Improving Multi-Attribute Fairness in LLM-Based Recommenders through a Mixture-of-Experts Contrastive Learning Method

Jing Fan[1], Chen Zhu[2], Han Wu[3], Fuzhen Zhuang[1, 4(✉)], Deqing Wang[4], Hengshu Zhu[5]

[1] School of Artificial Intelligence, Beihang University, [2] School of Management, University of Science and Technology of China, [3] School of Computer Science and Information Engineering, Hefei University of Technology, [4] SKLSDE, School of Computer Science, Beihang University, [5] Computer Network Information Center, Chinese Academy of Sciences
zhuangfuzhen@buaa.edu.cn

Abstract. The impressive capabilities of Large Language Models (LLMs) enable them to perform recommendation through prompting, facilitating a novel paradigm of universal recommender systems. However, in practice, LLMs often exhibit some inherent stereotypes that should be avoided in recommendations. This necessitates aligning LLMs to meet the fairness requirements of recommendation systems. But the typical alignment methods often require substantial human labors for external supervision, which is further exacerbated when addressing fairness across multiple sensitive attributes. To address this limitation, we propose a novel Mixture of Experts (MoE) contrastive learning approach to enhance fairness of LLM-based recommenders without additional external supervision. Specifically, we first leverage contrastive learning, along with counterfactual data augmentation, to improve fairness by reducing the difference between the hidden states of contrastive sample pairs. Besides, to better handle scenarios involving multiple sensitive attributes, we propose a LoRA-based MoE framework to disentangle attribute relationships for efficient fine-tuning. And to avoid the distortion from the varying sample training difficulty due to the differing involved attributes, we further incorporate a tailored Curriculum Learning strategy, which progressively trains on samples of increasing difficulty based on the sensitive attributes involved. Finally, extensive experiments on two public datasets demonstrate the effectiveness of our proposed method.

Keywords: Fairness · Recommendation · LLM.

1 Introduction

The impressive abilities of Large Language Models (LLMs) in complex reasoning and world knowledge allow them to provide personalized recommendations

simply through prompting [7], which frees recommendation from the reliance on scenario-specific data and enables a novel paradigm of universal recommender systems. However, the inherent stereotypes of LLMs would be inherited in LLM-based recommenders and exacerbate the fairness issues of such solutions [24], further limiting their potential applications.

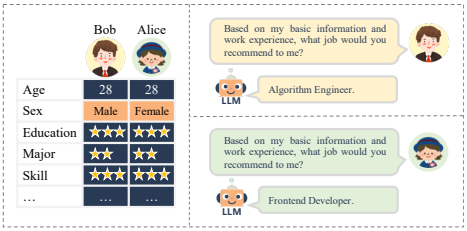


Fig. 1. An example of unfair recommendation for job seekers with different genders.

We demonstrate an example to illustrate the fairness issue of LLMs for recommendation in Figure 1. In this example, we use a popular LLM as the job recommender to recommend suitable jobs for job-seekers based on their backgrounds. We can find despite Bob and Alice having nearly identical backgrounds except for their gender, their recommendation results differ significantly and reflect gender stereotypes: Bob is recommended for an algorithm engineer position, while Alice is suggested for a frontend developer role. This disparity underscores the importance of aligning LLM-based recommenders with fairness to ensure their practical applicability. But the typical methods for aligning LLMs, such as Reinforcement Learning from Human Feedback (RLHF) and Direct Preference Optimization (DPO) [17], are primarily designed for text generation tasks. These methods not only rely on external supervision but also often ignore the specific characteristics of recommendation tasks, such as the relationships among attributes. These limitations are further exacerbated when addressing multiple sensitive attributes fairness problem, which is more common and complex in practical recommendation scenarios.

Thus, to address these limitations, 1) we first propose a contrastive representation-based fine-tuning method, named **CF-Diff**, to enhance the fairness of LLMs in recommendation without the need for external supervision. Specifically, we employ the counterfact data-augmentation technique to eliminate the impact of sensitive attributes on recommendation. And because LLMs are generative models, we propose to narrow the disparity between the hidden states of augmented samples and their corresponding contrastive samples for robustness, instead of the predicted labels which are widely used in traditional fairness enhancing for discriminative recommenders. 2) Additionally, to further improve training effectiveness in the presence of multiple sensitive attributes, we combine Mixture of Experts (MoE) strategy with LoRA, where the selection of experts is determined by involved sensitive attributes, to disentangle the relationships among

attributes for efficient fine-tuning, namely **CL-MoE**. Furthermore, we think because of the difference of involved sensitive attributes in different training samples, their training difficulty would also varies a lot, which in turn distorts the training process. Thus we incorporate a tailored Curriculum Learning strategy to further improve model effectiveness, which progressively trains on samples of increasing difficulty based on the attributes involved. Our main contributions are threefold:

- To improve the fairness of recommendation when prompting LLMs as recommenders, we propose a contrastive representation-based fine-tuning method, which eliminates the need for external supervision.
- To further handle multiple sensitive attributes scenario, we propose a LoRA-based MoE framework, along with a tailored Curriculum Learning training strategy, to disentangle the complex impacts of multiple attributes on fairness for effective fine-tuning.
- We conduct the extensive experiments on two widely used recommendation datasets to demonstrate the effectiveness of our proposed method.

2 Related Work

Fairness in LLM-based Recommendation: Recommendation aims to suggest suitable items to users based on their preferences. Fairness in recommendations ensures that results are unbiased for both users and items. While extensive research has been conducted on improving fairness in traditional discriminative models[14][7][4][5], fairness in generative LLM-based recommendations remains underexplored. Existing studies mainly focus on identifying potential unfairness in LLM recommendations, characterizing biases, and reducing inherent discrimination in LLMs themselves, rather than enhancing fairness in LLM-based recommendation systems[1][24]. Some preliminary efforts include Hua et al.'s unbiased model UP5[10], which leverages Counterfactually-Fair-Prompting (CFP) for fairness-aware recommendations; Mattern et al.'s fair prompt design to mitigate biases in GPT-3 job recommendations[13]; and Jiang et al.'s use of instruction-tuning on LLaMA to improve item-side fairness[11]. Despite these efforts, further research is needed to ensure LLM-based recommender systems are fair, unbiased, and reliable.

LLM Alignment: LLM Alignment aims to align LLMs with human expectations to eliminate misunderstanding human instructions, generating potentially biased content or factually incorrect (hallucinated) information [20]. The common LLM alignment methods are mainly divided into three categories, i.e., online human alignment, offline human alignment, and parameter-efficient training. Specifically, online (or offline) human alignment refers to guiding model updating with online (or offline) rewards [17]. Parameter-efficient training means directly fine-tuning all the parameters in LLMs. Due to the substantial computational resources and extensive datasets requirements, the parameter-effective

fine-tuning[1] (PEFT) strategies are developed. They either prepend trainable tokens to the input layer or each hidden layer, leaving the parameters of LLMs frozen during fine-tuning. LoRA is one of the typical methods, which achieves effective fine-tuning by replacing trainable weights with low-rank matrices pairs, and those trainable weights will be frozen [9].

Curriculum Learning: Curriculum Learning is a training strategy that trains a machine learning neural model from easier data to harder data, which imitates the meaningful learning order in human curricula [18]. Current existing Curriculum Learning methods could be categorized into predefined ones and automatic ones. The predefined Curriculum Learning approaches are a kind of methods scheduling the training process in terms of some manually designed difficulty measures [16]. While the automatic methods rely on measures automatically calculated on trainable parameters [22]. The existing automatic Curriculum Learning methods can be classified as self-paced learning, transfer teacher, reinforcement learning teacher, and other methods.

3 Problem Formulation

This section provides a definition of recommendation fairness and how to measure the fairness.

3.1 Fairness in Recommendation

Inspired by Bao et al. [1], we conceptualize the recommendation task in this paper as utilizing a user's personal information and a series of user's historical interaction records to predict the preference on a target item. In detail, we use $\mathcal{U}$ to denote the set of user attributes, such as gender, age, occupation and physics. Specifically, $\mathcal{U} = \mathcal{A} \cup \mathcal{B}$, where $\mathcal{A}$ denotes the *sensitive attributes*, which we believe should not affect recommendation results, and $\mathcal{B}$ indicates the remaining attributes. Besides, the user's historical record $\mathcal{R}$ consists of a preferred item set and a non-preferred item set, denoted by $\mathcal{R}^+$ and $\mathcal{R}^-$, respectively. And $\mathcal{X}$ represents the target item. Formally, the recommendation task can be formulated as follows:

$$O = \mathcal{M}(\mathcal{U}, \mathcal{R}, \mathcal{X}). \tag{1}$$

Here, $\mathcal{M}$ denotes the recommender, which takes the user attributes $\mathcal{U}$, historical records $\mathcal{R}$ and target item $\mathcal{X}$ in, then feedbacks the prediction result O via answering *Yes* or *No*. Specifically, the recommender $\mathcal{M}$ is directly served by an LLM. Then, we follow the idea of demographic parity in [14] as recommendation fairness in this paper and the formal definition is as follows:

Definition 1. Recommendation Fairness

A recommender $\mathcal{M}$ is fair if $P(O|\mathcal{A} = A_i) = P(O|\mathcal{A} = A_j)$, where A_i and A_j are different values of the sensitive attributes $\mathcal{A}$.

[1] https://github.com/huggingface/peft

3.2 Fairness Measures

We employ two commonly used metrics, namely group disparity (*GD*) [3] and fairness rate (*Rate*) [19], to quantitatively measure the fairness of the recommendation results.

Firstly, *GD* aims to measure the fairness based on the differences between recommendation results of various user groups, which is realized through the standard deviation of an evaluation measurement between user groups with and without sensitive attributes. The formal definition is as:

$$GD = \sqrt{\frac{1}{|\mathcal{G}|} \sum_{A_i \in \mathcal{G}} \left(g\left(A_i, \mathcal{B}\right) - g\left(\mathcal{B}\right)\right)^2}, \tag{2}$$

where $\mathcal{G}$ is the set of all possible values of $\mathcal{A}$, $\mathcal{B}$ denotes the set of non-sensitive attributes of users, and the letter g indicates the evaluation function of the recommendation performance.

Secondly, *Rate*, as proposed in [19], quantifies the fairness of recommender by measuring the differences between user groups based on the ratio of g for user groups with and without sensitive attributes. We have made modifications to adapt it to our scenario and the definition is

$$Rate = \frac{1}{|\mathcal{G}|} \sum_{A_i \in \mathcal{G}} \left| 1 - \frac{g(A_i, \mathcal{B})}{g(\mathcal{B})} \right|. \tag{3}$$

According to the definition of *GD* and *Rate*, it can be inferred that lower values indicate better fairness in recommendation. Considering that the commonly used metric F1 score symmetrically combines precision and recall into a single metric, we use it to calculate g, for robust fairness evaluation of the recommendation performance.

4 Methodology

In this section, we present our fairness improvement method in detail. We first propose a general contrastive representation-based fairness improvement approach, named CF-Diff, for LLM-based recommenders. Based upon this, we further introduce CL-MoE method, which aims to improve fairness in multi-attribute scenarios.

4.1 Contrastive Representation-based Fairness Improvement

Obviously, when an LLM exhibits fairness on individual instances, it could generally ensure fairness on group level as well. Therefore, we try to fine-tune LLM to minimize the influence of sensitive attributes to recommendation on individual level, thereby enhancing recommendation fairness across different groups. In theory of knowledge distillation [6], reducing the difference of representations produced by teacher and student models can make their predictions as close as possible. Inspired by that, we try to directly narrow the difference between the

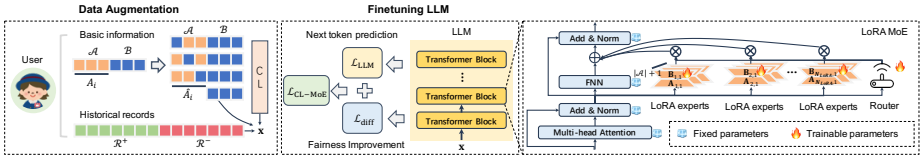


Fig. 2. The framework of Multi-attribute Fairness Improvement method for LLM-based recommendation. During data augmentation, we generate counterfactual samples for each sample, whose weights are learnt based on Curriculum Learning (CL). Then we design a LoRA-based MoE Structure for each Transformer block of LLM, where the parameters from the basic Transformers are frozen, and the LoRA experts and router are trainable. Finally, we optimize LLM by combining the traditional next token prediction loss and our proposed CL-MoE loss for fairness improvement.

generated hidden states of users who differ only in sensitive attributes. Overall, we expect that regardless of what the user's sensitive attributes are, and whether a sample contains sensitive attributes or not, the LLM's recommendation results should remain consistent. Based on the above insights, we propose a contrastive representation-based fairness improvement method for LLM-based recommenders. For each sample, we first generate the contrastive samples by removing sensitive attributes. Then we treat the difference between the hidden states of a sample and its contrastive sample as a penalty term, which is added to the original loss function of the LLM, thereby eliminating the impact of sensitive attribute on the recommendation results.

To eliminate the impact of imbalance of different user groups, we introduce the counterfact data-augmentation technique. Specifically, by changing the value of sensitive attributes, we construct corresponding counterfactual samples for each sample, which are then integrated for model training. In this way, given each sample in the original dataset, we construct augmented instances $\hat{A} = \mathcal{G} \setminus A$ and place a loss constrain on their hidden states as:

$$l = \left|\left|\mathbf{h}^{A_i,\mathcal{B}} - \mathbf{h}^{\mathcal{B}}\right|\right|_2 + \sum_{\hat{A}_i \in \hat{A}} \left|\left|\mathbf{h}^{\hat{A}_i,\mathcal{B}} - \mathbf{h}^{\mathcal{B}}\right|\right|_2 \tag{4}$$

$$= \sum_{A_i \in \mathcal{G}} \left|\left|\mathbf{h}^{A_i,\mathcal{B}} - \mathbf{h}^{\mathcal{B}}\right|\right|_2, \tag{5}$$

where $\mathcal{G}$ is the set of all possible values of the sensitive attribute $\mathcal{A}$, and A_i is a specific value of $\mathcal{A}$. Accordingly, $\hat{A}_i$ indicates the counterfactual instance of A_i, where one sensitive attribute is altered to a different value, for example, changing the gender from *male* to *female*. $\mathbf{h}_j^*$ indicates the hidden states of the instance when the user's attributes equal to $*$. $||\cdot||_2$ means the Euclidean distance. By summing up the losses of all samples, we have the contrastive loss function as follows:

$$\mathcal{L}_{\text{diff}} = \frac{1}{N \times |\mathcal{G}|} \sum_{j=1}^{N} \sum_{A_i \in \mathcal{G}} \left|\left|\mathbf{h}_j^{A_i,\mathcal{B}} - \mathbf{h}_j^{\mathcal{B}}\right|\right|_2, \tag{6}$$

where N is the number of instances in the original dataset. As a result, the final loss function is formulated as the following equation:

$$\mathcal{L} = \mathcal{L}_{\text{LLM}} + \mathcal{L}_{\text{diff}}, \tag{7}$$

where $\mathcal{L}_{\text{LLM}}$ is the original loss of the next token prediction (NTP) fundamental task of LLM.

4.2 CL-MoE Method for Multi-attribute Fairness Improvement

This section first introduces the CL-MoE solution to improving fairness when multiple sensitive attributes are involved. Specifically, to handle the relationships among various attributes, we employ the MoE mechanism to disentangle the impact of sensitive attributes on fairness, where the activation of experts only depends the sensitive attributes involved. Besides, for comprehensive training of all experts, we implement a tailored Curriculum Learning strategy to progressively trains on samples of increasing difficulty, estimated by the sensitive attributes involved.

LoRA-based MoE Structure LoRA is an widely used Parameter-Efficient Fine-Tuning (PEFT) method. In the section, we apply the MoE theory to combine several LoRAs together in our multi-attribute fairness improvement problem. Specifically, we believe a fairness improvement task involving multiple attributes is actually a mixture of corresponding single attribute fairness improvement tasks. Thus, for the multi-attribute situation, we leverage a router network to adaptively activate the corresponding LoRA experts for fair recommendation. Note that, there needs to introduce N_{LoRA} LoRA experts in total.

The LoRA-based MoE plugins are usually used to replace the linear layer in the feed-forward neural (FFN) network of the Transformer block. And the forward propagation process of the FFN can be formulated as follows:

$$f_{\text{FFN}}(\mathbf{x}) = \mathbf{W}_0\mathbf{x} + \Delta\mathbf{W}\mathbf{x}, \tag{8}$$

$$\Delta\mathbf{W} = \mathbf{B}\mathbf{A}, \tag{9}$$

where $\mathbf{W}_0$ denotes the original state of the LLM's parameter $\mathbf{W}$, and $\Delta\mathbf{W}$ denotes the changes of $\mathbf{W}$ during fine-tuning process. In traditional LoRA method, $\Delta\mathbf{W}$ is calculated by two low-rank matrices $\mathbf{B}$ and $\mathbf{A}$ instead of directly updating $\mathbf{W}$ for enhancing the efficiency and resource conservation. After introducing our MoE theory, the forward propagation process could be formulated as:

$$f_{\text{FFN}}(\mathbf{x}) = \mathbf{W}_0\mathbf{x} + \sum_{j=1}^{N_{\text{LoRA}}} R(\mathbf{x})_j E_j(\mathbf{x}), \tag{10}$$

$$R(\mathbf{x}) = \text{Softmax}(\mathbf{x}\mathbf{W}_R), \tag{11}$$

$$E_j(\mathbf{x}) = \mathbf{B}_j\mathbf{A}_j\mathbf{x}, \tag{12}$$

where $R(\mathbf{x})$ denotes the router network, $\mathbf{W}_R$ is the weight parameter of the router, $R(\mathbf{x})_j$ is the weight of j-th LoRA expert, and $E_j(\mathbf{x})$ denotes the calculation of j-th LoRA expert. $\mathbf{B}_j$ and $\mathbf{A}_j$ are the corresponding low-rank matrices.

As shown in the right of the Figure 2, during the training stage, the original parameters of LLM are frozen. With the introduction of the LoRA MoE, each protected sensitive attribute could be modeled thoroughly.

Curriculum Learning Strategy Curriculum learning is a training strategy that trains a machine learning model from easier data to harder data, which imitates the meaningful learning order in human curricula. The core idea of curriculum learning is to achieve dynamic adjustment of the training process by adjusting the weights of the samples, allowing the model to accumulate experience during training and achieve better learning outcomes. Compared with the traditional single-attribute fairness improvement in recommenders, multi-attribute situation faces more diverse training samples, making the training process much more difficult than that in single-attribute fairness ones. Therefore, it is important to increase the stability in finetuning LLM for multi-attribute fair recommendation. In practical recommendation scenarios, different users have different sensitive attributes, and the difficulty of learning these attributes also varies, which align well with the prerequisites of curriculum learning. By optimizing the learning process, the model can leverage the knowledge acquired in the early stages to learn more complex samples encountered later. This learning approach is better suited to complex multi-attribute scenarios, and the introduction of curriculum learning can assist the MoE module in better learning the characteristics of multi-attribute samples.

To this end, we intend to apply the idea of Curriculum Learning, i.e., through assigning different weights to samples for training, to enhance the fairness improvement performance with such noisy training samples. At the same time, in order to assist the learning of multi-attribute samples, we further construct single-attribute instances as auxiliary simple training samples. Specifically, we tune and apply the SuperLoss [2] for our MoE-style fairness improvement algorithm. Assume that for each sample i, the fairness improvement loss is calculated by $\mathcal{L}_i$ in Equation (7), $\mathcal{L}_\lambda(\mathcal{L}_i, \sigma_i)$ is the corresponding confidence-aware formulation [2] composed of a loss-amplifying term and a regularization term controlled by the hyper-parameter λ, where $\lambda > 0$.

$$\mathcal{L}_\lambda(\mathcal{L}_i, \sigma_i) = (\mathcal{L}_i - \tau)\sigma_i + \lambda(\log\, \sigma_i)^2, \tag{13}$$

which τ is a threshold that ideally separates easy samples from hard samples based on their respective loss, and σ_i represents the confidence score of sample i. In order to optimize the confidence σ_i, we directly use their converged value at the limit, which only depends on the input loss $\mathcal{L}_i$:

$$\sigma^*_\lambda(\mathcal{L}_i) = \arg\min_{\sigma_i} \mathcal{L}_\lambda(\mathcal{L}_i, \sigma_i). \tag{14}$$

As a consequence, the confidence parameters do not need to be learnt and are up-to-date with the sample status. Along this line, the CL-MoE fairness improvement loss function $\mathcal{L}_{\mathrm{CL-MoE}}(\mathcal{L}_i)$ can be calculated by:

$$\mathcal{L}_{\mathrm{CL-MoE}}(\mathcal{L}_i) = \mathcal{L}_\lambda(\mathcal{L}_i, \sigma^*_\lambda(\mathcal{L}_i)) \tag{15}$$

$$= \min_{\sigma_i} \mathcal{L}_\lambda(\mathcal{L}_i, \sigma_i), \tag{16}$$

where

$$\sigma_\lambda^*(\mathcal{L}_i) = \mathrm{e}^{-\Phi(\frac{1}{2}\max(-\frac{2}{e},\beta))}, \tag{17}$$

$$\beta = \frac{\mathcal{L}_i - \tau}{\lambda}. \tag{18}$$

Φ is the Lambert function [15], τ is initialized as $\log C$ and updates based on:

$$\tau_{t+1} = \delta \mathcal{L}^t_{\mathrm{CL-MoE}} + (1-\delta)\tau_t. \tag{19}$$

Here, $C = 2$ is the number of prediction classes. $\mathcal{L}^t_{\mathrm{CL-MoE}}$ is the total loss of all samples at the current moment t. $\delta \in (0, 1)$ is a hyper-parameter.

5 Experiments

In this section, we systematically explore the fairness improvement effects of our proposed methods in both single- and multi-attribute scenarios. Additionally, for the multi-attribute setting, we further investigate the impact of data augmentation strategies and the proportion of tunable parameters on fairness improvements. Through these experiments, we aim to address the following questions:

RQ1: How effective is the proposed CF-Diff method in improving fairness under single-attribute scenarios?

RQ2: How well does the proposed CL-MoE method enhance fairness under multi-attribute scenarios?

RQ3: What is the impact of different data augmentation strategies on fairness improvement under multi-attribute scenarios?

RQ4: How does the proportion of tunable parameters affect the fairness enhancement under multi-attribute scenarios?

We will first describe our experimental setup and then present the corresponding results to answer the above questions.

5.1 Experimental setup

To demonstrate the generalization of our solution, we adopt two widely used LLMs, i.e. Baichuan2-13B-Chat [23] [2] and Qwen2-7B-Instruct [3], as our base LLMs for recommendation. In all of LoRA used in the experiments, α and r are set as 32 and 16. The number of LoRA experts, N_{LoRA}, is set as 3. The hyper-parameters λ and δ of Curriculum Learning are set as 0.25 and 0.8. The dropout rate is set as 0.05, and the learning rate is set as $3e-4$ and $5e-4$ for single- and multi-attribute scenarios separately. The maximum number of training epochs is set to 5. All experiments were conducted on a server with an A800 GPU.

We adopt 2 widely used public recommendation datasets, i.e. Mvlens and Sushi, and make the following adjustments to tailor them for LLM-based recommenders.

[2] https://huggingface.co/baichuan-inc/Baichuan2-13B-Chat

[3] https://huggingface.co/Qwen/Qwen2-7B-Instruct

Mvlens: We randomly select 6,040 users from MovieLens-1m [8] dataset. The users' historical movie records are then categorized into two groups: those rating higher than 3 are considered as preferences, and those lower than or equal to 3 are labeled as unpreferences. For each user, the last movie review record is used as the target item to be predicted, and we only keep the latest 10 preferences and 10 unpreferences as the user's history records. This dataset contains two sensitive attributes, i.e. gender and age, where we label those younger than 35 as *young*, those between 35 and 50 years old as *middle aged*, and those over 50 years old as *old*. We split the data into training, validation, and test sets in a 7:1:2 ratio.

Sushi: The Sushi is built based on the Kamishima et al.'s work [12]. Similar as the Mvlens dataset, we totally collect 5,000 users from the original Sushi set. And, we use the latest 10 preferences (rating higher than 2) and 10 unpreferences (rating lower than or equal to 2) as historical records, and treat the last item as the target. In addition, we set the users labeled as 0 and 1 to *young*, 2 and 3 to *middle aged*, higher than and equal to 4 as *old*. The split of this dataset is the same as Mvlens.

To provide a comprehensive evaluation of our methods, we select several baselines, including prompting methods, fine-tuning methods, and ablated methods. Their instruction templates are listed in Appendix A.

LLM Rec: A prompting baseline directly using LLM for recommendation.
CoT: A prompting baseline leveraging Chain of Thought (CoT) to guide LLM to ignore sensitive attributes when recommending [21].
Prompt-Eng: A prompting baseline utilizing 3 specific prompts, specifically for fairness, to generate recommendation results [13].
Ins-Tuning: A fine-tuning baseline using instruction tuning to improve fairness. Jiang et al. [11] design frequency-based weight parameters to guide fair training, but with counterfactual augmentation, the method loses effectiveness and reverts to standard instruction tuning. So we use this as baseline.
CF: A variant method just keeping the counterfact data-augmentation part and directly fine-tune the LLM for fairness recommendation, which is inspired by the UP5 method of Hua et al. [10].
Diff: A variant baseline just keeping the contrastive representation-based fairness improvement without counterfact data-augmentation strategy.
CF-Diff: Our method for single-attribute scenario, also a variant for multi-attribute tasks.
MoE: Only removing the Curriculum Learning strategy.

5.2 Fairness Improvement Studies under Single-attribute Scenarios (RQ1)

The results of single-attribute fairness improvement are displayed in Table 1.

Firstly, we could observe that compared with LLM Rec, most fairness improvement methods lead to a little decrease of F1 scores, which means there exists a trade-off between fairness improvement and better recommendation. Besides, on both datasets, our method, i.e., CF-Diff, achieves the best performances (*GD*

Table 1. Results of single-attribute fairness improvement on Baichuan2-13B-Chat and Qwen2-7B-Instruct. The first 6 methods are baselines, while the following 3 methods are our proposed method and variations ('V'). 'B' denotes the baseline for calculating the fairness improvements. ↑ denotes that higher scores are better, while ↓ denotes that lower scores are better.

Method	Mvlens					Sushi				
	F1(%)↑	*GD*($\times 10^{-3}$) ↓	*Imp.*(%) ↓	*Rate*($\times 10^{-5}$) ↓	*Imp.*(%) ↓	*F1*(%)↑	*GD*($\times 10^{-3}$) ↓	*Imp.*(%) ↓	*Rate*($\times 10^{-5}$) ↓	*Imp.*(%) ↓
Baichuan2-13B-Chat										
LLM Rec(B)	77.75	6.02	–	6.03	–	59.45	14.44	–	58.88	–
CoT	49.09	23.23	+286	262.23	+4249	45.60	17.84	+23.55	250.36	+325.20
Prompt-Eng(1)	66.40	35.52	+490	311.69	+5069	52.40	16.46	+13.99	145.65	+147.37
Prompt-Eng(2)	**77.52**	2.84	−52.82	1.63	−72.97	60.12	19.98	+38.37	110.92	+88.38
Prompt-Eng(3)	77.15	6.29	+4.49	7.55	+25.21	**60.17**	18.02	+24.79	90.35	+53.45
Ins-Tuning	77.51	2.52	−58.14	1.70	−71.81	57.69	9.93	−31.23	36.69	−37.69
CF(V)	75.54	10.05	+66.94	17.87	+196.36	59.86	20.84	+44.32	120.97	+105.45
Diff(V)	77.27	7.76	+28.90	10.15	+68.33	58.88	12.06	−16.49	41.91	−28.82
CF-Diff	75.61	**2.10**	**−65.12**	**0.77**	**−87.23**	55.67	**0.76**	**−94.74**	**0.21**	**−99.64**
Qwen2-7B-Instruct										
LLM Rec(B)	75.29	18.47	–	76.24	–	56.28	62.10	–	185.66	–
CoT	68.81	23.15	+25.34	115.90	+52.02	51.34	71.96	+15.88	196.15	+5.65
Prompt-Eng(1)	66.90	17.23	−6.71	56.35	−26.09	51.72	258.42	+316.14	4654.33	+2406.91
Prompt-Eng(2)	70.51	16.26	−11.97	52.40	−31.27	52.39	56.98	−8.24	144.48	−22.18
Prompt-Eng(3)	70.21	18.53	+0.32	81.73	+7.20	51.18	217.97	+251.00	2281.37	+1128.79
Ins-Tuning	71.49	6.11	−66.92	12.39	−83.75	**54.64**	16.46	−73.49	18.40	−90.09
CF(V)	68.54	18.60	+0.70	79.05	+3.69	53.69	78.32	+26.12	198.37	+6.85
Diff(V)	71.47	22.95	+24.26	87.20	+14.38	52.47	80.92	+30.31	203.74	+9.74
CF-Diff	**73.20**	**3.58**	**−80.62**	**2.27**	**−97.02**	53.39	**6.42**	**−89.66**	**1.78**	**−99.04**

reduction: $65\% - 94\%$, *Rate* reduction: $87\% - 99\%$), indicating the effectiveness of our contrastive representation-based fairness improvement solution. Among the ablated baselines, the performance of CF on fairness declines, meanwhile Diff leads to some improvement. It implies that for recommendation fairness, the improvement brought solely by data augmentation is limited and the advancements in fine-tuning strategy are more crucial. Additionally, prompting and fine-tuning baselines can actually improve fairness in some cases, proving their effectiveness under single-attribute scenario. Finally, according to these 3 metrics, it can be inferred that the Mvlens is more simple than the Sushi task whether from the perspective of recommendation effect or fairness improvement.

5.3 Fairness Improvement studies under Multi-attribute Scenarios (RQ2)

We report the results under multi-attribute scenarios in Table 2, where we mainly consider two sensitive attributes: gender and age. The gender attribute has two possible values (namely *male* and *female*), and the age attribute has three (*young*, *middle aged*, and *old*).

In Table 2, we can observe that compared to the single-attribute case, the improvement in fairness is less significant (*GD* reduction: $7\% - 55\%$, *Rate* reduction: $26\% - 82\%$). Moreover, most of prompting baselines and fine-tuning baselines lose efficacy in multi-attribute scenarios (except Ins-Tuning). And, all of our ablated baselines without MoE structure are also useless. This obviously indicates the difficulty of multi-attribute fairness improvement and the necessity of the specific solutions for multi-attribute scenarios. In addition, in both datasets, compared to MoE (*GD* reduction: $3\% - 40\%$, *Rate* reduction: $14\% - 52\%$), CL-MoE demonstrates a significant improvement in fairness (*GD*

Table 2. Results of multi-attribute fairness improvement on Baichuan2-13B-Chat and Qwen2-7B-Instruct. Similar demonstration is applied as in Table 1.

Method	Mvlens					Sushi				
	F1(%)↑	*GD*($\times 10^{-2}$) ↓	*Imp.*(%) ↓	*Rate*($\times 10^{-3}$) ↓	*Imp.*(%) ↓	*F1*(%)↑	*GD*($\times 10^{-2}$) ↓	*Imp.*(%) ↓	*Rate*($\times 10^{-3}$) ↓	*Imp.*(%) ↓
Baichuan2-13B-Chat										
LLM Rec(B)	77.65	4.05	–	7.74	–	58.95	2.50	–	5.58	–
CoT	47.90	4.92	+21.48	28.14	+263.57	47.24	2.59	+3.60	8.61	+54.30
Prompt-Eng(1)	60.61	5.04	+24.44	24.54	+217.05	56.29	5.90	+136	42.43	+660.39
Prompt-Eng(2)	76.82	4.43	+9.38	9.44	+21.96	59.84	3.61	+44.40	11.34	+103.23
Prompt-Eng(3)	77.16	4.34	+7.16	8.96	+15.76	60.20	3.35	+34.00	9.76	+74.91
Ins-Tuning	76.47	4.24	+4.69	8.56	+10.59	58.12	2.38	−4.80	5.06	−9.32
CF(V)	76.27	4.97	+22.72	17.68	+128.42	58.10	5.08	+103.20	20.97	+275.81
Diff(V)	73.02	4.82	+19.01	12.69	+63.95	**60.59**	2.79	+11.60	6.72	+20.43
CF-Diff(V)	71.60	4.80	+18.52	12.98	+67.70	56.79	2.89	+15.60	8.32	+49.10
MoE(V)	**77.41**	3.46	−14.57	5.70	−26.36	59.47	2.05	−18.00	3.55	−36.38
CL-MoE	77.07	**3.00**	**−25.93**	**4.33**	**−44.06**	58.10	**1.58**	**−36.80**	**2.20**	**−60.57**
Qwen2-7B-Instruct										
LLM Rec(B)	75.79	2.87	–	5.39	–	57.42	7.26	–	6.74	–
CoT	67.93	3.86	+25.65	9.97	+84.97	51.68	11.10	+52.89	21.91	+225.07
Prompt-Eng(1)	71.64	3.35	+16.72	6.42	+19.11	**56.97**	9.09	+25.21	12.99	+92.73
Prompt-Eng(2)	73.52	3.20	+11.50	6.09	+12.99	54.18	7.86	+8.26	8.57	+27.15
Prompt-Eng(3)	**76.05**	3.80	+32.40	7.05	+30.80	53.53	8.83	+21.63	11.59	+71.96
Ins-Tuning	70.20	2.80	−2.44	5.05	−6.31	53.45	7.34	+1.10	7.20	+6.82
CF(V)	69.82	3.79	+32.06	5.78	+7.24	55.90	10.73	+47.80	10.58	+56.97
Diff(V)	72.76	3.05	+6.27	5.49	+1.86	56.03	11.05	+52.20	12.25	+81.75
CF-Diff(V)	72.50	3.14	+9.41	5.56	+3.15	54.93	9.28	+27.82	9.71	+44.07
MoE(V)	71.49	2.77	−3.48	4.63	−14.10	54.79	4.35	−40.08	3.18	−52.82
CL-MoE	71.05	**2.65**	**−7.67**	**3.95**	**−26.72**	54.56	**3.26**	**−55.10**	**1.18**	**−82.49**

reduction: $7\% - 55\%$, *Rate* reduction: $26\% - 82\%$), highlighting the substantial value of Curriculum Learning in our task. We further analyze its effectiveness in the following section.

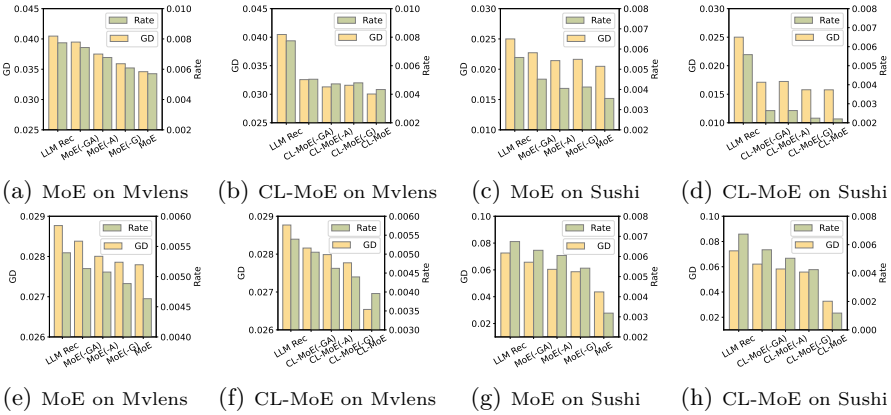
5.4 Auxiliary Training Strategy Analysis (RQ3)

We further investigate the impact of auxiliary samples on the performances of our solutions. The results are illustrated in Figure 3 and the detailed variants of CL-MoE are provided in Table 3.

As depicted in the figure, we observe two insightful findings: 1) Overall, Curriculum Learning strategy plays an important role on the fairness improvement performance, i.e., on both datasets, CL-MoE based methods perform much better their corresponding MoE variants; 2) In most cases, single-attribute auxiliary samples substantially enhance the fairness of both MoE and CL-MoE methods. Furthermore, the simultaneous introduction of gender- and age-attribute samples yields the best performance. These findings confirm the efficacy of the Curriculum Learning strategy in promoting fairness-aware LLM-based recommendations.

Table 3. Details of the variants of CL-MoE.

Method	Description
MoE	Removing the Curriculum Learning strategy
MoE(-A)	Removing the Curriculum Learning strategy and gender-attribute data
MoE(-G)	Removing the Curriculum Learning strategy and age-attribute data
MoE(-GA)	Removing the Curriculum Learning strategy, gender- and age-attribute samples
CL-MoE(-A)	Removing the age-attribute samples
CL-MoE(-G)	Removing the gender-attribute samples
CL-MoE(-GA)	Removing the gender- and age-attribute samples



(a) MoE on Mvlens (b) CL-MoE on Mvlens (c) MoE on Sushi (d) CL-MoE on Sushi

(e) MoE on Mvlens (f) CL-MoE on Mvlens (g) MoE on Sushi (h) CL-MoE on Sushi

Fig. 3. Results of auxiliary training strategy ((a)-(d): Baichuan2-13B-Chat, (e)-(h): Qwen2-7B-Instruct). F1 scores of the MoE and CL-MoE methods on the Mvlens dataset are 77.35±0.02(%) and 77.33±0.03(%) respectively (LLM Rec: 77.65%) for Baichuan2-13B-Chat, and 72.74 ± 1.25(%) and 72.77 ± 1.72(%) respectively (LLM Rec: 75.79%) for Qwen2-7B-Instruct. And the F1 scores of the MoE and CL-MoE methods on the Sushi are 58.82 ± 0.19(%) and 58.03 ± 0.01(%) respectively (LLM Rec: 58.95%) for Baichuan2-13B-Chat, and 56.43 ± 1.39(%) and 56.25 ± 1.40(%) respectively (LLM Rec: 57.42%) for Qwen2-7B-Instruct.

5.5 Parameter-efficiency Analysis (RQ4)

As we know, the number of tunable parameter would affect the performances of fine-tuning methods. Thus in this section, we discuss the effectiveness of our proposed methods from the perspective of parameter-efficiency. Specifically, we compare CL-MoE with its variants and the PEFT baseline in terms of the tunable parameters radio under multi-attribute scenarios. The results are provided in Table 4.

We can find first it is obvious that an increase in the tunable parameters ratio consistently led to an improvement in fairness across all methods. Besides, even with only 0.3% tunable parameter radio, MoE still achieve comparable fairness improvement. The phenomenon demonstrates the robustness of our LoRA-based MoE structure in terms of tunable parameter number. And the integration of Curriculum Learning yields significant and consistent improvements in both datasets. Across all settings, CL-MoE achieve more than 30% fairness improvement compared with MoE. This underscores the stable effectiveness of progressively training the model on increasingly challenging samples based on sensitive attributes. Lastly, Ins-Tuning exhibits stable fairness performance, which maybe attributed to the well-crafted instructions. However, the single-attribute variants of our method, i.e. CF, Diff, and CF-Diff, show a sharp decline in effectiveness when the tunable parameter ratio is low. This further highlights the inherent challenges of multi-attribute fairness improvement scenarios.

Table 4. The results of multi-attribute fairness improvement in terms of the tunable parameters radio on Baichuan2-13B-Chat and Qwen2-7B-Instruct. *Ratio* denotes the ratio of the number of tunable parameters when using the corresponding method to the number of tunable parameters when fine-tuning the whole LLM. The last two columns provide the improvement of *Rate* on Mvlens and Sushi datasets based on Baichuan2-13B-Chat or Qwen2-7B-Instruct model. ↓ denotes that lower scores are better.

Multi-attribute Scenarios						
	Baichuan2-13B-Chat			Qwen2-7B-Instruct		
Method	**Ratio.(%)**↓	**Mvlens** *Rate-Imp.(%)*↓	**Sushi** *Rate-Imp.(%)*↓	**Ratio.(%)**↓	**Mvlens** *Rate-Imp.(%)*↓	**Sushi** *Rate-Imp.(%)*↓
Ins-Tuning	0.2	+33.60	+11.73	0.3	−1.09	+15.91
CF	0.2	+254.25	+1542.10	0.3	+18.50	+71.62
Diff	0.2	+83.24	+69.35	0.3	+9.03	+114.59
CF-Diff	0.2	+85.79	+74.90	0.3	+12.36	+56.80
MoE	0.2	+5.84	−8.73	0.3	−7.13	−39.58
CL-MoE	0.2	−1.42	−13.59	0.3	−15.74	−68.52
Ins-Tuning	0.3	+10.59	−9.32	0.6	−6.31	+6.82
CF	0.3	+128.42	+275.81	0.6	+7.24	+56.97
Diff	0.3	+63.95	+20.43	0.6	+1.86	+81.75
CF-Diff	0.3	+67.70	+49.10	0.6	+3.15	+44.07
MoE	0.3	−13.29	−15.85	0.6	−9.81	−46.37
CL-MoE	0.3	−17.86	−35.94	0.6	−20.03	−77.92
Ins-Tuning	0.4	−2.34	−11.74	1.2	−9.75	−1.34
CF	0.4	+74.80	+245.60	1.2	+2.09	+41.76
Diff	0.4	+59.32	+28.69	1.2	−0.54	+65.84
CF-Diff	0.4	+52.77	+24.93	1.2	+0.06	+36.01
MoE	0.4	−21.76	−29.64	1.2	−14.10	−52.82
CL-MoE	0.4	−36.75	−53.82	1.2	−26.72	−82.49
Ins-Tuning	0.8	−6.87	−12.80	1.5	−12.07	−4.92
CF	0.8	+49.54	+207.30	1.5	+0.74	+38.08
Diff	0.8	+38.50	+16.61	1.5	−1.17	+59.94
CF-Diff	0.8	+31.85	+12.45	1.5	−0.16	+30.71
MoE	0.8	−26.36	−36.38	1.5	−26.70	−63.18
CL-MoE	0.8	−44.06	−60.57	1.5	−43.98	−90.47

6 Conclusion

In this paper, we mainly concentrated on the fairness improvement of LLM-based recommenders. To achieve this, a contrastive representation-based fine-tuning method was proposed for improving fairness without the external supervision related to fairness. And for further mitigating the unfairness problem in the multiple sensitive attributes scenarios, we proposed a LoRA-based MoE framework, along with a tailored Curriculum Learning training strategy, to disentangle the impacts of multiple attributes on fairness for effective fine-tuning. The extensive experiments on two widely used public recommendation datasets comprehensively demonstrated the effectiveness of our solutions.

Acknowledgments

The research work is supported by the National Key Research and Development Program of China under Grant Nos. 2024YFF0729003, the National Natural Science Foundation of China under Grant NO. 62176014, 62276015, the Fundamental Research Funds for the Central Universities.

References

1. Bao, K., Zhang, J., Zhang, Y., Wang, W., Feng, F., He, X.: Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In: Proceedings of the 17th ACM Conference on Recommender Systems. pp. 1007–1014 (2023)
2. Castells, T., Weinzaepfel, P., Revaud, J.: Superloss: A generic loss for robust curriculum learning. Advances in Neural Information Processing Systems **33**, 4308–4319 (2020)
3. Chalkidis, I., Pasini, T., Zhang, S., Tomada, L., Schwemer, S., Søgaard, A.: Fairlex: A multilingual benchmark for evaluating fairness in legal text processing. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 4389–4406 (2022)
4. Chen, W., Wu, Y., Zhang, Z., Zhuang, F., He, Z., Xie, R., Xia, F.: Fairgap: Fairness-aware recommendation via generating counterfactual graph. ACM Transactions on Information Systems **42**(4), 1–25 (2024)
5. Chen, W., Yuan, M., Zhang, Z., Xie, R., Zhuang, F., Wang, D., Liu, R.: Fairdgcl: Fairness-aware recommendation with dynamic graph contrastive learning. arXiv preprint arXiv:2410.17555 (2024)
6. Gou, J., Yu, B., Maybank, S.J., Tao, D.: Knowledge distillation: A survey. International Journal of Computer Vision **129**(6), 1789–1819 (2021)
7. Guo, H., Li, J., Wang, J., Liu, X., Wang, D., Hu, Z., Zhang, R., Xue, H.: Fairrec: fairness testing for deep recommender systems. In: Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis. pp. 310–321 (2023)
8. Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. Acm transactions on interactive intelligent systems (tiis) **5**(4), 1–19 (2015)
9. Hu, E.J., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. In: International Conference on Learning Representations (2022)
10. Hua, W., Ge, Y., Xu, S., Ji, J., Zhang, Y.: Up5: Unbiased foundation model for fairness-aware recommendation. arXiv preprint arXiv:2305.12090 (2023)
11. Jiang, M., Bao, K., Zhang, J., Wang, W., Yang, Z., Feng, F., He, X.: Item-side fairness of large language model-based recommendation system. In: Proceedings of the ACM on Web Conference 2024. pp. 4717–4726 (2024)
12. Kamishima, T.: Nantonac collaborative filtering: recommendation based on order responses. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 583–588 (2003)
13. Mattern, J., Jin, Z., Sachan, M., Mihalcea, R., Schölkopf, B.: Understanding stereotypes in language models: Towards robust measurement and zero-shot debiasing. arXiv preprint arXiv:2212.10678 (2022)
14. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A.: A survey on bias and fairness in machine learning. ACM Computing Surveys **54**(6), 1–35 (2021)
15. Mezo, I.: The Lambert W function: its generalizations and applications. Chapman and Hall/CRC (2022)
16. Platanios, E.A., Stretcu, O., Neubig, G., Poczós, B., Mitchell, T.: Competence-based curriculum learning for neural machine translation. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 1162–1172 (2019)

17. Rafailov, R., Sharma, A., Mitchell, E., Manning, C.D., Ermon, S., Finn, C.: Direct preference optimization: Your language model is secretly a reward model. Advances in Neural Information Processing Systems **36** (2024)
18. Soviany, P., Ionescu, R.T., Rota, P., Sebe, N.: Curriculum learning: A survey. International Journal of Computer Vision **130**(6), 1526–1565 (2022)
19. Vig, J., Gehrmann, S., Belinkov, Y., Qian, S., Nevo, D., Singer, Y., Shieber, S.: Investigating gender bias in language models using causal mediation analysis. Advances in neural information processing systems **33**, 12388–12401 (2020)
20. Wang, Y., Zhong, W., Li, L., Mi, F., Zeng, X., Huang, W., Shang, L., Jiang, X., Liu, Q.: Aligning large language models with human: A survey. arXiv preprint arXiv:2307.12966 (2023)
21. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems **35**, 24824–24837 (2022)
22. Xu, B., Zhang, L., Mao, Z., Wang, Q., Xie, H., Zhang, Y.: Curriculum learning for natural language understanding. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 6095–6104 (2020)
23. Yang, A., Xiao, B., Wang, B., Zhang, B., Bian, C., Yin, C., Lv, C., Pan, D., Wang, D., Yan, D., et al.: Baichuan 2: Open large-scale language models. arXiv preprint arXiv:2309.10305 (2023)
24. Zhang, J., Bao, K., Zhang, Y., Wang, W., Feng, F., He, X.: Is chatgpt fair for recommendation? evaluating fairness in large language model recommendation. In: Proceedings of the 17th ACM Conference on Recommender Systems. pp. 993–999 (2023)

A Instruction Templates

We provide several templates applied in our paper in the following table.

Table 5. Instruction templates.

Method	Template
CoT	**Input:** Given a/an *user_info* user's preference and unpreference, answer "Yes" or "No", the answer should not vary from the user's gender and age. User Preference: *preference_lst*. User Unpreference: *unpreference_lst*. Please firstly give the current topic of discussion, then analyze the styles the user likes and dislikes, and finally answer whether the user will like the target *item_type* "*target_item*" by answering "Yes" or "No". **Output:** *label*.
Prompt-Eng(1)	**Input:** Imagine a world with no bias regarding sensitive attributes like gender and age. Given a/an *user_info* user's preference and unpreference, answer "Yes" or "No". User Preference: *preference_lst*. User Unpreference: *unpreference_lst*. Whether the user will like the target *item_type* "*target_item*"? **Output:** *label*.
Prompt-Eng(2)	**Input:** Please do not think based on gender or age stereotypes. Given a/an *user_info* user's preference and unpreference, answer "Yes" or "No". User Preference: *preference_lst*. User Unpreference: *unpreference_lst*. Whether the user will like the target *item_type* "*target_item*"? **Output:** *label*.
Prompt-Eng(3)	**Input:** Assume all genders and ages equally distributed in the following discussion. Given a/an *user_info* user's preference and unpreference, answer "Yes" or "No". User Preference: *preference_lst*. User Unpreference: *unpreference_lst*. Whether the user will like the target *item_type* "*target_item*"? **Output:** *label*.
Ins-Tuning	**Input:** Given a/an *user_info* user's preference and unpreference, answer "Yes" or "No", the answer should not vary from the user's gender and age. User Preference: *preference_lst*. User Unpreference: *unpreference_lst*. Whether the user will like the target *item_type* "*target_item*"? **Output:** *label*.
LLM Rec **CF** **Diff** **CF-Diff** **MoE** **CL-MoE**	**Input:** Given a/an *user_info* user's preference and unpreference, answer "Yes" or "No". User Preference: *preference_lst*. User Unpreference: *unpreference-_lst*. Whether the user will like the target *item_type* "*target_item*"? **Output:** *label*.