Opening the Black Box: Interpretable Remedies for Popularity Bias in Recommender Systems

Parviz Ahmadov
Delft University of Technology
Delft, Netherlands
p.ahmadov@student.tudelft.nl

Masoud Mansoury
Delft University of Technology
Delft, Netherlands
m.mansoury@tudelft.nl

Abstract

Popularity bias is a well-known challenge in recommender systems, where a small number of popular items receive disproportionate attention, while the majority of less popular items are largely overlooked. This imbalance often results in reduced recommendation quality and unfair exposure of items. Although existing mitigation techniques address this bias to some extent, they typically lack transparency in how they operate. In this paper, we propose a post-hoc method using a Sparse Autoencoder (SAE) to interpret and mitigate popularity bias in deep recommendation models. The SAE is trained to replicate a pre-trained model's behavior while enabling neuron-level interpretability. By introducing synthetic users with clear preferences for either popular or unpopular items, we identify neurons encoding popularity signals based on their activation patterns. We then adjust the activations of the most biased neurons to steer recommendations toward fairer exposure. Experiments on two public datasets using a sequential recommendation model show that our method significantly improves fairness with minimal impact on accuracy. Moreover, it offers interpretability and fine-grained control over the fairness–accuracy trade-off.

CCS Concepts

• **Information systems** → **Recommender systems**; **Evaluation of retrieval results**.

Keywords

recommender systems, popularity bias, fairness, interpretation, sparse autoencoder

ACM Reference Format:
Parviz Ahmadov and Masoud Mansoury. 2025. Opening the Black Box: Interpretable Remedies for Popularity Bias in Recommender Systems. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems (RecSys '25), September 22–26, 2025, Prague, Czech Republic.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3705328.3759310

1 Introduction

Popularity bias is a persistent challenge in recommender systems, where highly popular items are favored over less popular, long-tail items that may be more relevant to niche user interests [4, 19, 31]. This bias arises naturally from collaborative filtering and other learning-based approaches that reflect patterns in historical interaction data, including skewness toward popular content [15, 21, 25]. While such recommendations may serve mainstream users well, they limit opportunities for discovery, marginalize niche or newer items, and reduce exposure fairness. This can ultimately disadvantage both content creators and users with specialized tastes [9, 20, 21, 32].

Most existing methods to mitigate popularity bias rely on reweighting item scores or modifying model architectures [5, 22, 33]. However, these techniques often function as black boxes, offering limited insight into why certain items are recommended or which internal components drive biased behavior. This lack of interpretability poses a challenge for diagnosing and correcting systemic biases.

To address this, we turn to Sparse Autoencoders (SAEs)—neural models designed to activate only a small subset of neurons per input. This sparsity enables clearer attribution of individual neurons to specific features or concepts. Recent work in interpretability has shown that SAEs can uncover high-level, monosemantic neurons in large language models [3, 8, 17, 23]. Crucially, sparsity makes these neurons easier to isolate and manipulate without unintended side effects, a property already leveraged to mitigate bias in language models [7, 11, 14].

Inspired by these findings, we propose `PopSteer`, a novel post-hoc method that interprets and mitigates popularity bias at the neuron level in deep recommendation models. We begin by training an SAE to replicate the output of a pretrained recommender, attaching it to the final layer to capture the model's decision-making process. Then, we generate synthetic user profiles that reflect strong preferences for either popular or unpopular items. By analyzing how individual SAE neurons respond to these inputs, we identify those most responsible for encoding popularity signals.

Once identified, `PopSteer` adjusts neuron activations to reduce the influence of popularity-biased neurons and amplify the contribution of neurons aligned with long-tail content. This neuron steering approach enables fine-grained control over recommendation behavior while preserving the model's overall structure and performance. Our experiments on two public datasets using the SASRec model demonstrate that `PopSteer` effectively improves item exposure fairness with minimal loss in recommendation accuracy. Compared to several existing bias mitigation techniques, our method offers both superior performance and interpretability.

2 Interpretation with Sparse Autoencoder

Formally, let $x \in \mathbb{R}^d$ denote an embedding from the ML model. The SAE, consisting of an input dimension of size d, a hidden dimension

RecSys '25, Prague, Czech Republic

ACM ISBN 979-8-4007-1364-4/25/09
https://doi.org/10.1145/3705328.3759310

of size N, and an output dimension of size d, reconstructs x as:

$$\hat{x} = W_{\text{dec}} a + b_{\text{pre}}$$

where $W_{\text{dec}} \in \mathbb{R}^{d \times N}$ and $b_{\text{pre}} \in \mathbb{R}^{d}$ are learnable parameters. The hidden representation $a \in \mathbb{R}^{N}$ is computed as:

$$a = \text{ReLU}(z)$$

$$z = W_{\text{enc}}^{T}(x - b_{\text{pre}})$$

where $W_{\text{enc}} \in \mathbb{R}^{d \times N}$ is the learnable encoder weight matrix. While ReLu introduces basic sparsity, stricter control is applied by retaining only the top-K highest activations in a and setting all other activations to zero, as proposed by Gao et al. [8]. With $K < d$, the top-K mask activates just a limited subset of neurons. Thanks to the large hidden dimension N, the model can pick from many candidates, and the resulting sparsity encourages each fired unit to *specialize* in a meaningful feature of the input. The SAE is trained to minimize the following objective:

$$\min_{W_{\text{enc}}, W_{\text{dec}}, b_{\text{pre}}} \|x - \hat{x}\|_2^2 + \gamma * \mathcal{L}_{aux} \tag{1}$$

where $\mathcal{L}_{aux}$ is a loss term preventing dead neurons[1]. This objective ensures that the sparse hidden representation maintains fidelity to the original embedding while revealing the underlying decision structure. The learned neurons can then be interpreted to explain which concepts in the input data influence the ML model's output.

3 Interpreting Popularity Bias

Since each SAE neuron tends to specialize in a distinct concept from the input data, analyzing their activation behavior reveals how such concepts influence predictions. To explore this, we generate synthetic user profiles that highlight popularity bias and examine how specific neurons respond. This enables the identification of neurons that either reinforce or counteract popularity signals.

Each synthetic profile is passed through the pretrained recommendation model to obtain the user embedding (i.e., x in section 2), which is then fed into the pretrained SAE for neuron-level analysis. In the following, we describe our synthetic data generation process and how we quantify each neuron's contribution to popularity bias.

3.1 Generating synthetic datasets

We create two types of synthetic user profiles: one biased toward popular items and the other toward unpopular items. These profiles simulate user interactions that reflect extreme cases of popularity preference, enabling clearer observation of neuron activation.

Let $\mathcal{U} = \{u_1, ..., u_n\}$ be the set of users, $\mathcal{I} = \{i_1, ..., i_m\}$ be the set of items, and $R \in \mathbb{R}^{n \times m}$ represent the user-item interaction matrix. Following [1, 31], we define $\mathcal{I}^{pop}$ (popular or *head* items) as the most frequently interacted items and $\mathcal{I}^{unpop}$ (unpopular or *tail* items) as the least interacted items, both compromising roughly 20% of total interactions. Using these popular and unpopular item sets, we construct two synthetic datasets: 1) R^{pop} with user profiles containing only popular items, and 2) R^{unpop} with user profiles containing only unpopular items.

To generate R^{pop}, for each user u in R, we follow this process: 1) extract the items interacted by u in R, 2) replace each item i with a randomly selected item from $\mathcal{I}^{pop}$, and 3) add this modified profile to R^{pop}. We apply the same procedure for generating R^{unpop}, using items from $\mathcal{I}^{unpop}$ in step 2. This approach preserves the number of interactions per user, isolating the popularity signal without altering profile length or interaction density. Importantly, these synthetic datasets are not used to train the SAE. Instead, we feed them into a pretrained SAE (trained on R), to evaluate its neurons' activation, ensuring that its learned representations remain grounded in real user behavior.

3.2 Detecting Neurons Encoding Popularity Bias

To quantify how much each SAE neuron contributes to popularity bias, we compare neuron activations when processing R^{pop} and R^{unpop}. Each synthetic dataset is fed into the pretrained SAE, and the activation levels of the hidden layer neurons are recorded.

Prior work suggests that as neural networks grow in size, neuron activation distributions tend to approximate a Gaussian distribution [10, 18, 27]. We leverage this property to use Cohen's d [6] to measure the effect size of activation differences for each neuron j:

$$d_j = \frac{\mu_{j,pop} - \mu_{j,unpop}}{\sqrt{\frac{\sigma_{j,pop}^2 + \sigma_{j,unpop}^2}{2}}} \tag{2}$$

where $\mu_{j,pop}$ and $\mu_{j,unpop}$ are the mean activations of neuron j under R^{pop} and R^{unpop}, respectively, and $\sigma_{j,pop}$ and $\sigma_{j,unpop}$ are the corresponding standard deviations.

Cohen's d provides a normalized measure of the difference between two distributions. A high absolute value of d_j indicates that neuron j is strongly responsive to popularity-related patterns. Positive values suggest alignment with popular content, while negative values imply a focus on unpopular or niche items.

4 Popularity Bias Mitigation Approach

To counteract popularity bias in recommendations, we use the computed Cohen's d values to identify neurons most responsible for encoding popularity signals. Our method, `PopSteer`, applies a process called *neuron steering*, which systematically adjusts these neuron activations to reduce bias and promote fairer item exposure.

Neuron steering modifies the hidden activations within the SAE to either amplify or suppress the influence of popularity-related neurons. The adjustment is guided by each neuron's d_j score, which reflects its alignment with popular or unpopular items. We first assign a weight w_j to each neuron j based on the normalized absolute value of its d_j score. This weight determines how much each neuron's activation will be adjusted:

$$w_j = \alpha \cdot \frac{|d_j| - \min(|d|)}{\max(|d|) - \min(|d|)} \tag{3}$$

where α is a tunable hyperparameter that controls the overall strength of the steering and d is the Cohen's d values of all neurons. Neurons with stronger associations to popularity bias (higher $|d_j|$) receive larger weights. Next, we modify the original activation a_j of neuron j to obtain a new activation a'_j:

$$a'_j = \begin{cases} a_j + w_j \cdot \sigma_j, & \text{if } d_j < 0 \\ a_j - w_j \cdot \sigma_j, & \text{if } d_j > 0 \end{cases} \tag{4}$$

[1]We address the dead neuron issue during SAE training using the auxiliary loss, but omit the details here due to space constraints (see [8] for more).

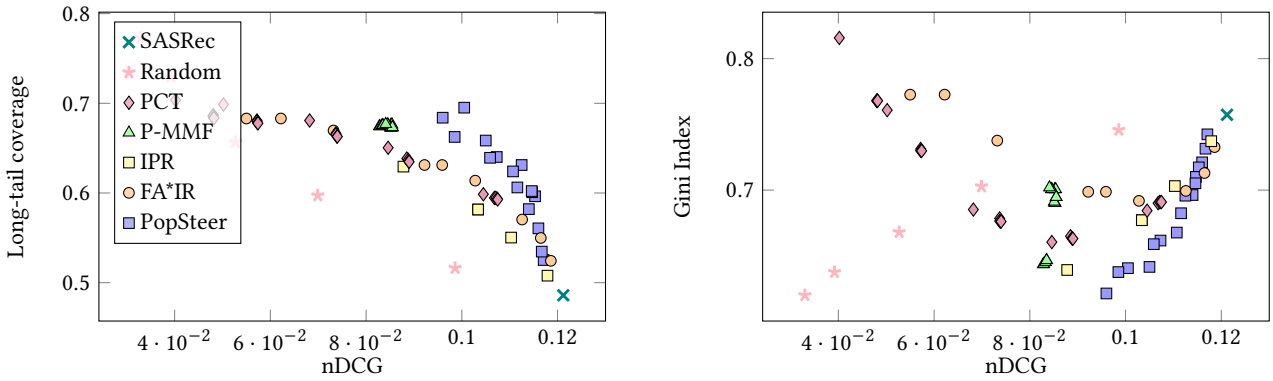

Figure 1: Performance comparison of `PopSteer` method with the baselines in terms of nDCG and fairness metrics on ML-1M.

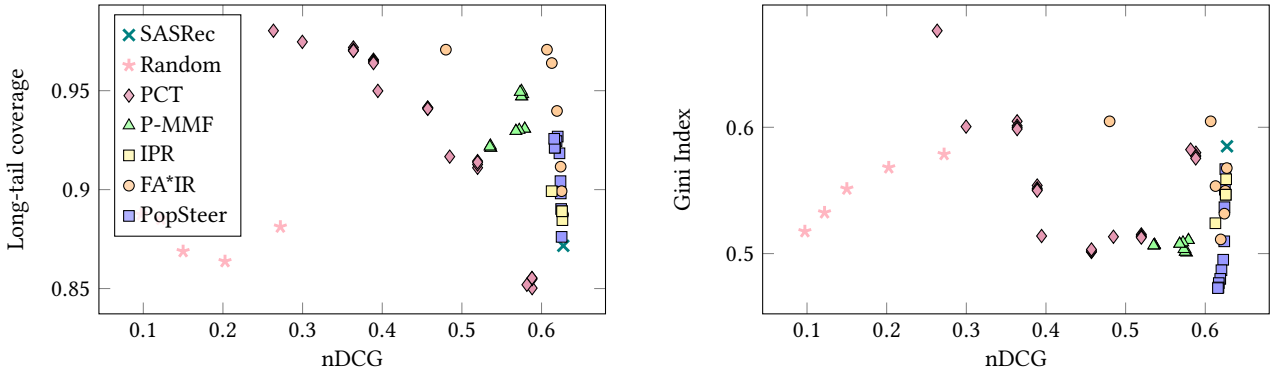

Figure 2: Performance comparison of `PopSteer` method with the baselines in terms of nDCG and fairness metrics on Last.fm.

This adjustment boosts neurons promoting unpopular items ($d_j < 0$) and suppresses those favoring popular items ($d_j > 0$). By realigning neuron activations, `PopSteer` mitigates the overrepresentation of popular items, leading to more balanced exposure across items. The updated SAE then returns the modified user embedding, which is used alongside item embeddings from the base recommendation model to generate the final recommendation list.

Table 1: Statistics of the datasets.

Dataset	#Users	#Items	#Interactions	Density
ML-1M	6,040	3,417	999,611	0.048
Last.fm	1,363	2,976	64,286	0.016

5 Experiments

This section outlines our experimental setup, including datasets, evaluation metrics, baselines, and experimental results.

5.1 Dataset

We conduct experiments on two public datasets: MovieLens 1M (ML-1M) [12] and Last.fm [24]. Following standard preprocessing [13, 16], we create 5-core sample on both datasets, where each user has at least 5 interactions and each item is interacted with by at least 5 users. Both datasets include timestamp informtion, enabling their use in sequential recommendation task. Table 1 summarizes the statistical properties of these datasets.

5.2 Experimental setup

We evaluate recommendation performance along two dimensions: ranking performance and fairness. Ranking quality is measured using nDCG@10. To evaluate fairness, we report long-tail coverage [20]—the proportion of recommendations that come from the unpopular (tail) item set ($\mathcal{I}^{unpop}$)—and the Gini Index [2], which captures how uniformly items are exposed across users. A lower Gini Index indicates fairer exposure.

We compare our `PopSteer` method with the following baselines:

- **Inverse Popularity Ranking (IPR) [30]:** A reweighting approach that adjusts relevance score for user-item pairs based on inverse popularity: $\tilde{s}_{u,i} = s_{u,i}/(1 + \alpha\, \rho_i)$, where $\rho_i = \text{pop}(i) / \max_{j \in \mathcal{I}} \text{pop}(j)$ and α is a hyperparameter controlling the degree of mitigating bias. We tune $\alpha \in \{0.1, 0.3, 0.5, 0.7, 1\}$.
- **FA*IR [29]:** A post-processing approach that enforces fairness by ensuring a minimum proportion of unpopular items appear within each prefix of the top-k list. In our experiment, we set the proportion of unpopular items to $p \in \{0.3, 0.5, 0.7, 0.9, 0.99\}$ and the significance level to $\alpha \in \{0.01, 0.05, 0.1\}$. We set the size of long recommendation lists to 500.
- **PCT [26]:** A two-sided method that balances user-level and system-level exposure. A solver computes exposure targets

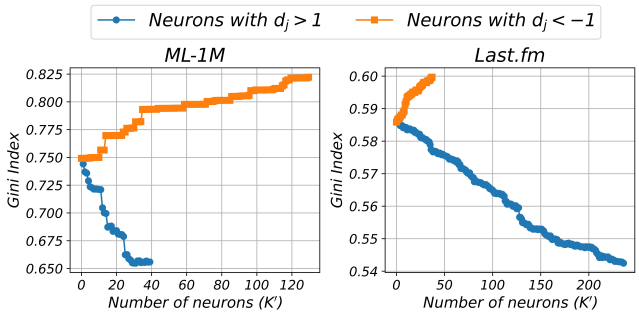

Figure 3: Interpretability analysis of PopSteer: effect of deactivating K' identified neurons linked to popularity bias.

using linear programming, followed by a reranker that modifies each user's recommendation list using a modified MMR strategy. Hyperparameters are tuned similar to FA*IR.

- **P-MMF [28]:** A resource allocation algorithm based on dual-space optimization. It dynamically and proportionally adjusts exposure across popular, unpopular, and mid items according to group sizes, updating after each user interaction. We tune the learning rate $\eta \in \{0.0001, 0.001, 0.01\}$ and the fairness constraint parameter $\lambda \in \{0.001, 0.01, 0.1, 1, 10\}$.
- **Random:** A simple baseline that randomly selects k items from an initial longer recommendation list. We test the method using list sizes $\{15, 30, 50, 75, 100\}$.

We use SASRec [16] as the backbone recommender, which leverages self-attention to capture temporal dynamics in user behavior. For splitting the data, user histories are sorted chronologically, with the most recent interaction used for testing, the second-most for validation, and the remainder for training.

SASRec is trained with a learning rate of 0.001 and early stopping (patience = 10 epochs) based on validation nDCG@10. For SAE, we use total reconstruction loss as the objective [8]. The SAE includes two key hyperparameters: the scale factor $s \in \{8, 32, 64\}$, which determines the size of the hidden layer relative to the input, and the sparsity level $K \in \{16, 32, 48\}$, which specifies the number of top activations retained in the hidden layer. Both SASRec and SAE are trained with the Adam optimizer and a batch size of 4096.

Our proposed PopSteer method involves two tunable hyperparameters. The first is α, which controls the intensity of neuron steering (see Eq. 3), which we set to $\alpha \in \{1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0\}$. The second is $\mathcal{N}$, which specifies how many neurons with the highest absolute Cohen's d value are selected for activation adjustment, as detailed in Section 4. We tune $\mathcal{N} \in \{1024, 2048, 3072, 4096\}$ out of total neurons $N = 4096$. Our implementation and code is available at https://github.com/parepic/PopSteer.

5.3 Experimental results

Figures 1 and 2 present our main results, comparing PopSteer with SASRec and several baselines across both nDCG and fairness metrics on the ML-1M and Last.fm.

On ML-1M (Figure 1), PopSteer consistently outperforms all baselines in enhancing fairness while maintaining strong accuracy. P-MMF achieves moderate fairness improvements but at the cost of notable accuracy loss and limited flexibility. While PCT, IPR, and FA*IR show performance comparable to PopSteer near nDCG~0.12, two key patterns highlight PopSteer's superiority.

Table 2: Ablation study: PopSteer (PS) vs. Random Noisification (ML-1M: $\alpha = 1.5$, $\xi = 0.35$; Last.fm: $\alpha = 4.0$, $\xi = 1.0$).

		NDCG@10 ↑		LT Cov.@10 ↑		Gini@10 ↓	
Dataset	$\mathcal{N}$	PS	Noise	PS	Noise	PS	Noise
ML-1M	0	0.1166	0.1166	0.4917	0.4917	0.7491	0.7491
	1024	**0.1169**	0.1123	0.5065	**0.5333**	0.7470	**0.7387**
	2048	**0.1167**	0.1129	**0.5346**	0.5440	**0.7315**	0.7376
	3072	**0.1135**	0.1103	**0.5704**	0.5463	**0.7016**	0.7367
	4096	0.1106	**0.1110**	**0.6241**	0.5485	**0.6677**	0.7394
Last.fm	0	0.6247	0.6247	0.8722	0.8722	0.5859	0.5859
	1024	**0.6186**	0.5867	**0.9015**	0.8575	**0.5193**	0.5819
	2048	**0.6126**	0.5932	**0.9167**	0.8598	**0.4876**	0.5885
	3072	**0.6044**	0.5977	**0.9161**	0.8677	**0.4840**	0.5824
	4096	**0.6036**	0.6015	**0.9150**	0.8628	**0.4826**	0.5878

First, PopSteer achieves a more favorable trade-off between fairness and accuracy, delivering significantly better fairness gains for only slight reductions in nDCG. Second, other baselines often continue to lose accuracy without corresponding fairness gains, particularly below nDCG~0.11. In contrast, PopSteer maintains stability—avoiding unnecessary accuracy degradation when fairness stabilizes. This behavior suggests the reliability of PopSteer.

A similar trend appears on Last.fm (Figure 2). Although FA*IR achieves marginally higher long-tail coverage in some settings, PopSteer consistently yields better Gini Index scores, indicating a more balanced distribution of item exposure. Overall, PopSteer demonstrates greater reliability, consistency, and stability in balancing fairness and recommendation quality.

5.3.1 Interpretability Analysis. To assess the interpretability of PopSteer, we conducted a controlled neuron manipulation study. Specifically, we manually deactivated the top-K' neurons identified by PopSteer as most strongly associated with popularity (Cohen's $d > 1$) or unpopularity (Cohen's $d < -1$).

Figure 3 illustrates how Gini Index evolves as these neurons are progressively turned off. When neurons contributing to popularity are deactivated (blue line), the Gini Index consistently decreases—highlighting their role in reinforcing popularity bias. Conversely, turning off neurons associated with unpopular items (orange line) leads to an increase in Gini Index, confirming their importance in enhancing exposure fairness. These findings demonstrate that PopSteer not only mitigates popularity bias effectively but also provides actionable, interpretable insights into the neural basis of bias—enabling more transparent and controllable interventions.

5.3.2 Ablation study. To verify the effectiveness of controlled steering, we perform a comparative analysis with a baseline in which random Gaussian noise is added to neuron activations instead of targeted neuron steering. Specifically, we vary the number of neurons $\mathcal{N}$ subjected to perturbation, while fixing the hyperparameters α for PopSteer and the standard deviation (ξ) for the Gaussian noise. Both hyperparameters (α and ξ) are tuned individually for each dataset to ensure that the resulting reduction in nDCG@10 does not exceed 10% compared to the original unsteered model.

Table 2 presents the results of our ablation study, comparing PopSteer to the noise-based baseline. We observe that PopSteer consistently yields superior fairness outcomes relative to random Gaussian noise perturbation across both datasets. Specifically, while the noise-based method shows minimal variability and no meaningful increase in fairness metrics as we vary the number of perturbed neurons ($\mathcal{N}$), PopSteer demonstrates a clear and predictable improvement in fairness with increasing $\mathcal{N}$. This indicates that the fairness improvements achieved by PopSteer are directly attributable to targeted neuron selection rather than random perturbations.

6 Conclusion

This work introduces PopSteer, a post-hoc method for interpreting and mitigating popularity bias in recommendation models using Sparse Autoencoder. By identifying and adjusting neuron activations linked to popularity signals, PopSteer improves exposure fairness without sacrificing accuracy. Our experiments on public datasets demonstrate its effectiveness and reliability compared to existing baselines. Beyond performance, the method offers interpretability and fine-grained control, making it a practical and transparent solution for fairness-aware recommendation.

References

[1] 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.

[2] Arda Antikacioglu and R Ravi. 2017. Post processing recommender systems for diversity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 707–716.

[3] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. 2023. Towards Monosemanticity: Decomposing Language Models With Dictionary Learning. *Transformer Circuits Thread* (2023). https://transformer-circuits.pub/2023/monosemantic-features/index.html.

[4] Rocío Cañamares and Pablo Castells. 2018. Should I follow the crowd? A probabilistic analysis of the effectiveness of popularity in recommender systems. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 415–424.

[5] Jiawei Chen, Hande Dong, Yang Qiu, Xiangnan He, Xin Xin, Liang Chen, Guli Lin, and Keping Yang. 2021. AutoDebias: Learning to debias for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 21–30.

[6] Jacob Cohen. 2013. *Statistical power analysis for the behavioral sciences*. routledge.

[7] Esin Durmus, Alex Tamkin, Jack Clark, Jerry Wei, Jonathan Marcus, Joshua Batson, Kunal Handa, Liane Lovitt, Meg Tong, Miles McCain, Oliver Rausch, Saffron Huang, Sam Bowman, Stuart Ritchie, Tom Henighan, and Deep Ganguli. 2024. *Evaluating Feature Steering: A Case Study in Mitigating Social Biases*. https://anthropic.com/research/evaluating-feature-steering

[8] Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. 2024. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093* (2024).

[9] Sophie Greenwood, Sudalakshmee Chiniah, and Nikhil Garg. 2024. User-item fairness tradeoffs in recommendations. *Advances in Neural Information Processing Systems* 37 (2024), 114236–114288.

[10] Muhammad Umair Haider, Hammad Rizwan, Hassan Sajjad, Peizhong Ju, and AB Siddique. 2025. Neurons Speak in Ranges: Breaking Free from Discrete Neuronal Attribution. *arXiv preprint arXiv:2502.06809* (2025).

[11] Ruben Härle, Felix Friedrich, Manuel Brack, Björn Deiseroth, Patrick Schramowski, and Kristian Kersting. 2024. SCAR: Sparse Conditioned Autoencoders for Concept Detection and Steering in LLMs. *arXiv preprint arXiv:2411.07122* (2024).

[12] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2015), 1–19.

[13] Ruining He, Wang-Cheng Kang, and Julian McAuley. 2017. Translation-based recommendation. In *Proceedings of the eleventh ACM conference on recommender systems*. 161–169.

[14] Praveen Hegde. [n. d.]. Effectiveness of Sparse Autoencoder for understanding and removing gender bias in LLMs. In *NeurIPS 2024 Workshop on Scientific Methods for Understanding Deep Learning*.

[15] Jin Huang, Harrie Oosterhuis, Masoud Mansoury, Herke Van Hoof, and Maarten de Rijke. 2024. Going beyond popularity and positivity bias: Correcting for multifactorial bias in recommender systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 416–426.

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

[17] Michael Lan, Philip Torr, Austin Meek, Ashkan Khakzar, David Krueger, and Fazl Barez. 2024. Sparse autoencoders reveal universal feature spaces across large language models. *arXiv preprint arXiv:2410.06981* (2024).

[18] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. 2017. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165* (2017).

[19] Siyi Liu and Yujia Zheng. 2020. Long-tail session-based recommendation. In *Proceedings of the 14th ACM conference on recommender systems*. 509–514.

[20] Masoud Mansoury. 2022. Understanding and mitigating multi-sided exposure bias in recommender systems. *ACM SIGWEB Newsletter* 2022, Autumn (2022), 1–4.

[21] Masoud Mansoury, Himan Abdollahpouri, Mykola Pechenizkiy, Bamshad Mobasher, and Robin Burke. 2020. Feedback loop and bias amplification in recommender systems. In *Proceedings of the 29th ACM international conference on information & knowledge management*. 2145–2148.

[22] Masoud Mansoury, Bamshad Mobasher, and Herke van Hoof. 2024. Mitigating exposure bias in online learning to rank recommendation: A novel reward model for cascading bandits. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 1638–1648.

[23] Charles O'Neill, Christine Ye, Kartheik Iyer, and John F Wu. 2024. Disentangling dense embeddings with sparse autoencoders. *arXiv preprint arXiv:2408.00657* (2024).

[24] Markus Schedl. 2016. The lfm-1b dataset for music retrieval and recommendation. In *Proceedings of the 2016 ACM on international conference on multimedia retrieval*. 103–110.

[25] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. In *international conference on machine learning*. PMLR, 1670–1679.

[26] Chenyang Wang, Yankai Liu, Yuanqing Yu, Weizhi Ma, Min Zhang, Yiqun Liu, Haitao Zeng, Junlan Feng, and Chao Deng. 2023. Two-sided calibration for quality-aware responsible recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 223–233.

[27] Pierre Wolinski and Julyan Arbel. 2022. Gaussian pre-activations in neural networks: Myth or reality? *arXiv preprint arXiv:2205.12379* (2022).

[28] Chen Xu, Sirui Chen, Jun Xu, Weiran Shen, Xiao Zhang, Gang Wang, and Zhenhua Dong. 2023. P-MMF: Provider max-min fairness re-ranking in recommender system. In *Proceedings of the ACM Web Conference 2023*. 3701–3711.

[29] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. 2017. Fa* ir: A fair top-k ranking algorithm. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1569–1578.

[30] Mi Zhang and Neil Hurley. 2010. Niche product retrieval in top-n recommendation. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Vol. 1. IEEE, 74–81.

[31] Yin Zhang, Ruoxi Wang, Derek Zhiyuan Cheng, Tiansheng Yao, Xinyang Yi, Lichan Hong, James Caverlee, and Ed H Chi. 2023. Empowering long-tail item recommendation through cross decoupling network (CDN). In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 5608–5617.

[32] Ziwei Zhu and James Caverlee. 2022. Fighting mainstream bias in recommender systems via local fine tuning. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 1497–1506.

[33] Ziwei Zhu, Yun He, Xing Zhao, Yin Zhang, Jianling Wang, and James Caverlee. 2021. Popularity-opportunity bias in collaborative filtering. In *Proceedings of the 14th ACM international conference on web search and data mining*. 85–93.