Addressing Popularity Bias in Recommender Systems: An Exploration of Self-Supervised Learning Models

ANASTASIIA KLIMASHEVSKAIA, MediaFutures, University of Bergen, Norway
MEHDI ELAHI, MediaFutures, University of Bergen, Norway
CHRISTOPH TRATTNER, MediaFutures, University of Bergen, Norway

The rapid growth of the volume and variety of online media content has made it increasingly challenging for users to discover fresh content that meets their particular needs and tastes. Recommender Systems are digital tools that support users in navigating the plethora of available items. While these systems may offer several benefits, they may also create or reinforce certain undesired effects, including *Popularity Bias*, i.e., a short list of popular items becoming more popular while a long list of unpopular ones becoming even more unpopular.

In this paper, we focus on this challenge and propose a novel recommendation approach that can generate accurate recommendations while effectively mitigating the popularity bias. Our proposed approach adopts models based on *Self-Supervised Learning (SSL)* that have recently drawn considerable attention in various application domains. Such models are known to enable recommender systems to exploit automatic mechanisms for data annotation hence providing self-supervisory signals for better training of the system from the available data. We considered various recommendation techniques based on the SSL model and compared their impact on popularity bias mitigation measured in terms of Average Recommendation Popularity (ARP), Gini-index, and Coverage. The results showed that SSL models could successfully mitigate the popularity bias while still maintaining the accuracy of the recommendation.

CCS Concepts: • **Information systems → Recommender systems**; **Personalization**; • **Computing methodologies → Learning from implicit feedback**.

Additional Key Words and Phrases: Recommender systems, Self-supervised learning, Evaluation, Beyond accuracy

ACM Reference Format:
Anastasiia Klimashevskaia, Mehdi Elahi, and Christoph Trattner. 2023. Addressing Popularity Bias in Recommender Systems: An Exploration of Self-Supervised Learning Models. In *Adjunct Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization (UMAP '23 Adjunct), June 26–29, 2023, Limassol, Cyprus.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3563359.3597409

1 INTRODUCTION

Nowadays, finding the right media to consume has become a major challenge for users due to the enormous volume, variety, and velocity of producing and sharing content online. YouTube, as an example of popular media-sharing applications, has about 1.5 billion active users who consume an incredible number of 5 billion videos per day. Users may feel overwhelmed when they need to choose from an unlimited selection of media, and may struggle to find fresh content that is relevant to their interests. Recommender systems can help address this challenge by generating personalized suggestions based on users' specific tastes and interests, rather than suggesting popular media based on generic mainstream tastes [18].

Permission to make digital or hard copies of part or all 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 third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2023 Copyright held by the owner/author(s).
Manuscript submitted to ACM

Despite the benefit of recommender systems, such systems may also introduce or reinforce certain undesired effects [5]. An example of such effects is *Popularity Bias* where a few popular items (a.k.a., short head) become more popular and many unpopular items (a.k.a., long tail) get more unpopular [1, 2, 12]. One possible approach to mitigate this bias is to generate and integrate additional data. The rationale behind this approach lies in the observation that the popularity bias may result from the lack of sufficient interaction data for unpopular items [19]. As a result, recommendation models may not be able to predict the relevance score for these items with high confidence. By incorporating additional data the model may be better equipped to address this problem and compensate for the missing data.

In recent times, a novel line of research has focused on recommendation based on *Self-Supervised Learning (SSL)* models [24]. In general, SSL models can be utilized when dealing with large amounts of unlabeled and unstructured data [16]. SSL models are often adopted to construct a combination of unsupervised pre-training and supervised fine-tuning. In the context of recommender systems, SSL models can enable these systems to better train on the available data by automatically re-annotate them. Hence, such a learning mechanism attempts to train the prediction model to form a new representation from the input data by auto-labeling them to ultimately obtain an additional supervisory signal from data. The experiments of early works using SSL models demonstrated promising results for the task of recommendation in different domains. However, these works have predominantly focused on the accuracy of recommendation.

In this paper, we go beyond the accuracy of recommendation and propose a novel approach based on SSL that can mitigate the challenge of popularity bias in recommender systems. We evaluated the proposed approach considering metrics indicative of popularity bias in the recommendation, i.e., Average Recommendation Popularity (ARP), Gini index, and Catalog Coverage. In addition to them, we considered common metrics for measuring accuracy and ranking quality, i.e., Hit-ratio, Precision, Recall, and nDCG.

We considered different SSL models and compared the recommendation generated by them against a classical Collaborative Filtering (CF) approach, i.e., Matrix Factoriztion (MF). The results of the experiments showed that recommendations generated by different SSL models can effectively mitigate popularity bias but to different extents while maintaining the accuracy of recommendation.

In summary, the present paper makes the following contributions:

- We propose a novel approach for popularity bias mitigation in recommender systems based on SSL models and evaluate the performance of this approach in comparison to the classical recommendation model (i.e., Matrix Factorization);
- We compare recommendations based on a range of SSL models including, SGL, BUIR, SSL4Rec, MixGCF, SimGCL, and XSimGCL; Our results demonstrated their superior performance in balancing the accuracy of recommendation while mitigating the popularity bias at the same time;
- We considered several metrics in our evaluation that can reflect on different aspects of the performance of recommender systems, i.e., accuracy, ranking quality, coverage and bias mitigation.

The remainder of the paper is organized as follows: In Section 2, we first review the related works on popularity bias in recommender systems and then discuss SSL models and approaches that employ them for recommendation. In Section 3, we describe the details of our models and experimental methodology. In Section 4, we explain the results of the experiments and in Section 5, provide a summary of findings and future plans.

2 RELATED WORK

In this section, we discuss related works on two different lines of research, i.e., mitigation strategies for popularity bias, and recommendation approaches based on Self-supervised Learning (SSL) models.

2.1 Popularity Bias Mitigation

Recommender systems are susceptible to a well-known type of bias so-called popularity bias, which is believed to be inherited from the input data. This type of bias manifests itself in the presence of a small number of popular items referred to as the "short head", and a large number of less popular (or unpopular) items referred to as the "long tail". By promoting the unpopular items in the long tail, these systems can increase their popularity, providing various benefits to users and media platforms. Examples of such benefits are discovery of interesting less-explored items, and enhancing the profit of platforms (e.g., Twitter and YouTube), due to the increase in the user engagement and content consumption.

One of the notable approaches for popularity bias mitigation was introduced by [3] where the performance of a user-specific approach for the mitigation of popularity bias was evaluated. The proposed approach was called Calibrated Popularity (CP) and it was focused on minimizing the distance between User Popularity Deviation (UPD) [11, 20]. The results of the experiment showed that the proposed approach was effective in mitigating the bias on the *user level* by taking into account the individual tendencies of users toward the popularity of items. In a more recent work [6], the authors investigated the impact of different recommendation techniques on the popularity bias not only on the user level but also on the *item level*. The results examined the bias in various application domains (e.g., movies and micro-blogging) and showed that recommendation techniques may increase the bias on the user level while reducing it on the item level. In [13], the authors extended the idea and investigated the popularity bias from *platform level* perspective. The outcome of their extensive evaluation showed that certain approaches can be effective in the bias mitigation on the platform level while others (e.g., CP) may focus on reducing the bias mainly on the user level.

2.2 Self-supervised Learning (SSL) Models

SSL models originally come from the Machine Learning community and are known to be effective in leveraging unlabeled, unstructured data and addressing the challenge in the relevant research fields [16]. These models represent a combination of supervised and unsupervised learning, where only a part of the input data may be initially labeled. More particularly, these models are frequently constructed as the combined form of unsupervised *pre-training* and supervised *fine-tuning*. The approach, in this case, is then aims to obtain supervisory signals from the date and incorporated them for any task at hand. This type of machine learning models attempts to learn a general representation from the input data and auto-label them for further use.

SSL models also found applications in image classification task [10]. Due to the absence of labels for the images in the data, the intuition behind learning models (e.g., neural models) in this research field is to usually assume that each image representing its own particular class (label). However, a more robust learning can be gone beyond that and by adopting SSL, data can be better augment in learning process [7]. In Natural Language Processing (NLP), language models were also benefited from SSL. For example, Generative Pre-trained Transformer (GPT) [17] generates natural language based on token distribution estimation and predicting the likelihood of certain language tokens occurring together in a sentence or phrase. This form of learning and prediction resembles itself as a particular case of SSL. Bidirectional Encoder Representations from Transformers (BERT) [4] is another example of an NLP application that uses a language model pre-training objective and can be fine-tuned with an additional output layer. This method also similarly utilizes the extended knowledge about the sequential nature of language for learning and predicting the new sequences. SSL approaches are quite versatile and allow exploiting auto-labeled and different modalities such as textual, audio or image [16]. Contrastive Predictive Coding (CPC) is a particular example of such models that is designed to exploit different modalities in the data with the same principle of training SSL. SSL models are overall powerful tools that can optimize the efficiency of such data and improve robustness of the model.

2.3 Recommendation based on Self-supervised Learning (SSL)

Similar to the noted research fields, SSL has also gained attention in recommendation domain over the past years and been used to construct better representations of recommended items from both their content and context point of views. Many of these models enabled data augmentation with NLP, graph learning techniques for better recommendation accuracy. Such SSL based recommendation models have been shown to be a robust approach with accurate results, with certain limitations nonetheless [10].

The quality of the aforementioned data augmentation is highly dependent on its choice of it, which is mostly defined by heuristics. A possible method to identify a proper data augmentation for a specific application domain with specific data is usually based on trial and error. Another drawback of self-supervised recommendation is the typical black box nature of these models, which makes them be not suitable for generation explanations to the recommendation output. Last but not least, as with many other recommendation approaches, self-supervised recommendation models are still vulnerable to attacks and manipulation with data poisoning. For more details on all the existing approaches one is advised to address the survey in [24] where authors collected extensive information in a systematic literature review on applications of SSL models for recommendation.

In contrast to these prior works that have centered on exploring the feasibility of generating recommendations based on SSL models, our work differs in its target research focus. More particularly, we address the challenge of popularity bias mitigation in recommender systems that are reliant upon SSL models. To the best of our knowledge, the literature on this topic is scarce, if not entirely absent. Therefore, our proposed approach fills a gap in the current research in this field.

3 METHODOLOGY

In this section, we explain the methodology we employed to perform our research work.

3.0.1 SSL Models Utilized for Recommendation. We have considered the following set of SSL models to generate recommendation:

- **SGL** [21]: builds user-item graph on top of the user interaction data and uses it for the task of reocmmendation. The idea behind the model is to enhance the classical *supervised* recommendation by using a *self-supervised* extension. This can be performed by devising different types of data augmentation to construct an auxiliary contrastive. This is shown to result in improved robustness and accuracy in the recommendation.
- **BUIR** [15]: employs asymmetric graph encoders (labeled as “online” and “target” networks) to supervise each other without negative sampling. One of the networks receives a user-item pair and user representation and then learns to predict item representation with the help of the other network, and vice versa.
- **SSL4Rec** [22]: aims to enhance item representation by leveraging latent relationships between item features. The model is specifically designed to deal with the label sparsity in the data by learning better relationships among items. The model employs a novel data augmentation adopting the correlations among data.

- **MixGCF** [9]: leverages Graph Neural Networks (GNN) and builds a user-item graph structure of the user interaction data. More specifically, the model trains a GNN and uses it to generate recommendation. The trained GNN uses (negative) sampling from data by using a mechanism called the hop mixing technique, i.e., making the synthetic negative data by aggregating embeddings from different layers of neighborhoods. This is shown to consistently improve the recommendation.
- **SimGCL** [25]: employees Graph Contrastive Learning (GCL) primarily to tackle the data sparsity in recommender systems. The model analyzes the input data for a recommender system and extracts self-supervised signals from them to enhance the model training. More particularly, the model operates by learning evenly distributed user-item representations from the data that was shown to improve accuracy and efficiency in recommendation.
- **XSimGCL** [23]: is an extended version of SimGCL model mainly to put emphasis on promoting the recommendation of long-tail items. This model tends to discard the common graph augmentations methods due to their ineffectiveness and instead employs a simple data augmentation to generate labels for better model training.

As a baseline and for the sake of comparison, we considered the classical collaborative filtering approach, i.e., Matrix Factorization (MF)[14]. For the implementation of the models, we relied on an open-source recommendation framework called SELFRec[1]. This framework offered implementations of the considered recommendation models. All the models were trained with parameters provided by the framework, with additional optimization with PyTorch's Adam optimizer[2]. We further optimized parameters such as *batch size*.

3.0.2 Dataset. The models were trained on a commonly used movie recommendation dataset MovieLens 1M [8] containing 1,000,209 anonymous ratings of approximately 3,900 movies provided by 6,040 users. The training and testing sets were made by following the common methodology (7:3 split), i.e., splitting the dataset into 70% training and 30% testing sets. After training the model, lists of 10 and 20 most relevant recommended items are generated for each user. The quality of the recommendation in terms of various metrics was then estimated by comparing the recommendations and the items in the test set. Due to space limitations we only report the result of the recommendation of size 10 since we obtained very similar results for the recommendation of size 20.

3.0.3 Metrics. Two groups of metrics were used for performance evaluation — (i) metrics for measuring the level of popularity bias in the recommendation, and, (ii) metrics measuring the accuracy and ranking quality of the recommendation. For the former group, we considered and implemented the following metrics:

- **Average Recommendation Popularity (ARP)**: estimates the popularity of recommendations produced by the model. The popularity of an item can be defined as the number of interactions it has in the dataset divided by the total number of users. Knowing each item's popularity at a moment of time, one can calculate the average popularity of recommendation list generated for a user, and then take another average over all the users in the dataset.
- **Gini Index**: is commonly adopted to estimate the distribution of exposure that each item in the system receives due to being recommended. The metric ranges from 0 to 1 with 0 meaning an absolutely equitable distribution of exposure and 1 being the case of absolute inequality. Assessing this metric can help one to check whether there is only a small group of items being recommended repeatedly or whether there is an actual diversity.

[1]https://github.com/Coder-Yu/SELFRec
[2]https://pytorch.org/docs/stable/generated/torch.optim.Adam.html

- **Catalog Coverage**: checks how many items from the whole system catalog have been recommended at least to one user. This value is normalized by dividing it by the total number of items available and ranges from 0 to 1 as well, with the extreme cases of 1 meaning the absolute coverage of all items and 0 indicating at no item coverage.

We implemented the above metrics since the original framework did not include them in the implementation. For the latter group, we considered Hit Ratio, Precision, Recall and Normalized Discounted Cumulative Gain (nDCG), provided by the framework.

4 RESULTS

The results of the experiment are presented in Table 1 which compares the performance of different SSL models for the task of Top-10 recommendation with respect to different evaluation metrics. The directional arrows in the table indicate whether a superior or inferior value is desirable for the metric.

Table 1. Performance comparison of different models adopted for Top-10 item recommendation task. The directional arrows in the table indicate whether a superior or inferior value is desirable for the metric. Highlighted values are the best among all the considered models.

Rec.	Model	Accuracy & Ranking Quality				Popularity Bias		
		Hit Ratio↑	Precision↑	Recall↑	nDCG↑	Gini Index↓	ARP↓	Coverage↑
CF	**MF**	**0.114**	**0.220**	**0.158**	**0.266**	0.880	0.233	0.495
SSL	**SGL**	0.012	0.024	0.014	0.026	**0.632**	0.090	0.709
	BUIR	0.107	0.206	0.139	0.253	0.982	0.352	0.120
	SSL4Rec	0.022	0.043	0.037	0.052	0.861	**0.081**	0.519
	MixGCF	0.075	0.145	0.092	0.173	0.866	0.253	**0.742**
	SimGCL	0.068	0.130	0.108	0.162	0.836	0.145	0.435
	XSimGCL	0.076	0.146	0.117	0.181	0.830	0.147	0.461

As shown, in terms of accuracy and ranking metrics, surprisingly, the classical collaborative filtering approach (i.e., MF model) achieved the best results although the observed values were very similar to the ones observed for the best-performing SSL model (i.e., BUIR). For the MF model, the values of Hit Ratio, Precision, Recall, and nDCG were 0.114, 0.220, 0.158, and 0.266, while for the BUIR model, these values were 0.107, 0.206, 0.139, and 0.253, respectively. SGL model demonstrates the worst performance in terms of accuracy and ranking quality of the recommendation.

Contrary to accuracy results, SSL models performed the best in mitigating the popularity bias in the recommendation. Among the SSL models, SSL4Rec, SGL, and MixGCF were those that achieved superior results and outperformed other models in terms of metrics indicative of popularity bias.

According to the results, for the average recommendation popularity (ARP) metric, the best score was observed for SSL4Rec model with the value of 0.081. This is a significant improvement (nearly 3 times) in comparison to the MF model with an ARP score of 0.233. In terms of Gini index, the best score was obtained by SGL model with a value of 0.632, about 28% improvement compared to the MF model with a value of 0.880. For the Catalog Coverage, the best score was observed for MixGCF model with the value of 0.742. This value presents about 33% improvement compared to the MF model with a value of 0.495. Surprisingly, the worst performance for these metrics was observed for BUIR model.

Overall, our results demonstrated the effectiveness of SSL models in mitigating popularity bias in recommender systems as well as their potential in maintaining the quality of the recommendation in terms of accuracy and ranking quality. It is worth noting that our results showed surprising observations in certain cases where improving a model resulted in both better accuracy and popularity bias mitigation in the recommendation. An example of such cases is XSimGCL model which is an upgraded version of SimGCL model. According to the results, XSimGCL outperformed SimGCL in terms of nearly all metrics which is contrary to the commonly accepted trade-off between accuracy and bias mitigation in recommender systems.

5 SUMMARY OF FINDINGS & FUTURE WORK

In this paper, we address the challenge of popularity bias in recommender systems by proposing a novel recommendation approach based on Self-Supervised Learning (SSL) models. While prior works have found compelling evidence for the excellence of SSL models in generating accurate recommendation, there exists a gap concerning the potential of SSL models to address issues such as bias in the recommendation. In order to evaluate our proposed approach, we considered several SSL models and compared their performance in terms of several metrics indicative of popularity bias, i.e., Gini index, Average Recommendation Popularity (ARP), and Catalog Coverage. The results showed the effectiveness of SSL models in mitigating popularity bias while maintaining a good level of recommendation accuracy.

As future plans, we would like to extend our evaluation to encompass a larger dataset, including a more diverse range of application domains. Our current analysis may be limited by its focus on a particular dataset and hence a more comprehensive evaluation could provide valuable insights. Moreover, we plan to conduct a real user study to compare the performance of different recommendation approaches based on SSL models in terms of popularity bias. By recruiting participants, we will study the extent to which the SSL models perform well in recommending both popular and niche items.

ACKNOWLEDGMENTS

This work was in part supported by the Research Council of Norway with funding to MediaFutures: Research Centre for Responsible Media Technology and Innovation, through the Centre for Research-based Innovation scheme, project number 309339.

REFERENCES

[1] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. Controlling popularity bias in learning-to-rank recommendation. In *ACM RecSys '17*, pages 42–46, 2017.

[2] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. Managing popularity bias in recommender systems with personalized re-ranking. In *FLAIRS '19*, pages 413–418, 2019.

[3] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, Bamshad Mobasher, and Edward Malthouse. User-centered evaluation of popularity bias in recommender systems. In *Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization*, pages 119–129, 2021.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

[5] Mehdi Elahi, Dietmar Jannach, Lars Skjærven, Erik Knudsen, Helle Sjøvaag, Kristian Tolonen, Øyvind Holmstad, Igor Pipkin, Eivind Throndsen, Agnes Stenbom, et al. Towards responsible media recommendation. *AI and Ethics*, pages 1–12, 2022.

[6] Mehdi Elahi, Danial Khosh Kholgh, Mohammad Sina Kiarostami, Sorush Saghari, Shiva Parsa Rad, and Marko Tkalcic. Investigating the impact of recommender systems on user-based and item-based popularity bias. *Information Processing & Management*, 58:102655, 2021.

[7] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.

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

[9] Tinglin Huang, Yuxiao Dong, Ming Ding, Zhen Yang, Wenzheng Feng, Xinyu Wang, and Jie Tang. Mixgcf: An improved training method for graph neural network-based recommender systems. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 665–674, 2021.

[10] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.

[11] Michael Jugovac, Dietmar Jannach, and Lukas Lerche. Efficient optimization of multiple recommendation quality factors according to individual user tendencies. *Expert Systems With Applications*, 81:321–331, 2017.

[12] Anastasiia Klimashevskaia. Popularity bias as ethical and technical issue in recommendation: A survey. In *Norsk IKT-konferanse for forskning og utdanning*, number 1, 2022.

[13] Anastasiia Klimashevskaia, Mehdi Elahi, Dietmar Jannach, Christoph Trattner, and Lars Skjærven. Mitigating popularity bias in recommendation: Potential and limits of calibration approaches. In *Advances in Bias and Fairness in Information Retrieval: Third International Workshop, BIAS 2022, Stavanger, Norway, April 10, 2022, Revised Selected Papers*, pages 82–90. Springer, 2022.

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

[15] Dongha Lee, SeongKu Kang, Hyunjun Ju, Chanyoung Park, and Hwanjo Yu. Bootstrapping user and item representations for one-class collaborative filtering. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 317–326, 2021.

[16] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):857–876, 2021.

[17] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[18] Francesco Ricci, Lior Rokach, and Bracha Shapira. Recommender systems: introduction and challenges. *Recommender systems handbook*, pages 1–34, 2015.

[19] Yuta Saito. Asymmetric tri-training for debiasing missing-not-at-random explicit feedback. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 309–318, 2020.

[20] Harald Steck. Calibrated recommendations. In *ACM RecSys '18*, page 154–162, 2018.

[21] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 726–735, 2021.

[22] Tiansheng Yao, Xinyang Yi, Derek Zhiyuan Cheng, Felix Yu, Ting Chen, Aditya Menon, Lichan Hong, Ed H Chi, Steve Tjoa, Jieqi Kang, et al. Self-supervised learning for large-scale item recommendations. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 4321–4330, 2021.

[23] Junliang Yu, Xin Xia, Tong Chen, Lizhen Cui, Nguyen Quoc Viet Hung, and Hongzhi Yin. Xsimgcl: Towards extremely simple graph contrastive learning for recommendation. *arXiv preprint arXiv:2209.02544*, 2022.

[24] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Jundong Li, and Zi Huang. Self-supervised learning for recommender systems: A survey. *arXiv:2203.15876*, 2022.

[25] Junliang Yu, Hongzhi Yin, Xin Xia, Lizhen Cui, and Quoc Viet Hung Nguyen. Graph augmentation-free contrastive learning for recommendation. *arXiv preprint arXiv:2112.08679*, 2021.