF-Integrated LLM recommenders and show that, despite sharing the same pre-trained CF item embeddings, their recommendation behavior differs substantially from that of the base CF model.
- We show that their performance is highly dataset-dependent, with gains mainly occurring on popularity-skewed datasets alongside lower diversity and stronger popularity bias.
- We propose a simple regularization method that mitigates dimensional collapse, improving recommendation diversity as well as the overall recommendation performance.

2 Preliminary

Problem Definition: Sequential Recommendation. Given a set of M users $\mathcal{U}$ and a set of N items $\mathcal{I}$, sequential recommendation aims to predict the next item a user is likely to interact with based on their historical interactions. Let $\mathcal{S}_u = (i_{u,1}, i_{u,2}, \ldots, i_{u,|\mathcal{S}_u|})$ denote the sequence of items that user u has interacted with, where $|\mathcal{S}_u|$ is the length of the sequence $\mathcal{S}_u$. The goal of sequential recommendation is to model the probability $P(i_{u,|\mathcal{S}_u|+1} \mid \mathcal{S}_u)$.

In traditional CF-based sequential recommenders (e.g., SASRec [24], GRU4Rec [18], BERT4Rec [37]), each item $i \in \mathcal{I}$ is typically represented as a high-dimensional ID-based embedding vector $\mathbf{e}_i \in \mathbb{R}^{d_{CF}}$. These embeddings are trained to capture collaborative signals and sequential patterns.

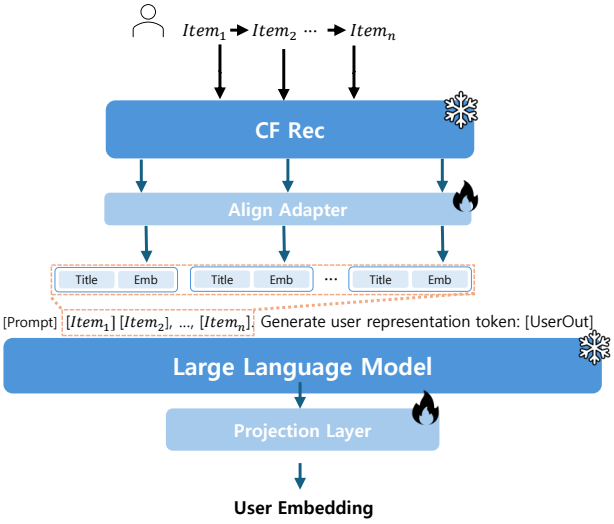

Figure 2: Overview of CF-Integrated LLMRecs, illustrated with the retrieval-based variant, adapted from [27]. [Prompt] denotes an instruction prompt prepended to the hybrid LLM input; a detailed prompt template is provided in Appendix A

2.1 CF-Integrated LLMRecs

CF-Integrated LLMRecs aim to combine the textual understanding of LLMs with CF knowledge by injecting pre-trained CF item embeddings into the LLM token space. As illustrated in Figure 2, the resulting LLM hidden states are then projected to obtain user embeddings.

Adapter. A trainable adapter $f_\theta : \mathbb{R}^{d_{\text{CF}}} \to \mathbb{R}^{d_{\text{LLM}}}$ maps each item's CF embedding into the LLM token space:

$$\mathbf{z}_i = f_\theta(\mathbf{e}_i), \qquad \forall i \in \mathcal{I}. \tag{1}$$

The adapter is applied to all items in advance, and the resulting adapted embeddings are later retrieved according to a user's interaction sequence.

Hybrid Prompt. The adapted embeddings $\mathbf{z}_i$ are paired with the corresponding title token embeddings $\mathbf{t}_i$ and interleaved with an instruction prompt to form a hybrid prompt fed into the LLM:

$$\mathbf{P}_u = [\texttt{prompt};\ (\mathbf{t}_{i_{u,1}}, \mathbf{z}_{i_{u,1}});\ \ldots;\ (\mathbf{t}_{i_{u,l_u}}, \mathbf{z}_{i_{u,|\mathcal{S}_u|}})]. \tag{2}$$

Next-Item Prediction. LLaRA [30] and A-LLMRec [26] adopt a generative approach, where candidate items are provided, and the LLM generates the next item's title. LLM-SRec [27] adopts a retrieval approach, where the LLM produces a user representation

that is compared against item representations (obtained analogously through the LLM) for next-item prediction. In our analysis, we adopt all three CF-Integrated LLMRecs (i.e., LLaRA, A-LLMRec, and LLM-SRec) with SASRec [24] as the shared base CF model, following their original implementations. We used pretrained LLaMA 3.2 (3B-Instruct) as the LLM backbone for all CF-Integrated LLMRecs.

2.2 Dimensional Collapse and Effective Rank

Dimensional Collapse (DC) is a phenomenon where learned representations are confined to a significantly lower-dimensional subspace than their nominal capacity, restricting the model's ability to encode diverse and fine-grained information [22, 43]. To quantify DC, we use the Effective Rank [31, 36], which measures how uniformly the singular values of the embedding matrix are distributed. A higher effective rank indicates that the embedding space is well-utilized across multiple dimensions, while a lower value signals that a few dominant dimensions capture most of the variance, indicating dimensional collapse.

3 Revisiting CF-Integrated LLMRecs

In this section, we revisit a common assumption underlying CF-Integrated LLMRecs. These methods assume that, by projecting pre-trained CF item embeddings into the token space of an LLM, the model can effectively preserve and exploit the collaborative knowledge encoded in the original base CF model. If this assumption holds, a CF-Integrated LLMRecs should, at minimum, correctly recommend for every user already handled correctly by the base CF model, while ideally extending correct recommendations to additional users through textual knowledge.

To validate this assumption, we first compare CF-Integrated LLMRecs with SASRec in terms of Hit@10 user overlap and overall recommendation performance across diverse datasets (Section 3.1). We then analyze their recommendation behavior in greater detail to understand where the observed differences arise (Section 3.2). Note that in this section, we use the following datasets and evaluation protocol for experiments.

Datasets. We conduct experiments on six Amazon Review datasets. We adopt the 2023 release [19] following LLM-SRec [27], covering Electronics, CDs and Vinyl, Baby Products, Musical Instruments, and Books. For Toys and Games, which is available in both releases, we use the 2018 version [33] to ensure direct comparability with A-LLMRec's original evaluation. The statistics are shown in Table 1.

Evaluation Protocol. We adopt the leave-one-out strategy [24, 37, 45], using the last item of each user's interaction sequence for testing and the second-to-last for validation. To reflect real-world deployment settings, we rank each user's ground-truth item against all items rather than a sampled set of negatives as in prior work [26, 27, 30]. Since the generative approach adopted by LLaRA and A-LLMRec cannot accommodate the full item candidate within the LLM's context window, we unify all three methods under the retrieval approach following LLM-SRec [27]. We adopt SASRec [24] as the shared base CF model across all CF-Integrated LLMRecs.

3.1 Does CF Integration Consistently Help?

Recall that existing CF-Integrated LLMRecs, such as A-LLMRec, LLaRA, and LLM-SRec, are motivated by the idea that injecting collaborative signals from a pre-trained CF model into an LLM enables the model to leverage CF knowledge for recommendation. Hence, if CF integration truly works as intended, its benefit should be observed not only as improved accuracy, but also as a substantial preservation of the users correctly served by the base CF model, i.e., SASRec, along with reasonably consistent gains across datasets. To test whether such preservation actually holds, we conduct a user-level overlap analysis: for each CF-Integrated LLMRec, we identify the set of users for whom the model achieves a hit at rank 10 (Hit@10), and measure how much this set overlaps with the corresponding hit-user set of SASRec.

Table 1: Statistics of datasets after preprocessing.

Dataset	# Users	# Items	# Interactions
Electronics	27,601	31,533	256,895
Baby Products	11,560	10,239	98,545
Musical Instruments	12,986	13,606	149,744
CDs and Vinyl	18,481	30,951	256,418
Books	22,499	40,289	270,479
Toys and Games	28,587	42,604	330,344

Table 2: Recommendation Behavioral Overlap Analysis: Intersection of Hit Users between SASRec and CF-Integrated LLMRecs. "∩ SASRec" denotes the number of users for whom both SASRec and the CF-Integrated LLMRec correctly recommend the ground-truth item in the top-10.

		Electronics	Baby	CDs	Books
SASRec	# Hit Users	128	135	435	553
A-LLMRec	# Hit Users	147	143	176	274
	∩ SASRec	57	11	35	80
	Overlap ratio	39%	7.7%	20%	29%
LLaRA	# Hit Users	153	144	99	221
	∩ SASRec	17	7	10	48
	Overlap ratio	11%	5%	10%	22%
LLM-SRec	# Hit Users	219	162	200	174
	∩ SASRec	46	15	50	49
	Overlap ratio	21%	9%	25%	28%

In Table 2, we find that the user-level overlap between SASRec and CF-Integrated LLMRecs is surprisingly low. More precisely, the overlap ratio — the proportion of CF-Integrated LLMRecs' Hit@10 users that also appear among SASRec's Hit@10 users — is remarkably low across all models and datasets, ranging from 5% to 39%. This indicates that CF-Integrated LLMRecs succeed on a largely different set of users than SASRec, despite sharing the same item embeddings.

Given this behavioral divergence, one might still expect CF-Integrated LLMRecs to compensate for the lack of preservation by delivering consistently stronger accuracy across datasets. To examine whether this is the case, Table 3 compares the recommendation performance of SASRec and three CF-Integrated LLMRecs across six datasets. We have the following observations: 1) The benefit of CF integration is highly inconsistent across datasets. CF-Integrated LLMRecs outperform SASRec on Electronics, Baby Products, and Musical Instruments, but substantially underperform on CDs and Vinyl, Books, and Toys and Games. In some cases, the gap is large; for example, on CDs and Vinyl, SASRec achieves an NDCG@10 of 0.0238, more than twice that of the best CF-Integrated LLMRec

Table 3: Performance comparisons between SASRec and CF-Integrated LLMRecs across six datasets.

Dataset	Model	Hit@10	Hit@20	NDCG@10	NDCG@20
Electronics	SASRec	0.0128	0.0220	0.0067	0.0091
	A-LLMRec	0.0147	0.0242	0.0066	0.0090
	LLaRA	0.0153	0.0238	0.0066	0.0088
	LLM-SRec	**0.0219**	**0.0366**	**0.0111**	**0.0148**
Baby	SASRec	0.0149	0.0252	0.0068	0.0094
	A-LLMRec	0.0152	0.0246	0.0069	0.0092
	LLaRA	0.0153	0.0242	0.0075	0.0097
	LLM-SRec	**0.0172**	**0.0330**	**0.0080**	**0.0120**
Musical	SASRec	0.0201	0.0375	0.0100	0.0146
	A-LLMRec	0.0211	0.0364	0.0099	0.0138
	LLaRA	0.0236	0.0395	0.0122	0.0162
	LLM-SRec	**0.0269**	**0.0423**	**0.0137**	**0.0175**
CDs	SASRec	**0.0435**	**0.0610**	**0.0238**	**0.0275**
	A-LLMRec	0.0176	0.0317	0.0089	0.0125
	LLaRA	0.0099	0.0172	0.0051	0.0069
	LLM-SRec	0.0200	0.0329	0.0097	0.0130
Books	SASRec	**0.0557**	**0.0796**	**0.0311**	**0.0372**
	A-LLMRec	0.0274	0.0470	0.0141	0.0190
	LLaRA	0.0221	0.0353	0.0105	0.0138
	LLM-SRec	0.0174	0.0302	0.0089	0.0121
Toys	SASRec	**0.0488**	**0.0624**	**0.0370**	**0.0404**
	A-LLMRec	0.0256	0.0367	0.0152	0.0180
	LLaRA	0.0268	0.0418	0.0144	0.0182
	LLM-SRec	0.0301	0.0431	0.0184	0.0217

(0.0097). 2) Their gains do not appear to come from preserving the recommendation strengths of SASRec even on the datasets where CF-Integrated LLMRecs achieve higher accuracy. The low Hit@10 overlap on these datasets indicates that CF-Integrated LLMRecs improve accuracy by succeeding on a largely disjoint set of users, rather than by strengthening the successful predictions of SASRec.

These results raise two key questions: (1) when performance gains are observed, what drives them? and (2) under what dataset conditions does CF integration lead to such gains? To answer these questions, we next analyze the characteristics of the items that CF-Integrated LLMRecs rank highly, and which dataset properties are associated with performance gains or degradation in Section 3.2.

3.2 Where Do the CF-Integrated LLMRec's Performance Gains Come From?

3.2.1 What Drives the Gains? Given the low user-level overlap with SASRec, we hypothesize that CF-Integrated LLMRecs succeed on a different set of users because the textual understanding ability of LLMs helps rank sparsely interacted, unpopular items that SASRec fails to rank highly. To investigate this, we stratify items into Popular (top 5%) and Unpopular (bottom 50%) groups by training-set frequency, and report both Hit@10 and average rank (the average position of the ground-truth item among all candidates). The former captures top-k recommendation performance, while the latter captures ranking improvements that do not reach the top-10 but may still reflect meaningful modeling of unpopular items.

Table 4: Hit@10 and Average Rank stratified by item popularity. Popular: top 5%, Unpopular: bottom 50% by training-set frequency. Hit@10: higher is better. AvgRank: lower is better.

		Popular (top 5%)		Unpopular (bottom 50%)	
Dataset	Model	Hit@10 ↑	AvgRank ↓	Hit@10 ↑	AvgRank ↓
Electronics	SASRec	0.0199	2,613	**0.0010**	16,213
	A-LLMRec	0.0447	2,259	0.0009	12,165
	LLaRA	0.0553	4,598	0.0000	12,249
	LLM-SRec	**0.0797**	**1,888**	0.0002	**11,399**
Baby	SASRec	0.0329	772	**0.0041**	5,542
	A-LLMRec	0.0500	570	0.0000	5,822
	LLaRA	0.0506	1,374	0.0031	6,372
	LLM-SRec	**0.0588**	**491**	0.0018	**4782**
Musical	SASRec	0.0442	1,069	**0.0047**	6,345
	A-LLMRec	0.0363	984	0.0024	4,574
	LLaRA	0.0646	1,531	0.0039	**3,680**
	LLM-SRec	**0.0723**	**828**	0.0047	4,673
CDs	SASRec	**0.0694**	2,279	**0.0105**	10,877
	A-LLMRec	0.0408	**2,253**	0.0048	7,141
	LLaRA	0.0242	3,433	0.0041	6,602
	LLM-SRec	0.0489	2,479	0.0046	**6,511**
Books	SASRec	**0.0684**	2,209	**0.0214**	14,116
	A-LLMRec	0.0694	**1,708**	0.0048	10,636
	LLaRA	0.0412	3,457	0.0082	**6447**
	LLM-SRec	0.0499	2,126	0.0020	11,481
Toys	SASRec	**0.0512**	3,382	**0.0112**	14,568
	A-LLMRec	0.0658	3,911	0.0113	11,314
	LLaRA	0.0681	5,420	0.0118	**7,952**
	LLM-SRec	0.0748	**3,022**	0.0130	11622

Contrary to our hypothesis, Table 4 shows that CF-Integrated LLMRecs do not consistently outperform SASRec on unpopular items under Hit@10. In particular, on the three datasets where CF-Integrated LLMRecs outperform SASRec (Electronics, Baby Products, and Musical Instruments) as shown in Table 3, we observe in Table 4 that their Hit@10 gains are mainly driven by popular items, while their Hit@10 on unpopular items remains comparable to, or even lower than, that of SASRec. Under AvgRank, however, CF-Integrated LLMRecs consistently outperform SASRec on unpopular items across all six datasets, with the average rank on Books dropping from 14,116 to 6,447.

These results suggest that while the textual knowledge of LLMs provides a useful signal for unpopular items—pushing them to generally higher positions in the ranked list, as reflected by the substantial AvgRank improvements—this signal is typically not strong enough to elevate them into the top-10, where Hit@10 is determined. In other words, LLMs help unpopular items rank higher overall, but rarely high enough to be actually recommended. This indicates that the Hit@10 gains of CF-Integrated LLMRecs observed in Table 3 do not primarily come from better recommendation of unpopular items. Rather, they are more likely to arise because CF-Integrated LLMRecs rank popular items more effectively within the top-10 than SASRec.

To verify this, we next analyze the actual items in the top-10 recommendation lists directly along two complementary axes: **(1) popularity concentration** (Pop@5%), defined as the proportion

Table 5: Popularity bias and catalog coverage of top-10 recommendations across all six datasets. Pop@5% denotes the proportion of recommendations occupied by the top 5% most popular items (by training-set frequency). *Cov.* is the number of unique items appearing in the union of the top-10 recommendation lists across all users.

	Electronics		Baby Products		Musical Inst.		CDs and Vinyl		Books		Toys and Games	
	Pop@5%	Cov.	Pop@5%	Cov.	Pop@5%	Cov.	Pop@5%	Cov.	Pop@5%	Cov.	Pop@5%	Cov.
SASRec	54.4	16,612	49.0	6,446	43.0	14,922	33.7	15,214	44.8	15,635	35.8	22,721
A-LLMRec	54.4	3,792	78.5	71	62.9	1,258	30.1	3,730	46.9	5,735	20.9	9,807
LLaRA	88.8	14	86.0	60	43.3	1,890	28.1	1,437	30.2	3,370	25.6	9,788
LLM-SRec	85.5	1,618	88.6	118	64.5	1,287	29.8	7,710	53.7	976	26.5	11,993

of recommended items that belong to the top 5% most popular items in the training data, and **(2) catalog coverage** [17, 23], defined as the number of unique items appearing in the union of the top-10 recommendation lists across all users [3]. Intuitively, if the model recommends largely overlapping items to different users (e.g., suggesting a similar top-10 list to User 1 and User 2), these lists collapse into a small union, yielding low catalog coverage. In this sense, catalog coverage directly reflects the diversity of recommendations at the system level: a low coverage indicates that the model concentrates its recommendations on a narrow subset of the catalog, repeatedly exposing the same items to different users, and thus corresponds to low recommendation diversity. For brevity, we denote catalog coverage as Cov. in all result tables.

As shown in Table 5, CF-Integrated LLMRecs exhibit a severe reduction in recommendation diversity (i.e., low catalog coverage) across all six datasets. For example, on Baby Products, LLM-SRec recommends only 118 unique items, whereas SASRec recommends 6,446. Meanwhile, we found that such loss in recommendation diversity is accompanied by amplified popularity concentration (i.e., high Pop@5%) only on Electronics, Baby Products, and Musical Instruments, where CF-Integrated LLMRecs also achieve higher overall accuracy as shown in Table 3. On these datasets, the proportion of recommendations occupied by the top 5% most popular items rises sharply, reaching as high as 88.6% compared with 49.0% for SASRec in Baby Products. In contrast, on the remaining datasets, CF-Integrated LLMRecs still exhibit substantial loss in recommendation diversity (i.e., low coverage) but without a corresponding concentration on highly popular items (i.e., low Pop@5%), which is accompanied by their inferior recommendation performance compared with SASRec. In summary, these results suggest that *CF-Integrated LLMRecs' observed gains in recommendation performance (Table 3) are better explained by stronger concentration on popular items within the top-10 recommendation lists than by improved recommendation of unpopular items.*

3.2.2 When Does Popularity-Oriented Behavior Help? To further understand when this popularity-oriented behavior becomes beneficial, we analyze the popularity concentration of the training data itself. To characterize this, we report two dataset-level statistics: 1) Top 5% Popularity, the proportion of total interactions accounted for by the top 5% most popular items, 2) Gini, which quantifies the inequality of the popularity distribution. Formally, let $p(i_k)$ denote the normalized interaction frequency of item i_k in the training set, where items are sorted in ascending order of frequency. The Gini coefficient is then defined as:

$$G = 1 - \frac{1}{N-1} \sum_{k=1}^{N} (2k - N - 1)\, p(i_k), \tag{3}$$

where $p(i_k) = \frac{f_{i_k}}{\sum_{j \in I} f_j}$ and f_{i_k} is the number of training interactions of item i_k. Higher values of G indicate a more skewed interaction distribution, where a smaller subset of items accounts for a larger share of the training interactions [2, 11].

Table 6: Popularity concentration of the training data. Higher Top 5% Popularity and Gini Coefficient indicate stronger popularity skew. Electronics, Baby, and Musical exhibit stronger popularity skew than CDs, Books, and Toys.

	Electronics	Baby	Musical	CDs	Books	Toys
Top 5% Popularity	35.0%	35.1%	33.9%	23.2%	27.7%	29.9%
Gini Coefficient	0.547	0.578	0.539	0.431	0.459	0.511

Table 6 shows that the three datasets where CF-Integrated LLMRecs outperform SASRec (i.e., Electronics, Baby Products, and Musical Instruments) also exhibit relatively stronger concentration of interactions on popular items, with the Top 5% Popularity accounting for 33.9%–35.1% of all interactions. In contrast, the datasets where CF-Integrated LLMRecs underperform SASRec show substantially lower top-5% concentration, ranging from 23.2% to 29.9%. A similar tendency is observed in the Gini coefficients, indicating that interaction distributions in the gain datasets (i.e., Electronics, Baby Products, and Musical Instruments) are overall more skewed toward a small set of popular items. These results suggest that *popularity amplification is more likely to translate into accuracy gains when the underlying training data is already heavily concentrated on popular items.*

Summary. Based on our observations, we argue that the gains of CF-Integrated LLMRecs are not mainly driven by better recommendations of unpopular items. Rather, their gains emerge when the collapse of recommendations onto a narrower item set aligns with datasets whose interaction distributions are already strongly skewed toward popular items. Consistent with Table 6, the datasets where CF-Integrated LLMRecs outperform SASRec exhibit higher popularity concentration in the training data, whereas the datasets where they underperform are relatively more uniform. In other words, we argue that CF integration yields gains not by faithfully extending the collaborative strengths of the base CF model, but often by amplifying popularity bias under dataset conditions where such bias happens to be beneficial.

Table 7: Effective rank of item embeddings. SASRec operates in 64 dimensions, while all CF-Integrated LLMRecs operate in 128 dimensions. A lower effective rank indicates more severe dimensional collapse.

Dataset	SASRec	A-LLMRec	LLaRA	LLM-SRec
Electronics	61.98	3.71	2.89	14.94
Baby Products	62.28	2.47	4.29	10.31
Musical Inst.	62.36	2.77	9.60	11.40
CDs and Vinyl	61.71	3.31	10.30	5.32
Books	61.54	4.86	5.60	8.32
Toys and Games	62.35	8.16	15.47	11.02

Table 8: Effective rank of item embeddings before and after applying the proposed orthogonal regularization. Higher effective rank indicates that more dimensions are effectively utilized; nominal dimension is 128.

	A-LLMRec		LLaRA		LLM-SRec	
Dataset	Base	+Reg	Base	+Reg	Base	+Reg
Electronics	3.71	5.27	2.89	23.26	14.94	28.29
Baby	2.47	6.56	4.29	17.48	10.31	19.89
Musical	2.77	29.63	9.60	12.58	11.40	29.15
CDs	3.31	19.23	10.30	15.76	5.32	20.16
Books	4.86	7.32	5.60	16.75	8.32	24.24
Toys	8.16	16.97	15.47	24.70	11.02	33.72

4 Diagnosing and Mitigating Dimensional Collapse

In Section 3.2, we observed that CF-Integrated LLMRecs consistently suffer from reduced recommendation diversity. On datasets with skewed interaction distributions, the issue is further accompanied by amplified popularity bias. This naturally raises the next question: what happens inside CF-Integrated LLMRecs during training that leads to such behavior? Motivated by prior work showing that dimensional collapse provides a useful lens for analyzing popularity-oriented recommendation behavior [43], we examine whether CF-Integrated LLMRecs also exhibit dimensional collapse in their item embedding space (Section 4.1). We then propose a simple regularization strategy to mitigate this collapse (Section 4.2).

4.1 Dimensional Collapse in CF-Integrated LLMRecs

To quantify the extent of dimensional collapse in CF-Integrated LLMRecs, we measure the effective rank of the item embedding matrix. Specifically, we use effective rank as a diagnostic measure of how many dimensions are effectively utilized in the embedding space [31, 36, 43]; lower values indicate that the representations are concentrated along a small number of dominant directions, reflecting more severe collapse.

Table 7 reports the effective rank of the item embedding matrix for each model and dataset. Across all six datasets, CF-Integrated LLMRecs exhibit a significant drop in effective rank relative to SASRec, despite operating in a higher-dimensional embedding space (128 vs. 64). While SASRec maintains an effective rank close to its nominal capacity (61–62), CF-Integrated LLMRecs collapse to values between 2 and 15, indicating that their item representations concentrate on a very small number of dominant directions. This pattern holds uniformly across datasets, regardless of whether CF-Integrated LLMRecs outperform SASRec, mirroring the universal diversity loss observed in Table 5.

We acknowledge that effective rank is a geometric property of the embedding space rather than a direct measure of recommendation diversity — items could in principle remain separable even within a low-rank subspace. Nevertheless, the severity of the collapse observed here (to fewer than 10 dimensions in most cases, from a nominal 128) imposes a practical limit on how finely items can be differentiated, consistent with the universal loss in the recommendation diversity reported in Table 5.

4.2 Mitigating Dimensional Collapse with Regularization

The analyses in Sections 3 and 4.1 suggest that CF-Integrated LLMRecs exhibit two coupled phenomena: reduced recommendation diversity and severe dimensional collapse in the item embedding space. This naturally raises the next question: *can mitigating dimensional collapse restore recommendation diversity?* If dimensional collapse indeed plays an important role in the observed behavior, then alleviating it might restore the geometry of the embedding space and thereby preserve recommendation diversity.

To this end, we introduce a simple yet effective orthogonal regularization that encourages the learned item embeddings to preserve their geometric spread and avoid collapse into a few dominant directions. As illustrated in Figure 2, CF-Integrated LLMRecs require a projection layer to map the hidden representation of the LLM ($\mathbb{R}^{d_{LLM}}$) to the recommendation space ($\mathbb{R}^{d_{CF}}$), where $d_{LLM} \gg d_{CF}$ (e.g. 3072 → 128). This dimensionality reduction is a critical bottleneck: when the projection layer collapses its output into a low-dimensional subspace, the resulting item representations lose the diversity needed for effective recommendation. To prevent this, we encourage the row vectors of the projection weight matrix to remain orthogonal by penalizing the deviation of its Gram matrix from the identity matrix:

$$\mathcal{L}_{ortho} = |\mathbf{W}\mathbf{W}^{\top} - \mathbf{I}|_F \tag{4}$$

where $\mathbf{W}$ denotes the weight matrix of a linear layer in the projection module and $|\cdot|_F$ denotes the Frobenius norm. We sum the regularization term over all layers in the projection module. The total training objective becomes:

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda \mathcal{L}_{ortho} \tag{5}$$

where $\mathcal{L}_{rec}$ is the original recommendation loss and λ controls the strength of the regularization. By encouraging the projection weights to preserve diversity across output dimensions, this regularization helps prevent item embeddings from collapsing into a few dominant directions. Note that we fix the λ to 0.01 for all experiments, as extensive tuning is prohibitively expensive for LLM training.

5 Experiments

In this section, we first examine whether the proposed regularization effectively alleviates dimensional collapse (Section 5.1), and

Table 9: Recommendation performance and diversity comparison across datasets. *Cov.* denotes the number of unique items appearing in the union of the top-10 recommendation lists across all users. Electronics, Baby Products, and Musical Instruments are relatively more popularity-skewed than CDs and Vinyl, Books, and Toys and Games (See Table 6).

	Electronics					**CDs and Vinyl**				
Model	Hit@10	Hit@20	NDCG@10	NDCG@20	Cov.	Hit@10	Hit@20	NDCG@10	NDCG@20	Cov.
SASRec	0.0128	0.0220	0.0067	0.0091	16,612	**0.0435**	**0.0610**	**0.0238**	**0.0275**	15,214
A-LLMRec	0.0147	0.0242	0.0066	0.0090	3,792	0.0176	0.0317	0.0089	0.0125	3,730
A-LLMRec + Reg	0.0155	0.0248	0.0067	0.0091	3,946	0.0226	0.0375	0.0107	0.0144	6,544
LLaRA	0.0153	0.0238	0.0066	0.0088	14	0.0099	0.0172	0.0051	0.0069	1,437
LLaRA + Reg	0.0243	0.0389	0.0110	0.0147	1,021	0.0163	0.0259	0.0079	0.0103	4,672
LLM-SRec	0.0219	0.0366	0.0111	0.0148	1,618	0.0200	0.0329	0.0097	0.0130	7,710
LLM-SRec + Reg	**0.0241**	**0.0361**	**0.0124**	**0.0154**	1,766	0.0245	0.0417	0.0119	0.0162	8,294
	Baby Products					**Books**				
Model	Hit@10	Hit@20	NDCG@10	NDCG@20	Cov.	Hit@10	Hit@20	NDCG@10	NDCG@20	Cov.
SASRec	0.0149	0.0252	0.0068	0.0094	6,446	**0.0557**	**0.0796**	**0.0311**	**0.0372**	15,635
A-LLMRec	0.0152	0.0246	0.0069	0.0092	71	0.0274	0.0470	0.0141	0.0190	5,735
A-LLMRec + Reg	0.0196	0.0318	0.0094	0.0125	4,041	0.0295	0.0506	0.0147	0.02	7,089
LLaRA	0.0153	0.0242	0.0075	0.0097	60	0.0221	0.0353	0.0105	0.0138	3,370
LLaRA + Reg	0.0211	0.0338	0.0106	0.0138	64	0.0304	0.0507	0.0155	0.0206	8,298
LLM-SRec	0.0172	0.0330	0.0080	0.0120	118	0.0174	0.0302	0.0089	0.0121	976
LLM-SRec + Reg	**0.0290**	**0.0461**	**0.0142**	**0.0184**	821	0.0387	0.0604	0.0209	0.0264	13,665
	Musical Instruments					**Toys and Games**				
Model	Hit@10	Hit@20	NDCG@10	NDCG@20	Cov.	Hit@10	Hit@20	NDCG@10	NDCG@20	Cov.
SASRec	0.0201	0.0375	0.0100	0.0146	14,922	**0.0488**	**0.0624**	**0.0370**	**0.0404**	22,721
A-LLMRec	0.0211	0.0364	0.0099	0.0138	1,225	0.0256	0.0367	0.0152	0.0180	9,807
A-LLMRec + Reg	0.0207	0.0362	0.0100	0.0138	3,749	0.0296	0.0404	0.0177	0.0204	10,050
LLaRA	0.0236	0.0395	0.0122	0.0162	1,890	0.0268	0.0418	0.0144	0.0182	9,788
LLaRA + Reg	0.0243	0.0378	0.0127	0.0161	2,138	0.0265	0.0416	0.0147	0.0185	10,929
LLM-SRec	0.0269	0.0423	0.0137	0.0175	1,287	0.0301	0.0431	0.0184	0.0217	11,993
LLM-SRec + Reg	**0.0337**	**0.0506**	**0.0173**	**0.0215**	1,793	0.0361	0.0457	0.0261	0.0285	13,982

then evaluate how such geometric changes affect recommendation diversity and overall recommendation performance (Section 5.2). We further examine the effect of the hyper-parameter λ (Section 5.3) and whether the benefit of the proposed regularization generalizes to other CF base models beyond SASRec (Section 5.4). Lastly, we analyze whether the regularization helps preserve the CF knowledge encoded in the base CF model by examining the user-level behavioral overlap with SASRec before and after applying regularization (Section 5.5).

5.1 Effective Rank

To verify that the proposed regularization indeed mitigates dimensional collapse in CF-Integrated LLMRecs, we compare the effective rank of the item embedding matrix before and after applying the regularizer. Table 8 reports the results. Overall, the regularized models consistently exhibit higher effective rank than their unregularized counterparts, indicating that the item embeddings make use of a broader subspace instead of collapsing onto only a few dominant directions. Figure 3 qualitatively corroborates this observation. The item embeddings of regularized LLM-SRec appear less concentrated than those of the unregularized model, which is consistent with the geometric recovery indicated by effective rank.

It is worth noting that the proposed regularization does not directly optimize recommendation diversity. Instead, it acts on the geometry of the representation space. Therefore, the increase in effective rank provides evidence that the regularizer successfully alleviates the internal collapse identified in Section 4.1.

5.2 Recommendation Diversity and Performance

Here, we examine whether the geometric recovery leads to improvements in recommendation diversity and overall recommendation quality. Table 9 reports the recommendation diversity and performance of CF-Integrated LLMRecs with and without the proposed orthogonal regularization across all six datasets. Overall, adding the proposed regularization consistently improves recommendation diversity (i.e., higher Cov.) relative to the unregularized counterparts. Together with the effective-rank results in Section 5.1, this trend is consistent with the idea that mitigating dimensional collapse helps recover recommendation diversity. Notably, the regularization also improves recommendation performance. As illustrated in Figure 3, orthogonal regularization alleviates the compressed geometry of

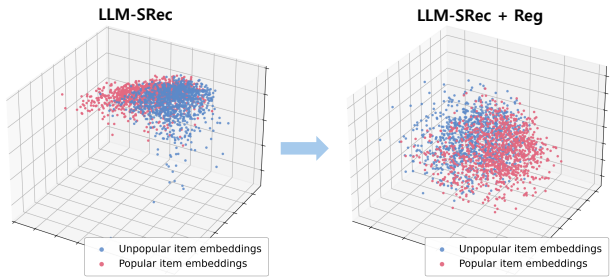

Figure 3: 3D PCA Visualization of item embeddings from LLM-SRec on Baby Products before and after applying the proposed regularization. We visualize items from the top 10% most popular and the bottom 10% least popular groups.

the item embedding space and allows the embeddings to spread over a broader subspace. This makes items more distinguishable, which in turn could help improve recommendation performance.

At the same time, we observed that the relationship between geometric recovery and diversity recovery is not strictly proportional across all models and datasets. In some cases, substantial improvement in effective rank is accompanied by only a modest increase in recommendation diversity. A representative example is LLaRA on Baby Products, where regularization clearly improves both effective rank and recommendation performance, while catalog coverage increases only marginally (60 $\rightarrow$ 64). This indicates that effective rank should be viewed as a geometric diagnostic of collapse rather than a direct surrogate for recommendation diversity. Instead, the main evidence is intervention-based: after applying regularization, collapse is alleviated, and recommendation diversity generally improves, although the magnitude of that recovery varies across models and datasets. These findings suggest that dimensional collapse is an important contributing factor to the diversity loss of CF-Integrated LLMRecs, even if it does not by itself determine the final diversity level. In summary, the magnitude of recovery varies across models and datasets, but the overall trend is clear: *orthogonal regularization reliably pushes the embedding space away from the severely collapsed regime.*

5.3 Sensitivity Analysis on λ

The orthogonal regularization weight λ in Equation 5 controls the trade-off between the recommendation objective and the geometric constraint on the projection layer. While we fixed λ=0.01 in the main experiments due to the high cost of LLM training, this section examines how the recommendation behavior varies with the strength of the regularization. We vary $\lambda \in \{0, 0.001, 0.01, 0.03, 0.05, 0.1\}$, where λ=0 corresponds to the unregularized baseline, using LLM-SRec as the representative CF-Integrated LLMRec on Baby Products (popularity-skewed) and CDs and Vinyl (more balanced).

On the popularity-skewed Baby Products, recommendation performance peaks at λ=0.01 (HR@10 0.0290, NDCG@10 0.0142) and degrades for larger values ($\lambda \geq 0.03$). On the more balanced CDs and Vinyl, in contrast, performance continues to improve as λ grows, reaching its highest value at λ=0.1 (HR@10 0.0297, NDCG@10 0.0148). This contrast suggests that on skewed datasets, moderate regularization is beneficial, as stronger regularization disrupts the popularity aligned signal on which accuracy partly relies in such settings; on balanced datasets, the model continues to benefit from stronger regularization.

Table 10: Sensitivity of LLM-SRec to the regularization strength λ on Baby Products and CDs and Vinyl. Best per dataset in bold.

	Baby Products			CDs and Vinyl		
λ	HR@10	NDCG@10	Cov.	HR@10	NDCG@10	Cov.
0	0.0172	0.0080	118	0.0200	0.0097	7,710
0.001	0.0235	0.0118	1,478	0.0243	0.0123	11,110
0.01	**0.0290**	**0.0142**	821	0.0245	0.0119	8,294
0.03	0.0202	0.0099	1,284	0.0252	0.0130	11,748
0.05	0.0195	0.0098	549	0.0279	0.0146	12,233
0.1	0.0210	0.0107	531	**0.0297**	**0.0148**	13,784

We further observe that the λ that maximizes accuracy does not necessarily coincide with the one that maximizes coverage. On Baby Products, λ=0.001 yields the highest catalog coverage (1,478) while the best recommendation performance is obtained at λ=0.01 with substantially lower coverage (821). This indicates that, beyond a certain point, additional diversity does not translate into accuracy gains—particularly on skewed datasets, where useful predictions tend to remain concentrated on a smaller subset of items. Despite these dataset dependent patterns, λ=0.01 consistently improves over the unregularized baseline (λ=0) on both datasets and across all metrics (e.g., HR@10 0.0172 $\rightarrow$ 0.0290 on Baby, and 0.0200 $\rightarrow$ 0.0245 on CDs), suggesting that it serves as a reasonable single default. At the same time, the analysis indicates additional headroom from per-dataset tuning, particularly on balanced datasets where larger λ values are beneficial.

5.4 Generalization to Other CF Base Model

To examine whether the proposed regularization generalizes beyond SASRec, we further apply it to LLM-SRec with GRU4Rec [18] as the base CF model. Note that we opt for LLM-SRec because, among the CF-Integrated LLM recommenders, it consistently shows the strongest performance in our experiments. As shown in Table 11, the proposed regularization exhibits a similar trend to that observed in Section 5.2: it generally improves recommendation diversity by increasing catalog coverage, while also enhancing overall recommendation performance. These results suggest that the benefit of the proposed regularization is not limited to a specific base CF model but extends to other CF recommenders as well.

5.5 CF Knowledge Preservation

We further investigate whether orthogonal regularization helps preserve the CF knowledge encoded in the item embeddings from the base CF model by mitigating dimensional collapse. Specifically, we compare the user-level behavioral overlap with SASRec before and after applying regularization. As shown in Table 12, the overlap with SASRec generally increases across models and datasets when regularization is applied. This suggests that alleviating dimensional collapse improves the preservation of the collaborative filtering knowledge encoded in the injected CF item embeddings, thereby

Table 11: Generalization of the proposed regularization to GRU4Rec as the base CF model.

Dataset	Model	Hit@10	Hit@20	NDCG@10	NDCG@20	Cov.
Electronics	GRU4Rec	0.0212	0.0310	0.0121	0.0146	10,499
	LLM-SRec	0.0264	0.0392	0.0138	0.0170	2,411
	LLM-SRec + Reg	**0.0268**	**0.0390**	**0.0146**	**0.0176**	3,199
Baby	GRU4Rec	0.0190	0.0316	0.0100	0.0131	6,670
	LLM-SRec	0.0271	0.0466	0.0131	0.0179	796
	LLM-SRec + Reg	**0.0280**	**0.0450**	**0.0146**	**0.0189**	1,311
Musical	GRU4Rec	0.0283	0.0449	0.0153	0.0194	8,812
	LLM-SRec	0.0352	0.0527	0.0185	0.0229	2,648
	LLM-SRec + Reg	**0.0368**	**0.0585**	**0.0195**	**0.0249**	2,662
CDs	GRU4Rec	**0.0448**	**0.0625**	**0.0267**	**0.0311**	17,334
	LLM-SRec	0.0190	0.0325	0.0097	0.0131	1,202
	LLM-SRec + Reg	0.0340	0.0546	0.0174	0.0226	8,702
Books	GRU4Rec	**0.0602**	**0.0816**	**0.0373**	**0.0427**	19,929
	LLM-SRec	0.0359	0.0597	0.0170	0.0230	10,772
	LLM-SRec + Reg	0.0422	0.0605	0.0241	0.0286	14,061
Toys	GRU4Rec	**0.0643**	**0.0753**	**0.0495**	**0.0522**	18,589
	LLM-SRec	0.0355	0.0508	0.0204	0.0243	12,320
	LLM-SRec + Reg	0.0481	0.0624	0.0314	0.0350	15,741

making the recommendation behavior of CF-Integrated LLMRecs more aligned with that of the base CF model.

6 Related Work

LLM-based Recommender Systems. Early research on LLM-based recommender systems primarily explored zero-shot and in-context learning approaches [12, 16, 38]. Building on this, TALLRec [5] utilized Parameter-Efficient Fine-Tuning (PEFT) [21] to bridge the gap between general language modeling and recommendation-specific objectives. Despite these advances, text-centric models often underperform in warm scenarios compared to traditional ID-based models because they fail to explicitly capture collaborative knowledge [26]. To address this, recent research has shifted toward CF-Integrated LLMRec frameworks [26, 27, 30, 42]. These approaches share a common architecture of aligning pre-trained CF embeddings with the token space of LLMs to incorporate pre-trained CF knowledge. More specifically, LLaRA [30] and CoLLM [42] integrate item embeddings from SASRec with textual features via an alignment adapter to inject CF knowledge to LLMs. A-LLMRec [26] employs latent space matching with reconstruction objectives to align CF and textual embeddings, and LLM-SRec [27] proposes a knowledge distillation framework to transfer the ability to understand the sequential patterns of interaction histories.

Dimensional Collapse in Recommender Systems. Recommendation requires distinguishing a large number of candidate items in the representation space; when learned representations collapse into a low-dimensional subspace, item embeddings become less discriminative, which can limit fine-grained ranking and reduce recommendation diversity [7, 10, 14, 43]. Zhang et al. [43] relate popularity bias in graph collaborative filtering to dimensional collapse and propose a decorrelation-based objective for mitigation. Cui et al. [10] show that, in LLM-enhanced sequential recommendation, projecting high-dimensional language representations into the item space can induce severe spectral collapse, and propose a spectral-aware adapter to address this issue. Motivated by this line of research, we examine dimensional collapse in CF-Integrated LLMRecs as a diagnostic lens for understanding their observed reduction in recommendation diversity.

Table 12: Hit@10 Users Overlap with SASRec after applying regularization. Higher values indicate that the regularized model correctly recommends the ground-truth item for more users who are also correctly served by SASRec.

Model	Electronics		Baby Products		CDs and Vinyl		Books	
	Base	+Reg	Base	+Reg	Base	+Reg	Base	+Reg
A-LLMRec	57	50	11	20	35	75	80	109
LLaRA	17	38	7	18	10	30	48	57
LLM-SRec	46	46	15	33	50	74	49	209

Popularity Bias and Diversity in Recommendation. In recommender systems, recommendation quality is not fully captured by accuracy alone, since models may repeatedly expose a narrow set of popular items while overlooking a large portion of the catalog [3, 17, 39]. Prior work has studied this issue through both popularity bias and recommendation diversity, as recommendation concentration on head items often accompanies limited exposure to long-tail items and reduced catalog-level diversity [6, 43]. Accordingly, alleviating popularity bias and broadening recommendation diversity have been considered important goals for improving recommendation quality beyond accuracy [44]. Existing studies have explored reranking [3], regularization [1], and representation-level debiasing [6] methods toward this end. Complementary to this literature, we show that CF-Integrated LLM recommenders suffer from severe diversity loss and tend to amplify concentration on popular items under popularity-skewed datasets.

Revisiting Collaborative Information in Recommender Systems. Recent studies revisited what collaborative information is and how it can be utilized in recommendation models [34, 41]. Zhang et al. [41] systematically investigate collaborative information itself, clarifying it through item co-occurrence patterns and analyzing its manifestation and impact on recommendation performance. Complementarily, Pouryousef and Montazer [34] examine whether LLMs can reason over collaborative signals, showing that naive prompting is insufficient, while retrieving similar-user interactions and presenting them in a structured format substantially improves recommendation quality. Complementing these efforts, we focus on the CF-Integrated LLMRec paradigm and show that injecting pre-trained CF embeddings into LLMs does not guarantee faithful preservation of collaborative knowledge, a failure we attribute to dimensional collapse.

7 Conclusion

In this work, we revisited the common assumption that CF-Integrated LLMRecs preserve and exploit the collaborative knowledge encoded in pre-trained CF embeddings. Through a systematic analysis across six datasets and three representative models, we showed that CF-Integrated LLMRecs recommend to a largely disjoint set of users from the base CF model, and that their performance gains are inconsistent across datasets. Our analysis revealed that these gains are not driven by improved recommendations of unpopular items, but rather by amplified popularity concentration that happens to align with popularity-skewed datasets. We further traced this behavior to dimensional collapse in the item embedding space, where the effective rank drops sharply despite a higher nominal dimensionality. To mitigate this, we proposed a simple orthogonal regularization

that alleviates collapse, restores recommendation diversity, and generally improves recommendation accuracy.

Limitations. Our study has several limitations that open avenues for future work. First, we do not establish a complete causal link between dimensional collapse and recommendation diversity; effective rank is a geometric diagnostic rather than a direct surrogate, and improvements in one do not always translate into improvements in the other. Second, the proposed regularization does not always fully close the gap to the base CF model, suggesting that factors beyond collapse contribute to the observed degradation.

References

[1] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2017. Controlling popularity bias in learning-to-rank recommendation. In *Proceedings of the eleventh ACM conference on recommender systems*. 42–46.

[2] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, Bamshad Mobasher, and Edward Malthouse. 2021. User-centered evaluation of popularity bias in recommender systems. In *Proceedings of the 29th ACM conference on user modeling, adaptation and personalization*. 119–129.

[3] Gediminas Adomavicius and YoungOk Kwon. 2011. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Transactions on Knowledge and Data Engineering* 24, 5 (2011), 896–911.

[4] Pegah Malekpour Alamdari, Nima Jafari Navimipour, Mehdi Hosseinzadeh, Ali Asghar Safaei, and Aso Darwesh. 2020. A systematic study on the recommender systems in the E-commerce. *Ieee Access* 8 (2020), 115694–115716.

[5] 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.

[6] Miaomiao Cai, Lei Chen, Yifan Wang, Haoyue Bai, Peijie Sun, Le Wu, Min Zhang, and Meng Wang. 2024. Popularity-aware alignment and contrast for mitigating popularity bias. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 187–198.

[7] Huiyuan Chen, Vivian Lai, Hongye Jin, Zhimeng Jiang, Mahashweta Das, and Xia Hu. 2024. Towards mitigating dimensional collapse of representations in collaborative filtering. In *Proceedings of the 17th ACM international conference on web search and data mining*. 106–115.

[8] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.

[9] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.

[10] Yu Cui, Feng Liu, Zhaoxiang Wang, Changwang Zhang, Jun Wang, Can Wang, and Jiawei Chen. 2026. SpecTran: Spectral-Aware Transformer-based Adapter for LLM-Enhanced Sequential Recommendation. *arXiv preprint arXiv:2601.21986* (2026).

[11] Farzad Eskandanian and Bamshad Mobasher. 2020. Using stable matching to optimize the balance between accuracy and diversity in recommendation. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*. 71–79.

[12] Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. 2023. Chat-rec: Towards interactive and explainable llms-augmented recommender system. *arXiv preprint arXiv:2303.14524* (2023).

[13] Jyotirmoy Gope and Sanjay Kumar Jain. 2017. A survey on solving cold start problem in recommender systems. In *2017 International Conference on Computing, Communication and Automation (ICCCA)*. IEEE, 133–138.

[14] Xingzhuo Guo, Junwei Pan, Ximei Wang, Baixu Chen, Jie Jiang, and Mingsheng Long. 2023. On the embedding collapse when scaling up recommendation models. *arXiv preprint arXiv:2310.04400*.

[15] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[16] Zhankui He, Zhouhang Xie, Rahul Jha, Harald Steck, Dawen Liang, Yesu Feng, Bodhisattwa Prasad Majumder, Nathan Kallus, and Julian McAuley. 2023. Large language models as zero-shot conversational recommenders. In *Proceedings of the 32nd ACM international conference on information and knowledge management*. 720–730.

[17] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)* 22, 1 (2004), 5–53.

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

[19] Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. 2024. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952* (2024).

[20] 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.

[21] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *Iclr* 1, 2 (2022), 3.

[22] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. 2021. Understanding dimensional collapse in contrastive self-supervised learning. *arXiv preprint arXiv:2110.09348*.

[23] Marius Kaminskas and Derek Bridge. 2016. Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 7, 1 (2016), 1–42.

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

[25] Kibum Kim, Dongmin Hyun, Sukwon Yun, and Chanyoung Park. 2023. Melt: Mutual enhancement of long-tailed user and item for sequential recommendation. In *Proceedings of the 46th international ACM SIGIR conference on Research and development in information retrieval*. 68–77.

[26] Sein Kim, Hongseok Kang, Seungyoon Choi, Donghyun Kim, Minchul Yang, and Chanyoung Park. 2024. Large language models meet collaborative filtering: An efficient all-round llm-based recommender system. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1395–1406.

[27] Sein Kim, Hongseok Kang, Kibum Kim, Jiwan Kim, Donghyun Kim, Minchul Yang, Kwangjin Oh, Julian McAuley, and Chanyoung Park. 2025. Lost in Sequence: Do Large Language Models Understand Sequential Recommendation?. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*. 1160–1171.

[28] Sein Kim, Namkyeong Lee, Donghyun Kim, Minchul Yang, and Chanyoung Park. 2023. Task relation-aware continual user representation learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1107–1119.

[29] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[30] Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. 2024. Llara: Large language-recommendation assistant. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1785–1795.

[31] Kanglin Liu, Wenming Tang, Fei Zhou, and Guoping Qiu. 2019. Spectral regularization for combating mode collapse in gans. (2019), 6382–6390.

[32] Manal Loukili, Fayçal Messaoudi, and Mohammed El Ghazi. 2023. Machine learning based recommender system for e-commerce. *IAES International Journal of Artificial Intelligence* 12, 4 (2023), 1803–1811.

[33] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*. 188–197.

[34] Shahrooz Pouryousef and Ali Montazeralghaem. 2025. Do LLMs Understand Collaborative Signals? Diagnosis and Repair. *arXiv preprint arXiv:2505.20730*.

[35] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*.

[36] Olivier Roy and Martin Vetterli. 2007. The effective rank: A measure of effective dimensionality. (2007), 606–610.

[37] 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.

[38] Lei Wang and Ee-Peng Lim. 2023. Zero-shot next-item recommendation using large pretrained language models. *arXiv preprint arXiv:2304.03153* (2023).

[39] Jacek Wasilewski and Neil Hurley. 2016. Incorporating Diversity in a Learning to Rank Recommender System.. In *FLAIRS*. 572–578.

[40] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys (CSUR)* 52, 1 (2019), 1–38.

[41] Xiaokun Zhang, Zhaochun Ren, Bowei He, Ziqiang Cui, and Chen Ma. 2026. Have We Really Understood Collaborative Information? An Empirical Investigation. In *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining*. 975–984.

[42] Yang Zhang, Fuli Feng, Jizhi Zhang, Keqin Bao, Qifan Wang, and Xiangnan He. 2025. Collm: Integrating collaborative embeddings into large language models for recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2025).

[43] Yifei Zhang, Hao Zhu, Zixing Song, Piotr Koniusz, Irwin King, et al. 2023. Mitigating the popularity bias of graph collaborative filtering: A dimensional collapse perspective. *Advances in Neural Information Processing Systems* 36 (2023), 67533–67550.

[44] Yuying Zhao, Yu Wang, Yunchao Liu, Xueqi Cheng, Charu C Aggarwal, and Tyler Derr. 2025. Fairness and diversity in recommender systems: a survey. *ACM Transactions on Intelligent Systems and Technology* 16, 1 (2025), 1–28.

[45] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management*. 1893–1902.

A Prompt Template

We provide the detailed prompt template corresponding to `[Prompt]` in Figure 2, together with an example of the hybrid input sequence. Following prior CF-Integrated LLMRecs [26, 27, 30], we represent each interacted item using both its textual title embedding and the adapted CF item embedding. The prompt is used to instruct the LLM to aggregate a user's historical interaction sequence and produce a user representation token, which is later projected into the recommendation space for retrieval-based next-item prediction.

A.1 User Prompt

The user prompt describes the user's historical purchase sequence in chronological order, as shown in Figure 4. For each interacted item, we include its timestamp, item title, and the corresponding adapted CF item embedding. We append a learnable token, denoted as `[UserOut]`, to aggregate information from the user's interaction sequence. The last hidden state corresponding to `[UserOut]` is used as the user representation for recommendation.

LLaRA / LLM-SRec
This user has made a series of purchases in the following order: [History Item List: No.\# Time: YYYY/MM/DD Title: Item Title, Item Embedding]. Based on this sequence of purchases, generate user representation token: [UserOut].
A-LLMRec
This is user representation from recommendation models: [User Representation], and this user has made a series of purchases in the following order: [History Item List: No.\# Time: YYYY/MM/DD Title: Item Title, Item Embedding]. Based on this sequence of purchases and user representation, generate user representation token: [UserOut].

Figure 4: Example user prompt for generating the user representation token.

A.2 Item Prompt

The item prompt describes each candidate item using its textual title and adapted CF item embedding, as shown in Figure 5. We append a learnable token, denoted as `[ItemOut]`, to aggregate information from the item-side input. The last hidden state corresponding to `[ItemOut]` is used as the item representation. During retrieval-based recommendation, we compute recommendation scores between the user representation and item representations to rank candidate items.

Item Prompt
The item title and item embedding are as follows: "Title": Item Title, Item Embedding, then generate item representation token: [ItemOut]

Figure 5: Example item prompt for generating the item representation token.

B Implementation Details

B.1 Computing Environment

All experiments are conducted on a single NVIDIA L40S (48GB) GPU with an Intel Xeon Gold 6530 CPU. Models are implemented in PyTorch 2.7.1 with HuggingFace Transformers 4.52.4 on Python 3.10.

B.2 Base CF Model Hyperparameters

We follow the original setups of SASRec [24] and GRU4Rec [18]. Both models use a hidden dimension of 64 with dropout 0.2, trained for up to 200 epochs with early stopping on validation NDCG@10.

B.3 CF-Integrated LLMRec Training Configuration

For all three CF-Integrated LLMRecs, we use the pre-trained CF item embeddings of dimension d_{CF}=64 and project them into the LLM token space (d_{LLM}=3072) via the alignment adapter, following the released code of each method. The projection layer maps the LLM hidden state back to d_{CF}=128 for retrieval-based next-item prediction. The orthogonal regularization weight is fixed to λ=0.01 throughout the main experiments. Other hyperparameters follow the original implementation of each baseline.

C Ablation on Regularization Placement

The orthogonal regularization can, in principle, be applied to any linear module in the model. To understand which component is most responsible for mitigating dimensional collapse, we ablate the placement of the regularization across two natural candidates: (i) the *alignment adapter* $f_\theta : \mathbb{R}^{64} \to \mathbb{R}^{3072}$, which maps pre-trained CF item embeddings into the LLM token space, and (ii) the *projection layer* $\mathbb{R}^{3072} \to \mathbb{R}^{128}$, which maps the LLM hidden state back to the recommendation space (used in our main experiments). For both variants we fix λ=0.01, the value used in our main experiments.

Table 13: Ablation on regularization placement using LLM-SRec with λ=0.01. "Adapter only" applies the orthogonal penalty to the alignment adapter f_θ; "Projection only" applies it to the projection layer (default in our main experiments).

	HR@10	HR@20	NDCG@10	NDCG@20	Cov.
Baby Products					
None (baseline)	0.0172	0.0330	0.0080	0.0120	118
Adapter	0.0183	0.0349	0.0086	0.0127	320
Projection (Ours)	**0.0290**	**0.0461**	**0.0142**	**0.0184**	**821**
CDs and Vinyl					
None (baseline)	0.0200	0.0329	0.0097	0.0130	7,710
Adapter	0.0227	0.0389	0.0111	0.0151	8,270
Projection (Ours)	**0.0245**	**0.0417**	**0.0119**	**0.0162**	**8,294**

Observations. On both datasets, applying the orthogonal regularization to the projection layer outperforms applying it to the adapter across all metrics. The gap is especially pronounced on Baby Products, where the adapter only variant yields only a marginal improvement over the unregularized baseline (HR@10 0.0172 $\to$ 0.0183, NDCG@10 0.0080 $\to$ 0.0086), whereas the projection only variant achieves a substantial gain (HR@10 0.0172 $\to$ 0.0290, NDCG@10 0.0080 $\to$ 0.0142). On CDs and Vinyl, both placements improve over the baseline, but projection only consistently dominates.

Why projection but not the adapter? The orthogonal penalty $\|WW^\top - \mathbf{I}\|_F$ is well posed only when the linear layer satisfies $d_{\text{out}} \le d_{\text{in}}$, since the rows of W lie in $\mathbb{R}^{d_{\text{in}}}$ and at most d_{in} orthonormal vectors can exist in this space. The projection layer satisfies this condition by construction ($\mathbb{R}^{3072} \to \mathbb{R}^{2048} \to \mathbb{R}^{128}$), So the penalty is an achievable target. The alignment adapter, in contrast, maps $\mathbb{R}^{64}$ to $\mathbb{R}^{3072}$, so its first linear layer has d_{out}=3072 $\gg$ d_{in}=64. In this regime $\|WW^\top - \mathbf{I}\|_F$ has a strictly positive lower bound: at most 64 rows can be made orthonormal, and the remaining 3,008 directions are never spanned by the 64-dimensional input distribution. The

optimizer is thus driven toward an unreachable target, producing gradients that disperse weights along directions orthogonal to the actual input signal directions that carry no useful collaborative information. This is consistent with the empirical degradation we observe in Table 13 and reinforces the motivation in Section 4.2 that the projection layer—being the dimensionality reduction bottleneck—is the appropriate place to apply the orthogonal constraint.