UP5: Unbiased Foundation Model for Fairness-aware Recommendation

Wenyue Hua, Yingqiang Ge, Shuyuan Xu, Jianchao Ji, Yongfeng Zhang
Department of Computer Science, Rutgers University, New Brunswick, NJ 08854
wenyue.hua,yingqiang.ge,shuyuan.xu,jianchao.ji,yongfeng.zhang@rutgers.edu

Abstract

Recent advances in Foundation Models such as Large Language Models (LLMs) have propelled them to the forefront of Recommender Systems (RS). Despite their utility, there is a growing concern that LLMs might inadvertently perpetuate societal stereotypes, resulting in unfair recommendations. Since fairness is critical for RS as many users take it for decision-making and demand fulfillment, this paper focuses on user-side fairness for LLM-based recommendation where the users may require a recommender system to be fair on specific sensitive features such as gender or age. In this paper, we dive into the extent of unfairness exhibited by LLM-based recommender models based on both T5 and LLaMA backbones, and discuss appropriate methods for promoting equitable treatment of users in LLM-based recommendation models. We introduce a novel Counterfactually-Fair-Prompt (CFP) method towards Unbiased Foundation mOdels (UFO) for fairness-aware LLM-based recommendation. Experiments are conducted on two real-world datasets, MovieLens-1M and Insurance, and compared with both matching-based and sequential-based fairness-aware recommendation models. Results show that CFP achieves better recommendation performance with a high level of fairness. Source code is anonymously released for reproducibility[1].

1 Introduction

Large Language Model (LLM) has revolutionized the research in NLP (Brown et al., 2020; Bubeck et al., 2023), and its application on Recommender Systems (RS) also attracts soaring interest (Fan et al., 2023; Li et al., 2023a; Chen et al., 2023; Lin et al., 2023; Liu et al., 2023). Recommender Systems (Bobadilla et al., 2013) are algorithms designed to personalize contents or items for individual users based on their preferences. Through

[1]Code and data: https://github.com/agiresearch/UP5

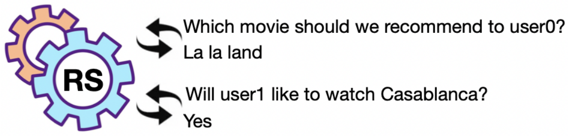

Figure 1: Toy examples of the input-output for prompt-driven LLM-based recommendation models.

personalized natural language prompts (Geng et al., 2022), Large Language Models can serve as a backbone for RS (LLM4RS) to generate personalized recommendations based on user and item information. Figure 1 shows a toy input-output example of prompting LLM-based recommender systems for personalized recommendation.

This paper delves into the fairness of LLM-based recommendation, a significant concern of RS due to its influence on individual decision-making (Li et al., 2023b; Amigó et al., 2023; Ge et al., 2021; Deldjoo et al., 2021; Abdollahpouri et al., 2020; Ekstrand et al., 2019a; Shrestha and Yang, 2019). Specifically, we aim to address user-side counterfactual fairness (Leonhardt et al., 2018; Sonboli et al., 2021; Rahmani et al., 2022; Li et al., 2021; Wu et al., 2021) in RS. We ensure that the RS generates recommendations without factoring in the sensitive attributes that users wish to remain undisclosed. For instance, in a movie recommender system, users may seek recommendations that are not influenced by sensitive attributes such as race, gender, or age. For example, an elderly user may also want to watch younger generation movies to catch up with the times, and thus the user does not want to be discriminated on their age in terms of movie recommendation. As a result, recommender systems should allow users to convey their sensitive preferences and consider these criteria for generating recommendations, rather than solely relying on the recommendation model's determination.

In traditional RS, each user is modeled either as a single embedding (in matching models) (Menon

Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 1899–1912

and Williamson, 2018; Zhang et al., 2017; Liang et al., 2018; Yi et al., 2019; Cheng et al., 2016; Koren et al., 2009) such that whether an item should be recommended is computed by the similarity between item embedding and user embedding, or as a sequence of item embeddings from the user's interaction history (in sequential models) (Hidasi and Karatzoglou, 2018; Kang and McAuley, 2018; Sun et al., 2019; Hidasi et al., 2015; Wu et al., 2017; Yu et al., 2016) such that the model will generate the next item based on the history. However, in the context of LLM-based recommendation, the user's information is not consolidated into a singular user embedding or a sequence of item embeddings, thus rendering traditional methods inapplicable. As a result, this paper explores methods to remove sensitive information from LLM-based recommendation models for fairness-aware recommendation. Since LLM-based recommendation models contain a large number of parameters storing a rich amount of knowledge for both language understanding and personalized recommendation, to remove unfairness from such models, three challenges need to be addressed: 1) efficient training and inference of the attribute-specific fairness-aware models for each sensitive attribute and their combinations, 2) avoiding training separate models for each combination of sensitive attributes due to a potentially exponential growth in attribute combinations, and 3) minimizing performance decrease on recommendations, as user attributes could be important for the recommendation performance.

In this work, we first explore three methods to probe the unfairness of LLM-based recommendation. Then, we present the Counterfactually-Fair-Prompt (CFP) method to mitigate the user-side unfairness and propose a fairness-aware foundation model, wherein sensitive user attributes, such as gender, age, occupation, etc., can be either removed or preserved based on each user's preference. We experiment on two datasets which contain sensitive attributes, *MovieLens-1M* and *Insurance*, for fairness research, showing the effectiveness of our model in eliminating unfairness while maintaining a high level of recommendation performance.

The paper proceeds as follows: Section 2 presents an overview of the related work on fairness in LLM and RS; Section 3 briefly introduces the preliminary of LLM-based recommendation and its fairness motivation; Section 4 introduces the proposed CFP model. Section 5 presents the experimental results for both single-attribute fairness and combined-attribute fairness. Section 6 provides ablation studies and hyperparameter sensitivity analysis. Section 7 concludes the paper.

2 Related Work

Fairness of Recommender Systems. Since recommender systems involve various stakeholders such as users, item providers, and the platform itself, fairness is a multi-sided concept in recommender systems (Li et al., 2023b; Wang et al., 2023; Ekstrand et al., 2019b). For user-side fairness, especially counterfactual fairness, it is usually defined as whether recommendations for a user are made independently of the user's sensitive attributes, which is measured by determining whether the recommendation outcomes for a given user are equivalent in both the factual and counterfactual scenarios with respect to a specific attribute (Ge et al., 2022; Dong et al., 2020; Li et al., 2021). In the context of RS, a counterfactual world is an alternate scenario in which the user's sensitive attributes are manipulated while all other attributes independent of the sensitive attributes are held constant, as defined in the following (Li et al., 2021):

Definition 2.1 (Counterfactually fair recommendation) *An RS is counterfactually fair iff. for any possible user u with features $X = x$ and $K = k$, where K are the user's sensitive attributes and X are the attributes that are causally independent of K,*

$$P(L_k|X = x, K = k) = P(L_{k'}|X = x, K = k) \quad (1)$$

holds for all L and any value k attainable by K, where L is the recommendation list for user u.

A sufficient condition for RS to be counterfactually fair is to remove the user's sensitive information when generating recommendations so that the recommendation outcome remains unchanged across various counterfactual scenarios (Li et al., 2021; Wu et al., 2022), which is ultimately similar to the fairness of language models except that we focus on user representations other than attribute-related words. Li et al. and Wu et al. explored personalized counterfactual fairness for traditional RS, where (Li et al., 2021) is developed for matching-based RS while (Wu et al., 2022) is for sequential-based RS. However, counterfactual fairness for LLM-based RS has largely been unexplored, which has unique challenges to solve as we mentioned before. Furthermore, existing methods are not directly applicable to LLM-based recommendation.

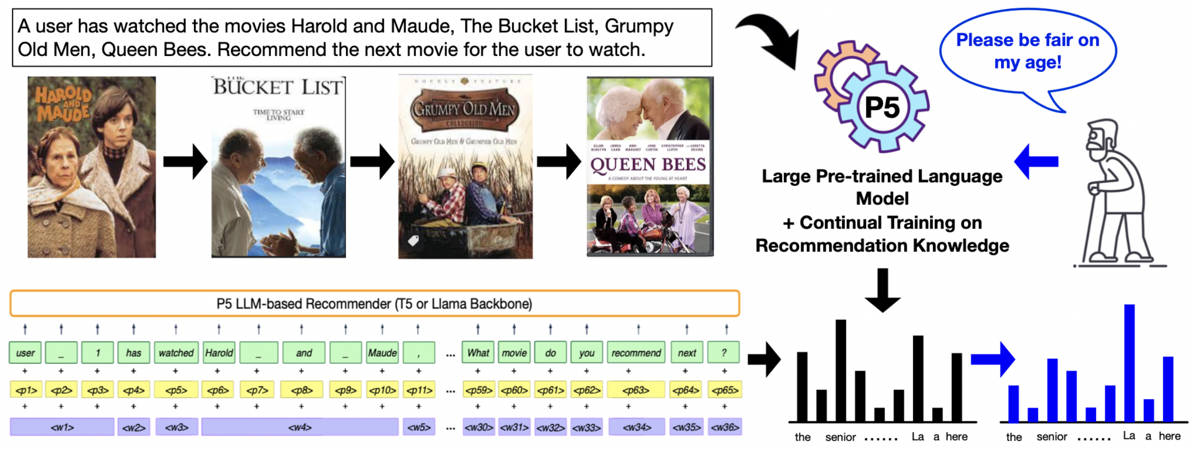

Figure 2: Counterfactual fairness of LLM-based recommendation given the user's choice of sensitive attribute.

For example, Li et al. requires updating all parameters in the model for each feature, which is not parameter-efficient and thus unsuitable for large language models. Wu et al. appends a prefix prompt and an adapter to the model for improving fairness on sequential recommendation. However, for each attribute combination, a new prefix prompt and a new adapter must be trained from scratch, and thus the method cannot properly handle the exponential combination of attributes. As a result, developing fairness-aware methods for LLM-based recommendation is highly needed.

Fairness of Large Language Models. Fairness of language models is usually concerned with whether embeddings for attribute-related words such as gender-related words are associated with stereotypes (Ravfogel et al., 2020). Recent studies have highlighted the potential of unfairness in the pre-training data of LLMs, which leads to the generation of harmful or offensive content, including discrimination against marginalized groups. Consequently, there has been an increased research focus on addressing the harmfulness issues of LLMs, with a particular emphasis on unfairness. In a study conducted by Zhuo et al., the fairness of LLMs was examined using two datasets specifically designed to assess bias in the context of general question answering and text generation tasks. Another research effort by Sun et al. evaluated the safety of Chinese LLMs, including an examination of fairness. The study involved observing the frequency of harmful information present in the responses generated by LLMs. This approach provided insights into the potential unfairness and its impact on the safety of these models. (Zhang et al., 2023) and (Li and Zhang, 2023) tested the fairness of ChatGPT on recommendation, education, medical and legal tasks, though they did not provide solutions for the unfairness problems. There also exist several benchmark datasets that are used to better evaluate the unfairness and other harmfulness of LLMs, such as RedTeamingData (Ganguli et al., 2022) and HELM (Liang et al., 2022). While there have been numerous investigations into the fairness of LLMs within the field of NLP, there is currently a gap of research in terms of addressing the fairness problems of LLM-based recommender systems.

3 Preliminary of LLM-based Recommendation

Foundation Models such as Large Language Lodels (LLMs), e.g., BERT (Devlin et al., 2018), Llama (Touvron et al., 2023), T5 (Raffel et al., 2020), and GPT-3 (Brown et al., 2020), have been shown to effectively learn rich semantics from web-scale data and transfer knowledge in pre-training data to various downstream NLP tasks. For recommender systems, P5 (Geng et al., 2022; Xu et al., 2023) stands as a seminal framework for foundational recommendation models, grounded in the architecture of LLM backbone models, including both encoder-decoder configuration T5 (Raffel et al., 2020) and decoder-only model Llama (Touvron et al., 2023). By integrating various recommendation tasks—ranging from item generation, recommendation explanation, to rating prediction—P5 enhances the adaptability of contemporary recommendation methodologies.

In our research, we employ both T5 (Raffel et al., 2020) and OpenLlama (Geng and Liu, 2023) backbones within the P5 framework to execute experiments targeting unfairness mitigation. In this particular section, we train P5 and probe its fairness problem to motivate the fairness research for LLM-based recommendation. More specifically, we train P5 on two tasks: direct recommendation and sequential recommendation. Direct recommendation generates recommendations without any user-item interaction history in the input prompt, while sequential recommendation explicitly involves user-item interaction histories. We use the simple and effective sequential ID indexing method for both tasks (Hua et al., 2023). The prompt for each task is presented in the following square box.

Direct Recommendation
Input: Which movie user_{{user_ID}} would like to watch among the following candidates? {{List of 100 candidate movies}}. **Output**: {{movie_ID}}
Sequential Recommendation
Input: User_{{user_ID}} has already watched the following movies {{the sequence of movie IDs this user watched}}. Which movie user_{{user_ID}} would like to watch next?
Output: {{movie_ID}}

Motivating Fairness Concerns. As presented above, P5 does not explicitly involve any sensitive-attribute-related textual description for users. However, it can still implicitly infer user sensitive attributes and possibly use it for recommendation, even though users may not want to include such sensitive attributes when generating recommendations for them. We use three methods for probing the user attributes from the LLM: 1) eliciting the attributes through in-context learning based on manually designed prompts, 2) generating attributes by tuning soft probing prompts, and 3) training a classifier on the embeddings corresponding to the user tokens in the input.

Figure 3 presents the AUC score of predicting the sensitive attributes (gender, age, occupation, and marital_status) from the LLM on Movielens and Insurance datasets, while more details of the implementation and results are presented in the Appendix A. Experimental results show that both the soft prompt tuning and the classification methods can detect user-sensitive attributes from the LLM, though manual prompts fail. The classification and soft prompt tuning methods both generate above-random predictions on user attributes. This result implies that even though the training and tuning process of LLM-based recommendation does not directly involve users' sensitive attributes, such sensitive information is still inferred by the LLM and embeded in the LLM parameters for generating recommendations, though users may not want

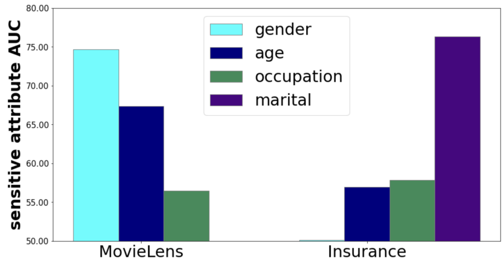

Figure 3: Inferring sensitive attribute information from LLM-based recommendation model.

their recommendations to be influenced by certain sensitive attributes. As a result, it is important to develop sensitive mitigation methods so as to enable counterfactually fair LLM-based recommendation, which we will introduce in the following sections.

4 Counterfactually-Fair Prompting

We propose a Counterfactually-Fair-Prompt (CFP) method to mitigate the unfairness of LLM-based recommendation, resulting in the development of a fair and accurate recommendation foundation model. Our approach is 1) personalized, since each user can choose the attributes that they wish to be treated fairly on, and 2) space and time efficient, since our approach does not require retraining the entire foundation model and only needs to train the prefix prompts. The key idea of the CFP method is to train a counterfactually-fair prompt (CFP): For encoder-decoder LLM, we need an encoder prompt p_{enc} to remove sensitive attributes and a decoder prompt p_{dec} to preserve the model performance; For decoder-only LLM, we only need a decoder prompt. Our goal is to learn such CFP so that sensitive information in the user token embeddings is removed by simply concatenating the CFP with the original input prompt.

CFPs are trained by adversarial learning (Lowd and Meek, 2005; Chakraborty et al., 2018; Zhao et al., 2022). Adversarial learning requires a discriminator module (Wang and Yu, 2019) aiming at precise extraction of attribute values from embeddings, while CFP aims at obfuscation of the discriminator's efforts. Thus, the stronger the discriminator, the more effectively we can clean sensitive information from embeddings. According to the probing experiments in Section 3, the multi-class classifier is a stronger prober than other approaches. Thus, we utilize the classifier as the discriminator in adversarial learning. Figure 4 shows the model architecture. We also present the results of using the soft probing prompt as a discriminator in Section 6 for comprehensiveness.

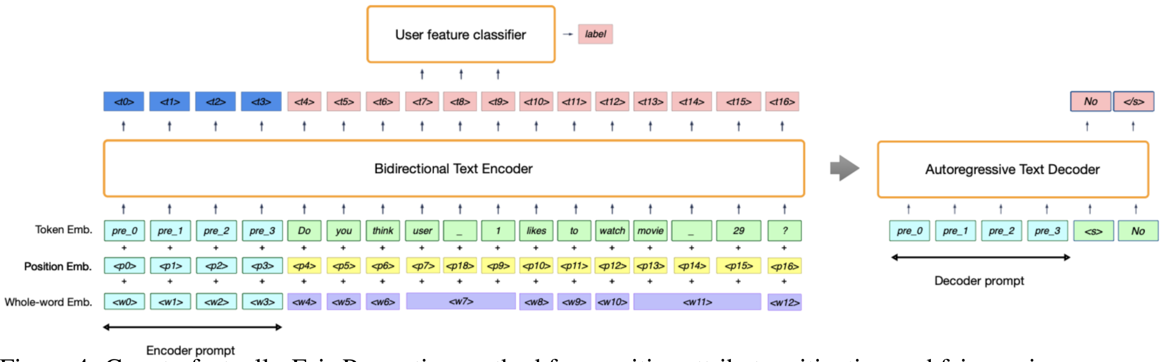

Figure 4: Counterfactually-Fair-Prompting method for sensitive attribute mitigation and fairness improvement

The model training involves an iterative process where the CFP and the classifier are optimized in succession. For each attribute k, we denote the recommendation loss as L^k_{rec} and the discriminator loss as L^k_{dis}. Let $\mathcal{M}$ denote the recommendation foundation model and $\mathcal{C}_k$ as the classifier. L^k_{rec} is a negative log-likelihood loss that encourages generating the correct item y:

$$L^k_{rec} = -\sum\nolimits_{j=1}^{|y|} \log P(y_j | p_{enc_k} \circ x, p_{dec_k} \circ y_{0:j-1}, \mathcal{M}) \tag{2}$$

L^k_{dis} is a Cross-Entropy Loss (CEL) that encourages predicting user attribute k correctly based on the average of user-relevant token embeddings $\mathcal{E}$ (e.g., the tokens "user", "_", and "1" in Figure 4) conditioned on p_{enc_k}. Denoting u as the user, and c_u the correct attribute value for the user, L_{dis} is:

$$L^k_{dis} = \text{CEL}(c_u, \mathcal{C}_k(\text{mean}(\mathcal{E}_u))) \tag{3}$$

The adversarial loss L_k for each attribute k is defined as below, where λ_k denotes the discriminator weight for attribute k:

$$L_k = \sum\nolimits_u L^k_{rec} - \lambda_k \cdot L^k_{dis} \tag{4}$$

The training algorithm is presented in Appendix C.

4.1 Prompt Mixture

Users may seek recommendations that remain impartial to several attributes at the same time. For instance, they may want a model to overlook details like gender and marital status but still value recommendations that resonate with movie preferences typical for their age group. Consequently, CFPs must possess the capacity to exclude several attributes in tandem. An elementary approach might involve developing a prompt for every possible attribute combination, but this is operationally taxing given the exponential growth in the number of combinations.

To solve the challenge, we propose a Prompt Mixture (PM) module. This module comprises a singular attention layer that combines the embeddings from various single-attribute CFPs to integrate user preferences. The attentional framework offers flexibility regarding input length, allowing for the integration of a variable number of CFPs, each potentially of distinct lengths. The PM is adept at processing information from different CFPs, masking sensitive user information while preserving other relevant details within the model-generated hidden states. This positions the PM as an invaluable instrument for a user-controllable LLM-based recommendation model since users have the freedom to choose different sensitive feature combinations, facilitating the assimilation of multifaceted user stipulations without the necessity for specialized model training for each unique combination of requirements (Figure 5).

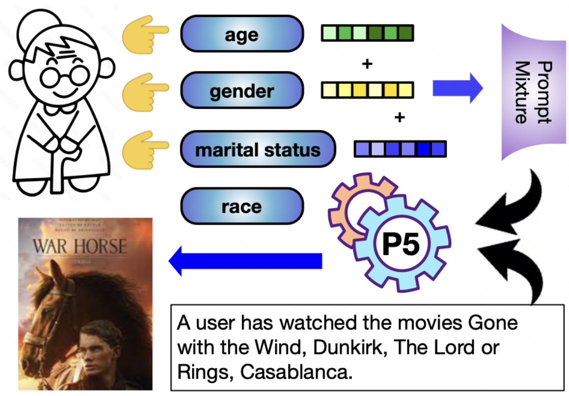

Figure 5: Prompt Mixture over CFPs from 3 attributes

Similar to single attribute prompt learning introduced above, PM is also trained based on adversarial learning, where each optimization step includes a random combination of sensitive attributes selected to be removed. PM takes a concatenation of multiple single-attribute prefix prompts as input and generates a new prompt, which is optimized to simultaneously decrease the recommendation loss and increase the sum of discriminator loss of multiple classifiers. The loss function for one step with a set of randomly selected attributes $\mathbf{K}$ is:

$$L_{\mathbf{K}} = \sum_u (L_{rec}^{\mathbf{K}} - \Sigma_{k \in \mathbf{K}} \lambda_k \cdot L_{dis}^{k}) \qquad (5)$$

5 Experiments

This section presents the experimental results of CFP on a variety of metrics, including recommendation performance and fairness level. The results show the model's ability to achieve fairness in both single-attribute and multi-attribute scenarios.

5.1 Experimental Setup

Datasets Experiments are conducted on the MovieLens-1M dataset and Insurance dataset:
MovieLens-1M(Harper and Konstan, 2015): The dataset contains user-movie interactions and user profile information: gender, age, and occupation. Gender is a binary feature, occupation is a twenty-one-class feature, and age is a seven-class feature.
Insurance[2]: The dataset contains user-insurance interactions. The user profile contains four features: gender, marital status, age, and occupation. Gender is a binary feature, marital status is a seven-class feature, occupation is a six-class feature, and age is a five class feature.

Evaluation Metrics To evaluate direct recommendation and sequential recommendation tasks, one correct item is predicted among 100 randomly selected negative samples for both tasks. The metrics are Hit@k for k in $\{1, 3, 10\}$. We adopt the commonly used leave-one-out strategy (for each user, treat the second-to-last interacted item to be the validation item and the last interacted item to be the test item) to create the training, validation, and test datasets. We adopt AUC for user attribute classification to evaluate whether sensitive attributes are involved in recommendations.

LLM Backbone We train the LLM recommendation model under the P5 paradigm (Geng et al., 2022) using both T5-Base (Raffel et al., 2020) and OpenLlama-3B (Geng and Liu, 2023) backbones. We present results based on T5 in this section as the main results for comparison, and detailed results for the OpenLlama experiments are presented in the Appendix B.

[2]https://www.kaggle.com/datasets/mrmorj/insurance-recommendation

Dataset	MovieLens			Insurance		
Model	PMF	SimpleX	P5	PMF	SimpleX	P5
↑ Hit@1	19.91	17.94	**20.57**	70.20	76.50	**82.53**
↑ Hit@3	38.66	**38.79**	38.38	75.23	80.12	**92.68**
↑ Hit@10	65.69	65.69	**67.31**	90.04	91.41	**98.89**
↓ AUC (G)	80.22	75.52	**74.71**	52.04	53.34	**50.11**
↓ AUC (A)	82.37	79.39	**67.40**	57.94	56.87	**50.09**
↓ AUC (O)	61.32	59.40	**56.50**	58.25	57.12	**53.28**
↓ AUC (M)	–	–	–	71.30	**68.85**	69.25

Table 1: Results of matching-based recommendation, G means Gender, A means Age, O means Occupation, and M means Marital Status (%).

Dataset	MovieLens			Insurance		
Model	SAS	BERT	P5	SAS	BERT	P5
↑ Hit@1	28.39	29.30	**30.34**	77.26	81.20	**84.56**
↑ Hit@3	**53.89**	49.06	49.26	85.15	93.33	**93.99**
↑ Hit@10	**76.32**	70.06	67.40	95.76	98.78	**98.98**
↓ AUC (G)	91.90	78.52	**74.71**	73.23	61.20	**50.13**
↓ AUC (A)	92.06	73.35	**67.40**	57.93	**54.34**	56.92
↓ AUC (O)	76.57	64.79	**56.50**	88.04	**54.30**	57.87
↓ AUC (M)	–	–	–	76.61	**76.11**	76.37

Table 2: Results of sequential recommendation, G is Gender, A is Age, O is Occupation, and M is Marital Status (%). SAS is SASRec and BERT is Bert4Rec.

Baselines We adopt four SOTA fairness-aware models as baselines: Li et al.'s Counterfactual-filter method over PMF (C-PMF) and SimpleX (C-SX), and Wu et al.'s Selective-prompt-adapter method on SASRec (S-SAS) and BERT4Rec (S-B4). PMF (Mnih and Salakhutdinov, 2007; Menon and Williamson, 2018) is the Probabilistic Matrix Factorization model that adds Gaussian prior into the user and item latent factor distributions for matrix factorization. SimpleX (Mao et al., 2021) is a contrastive learning model based on cosine contrastive loss which has achieved state-of-the-art performance on recommendation performance. Li et al.'s unfairness-removing filters are applied right after the user embedding computed by PMF and SimpleX, which creates C-PMF and C-SX. SASRec (Kang and McAuley, 2018) is a sequential recommendation model based on left-to-right self-attention mechanism. BERT4Rec (Sun et al., 2019) is a bidirectional sequential recommendation model based on BERT. Wu et al.'s prompts are appended to item sequences and adaptors are inserted into each Transformer encoder block in SASRec and BERT4Rec, which creates S-SAS and S-BERT.

Implementation Details The model hyper-parameters are selected within the following range: discriminator weight $\lambda \in \{1, 5, 10, 100\}$, prefix length $\in \{5, 15, 30\}$, batch size = 16, number of steps $T \in \{10, 20\}$ to update $\mathcal{C}$ on L_{dis} or prefix prompt $\mathcal{P}$ on L_{rec}, number of batches $R \in \{20\}$ to update prefix prompt $\mathcal{P}$ on adversarial loss L.

Dataset	MovieLens									Insurance								
Attribute	Gender			Age			Occupation			Age			Marital			Occupation		
Model	C-PMF	C-SX	CFP	C-PMF	C-SX	CFP	C-PMF	C-SX	CFP	C-PMF	C-SX	CFP	C-PMF	C-SX	CFP	C-PMF	C-SX	CFP
↑ Hit@1	**16.73**	13.96	16.38	17.42	13.87	**21.22**	15.60	14.06	**21.00**	67.61	71.14	**82.53**	66.68	71.50	**81.03**	68.51	71.09	**82.53**
↑ Hit@3	34.03	29.56	**35.04**	34.20	29.61	**39.22**	34.36	29.56	**38.50**	73.25	83.23	**92.68**	74.23	83.00	**90.58**	74.09	82.23	**92.68**
↑ Hit@10	65.32	56.02	**65.82**	65.18	55.42	**67.30**	65.33	56.02	**69.49**	85.98	92.65	**98.89**	85.99	96.50	**97.66**	85.95	93.27	**98.89**
↓ AUC	56.62	70.80	**54.19**	62.55	79.26	**52.91**	56.01	57.02	**50.00**	50.81	51.26	**50.09**	**52.10**	56.23	52.19	54.40	**52.09**	53.28

Table 3: Results of single-attribute fairness-aware prompting on matching-based models (%)

Dataset	MovieLens									Insurance								
Attribute	Gender			Age			Occupation			Age			Marital			Occupation		
Model	S-SAS	S-B4	CFP	S-SAS	S-B4	CFP	S-SAS	S-B4	CFP	S-SAS	S-B4	CFP	S-SAS	S-B4	CFP	S-SAS	S-B4	CFP
↑ Hit@1	20.87	23.48	**26.82**	22.95	27.98	**31.23**	18.90	24.33	**31.66**	69.40	81.20	**82.08**	70.10	75.33	**80.63**	70.09	81.20	**82.62**
↑ Hit@3	41.64	42.09	**45.18**	44.10	49.32	**51.18**	20.84	43.29	**50.73**	80.05	**93.33**	92.62	80.38	84.54	**90.16**	80.38	**93.33**	92.65
↑ Hit@10	60.82	62.43	**64.38**	66.00	69.38	**67.70**	43.87	59.74	**67.45**	88.34	**98.78**	98.37	88.49	94.34	**98.38**	88.91	**98.78**	98.54
↓ AUC	59.72	58.33	**54.19**	60.20	67.33	**52.91**	67.27	60.36	**50.00**	57.48	53.34	**51.23**	66.51	69.11	**50.03**	86.66	54.30	**50.82**

Table 4: Results of single-attribute fairness-aware prompting on sequential models (%)

5.2 Overall Results of the CFP Model

This subsection presents the overall results.

Overall Performance Table 1 and Table 2 present the recommendation performance and unfairness of the baseline models for direct recommendation and sequential recommendation respectively. The first 3 rows on each table are the recommendation performance and the last 4 rows show the extent of unfairness. From the result, we see that LLM-based recommendation model (P5) performs better than other models on both datasets.

Single-Attribute Scenario We compare the CFP model with fair matching-based models C-PMF and C-SX in Table 3 and fair sequential-based models S-SASRec and S-BERT4Rec in Table 4, since both frameworks provide solutions in single-attribute scenarios. CFP outperforms both fair matching-based and sequential-based models in terms of both AUC and recommendation accuracy. The AUC of CFP is close to 50%, indicating a high level of fairness since the model is unable to inferr users' sensitive attributes, and the negative impact on recommendation performance is minimal compared to other models.

Multi-Attribute Scenario We also provide experiment results on multi-attribute fairness treatment, as shown in Table 5 and Table 6. The attribute row denotes the set of attributes to be removed, where "G" represents "gender," "A" represents "age," "O" represent "occupation," and "M" represents "marital status". Two or more attributes together such as "GA" means that the sensitive attributes need to be removed at the same time. We compare our CFP model with the two matching-based fairness baselines C-PMF and C-SX from Li et al., since the sequential fairness baselines from Wu et al. are unable to handle mutiple attributes. We report the recommendation performance and the average AUC for the targeted user attributes in Table 5 (MovieLens) and Table 6 (Insurance). We can see that our CFP method under prompt mixture is an effective method to combine the single-attribute prefix prompts, achieving fairness and meanwhile maintaining high recommendation performance.

6 Detailed Analysis

This section discusses the effect of different model designs of the CFP method. We experiment on 1) how hyperparameters such as prompt length and discriminator weights affect the performance, and 2) how the choice of discriminator (classifier or soft probing prompt) affects the performance.

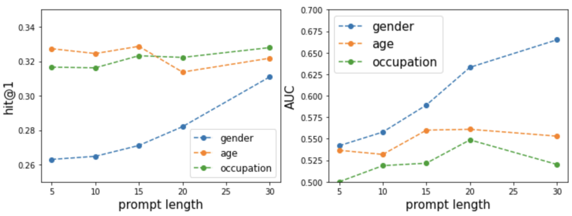

Figure 6: Different prompt length on MovieLens

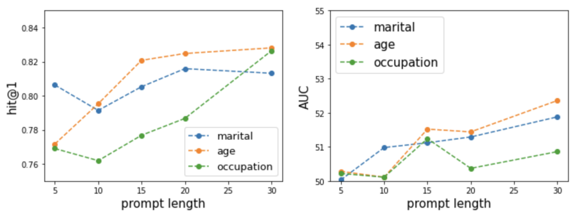

Figure 7: Different prompt length on Insurance

Hyperparameter Sensitivity In this section, we study the effect of prompt length (5, 10, 15, 30) and discriminator weight (0.1, 1, 10, and 100) on both recommendation performance (Hit@1 on sequential recommendation) and attribute detection performance (AUC). Figure 6 and 7 present the effects

Model	GA			GO			AO			GAO		
Attribute	C-PMF	C-SX	CFP	C-PMF	C-SX	CFP	C-PMF	C-SX	CFP	C-PMF	C-SX	CFP
↑ Hit@1	14.93	15.61	**16.33**	15.25	15.53	**18.67**	14.84	15.43	**21.37**	15.09	15.67	**20.18**
↑ Hit@3	32.11	31.79	**37.48**	32.70	31.84	**39.02**	31.83	31.87	**39.83**	32.58	31.85	**38.79**
↑ Hit@10	60.51	58.82	**66.89**	60.58	58.78	**66.39**	59.51	58.71	**68.40**	60.75	58.87	**66.78**
↓ Avg. AUC	58.03	70.25	**54.22**	56.57	60.90	**52.10**	56.57	64.41	**50.00**	56.54	65.19	**53.21**

Table 5: Results of multi-attribute fairness-aware prompting on MovieLens dataset (%)

Model	AO			AM			MO			AMO		
Attribute	C-PMF	C-SX	CFP	C-PMF	C-SX	CFP	C-PMF	C-SX	CFP	C-PMF	C-SX	CFP
↑ Hit@1	63.68	71.58	**79.00**	62.27	71.23	**80.91**	62.44	71.11	**78.30**	64.38	72.30	**81.63**
↑ Hit@3	70.55	80.50	**89.22**	69.78	79.18	**90.97**	69.39	81.22	**88.45**	70.11	81.78	**91.52**
↑ Hit@10	84.88	93.61	**97.66**	83.85	93.22	**98.73**	84.88	93.52	**97.33**	85.90	93.35	**97.37**
↓ Avg. AUC	58.38	55.98	**50.80**	55.60	59.97	**50.79**	57.86	59.79	**50.64**	57.44	58.43	**50.74**

Table 6: Results of multi-attribute fairness-aware prompting on Insurance dataset (%)

of prefix prompt length on MovieLens and Insurance, respectively. In general, longer prefix length hurts fairness but improves the recommendation performance. Figure 8 and 9 present the results under different discriminator weight λ, showing that larger weights bring better fairness but hurt the recommendation performance since the fairness term dominates the loss. Results indicate that we need to choose the prompt length and discriminator weight carefully to balance the fairness-recommendation trade-off.

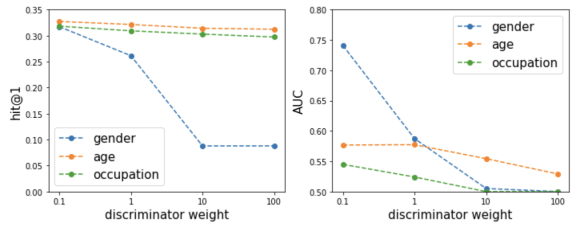

Figure 8: Different discriminator weight on MovieLens

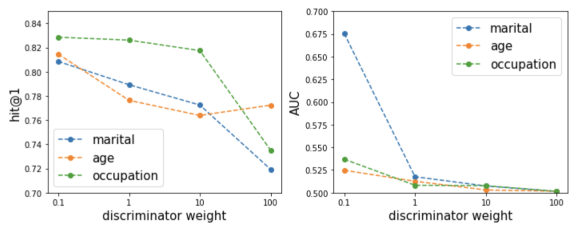

Figure 9: Different discriminator weight on Insurance

Soft Probing Prompt as Discriminator This section discusses whether we can use soft probing prompt as the discriminator in adversarial training to improve fairness. According to the motivating experiments on probing fairness of LLMs (Section 3), soft probing prompt is a weaker tool to extract user attribute information compared with multi-class classier. To further validate this, we train the CFP using soft probing prompt as the discriminator. To test the effectiveness of the trained prompts, we append the trained CFP in front of the model inputs and then use 1) soft probing prompt and 2) multi-class classifier to extract user attribute information. We present the results on the Insurance dataset targeting the marital status attribute under different lengths of the CFP in Figure 10, and other dataset and attributes have similar observations. We see that 1) the probing prompts cannot extract any user attribute since its AUC is close to 50%, while the classifier can still extract non-trivial sensitive attribute information from the LLM. 2) longer CFPs are more effective in removing sensitive attributes, since the classifier can extract less information, while AUCs for probing prompts are always around 50%. As a result, this result shows that to train CFPs, it is better to use the classifier instead of soft probing prompt as the discriminator.

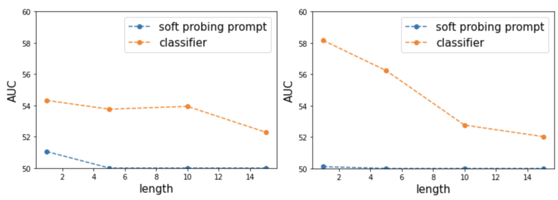

Figure 10: Effect of different lengths on AUC using soft probing prompt and classifier for probing

7 Conclusion and Future Work

This paper explores the unfairness issue of LLM for recommendation by first probing the unfairness issue of LLM-based recommendation models, and then proposing a novel CFP method to mitigate the issue, enabling a fair recommendation foundation model. In the future, we will explore fairness in other aspects of LLM-based recommendation, such as explanation generation and conversational recommendation. We are also committed to developing user-friendly interfaces and algorithms that are responsive to user specifications for user controllable fairness without compromising the system's performance or user experience.

Limitation

The paper investigates unfairness issues in large language models for recommender systems. However, the paper still has several limitations. In particular, though we explored fairness of LLM-based recommendation over several sensitive features such as gender, age, and occupation, we did not study the bias problems with regard to historically disadvantaged groups. The reason is because we are not aware of the availability of any dataset containing such sensitive feature information. In the future, when such dataset becomes available, we plan to extend our exploration on the fairness of LLM-based recommendation over such features.

Ethical Consideration

Our method is proposed to increase the fairness of recommendation performance for users. It will unlikely lead to negative societal impacts.

References

Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. 2020. The connection between popularity bias, calibration, and fairness in recommendation. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 726–731.

Enrique Amigó, Yashar Deldjoo, Stefano Mizzaro, and Alejandro Bellogín. 2023. A unifying and general account of fairness measurement in recommender systems. *Information Processing & Management*, 60(1):103115.

Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. 2013. Recommender systems survey. *Knowledge-based systems*, 46:109–132.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. 2018. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*.

Jin Chen, Zheng Liu, Xu Huang, Chenwang Wu, Qi Liu, Gangwei Jiang, Yuanhao Pu, Yuxuan Lei, Xiaolong Chen, Xingmei Wang, et al. 2023. When large language models meet personalization: Perspectives of challenges and opportunities. *arXiv:2307.16376*.

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

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. *International Conference on Learning Representations (ICLR) 2021*.

Yashar Deldjoo, Vito Walter Anelli, Hamed Zamani, Alejandro Bellogin, and Tommaso Di Noia. 2021. A flexible framework for evaluating user and item fairness in recommender systems. *User Modeling and User-Adapted Interaction*, pages 1–55.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Zhenhua Dong, Hong Zhu, Pengxiang Cheng, Xinhua Feng, Guohao Cai, Xiuqiang He, Jun Xu, and Jirong Wen. 2020. Counterfactual learning for recommender system. In *Fourteenth ACM Conference on Recommender Systems*, pages 568–569.

Michael D Ekstrand, Robin Burke, and Fernando Diaz. 2019a. Fairness and discrimination in recommendation and retrieval. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 576–577.

Michael D Ekstrand, Robin Burke, and Fernando Diaz. 2019b. Fairness and discrimination in retrieval and recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1403–1404.

Wenqi Fan, Zihuai Zhao, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Jiliang Tang, and Qing Li. 2023. Recommender systems in the era of large language models (llms). *arXiv:2307.02046*.

Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. 2022. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*.

Yingqiang Ge, Shuchang Liu, Ruoyuan Gao, Yikun Xian, Yunqi Li, Xiangyu Zhao, Changhua Pei, Fei Sun, Junfeng Ge, Wenwu Ou, et al. 2021. Towards long-term fairness in recommendation. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 445–453.

Yingqiang Ge, Juntao Tan, Yan Zhu, Yinglong Xia, Jiebo Luo, Shuchang Liu, Zuohui Fu, Shijie Geng, Zelong Li, and Yongfeng Zhang. 2022. Explainable fairness in recommendation. *arXiv preprint arXiv:2204.11159*.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the Sixteenth ACM Conference on Recommender Systems*.

Xinyang Geng and Hao Liu. 2023. Openllama: An open reproduction of llama.

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

Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 843–852.

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

Wenyue Hua, Shuyuan Xu, Yingqiang Ge, and Yongfeng Zhang. 2023. How to Index Item IDs for Recommendation Foundation Models. In *Proceedings of the 1st International ACM SIGIR Conference on Information Retrieval in the Asia Pacific (SIGIR-AP)*.

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

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

Jurek Leonhardt, Avishek Anand, and Megha Khosla. 2018. User fairness in recommender systems. In *Companion Proceedings of the The Web Conference 2018*, pages 101–102.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *EMNLP*.

Lei Li, Yongfeng Zhang, Dugang Liu, and Li Chen. 2023a. Large language models for generative recommendation: A survey and visionary discussions. *arXiv:2309.01157*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Yunqi Li, Hanxiong Chen, Shuyuan Xu, Yingqiang Ge, Juntao Tan, Shuchang Liu, and Yongfeng Zhang. 2023b. Fairness in recommendation: Foundations, methods, and applications. *ACM Transactions on Intelligent Systems and Technology*, 14(5):1–48.

Yunqi Li, Hanxiong Chen, Shuyuan Xu, Yingqiang Ge, and Yongfeng Zhang. 2021. Towards personalized fairness based on causal notion. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1054–1063.

Yunqi Li and Yongfeng Zhang. 2023. Fairness of chatgpt. *arXiv:2305.18569*.

Nan Liang, Hai-Tao Zheng, Jin-Yuan Chen, Arun Kumar Sangaiah, and Cong-Zhi Zhao. 2018. Trsdl: Tag-aware recommender system based on deep learning–intelligent computing systems. *Applied Sciences*, 8(5):799.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Xiangyang Li, Chenxu Zhu, Huifeng Guo, Yong Yu, Ruiming Tang, et al. 2023. How can recommender systems benefit from large language models: A survey. *arXiv:2306.05817*.

Peng Liu, Lemei Zhang, and Jon Atle Gulla. 2023. Pre-train, prompt and recommendation: A comprehensive survey of language modelling paradigm adaptations in recommender systems. *arXiv:2302.03735*.

Daniel Lowd and Christopher Meek. 2005. Adversarial learning. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 641–647.

Kelong Mao, Jieming Zhu, Jinpeng Wang, Quanyu Dai, Zhenhua Dong, Xi Xiao, and Xiuqiang He. 2021. Simplex: A simple and strong baseline for collaborative filtering. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 1243–1252.

Aditya Krishna Menon and Robert C Williamson. 2018. The cost of fairness in binary classification. In *Conference on Fairness, Accountability and Transparency*, pages 107–118. PMLR.

Andriy Mnih and Russ R Salakhutdinov. 2007. Probabilistic matrix factorization. *Advances in neural information processing systems*, 20.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Hossein A Rahmani, Mohammadmehdi Naghiaei, Mahdi Dehghan, and Mohammad Aliannejadi. 2022. Experiments on generalizability of user-oriented fairness in recommender systems. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2755–2764.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. Null it out: Guarding protected attributes by iterative nullspace projection. *arXiv preprint arXiv:2004.07667*.

Yash Raj Shrestha and Yongjie Yang. 2019. Fairness in algorithmic decision-making: Applications in multi-winner voting, machine learning, and recommender systems. *Algorithms*, 12(9):199.

Nasim Sonboli, Jessie J Smith, Florencia Cabral Berenfus, Robin Burke, and Casey Fiesler. 2021. Fairness and transparency in recommendation: The users' perspective. In *Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization*, pages 274–279.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450.

Hao Sun, Zhexin Zhang, Jiawen Deng, Jiale Cheng, and Minlie Huang. 2023. Safety assessment of chinese large language models. *arXiv preprint arXiv:2304.10436*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Huaxia Wang and Chun-Nam Yu. 2019. A direct approach to robust deep learning using adversarial networks. *arXiv preprint arXiv:1905.09591*.

Yifan Wang, Weizhi Ma, Min Zhang, Yiqun Liu, and Shaoping Ma. 2023. A survey on the fairness of recommender systems. *ACM Transactions on Information Systems*, 41(3):1–43.

Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. 2017. Recurrent recommender networks. In *Proceedings of the tenth ACM international conference on web search and data mining*, pages 495–503.

Yao Wu, Jian Cao, Guandong Xu, and Yudong Tan. 2021. Tfrom: A two-sided fairness-aware recommendation model for both customers and providers. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1013–1022.

Yiqing Wu, Ruobing Xie, Yongchun Zhu, Fuzhen Zhuang, Ao Xiang, Xu Zhang, Leyu Lin, and Qing He. 2022. Selective fairness in recommendation via prompts. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2657–2662.

Shuyuan Xu, Wenyue Hua, and Yongfeng Zhang. 2023. Openp5: Benchmarking foundation models for recommendation. *arXiv preprint arXiv:2306.11134*.

Baolin Yi, Xiaoxuan Shen, Hai Liu, Zhaoli Zhang, Wei Zhang, Sannyuya Liu, and Naixue Xiong. 2019. Deep matrix factorization with implicit feedback embedding for recommendation system. *IEEE Transactions on Industrial Informatics*, 15(8):4591–4601.

Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A dynamic recurrent model for next basket recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 729–732.

Jizhi Zhang, Keqin Bao, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Is chatgpt fair for recommendation? evaluating fairness in large language model recommendation. *RecSys*.

Yongfeng Zhang, Qingyao Ai, Xu Chen, and W Bruce Croft. 2017. Joint representation learning for top-n recommendation with heterogeneous information sources. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1449–1458.

Weimin Zhao, Sanaa Alwidian, and Qusay H Mahmoud. 2022. Adversarial training methods for deep learning: A systematic review. *Algorithms*, 15(8):283.

Terry Yue Zhuo, Yujin Huang, Chunyang Chen, and Zhenchang Xing. 2023. Exploring ai ethics of chatgpt: A diagnostic analysis. *arXiv preprint arXiv:2301.12867*.

APPENDIX

A Probing Unfairness in LLM-based RS

Probing the user attributes out of LLM is a non-trivial task in LLM-based RS because each user does not have one specific user embedding. In this section, we illustrate three methods to detect unfairness of LLM-based RS. The results show that even if the training data does not explicitly use user-sensitive attributes, LLM-based RS still implicitly infers user information and possibly leaks it.

In general, there are three distinct methodologies for probing user attributes in LLM: (1) eliciting attributes through in-context learning utilizing interpretable discrete prompts that are manually designed, (2) eliciting attributes through the training of tunable prompts, and in this paper, we adopt soft prompts which are more amenable to optimization compared with discrete prompts, (3) training a classifier on embeddings generated for user tokens that appear in the input prompts. The three subsections below show how much user attribute information is encoded and how they can be probed by the three methods above.

A.1 Manually-Designed Prompt

In the first method, we directly adopt manually-designed discrete prompts using in-context learning to probe user sensitive attributes out of the LLM. We use questions about users with (or without) their item interaction history and expect reasonable answers when multiple examples are appended in the input.

More specifically, we test two types of manual prompts: direct prompts and in-context learning prompts. The direct prompt directly asks the LLM about a user's sensitive attribute, as shown by the following example, one without user-item interaction and one with user-item interaction.

Discrete Prompt without User-Item Interaction
Input: What is the {{attribute}} for user_{{user_id}}? **Output**: {{user attribute value}}
Discrete Prompt with User-Item Interaction
Input: User_{{user_id}} has watched movies (or bought insurance) {{sequence of movie (or insurance) IDs}}. What is the {{attribute}} of user_{{user_id}}? **Output**: {{user attribute value}}

The attribute can be gender, age, occupation or marital status provided by MovieLens and Insurance datasets. The answer template is simply the value of the questioned attribute, such as female / male, above / below 55 years old, or single / married. We constrain the output generated from the decoder based on constrained token generation over all possible values of the questioned attribute (De Cao et al., 2021).

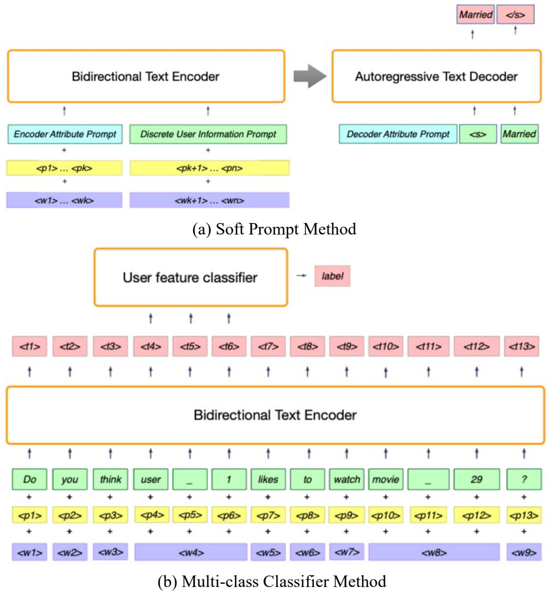

Figure 11: Details for Probing Methods

For in-context learning prompts, contextual examples, which are question-answer pairs of randomly sampled known users, are appended before the question. We use as many contextual examples as the maximum input length allows. The following example presents in-context learning prompts for the MovieLens dataset with and without user-item interaction information. We use gray color to differentiate the context from the question.

In-context Learning Example w/o User-Item Interaction
Input: What is the gender of user_1? Female. What is the gender of user_2? Male. What is the gender of user_3? Female. What is the gender of user_4? Female. What is the gender of user_5? Male. What is the gender of user_10? **Output**: Male
In-context Learning Example w/ User-Item Interaction
Input: User_1 has watched movies 17, 1991, 29, 3039, 890. What is the gender of user_1? Female. User_2 has watched movies 29, 1084, 27, 93, 781. What is the gender of user_2? Male. User_10 has watched movies 136, 798, 2778, 1894, 1. What is the gender of user_10? **Output**: Male.

We measure the performance of probing user sensitive attributes from LLM using AUC and results are presented in Table 7. We notice that the AUC is either 50% or slightly above 50%, indicating that the prediction result is no better than random guessing. Thus even if there is user sensitive information encoded in LLM such as P5 (see the next two subsections), direct prompting cannot elicit it. The reason may be that the model is trained using numerical user and item identifiers

MovieLens	Gender	Age	Occupation	–
w/ interaction	50.33	50.09	50.00	–
w/o interaction	50.26	50.00	50.00	–
Insurance	Gender	Age	Occupation	Marital
w/ interaction	50.00	50.33	50.47	50.20
w/o interaction	50.00	50.00	50.00	50.00

Table 7: Manually-Designed Prompt AUC (%)

rather than natural language labels or descriptions and does not include any additional user or item metadata. Therefore, prompts designed using natural language may not align with the numerical representations used in the model's training. Manual prompts' failure can be considered as an advantage of LLM-based RS, as user attributes will not be leaked too easily.

A.2 Soft Probing Prompt Tuning

In the second method, we adopt tunable prompts proposed in Lester et al. to explore soft prompt tuning with a frozen pre-trained LLM-based RS to elicit attributes. Each attribute has one soft probing prompt trained, which is tailored to act as a question, guiding the model to produce desired outcomes. Soft probing prompts can be optimized end-to-end over a training dataset and can condense information by learning from the training. The model structure is presented in Figure 11(a). The encoder input is a concatenation of an encoder attribute prompt and an untunable discrete prompt, where the discrete prompt part includes the target user and relevant user-item interaction history, as shown below:

> User user_{{user_id}} has watched movies (or bought insurances) {{sequence of item IDs}}.

The decoder attends to the decoder attribute prompt, the previously generated tokens, and the encoder hidden state to predict the probability distribution of future tokens. The encoder attribute prompt and decoder attribute prompt are generated respectively by a two-layer multi-layer perceptron (MLP) and a three-layer MLP as proposed in (Li and Liang, 2021). The prompts are tuned by minimizing the negative log-likelihood of the attribute value tokens y conditioned on the input text x and the soft probing prompts p in an end-to-end manner:

$$L = -\sum_{j=1}^{|y|} \log P(y_j|y_{<j}, x, p) \qquad (6)$$

For answer generation, we also apply the constrained generation as in manual prompting.

MovieLens	gender	age	occupation	–
	70.84	64.60	56.50	–
Insurance	gender	age	occupation	marital
	50.00	51.80	50.00	70.28

Table 8: Soft Probing Prompt Tuning AUC (%)

MovieLens	gender	age	occupation	–
	74.71	67.40	53.47	–
Insurance	gender	age	occupation	marital
	50.13	56.92	57.87	76.37

Table 9: Multi-class Classifier AUC (%)

In experiments, we create separate train and test datasets by dividing all users into two groups in a 9:1 ratio, and generating a unique discrete attribute prompt for each user in the process. Experimental results on MovieLens and Insurance datasets are shown in Table 8. We notice that using soft probing prompt tuning does generate non-trivial predictions on user attributes, especially on MovieLens dataset, indicating that LLM-based RS does encode user attributes and leaks personal information.

A.3 Multi-Class Classifier

The third probing method trains a multi-class classifier on the user token embeddings generated by the encoder for all input sentences in the training set. The model structure is presented in Figure 11(b), where the classifier is a seven-layer multi-layer perceptron (MLP) network trained by standard cross-entropy loss. Tables 9 presents the AUC results. The non-trivial AUC scores indicate that LLM-based RS also suffers from user information leakage, similar to other RS models. We also observe that the AUC scores obtained from the trained classifier tend to be higher than those obtained through soft probing prompt tuning. This suggests that training a classifier is a more effective probing method of user sensitive attributes from LLMs than training soft probing prompts. This observation highlights that the cross-entropy loss over multiple classes is better suitable than the negative log-likelihood loss over the entire vocabulary. This observation is leveraged in our design of fairness-aware foundation model architecture.

A.4 Summary of Probing LLM-RS Unfairness

This section demonstrates three possible methods to elicit user sensitive attributes from LLM-based RS: manually-designed discrete prompts, soft probing prompts, and multi-class classifier. The latter two successfully generate non-trivial user attribute values among the three methods. Figure 3 illustrates the degree of unfairness on LLM models trained on MovieLens and Insurance datasets, measured by the AUC of label prediction. The model on MovieLens is unfair on gender, age, and slightly on occupation, while the model on Insurance is unfair on the marital status the most.

B Results on P5-OpenLlama-3B

This appendix presents all the experiment results of the P5 recommendation paradigm under the Openllama-3B backbone. The observations here are largely consistent with that under the T5 backbone.

Table 10 and Table 11 present the recommendation performance and AUC scores.

Dataset	MovieLens	Insurance
↑ Hit@1	22.79	83.01
↑ Hit@3	35.97	87.95
↑ Hit@10	62.18	87.95
↓ AUC (G)	73.39	50.49
↓ AUC (A)	59.59	51.68
↓ AUC (O)	50.43	50.18
↓ AUC (M)	–	58.40

Table 10: Results of matching-based recommendation, G means Gender, A means Age, O means Occupation, and M means Marital Status (%).

Dataset	MovieLens	Insurance
↑ Hit@1	33.70	84.17
↑ Hit@3	46.92	87.23
↑ Hit@10	68.18	90.11
↓ AUC (G)	73.39	51.32
↓ AUC (A)	59.59	52.40
↓ AUC (O)	50.43	50.97
↓ AUC (M)	–	61.89

Table 11: Results of sequential-based recommendation, G means Gender, A means Age, O means Occupation, and M means Marital Status (%).

Tables 12 and 13 present the single-attribute fairness performance using single-attribute CFPs.

Dataset	MovieLens			Insurance		
Attribute	Gender	Age	Occupation	Age	Marital	Occupation
↑ Hit@1	20.78	22.08	22.79	83.01	82.74	83.01
↑ Hit@3	34.62	35.12	35.97	87.95	87.31	87.95
↑ Hit@10	59.14	60.97	62.18	87.95	87.92	87.95
↓ AUC	52.30	50.23	50.43	51.68	50.00	50.18

Table 12: Results of single-attribute fairness-aware prompting on matching-based models (%)

Tables 14 and 15 present the multi-attribute fairness-aware performance using prompt mixture over multiple CFPs.

Dataset	MovieLens			Insurance		
Attribute	Gender	Age	Occupation	Age	Marital	Occupation
↑ Hit@1	31.72	32.69	33.70	84.17	82.33	84.17
↑ Hit@3	44.60	45.72	46.92	87.23	86.14	87.23
↑ Hit@10	65.13	67.73	68.18	90.11	88.90	90.11
↓ AUC	54.38	52.25	50.43	52.40	50.23	50.97

Table 13: Results of single-attribute fairness-aware prompting on sequential models (%)

Model	GA	GO	AO	GAO
↑ Hit@1	22.13	20.78	22.08	22.13
↑ Hit@3	36.77	34.62	35.12	36.77
↑ Hit@10	60.08	59.14	60.97	60.08
↓ Avg. AUC	50.49	51.37	50.33	50.47

Table 14: Results of multi-attribute fairness-aware prompting on MovieLens dataset (%)

Model	AO	AM	MO	AMO
↑ Hit@1	84.17	82.33	82.33	82.33
↑ Hit@3	87.23	86.14	86.14	86.14
↑ Hit@10	90.11	88.90	88.90	88.90
↓ Avg. AUC	51.69	51.32	50.60	51.20

Table 15: Results of multi-attribute fairness-aware prompting on Insurance dataset (%)

C Pseudo Code for CFP Training

In this section, we provide the pseudo code of training the Counterfactually-Fair Prompts (CFP) for unbiased recommendation foundation model.

Algorithm 1 CFP Training

Require: Pretrained LLM4RS $\mathcal{M}$, Randomly initialized prefix prompt $\mathcal{P}$, Randomly initialized classifier $\mathcal{C}$, discriminator loss weight λ, number of epochs $Epoch\_num$, number of steps T to update $\mathcal{C}$ on L_{dis} or prefix prompt $\mathcal{P}$ on L_{rec}, number of batches R to update prefix prompt $\mathcal{P}$ on adversarial loss L

```
for epoch ← 1 to Epoch_num do
    for batch_num, batch do
        for i ∈ [1, T] do
            rec_loss, u_emb ← P(M,batch)
            dis_loss ← C(u_emb, label_u)
            L ← rec_loss - λ · dis_loss
            Optimize P based on L with M, C fixed
        end for
        if batch_num % R == 0 then
            for i ∈ [1, T] do
                rec_loss ← P(M,batch)
                Optimize P on rec_loss with M, C fixed
            end for
            for i ∈ [1, T] do
                rec_loss, u_emb ← P(M,batch)
                dis_loss ← C(u_emb, label_u)
                Optimize C on dis_loss with M, P fixed
            end for
        end if
    end for
end for
```