SteerRec: Popularity-Steerable Disentanglement for Long-tail Recommendation

Jiayue Wu[1,2], Chenxu Niu[1,2], Mingzhe Lu[1,2*], Qihao Wang[1,2], and Yue Hu[1,2]
[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{wujiayue, niuchenxu, lumingzhe, wangqihao, huyue}@iie.ac.cn
[*]Corresponding author

***Abstract*—Popularity bias remains a pervasive challenge in recommendation systems, where models tend to over-recommend a few head items while neglecting a vast number of niche items. Existing debiasing methods are typically computationally intensive, dependent on specific model architectures, and fail to fully disentangle popularity bias from intrinsic collaborative signals. To address this issue, we propose *SteerRec*, a novel two-stage framework designed for balanced and precise popularity-steered recommendation. We employ a plug-and-play *Disentanglement Adapter* to separate popularity bias from core item representations and an *Adaptive Gated Mechanism* to dynamically control debiasing intensity for each item, thereby generating refined embeddings that reconcile recommendation accuracy with long-tail fairness in the final results. Extensive experiments on three real-world datasets demonstrate that *SteerRec* significantly improves the exposure and recommendation performance of tail items while maintaining the overall recommendation quality.**

***Index Terms*—Recommendation System, Popularity Bias, Representation Learning**

I. Introduction

In recommender systems, Collaborative Filtering (CF) [1]–[4], especially Graph Neural Network (GNN)-based methods, has achieved remarkable success by capturing high-order user-item connectivity [5]–[8]. However, real-world data inevitably follows a long-tail distribution: a few popular items dominate interactions while most are rarely observed [9]. Models trained on such skewed data tend to entangle item popularity with intrinsic collaborative signals within embeddings. Consequently, the learned embeddings are dominated by popularity shortcuts, which suppresses the exposure of long-tail items and compromises the ability to model user preferences.

Current research on mitigating popularity bias in recommender systems primarily follows three paradigms. Weight adjustment and regularization-based methods, such as inverse propensity scoring (IPS) [10], [11] and fairness-aware calibration [12], reweight samples to balance the influence of popular (head) and long-tail items. By contrast, causal inference approaches [13]–[15], exemplified by MACR [15], employ causal graphs and counterfactual reasoning to disentangle item popularity from users' intrinsic preferences. A third line of work adopts representation-based techniques [16], [17], such as DAP [17], which mitigate popularity-related bias in graph convolutional networks by clustering nodes in the embedding space and subtracting similarity-weighted degree-based influences within each cluster after each graph convolution layer.

Although these methods have shown certain effectiveness, they still suffer from several limitations. First, most of them require retraining the backbone recommender models, which is computationally expensive and limits their practical applicability in large-scale systems. Second, some debiasing approaches are designed for specific model architectures, such as GNNs, and therefore cannot be easily generalized to other recommendation frameworks. Moreover, existing methods mainly perform superficial item-level adjustments, leaving popularity signals deeply entangled with the intrinsic collaborative signals in the learned embeddings.

To address these challenges, we propose *SteerRec*, a popularity-steerable recommendation framework as illustrated in Fig. 1. The framework consists of two core stages: **Popularity-Steered Representation Learning** and **Popularity-Steered Adaptation**. In the first stage, to isolate bias-related components, we propose a plug-and-play *Disentanglement Adapter* designed to extract latent popularity representations, thereby effectively decoupling popularity-driven noise from essential collaborative signals. In the second stage, we leverage the extracted popularity representations to perform debiasing. Recognizing that a uniform debiasing strategy is suboptimal due to the disproportionate impact of bias across different items, we design an *Adaptive Gated Mechanism*. This mechanism dynamically adjusts the debiasing intensity for each individual item, enabling precise and fine-grained control over the mitigation of popularity effects. Finally, *SteerRec* achieves a superior balance between recommendation accuracy and fairness, significantly enhancing the visibility of long-tail items without compromising overall predictive performance.

Our main contributions are summarized as follows:

- We propose *SteerRec*, a plug-and-play framework that disentangles popularity bias from item embeddings without retraining the backbone model, producing debiased representations that preserve intrinsic collaborative signals while suppressing popularity-driven artifacts.
- We introduce an adaptive gated steering mechanism that selectively adjusts popularity-related components of item embeddings. This allows for the precise and controlled regulation of each item's popularity effect.
- We propose a joint optimization strategy that integrates supervised bias alignment, preference smoothness, and predictive accuracy, ensuring that the final ranking reflects

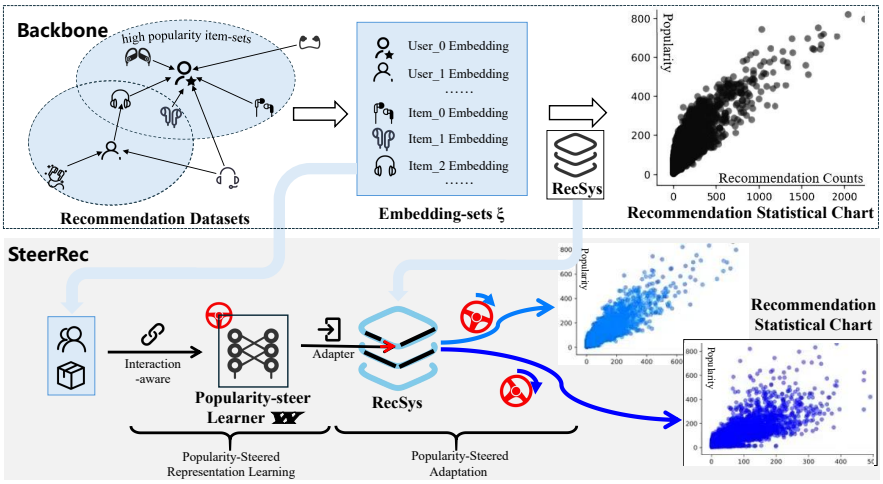


Fig. 1: The overview of the proposed *SteerRec* framework. Our framework first extracts popularity-driven components with a *Disentanglement Adapter*, then applies an *Adaptive Gated Mechanism* for controllable, item-specific debiasing.

intrinsic user preferences rather than exposure-driven dominance.

- Extensive experiments on three real-world datasets show that our method consistently enhances long-tail recommendation performance while maintaining competitive overall accuracy.

II. Preliminaries

Let $\mathcal{U} = \{u_1, u_2, \ldots, u_M\}$ and $\mathcal{I} = \{i_1, i_2, \ldots, i_N\}$ denote the sets of M users and N items, respectively. The historical interactions are represented by an implicit feedback matrix $\mathbf{R} \in \{0,1\}^{M \times N}$, where $y_{ui} = 1$ means an observed interaction (e.g., a click or purchase) between user u and item i, while $y_{ui} = 0$ indicates the absence of an observed interaction.

Modern CF backbones, such as LightGCN [8] and PureMF [18], learn d-dimensional embedding vectors $\mathbf{e}_u \in \mathbb{R}^d$ and $\mathbf{e}_i \in \mathbb{R}^d$ for each user $u \in \mathcal{U}$ and item $i \in \mathcal{I}$, respectively. The predicted preference score $\hat{y}_{ui}$ is then computed via the inner product:

$$\hat{y}_{ui} = \mathbf{e}_u^\top \mathbf{e}_i. \tag{1}$$

To optimize these embeddings, the Bayesian Personalized Ranking (BPR) loss [19] is widely employed. The loss function is formulated as:

$$L_{BPR} = -\sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{N}_u} \sum_{j \notin \mathcal{N}_u} \ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) \tag{2}$$

where $\mathcal{N}_u = \{i \in \mathcal{I} \mid y_{ui} = 1\}$ represents the set of items that user u has interacted with, $\sigma(\cdot)$ is the sigmoid function.

III. Methodology

As mentioned before, the learned item embedding $\mathbf{e}_i$ on skewed data often mix intrinsic collaborative signals with popularity bias, causing popular items to dominate and long-tail items to be underrepresented. To address this issue, we propose *SteerRec*, an adapter equipped with an adaptive gate that isolates and controls popularity bias, producing debiased representations that reconcile recommendation accuracy with exposure fairness.

A. Popularity-Steered Representation Learning

To effectively decouple popularity-driven components from item embeddings, we introduce a *Disentanglement Adapter*. This module isolates bias-related signals from collaborative features through a low-rank projection architecture.

Given an input item embedding $\mathbf{e}_i \in \mathbb{R}^d$, we project it through a low-rank transformation to extract the steering vector $\mathbf{v}_{\text{bias}}$. This vector captures the specific feature components most sensitive to interaction frequency:

$$\mathbf{v}_{\text{bias}} = \text{ReLU}(\mathbf{e}_i \mathbf{W}_{\text{down}}) \mathbf{W}_{\text{up}} \tag{3}$$

where $\mathbf{W}_{\text{down}} \in \mathbb{R}^{d \times r}$ and $\mathbf{W}_{\text{up}} \in \mathbb{R}^{r \times d}$ are trainable projection matrices. The rank constraint $r \ll d$ restricts the adapter's capacity, forcing it to focus on the most pervasive, high-density patterns. By concentrating on these dominant frequency-related signals, the adapter ensures that $\mathbf{v}_{\text{bias}}$ captures the popularity-driven components of the embedding, enabling the subsequent steering process to isolate genuine collaborative signals from popularity interference.

B. Popularity-Steered Adaptation

Upon isolating the bias component $\mathbf{v}_{\text{bias}}$, a naive debiasing strategy would be to directly subtract this popularity-related information from the original item embeddings. However, such a rigid approach is often suboptimal, as it may lead to over-correction by indiscriminately removing signals that reflect genuine item quality. To overcome this, we introduce an *Adaptive Gated Mechanism* to achieve item-specific regulation of popularity effects.

Specifically, we first define a popularity prior z_i to provide a global statistical reference:

$$z_i = \frac{\ln(P_i + 1) - \mathbb{E}[\ln(\mathbf{P} + 1)]}{\sqrt{\text{Var}[\ln(\mathbf{P} + 1)]}} \tag{4}$$

where P_i denotes the raw interaction count of item i. The terms $\mathbb{E}[\ln(\mathbf{P}+1)]$ and $\text{Var}[\ln(\mathbf{P}+1)]$ represent the expectation and variance of the log-transformed popularity across the entire item catalog, respectively.

To further control the debiasing intensity, we concatenate the original embedding $\mathbf{e}_i$ with the bias vector $\mathbf{v}_{\text{bias}}$ and feed them into a Multi-Layer Perceptron (MLP) to derive a gating coefficient $g_i \in (0, 1)$:

$$g_i = \sigma\left(\text{MLP}\left([\mathbf{e}_i \parallel \mathbf{v}_{\text{bias}}]\right)\right), \tag{5}$$

where $\parallel$ denotes the concatenation operation and $\sigma(\cdot)$ is the sigmoid activation function.

Finally, the debiased item representation $\mathbf{e}'_i$ is obtained by adaptively scaling and subtracting the normalized bias component:

$$\mathbf{e}'_i = \mathbf{e}_i - g_i \cdot z_i \frac{\mathbf{v}_{\text{bias}}}{\|\mathbf{v}_{\text{bias}}\|_2}. \tag{6}$$

The framework synthesizes the statistical prior z_i with the adaptive gate g_i, enabling autonomous control over the intensity of popularity steering.

C. *Preference Prediction*

Finally, we compute the preference score y'_{ui} between user u and item i using the debiased representations. Specifically, the prediction is formulated as the inner product of the user embedding $\mathbf{e}_u$ and the debiased item embedding $\mathbf{e}'_i$:

$$y'_{ui} = \mathbf{e}_u^\top \mathbf{e}'_i \tag{7}$$

It is worth noting that our disentanglement and steering processes are applied exclusively to the item embeddings. The rationale behind this design is that the predicted score in recommendation is inherently a relative measure of preference across items. By adjusting item representations while keeping user embeddings fixed, we can effectively reshape the relative ranking among items for each user without altering the underlying user preference space. This design not only simplifies the adaptation process but also avoids introducing additional bias or instability into user representations.

D. *Multi-task Learning Objectives*

The *SteerRec* framework is optimized through a joint objective function that balances recommendation accuracy with rigorous disentanglement of popularity signals.

1) Popularity Regression Loss ($\mathcal{L}_{pop}$): To guarantee that the steering vector $\mathbf{v}_{bias}$ effectively captures the underlying popularity information, we introduce a supervised popularity regression task. We posit that if $\mathbf{v}_{bias}$ is truly representative of popularity-driven factors, it should possess sufficient predictive power to reconstruct the item's relative popularity intensity.

Specifically, we employ a popularity predictor ψ, implemented as a lightweight two-layer Multi-Layer Perceptron (MLP), to map the isolated bias vector back to its standardized popularity indicator z_i. The identification loss is formulated as a Mean Squared Error (MSE):

$$\mathcal{L}_{pop} = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \|\psi(\mathbf{v}_{bias,i}) - z_i\|^2 \tag{8}$$

where $\mathcal{I}$ denotes the set of items in the training batch, and z_i is the normalized popularity defined in Eq. 4.

TABLE I: Statistics of Datasets

Features	Gowalla	Yelp2018	Amazon-book
#Users	29,858	31,668	52,643
#Items	40,981	38,048	91,599
#Train	810,128	1,237,259	2,380,730
#Test	217,242	324,147	603,378
Density	0.00084	0.00130	0.00062
Gini Coeff	0.4410	0.5315	0.4884
Popular Items	53.02%	59.46%	55.49%
Cold Items (<= 5 clicks)	1.48%	2.96%	4.80%

2) Ranking Smoothness Constraint ($\mathcal{L}_{smth}$): The Ranking Smoothness Constraint is designed to prevent popularity from dominating the predicted scores. We argue that if a user has interacted with both a popular item i and a niche item j, their predicted scores should be relatively close. A substantial score gap indicates that the model is over-relying on item frequency rather than intrinsic preference.

Specifically, for any user u and a pair of interacted items (i, j) where i is more popular than j, we define the smoothness loss as:

$$\mathcal{L}_{smth} = -\sum_{u=1}^{M} \sum_{i \in \mathcal{N}_u} \sum_{j \in \mathcal{N}_u, z_i > z_j} \ln \sigma\left(\gamma - (y'_{ui} - y'_{uj})\right) \tag{9}$$

where $\mathcal{N}_u$ denotes the set of items that user u has interacted with. In this formulation, $\gamma > 0$ acts as a margin that defines the maximum allowable score difference. By minimizing $\mathcal{L}_{smth}$, we encourage the model to reduce the disparity between popular and niche items within the user's positive interaction set, ensuring that the final ranking is driven by intrinsic collaborative signals rather than exposure-driven popularity dominance.

3) Debiased Recommendation Loss ($\mathcal{L}_{rec}$): To ensure predictive accuracy, we optimize the model using the Bayesian Personalized Ranking (BPR) criterion based on the debiased item representations. The goal is to ensure that for each user, the preference score for an interacted item i is higher than that of a non-interacted item j. The loss function is formulated as:

$$\mathcal{L}_{rec} = -\sum_{u=1}^{M} \sum_{i \in \mathcal{N}_u} \sum_{j \notin \mathcal{N}_u} \ln \sigma(y'_{ui} - y'_{uj}) \tag{10}$$

By optimizing $\mathcal{L}_{rec}$ using these debiased embeddings, the model learns to prioritize items based on intrinsic user-item relevance, effectively mitigating the interference of popularity-driven signals during the ranking process.

TABLE II: Performance Comparison on Three Datasets (Overall vs. Tail)

Datasets	Gowalla				Yelp2018				Amazon-book			
	Overall Test		*Tail Test*		*Overall Test*		*Tail Test*		*Overall Test*		*Tail Test*	
Methods	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
LightGCN	0.1819	0.1544	0.0422	0.0185	0.0632	0.0520	0.0087	0.0044	0.0417	0.0321	0.0087	0.0050
IPSCN	0.1325	0.1132	0.0477	0.0213	0.0473	0.0391	0.0136	0.0077	0.0285	0.0221	0.0118	0.0069
CausE	0.1334	0.1137	0.0485	0.0225	0.0492	0.0405	0.0141	0.0085	0.0299	0.0230	0.0127	0.0078
DICE	0.1337	0.1138	0.0493	0.0241	0.0505	0.0409	0.0132	0.0073	0.0348	0.0264	0.0121	0.0074
MACR	0.1635	0.1396	0.0588	0.0295	0.0522	0.0432	0.0131	0.0078	0.0382	0.0298	0.0132	0.0079
Tail	0.1647	0.1391	0.0628	0.0319	0.0570	0.0466	0.0154	0.0095	0.0369	0.0283	0.0151	0.0094
DAP	0.1672	0.1427	0.0708	0.0354	0.0562	0.0461	0.0218	0.0129	0.0414	0.0328	0.0166	0.0102
SteerRec	0.1650	0.1355	**0.0769**	**0.0395**	0.0510	0.0410	**0.0223**	**0.0137**	0.0414	0.0321	**0.0263**	**0.0174**

4) Joint Optimization: The final objective function integrates the aforementioned components with L_2 regularization on the adapter weights Φ:

$$L_{total} = L_{rec} + L_{pop} + L_{smth} + \alpha\|\Phi\|_2^2 \tag{11}$$

IV. Experiments

We conduct experiments to evaluate the performance of our proposed framework. Specifically, we aim to answer the following research questions:

- **RQ1:** How does *SteerRec* perform compared to state-of-the-art models in terms of both overall recommendation accuracy and performance on tail items?
- **RQ2:** Can *SteerRec* effectively alleviate popularity bias in terms of exposure distribution, and does the learned bias vector truly capture item popularity?
- **RQ3:** How do the each components in *SteerRec* contribute to the balance between accuracy and fairness?

A. Experimental Setup

1) Model and Datasets: We implemented our framework based on LightGCN [8], due to its efficiency and proven effectiveness in collaborative filtering. To evaluate the effectiveness of the proposed method, we conduct experiments on three widely used benchmark datasets: Gowalla, Yelp2018, and Amazon-Book. Gowalla contains user check-in data from a location-based social network, representing implicit user–venue interactions. Yelp2018 and Amazon-Book represent user interactions in business review and e-commerce platforms, respectively. These datasets vary in domain characteristics and sparsity levels, providing a comprehensive benchmark for evaluating recommendation performance. Detailed statistics are reported in Table I.

For a fair comparison, we follow the same methodology for constructing the test set in [8]. Specifically, each item is ranked by its interaction frequency in the training set, and the bottom 80% of them are defined as tail items. Based on this criterion, the *Tail Test Set* is composed of all test interactions involving tail items, whereas the *Overall Test Set* retains all interactions in the original test set.

2) Compared Methods: We compare our approach with the following methods:

- **IPSCN** [20]: A weighting-based method that employs Inverse Propensity Scoring with variance reduction to reweight items.
- **CausE** [21]: A causal framework that learns debiased embeddings by modeling exposure and click data separately via propensity-based regularizers.
- **DICE** [14]: A causal disentanglement method that partitions interactions into interest-driven and conformity-driven components.
- **MACR** [15]: A counterfactual inference framework designed to eliminate the direct causal effect of item popularity on ranking scores.
- **Tail-GNN** [16]: A representation-based model that enhances tail node embeddings by simulating missing neighborhood information.
- **DAP** [17]: A graph-based debiasing framework that intervenes in bias amplification within GNN layers through node clustering.

3) Evaluation Metrics: We report the all-ranking performance using two widely adopted metrics: Recall@20, which measures the coverage of relevant items among the top-20 recommendations, and NDCG@20, which further evaluates ranking quality by assigning higher weights to relevant items placed closer to the top. Together, these metrics offer a comprehensive assessment of recommendation accuracy and ranking quality.

4) Hyper-parameter Settings: We implement the proposed *SteerRec* framework using PyTorch and optimize the network parameters via the Adam optimizer. To ensure consistency and reproducibility, the embedding dimension d is fixed at 64, aligning with the pre-trained backbones. We employ a batch size of 4,096 and a fixed learning rate of 1×10^{-4}. The low-rank dimension r is set to 16 to balance model expressiveness with parameter efficiency. The margin hyperparameter γ in Eq. 9 is set to 0.2. The L_2 regularization coefficient α in Eq. 11 is specified as 1×10^{-4} to prevent overfitting. All models are trained for 100 epochs.

B. Performance Comparison (RQ1)

To validate the performance of the proposed *SteerRec* in addressing the long-tail recommendation problem, the comprehensive performance comparison against all baselines across three datasets is presented in Table II. Based on the experimental results, we have the following key observations:

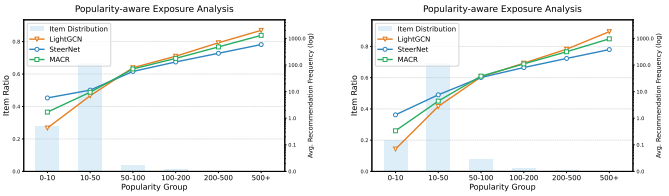
- **The substantial performance gap between overall and tail groups confirms the severity of popularity bias in existing recommendation methods.** As shown in

Table II, although LightGCN achieves superior results on the *Overall Test Set*, its performance drops significantly when evaluated on the tail item group specifically. Specifically, the Recall@20 and NDCG@20 on the *Tail Test Set* decrease by an average of 80.72% and 87.99%, respectively, compared to the *Overall Test Set*. This sharp contrast suggested that existing models tend to prioritize memorizing high-frequency patterns over capturing users' actual preferences for specialized items. Such results highlight the necessity of effective debiasing interventions. In contrast, our proposed method significantly narrowed the performance degradation to 48.71% of the Recall@20 and 60.93% of the NDCG@20, respectively.

- **SteerRec achieves a superior balance between accuracy and fairness, significantly boosting tail performance while preserving overall recommendation utility.** As observed in Table II, most conventional debiasing baselines (e.g., IPSCN, CausE, and DICE) suffer from a significant performance drop in the *Overall Test Set*. For instance, while LightGCN achieves an Overall Recall of 0.1819 on Gowalla, most baselines struggle to exceed 0.13. In contrast, *SteerRec* demonstrates superior robustness by achieving substantial gains in the *Tail Test* with minimal degradation of overall accuracy. While representation-based baselines like Tail and DAP show competitive results on specific datasets, *SteerRec* demonstrates superior stability and a more robust accuracy-fairness balance across all three datasets by consistently achieving the highest tail gains with minimal overall performance loss.
- **SteerRec adaptively reduces popularity bias, leading to substantial improvements on sparse, bias-separable datasets like Amazon-Book.** On the *Tail Test Set*, *SteerRec* achieves its largest improvement on Amazon-Book, with Tail Recall reaching 0.0263—a 202.3% gain over the LightGCN baseline (0.0087). This success can be attributed to two main factors. First, Amazon-Book has a large item catalog (91,599 items) with many extremely sparse items: 4.80% of items have five or fewer interactions, compared to 2.96% in Yelp2018 and 1.48% in Gowalla. Traditional models often treat these extremely sparse items as noise due to the lack of sufficient collaborative signals. Conversely, *SteerRec* utilizes the popularity prior z_i to identify these items and applies adaptive steering to compensate for their weak representations. Second, the bias in Amazon-Book is more separable: interactions reflect a clearer mapping between item attributes and user preferences than in datasets like Yelp2018, where external factors introduce complex confounding.

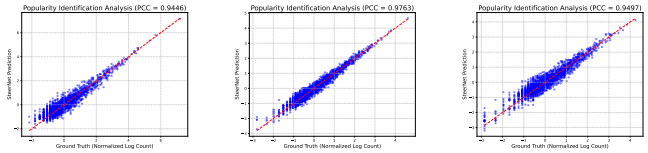
C. Popularity-aware Exposure Analysis (RQ2)

To investigate the impact of *SteerRec* on exposure fairness, we analyze the distribution of recommendation frequency across items with varying popularity levels on the three datasets. Specifically, we grouped items into six popularity intervals based on their interaction frequency in the training set: [0, 10), [10, 50), [50, 100), [100, 200), [200, 500), and [500+). For each group, we compute the average recommendation frequency and plot the results on a logarithmic scale in Fig. 2.



(a) Gowalla (b) Amazon-Book

Fig. 2: Popularity-aware exposure analysis on two datasets. Bars indicate the item ratio of each popularity group, while curves represent the average recommendation frequency.



(a) Gowalla (b) Yelp2018 (c) Amazon-Book

Fig. 3: Popularity identification analysis on three datasets. Each point represents an item, plotting the predicted popularity score against the ground-truth popularity label. The strong positive correlation indicates that the extracted bias vector captures meaningful popularity information.

As illustrated in Fig.2, the backbone LightGCN model suffers from pronounced popularity bias. It disproportionately favors head items, resulting in a recommendation frequency that far exceeds their presence in the catalog, while the vast majority of tail items remain under-exposed. While MACR provides a moderate correction to this imbalance, *SteerRec* achieves a more significant and systematic redistribution of exposure. Specifically, *SteerRec* effectively suppresses the over-representation of "super-head" items and substantially enhances the visibility of items in low-popularity brackets—most notably within the 0–10 and 10–50 interaction intervals. This demonstrates *SteerRec*'s superior ability to promote long-tail content while maintaining a balanced recommendation ecosystem.

These observations demonstrate that *SteerRec* effectively reshapes the exposure distribution to be more equitable. By re-steering the biased embeddings, the framework ensures a more balanced allocation of recommendation opportunities without over-relying on global item popularity.

D. Popularity Identification Analysis (RQ2)

To verify whether the extracted bias vector $\mathbf{v}_{\text{bias}}$ effectively captures item popularity information, we evaluate the correlation between the predicted popularity scores and the ground-truth popularity labels. Specifically, for each dataset,

TABLE III: Ablation study results on Gowalla

Methods	Overall Test		Tail Test	
	Recall@20	NDCG@20	Recall@20	NDCG@20
SteerRec	0.1650	0.1355	0.0769	0.0395
w/o L_{rec}	0.1629	0.1334	0.0775	0.0402
w/o L_{smth}	0.1819	0.1543	0.0427	0.0188
w/o L_{pop}	0.1658	0.1358	0.0750	0.0382

we randomly sample 2,000 items and compute the predicted scores based on the learned bias vectors. The results are visualized in Fig. 3, showing that the predicted scores exhibit a remarkably strong positive correlation with the ground-truth labels (PCC > 0.94 across all datasets). This indicates that the bias vectors learned by *SteerRec* successfully encode salient popularity features.

We observe that, while the predicted scores of the popularity predictor ψ closely match the ground-truth for head items, the variance of predictions is substantially higher for niche items. This indicates that latent representations of extremely low-frequency items are inherently uncertain due to data sparsity. To mitigate the propagation of such instability, we adopt a hybrid steering strategy that balances adaptivity with robustness. Specifically, for popular items, the *Adaptive Gated Mechanism* is used to flexibly adjust the influence of popularity bias. For low-popularity items, we use the deterministic, normalized popularity prior z_i. This dual-path design prevents error propagation in sparse regions while preserving fine-grained, data-driven control for frequently observed items, enabling stable and accurate debiasing across the entire popularity spectrum.

E. *Ablation Analysis (RQ3)*

To investigate the individual contribution of each optimization objective within *SteerRec*, we conduct an ablation study on the Gowalla dataset by removing key loss components: the Debiased Recommendation Loss ($\mathcal{L}_{rec}$), the Ranking Smoothness Constraint ($\mathcal{L}_{smth}$), and the Popularity Regression Loss ($\mathcal{L}_{pop}$). The performance variations are summarized in Table III, leading to the following insights:

- **Impact of** $\mathcal{L}_{rec}$: Removing $\mathcal{L}_{rec}$ leads to a marginal decline in overall performance (Recall@20: 0.1629 vs. 0.1650) but results in a slight improvement in tail-item metrics. This indicates that while L_{rect} is essential for maintaining global recommendation accuracy, its relaxation allows the model to prioritize tail-item steering, reflecting the trade-off between head-item precision and tail-item exposure.
- **Impact of** $\mathcal{L}_{smth}$: Removing $\mathcal{L}_{smth}$ causes overall Recall@20 to surge to 0.1819, yet tail performance collapses from 0.0769 to 0.0427 (comparable to LightGCN). This confirms that $\mathcal{L}_{smth}$ acts as a crucial regularizer that prevents the recommendation list from being dominated by high-probability head items. By narrowing the score gap between popular and niche items that share similar user interest, this constraint forces the model to prioritize intrinsic preferences over frequency-based shortcuts.
- **Impact of** $\mathcal{L}_{pop}$: Removing $\mathcal{L}_{pop}$ leads to a marginal increase in overall Recall@20 (0.1650 to 0.1658) but a simultaneous drop in tail performance (0.0769 to 0.0750). This performance shift indicates that without explicit popularity supervision, the model tends to favor head items to optimize global metrics. Thus, $\mathcal{L}_{pop}$ is essential for enforcing the structural alignment between the steering vectors and item popularity, ensuring that the disentangled components accurately capture bias patterns for equitable recommendation.

In summary, the joint optimization of $\mathcal{L}_{rec}$, $\mathcal{L}_{smth}$, and $\mathcal{L}_{pop}$ enables *SteerRec* to achieve a robust balance between overall accuracy and tail-item fairness, validating the necessity of each component in our steering mechanism.

TABLE IV: Performance on *Tail Test Set* Across Three Datasets

Datasets	Gowalla		Yelp2018		Amazon-book	
Methods	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
MF	0.0460	0.0221	0.0109	0.0062	0.0088	0.0049
SteerRec	0.0623	0.0328	0.0163	0.0097	0.0213	0.0142

F. *Generalization to Matrix Factorization*

To verify the broad applicability of *SteerRec*, we extend our framework beyond graph-based architectures to a standard Matrix Factorization (MF) backbone. Since *SteerRec* operates directly on the representation, it is fundamentally model-agnostic: any collaborative filtering (CF) paradigm that represents users and items as vectors can be seamlessly enhanced by our steering process. Table IV reports the performance on the *Tail Test Set* across the three datasets.

As shown in Table IV, integrating our *SteerRec* consistently improves both Recall@20 and NDCG@20 compared to the vanilla MF baseline. Specifically, on Gowalla, *SteerRec* achieves a Recall@20 of 0.0623 and NDCG@20 of 0.0328, representing relative gains of approximately 35% and 48% over MF, respectively. Similar improvements are observed on Yelp2018 and Amazon-book, indicating that our approach effectively mitigates popularity bias and enhances the recommendation of long-tail items.

V. Related Work

A. *Collaborative Filtering in Recommender Systems*

Collaborative filtering (CF) is a widely used approach in recommender systems that predicts user preferences based on past interactions. CF methods can be categorized into **memory-based** and **model-based** approaches [1]. Memory-based CF relies on similarities among users or items to generate recommendations, while model-based CF employs machine learning models to capture complex patterns in interaction data. Architectures such as CNNs, RNNs, and AutoEncoders have been widely adopted [2]–[4]. More recently, GNN–based methods have gained significant attention due to their ability to model high-order relations in user–item graphs [5]–[8].

B. *Popularity Bias in Recommendation*

Recommendation models often suffer from popularity bias, where popular items dominate interactions and overshadow long-tail items. Current efforts to tackle this issue can be broadly categorized into three types: Weight adjustment and regularization methods, such as inverse propensity scoring (IPS) and fairness-aware calibration, aim to rebalance the contributions of head and tail items [10], [11], [22]. Causal intervention–based techniques leverage causal graphs and counterfactual reasoning to separate popularity-induced effects from intrinsic user preferences [13], [14]. Representation-level techniques and re-ranking strategies either modify latent embeddings to reduce bias amplification [17] or adjust recommendation scores directly to enhance exposure for long-tail items. Despite these efforts, existing solutions are often limited by high computational overhead, dependency on specific model architectures. More importantly, they frequently fail to fully disentangle popularity bias from intrinsic collaborative signals, leading to sub-optimal trade-offs between accuracy and fairness.

VI. Summary

In summary, we tackle the challenge of popularity bias in recommendation systems by proposing *SteerRec*, a robust two-stage framework. Our approach first employs a *Disentanglement Adapter* to isolate popularity-driven noise from the core item representations, and then leverages an *Adaptive Gated Mechanism* to adjust the debiasing intensity for each item individually. This approach generates debiased embeddings that provide balanced representation for both popular and niche items. Experimental results on three real-world datasets show that *SteerRec* significantly improves performance on the *Tail Test Set* while reserving overall recommendation utility.

VII. Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant No. 2025M771524. The authors would also like to acknowledge the assistance of large language models, including ChatGPT and Gemini, which were used to support language polishing and preliminary drafting during the manuscript preparation process.

References

[1] Y. Li, K. Liu, R. Satapathy, S. Wang, and E. Cambria, "Recent Developments in Recommender Systems: A Survey," Jun. 2023.

[2] C. Choudhary, I. Singh, and M. Kumar, "Sarwas: Deep ensemble learning techniques for sentiment based recommendation system," *Expert Systems with Applications*, vol. 216, p. 119420, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417422024393

[3] H. An and N. Moon, "Design of recommendation system for tourist spot using sentiment analysis based on cnn-lstm," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, pp. 1653 – 1663, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:208093192

[4] T. Liu, Q. Wu, L. Chang, and T. Gu, "A review of deep learning-based recommender system in e-learning environments," *Artificial Intelligence Review*, vol. 55, pp. 5953 – 5980, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:249145178

[5] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering." New York, NY, USA: Association for Computing Machinery, 2019, p. 165–174.

[6] J. Sun, Y. Zhang, C. Ma, M. J. Coates, H. Guo, R. Tang, and X. He, "Multi-graph convolution collaborative filtering," *2019 IEEE International Conference on Data Mining (ICDM)*, pp. 1306–1311, 2019.

[7] X. Wang, H. Jin, A. Zhang, X. He, T. Xu, and T.-S. Chua, "Disentangled graph collaborative filtering," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA, 2020, p. 1001–1010.

[8] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation." New York, NY, USA: Association for Computing Machinery, 2020, p. 639–648.

[9] H. Abdollahpouri, R. Burke, and B. Mobasher, "Controlling Popularity Bias in Learning-to-Rank Recommendation," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*. Como Italy: ACM, Aug. 2017, pp. 42–46.

[10] D. Jannach, L. Lerche, I. Kamehkhosh, and M. Jugovac, "What recommenders recommend: an analysis of recommendation biases and possible countermeasures," *User Modeling and User-Adapted Interaction*, vol. 25, pp. 427–491, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:15621670

[11] T. Schnabel, A. Swaminathan, A. Singh, N. Chandak, and T. Joachims, "Recommendations as treatments: debiasing learning and evaluation," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, 2016, p. 1670–1679.

[12] H. Abdollahpouri, R. Burke, and B. Mobasher, "Controlling Popularity Bias in Learning-to-Rank Recommendation," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*. Como Italy: ACM, Aug. 2017, pp. 42–46.

[13] Y. Zhang, F. Feng, X. He, T. Wei, C. Song, G. Ling, and Y. Zhang, "Causal intervention for leveraging popularity bias in recommendation," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA, 2021, p. 11–20.

[14] Z. Zhao, J. Chen, S. Zhou, X. He, X. Cao, F. Zhang, and W. Wu, "Popularity bias is not always evil: Disentangling benign and harmful bias for recommendation," *IEEE Trans. on Knowl. and Data Eng.*, vol. 35, no. 10, p. 9920–9931, Oct. 2023.

[15] T. Wei, F. Feng, J. Chen, Z. Wu, J. Yi, and X. He, "Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender system," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery amp; Data Mining*, ser. KDD '21. ACM, Aug. 2021, p. 1791–1800. [Online]. Available: http://dx.doi.org/10.1145/3447548.3467289

[16] Z. Liu, T.-K. Nguyen, and Y. Fang, "Tail-GNN: Tail-Node Graph Neural Networks," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. Virtual Event Singapore: ACM, Aug. 2021, pp. 1109–1119.

[17] J. Chen, J. Wu, J. Chen, X. Xin, Y. Li, and X. He, "How Graph Convolutions Amplify Popularity Bias for Recommendation?" May 2023. [Online]. Available: https://doi.org/10.1007/s11704-023-2655-2

[18] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proceedings of the 21st International Conference on Neural Information Processing Systems*, ser. NIPS'07. Red Hook, NY, USA: Curran Associates Inc., 2007, p. 1257–1264.

[19] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," *ArXiv*, vol. abs/1205.2618, 2009. [Online]. Available: https://api.semanticscholar.org/CorpusID:10795036

[20] A. Gruson, P. Chandar, C. Charbuillet, J. McInerney, S. Hansen, D. Tardieu, and B. Carterette, "Offline Evaluation to Make Decisions About Playlist Recommendation Algorithms," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. Melbourne VIC Australia: ACM, Jan. 2019, pp. 420–428.

[21] S. Bonner and F. Vasile, "Causal embeddings for recommendation," in *Proceedings of the 12th ACM Conference on Recommender Systems*. New York, NY, USA: Association for Computing Machinery, 2018, p. 104–112.

[22] H. Abdollahpouri, R. Burke, and B. Mobasher, "Managing popularity bias in recommender systems with personalized re-ranking," 2019. [Online]. Available: https://arxiv.org/abs/1901.07555