Automatic Feature Fairness in Recommendation via Adversaries

Hengchang Hu*
hengchang.hu@u.nus.edu
National University of Singapore
Singapore

Yiming Cao†
caoy0035@e.ntu.edu.sg
Nanyang Technological University
Singapore

Zhankui He
zhh004@eng.ucsd.edu
UC, San Diego
United States

Samson Tan‡
samson.tmr@u.nus.edu
AWS AI Research & Education
United States

Min-Yen Kan
kanmy@comp.nus.edu.sg
National University of Singapore
Singapore

ABSTRACT

Fairness is a widely discussed topic in recommender systems, but its practical implementation faces challenges in defining sensitive features while maintaining recommendation accuracy. We propose *feature fairness* as the foundation to achieve equitable treatment across diverse groups defined by various feature combinations. This improves overall accuracy through balanced feature generalizability. We introduce unbiased feature learning through adversarial training, using adversarial perturbation to enhance feature representation. The adversaries improve model generalization for under-represented features. We adapt adversaries automatically based on two forms of feature biases: frequency and combination variety of feature values. This allows us to dynamically adjust perturbation strengths and adversarial training weights. Stronger perturbations are applied to feature values with fewer combination varieties to improve generalization, while higher weights for low-frequency features address training imbalances. We leverage the Adaptive Adversarial perturbation based on the widely-applied Factorization Machine (AAFM) as our backbone model. In experiments, AAFM surpasses strong baselines in both fairness and accuracy measures. AAFM excels in providing item- and user-fairness for single- and multi-feature tasks, showcasing their versatility and scalability. To maintain good accuracy, we find that adversarial perturbation must be well-managed: during training, perturbations should not overly persist and their strengths should decay.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Recommender System, Adversarial Training, Fair Recommendation

ACM Reference Format:
Hengchang Hu, Yiming Cao, Zhankui He, Samson Tan, and Min-Yen Kan . 2023. Automatic Feature Fairness in Recommendation via Adversaries. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region (SIGIR-AP '23), November 26–28, 2023, Beijing, China.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3624918.3625318

1 INTRODUCTION

Fairness in Recommendation Systems (RS) has garnered considerable attention. Various techniques have been employed, such as outcome re-ranking [9, 19, 29] and unbiased learning [2, 9, 27, 38, 40] (mitigating biases in the training process directly). However, the specific fairness requirements vary depending on the stakeholders and the specific needs of the application. The definition of fairness in user-centric or item-centric recommendations relies on the chosen sensitive features [34]. Prior studies [9, 19, 21, 41] only consider the chosen sensitive features either from users or items, which poses challenges in terms of fairness scalability when considering the other aspect. Furthermore, imposing constraints to achieve fairness often compromises overall recommendation accuracy, further constraining real-world applicability.

To enable flexible selection of sensitive features, we introduce generic **feature fairness** as our core guiding principle. It centers on features themselves, agnostic to whether features are from users or items. In this work, we examine two statistical biases for feature values (e.g., *student* or *male*): feature *frequency*, indicating the occurrence rate within its feature domain (e.g., user occupation, or user gender); and feature *combination variety*, representing the diversity of co-occurring samples with other features.

To investigate the biased outcomes resulting from skewed features, let us take a case of MovieLens. Our preliminary analysis focuses on two feature-defined user groups: *male+students* and *male+homemakers*. We use Factorization Machine [25] for modeling here. Figure 1 illustrates that the majority group of *male students* (representing 21.09% of users) consistently outperforms the average, while the minority group of *male homemakers* (representing 0.1%) exhibits below-average performance with significant fluctuations during training. This disparity in accuracy and stability results in unfairness. There are two causes: (1) Limited co-occurrence frequency of the *male* and *homemaker* features in training hinders the model's ability to capture their interactions. (2) Additionally, when there is a greater combination variety of gender for the *student* feature compared to the *homemaker* feature, the model struggles

[*] Corresponding author.
[†] Work done while interning at the National University of Singapore.
[‡] Work done while a Ph.D. student at the National University of Singapore.

SIGIR-AP '23, November 26–28, 2023, Beijing, China

ACM ISBN 979-8-4007-0408-6/23/11.
https://doi.org/10.1145/3624918.3625318

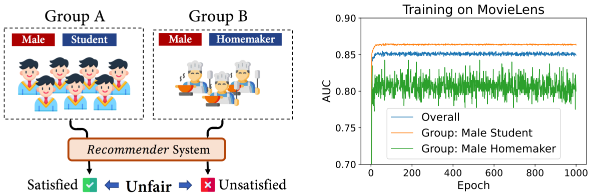

Figure 1: (left) Unfairness between two groups with sensitive features. (right) Biased validation accuracy between two data groups during training of Factorization Machine.

to recognize interactions between the *homemaker* feature and different gender values, resulting in poorer generalization.

In this work, we aim to utilize the two forms of feature biases to automatically (1) incorporate fairness considerations across diverse feature domains; and (2) ensure similar generalizability for different combinations of feature values.

Adversarial training [11] is a technique for augmenting model generalization [15], where the generalization derives from its robustness to unseen inputs. We thus adopt adversarial training to accommodate a variety of feature combinations. By integrating adversarial training into our regular training iterations, we enhance feature representations by perturbing them. However, applying this approach directly still poses issues.

First, existing approaches assume consistent perturbation intensities [8, 15] for all feature representations, but there are significant variations in sample outcomes associated with different features. Our method utilizes *combination variety* as the measure to determine the intensity of adversarial perturbation. We employ a formula that maps lower variety values to higher adversarial intensity, thereby enhancing the stability of targeted groups. To prevent excessive perturbation that overly distorts the original data representation, we map *variety* inversely proportional to a range of $0 \sim 1$. Second, conventional adversarial unbiased learning approaches often view accuracy and fairness as conflicting objectives [34]. As features often follow a long-tailed distribution, low-frequency features make up the majority of features. Hence, low-frequency features are important, so we prioritize their appropriate representation during training by assigning higher adversarial training weights. This balancing results in enhanced performance.

We instantiate the above-mentioned Adaptive Adversaries with the FM model as our backbone, or AAFM for short. Extensive experiments show that our method improves results by 1.9% in accuracy against baselines, while balancing group standard deviation by $\frac{7}{10}$ on fairness metrics. AAFM further demonstrates scalability, tackling fairness concerns for both users and items simultaneously. Additionally, as the number of feature domains in the data increases, our approach consistently tackles fairness at finer levels among diverse groups. This serves as a bridge between group and individual fairness, spanning datasets with one feature domain to those with a broader range of three feature domains. Our method's universal applicability to fairness issues offers a win–win outcome by promoting both fairness and accuracy.

In summary, our contributions are as follows: (i) Compared to user fairness and item fairness, we define our task as a more fundamental feature fairness objective. The feature fairness task aims to develop a parameter-efficient framework that flexibly provides feature-specific fairness for various combinations of user or item features. (ii) We introduce AAFM, an adversarial training method that leverages statistical feature bias for unbiased learning, combining the benefits of fairness and accuracy. (iii) Through experiment datasets with varying numbers of features, user- and item-centric settings, we validate the scalability and practicality of AAFM in real scenarios. The code is available at: https://github.com/HoldenHu/AdvFM

2 METHODOLOGY

In what follows, we first outline our task and delve into the issue of feature fairness, which arises due to two biases. We then provide our solution — Adversarial Factorization Machines which applies the fast gradient method to construct perturbations over feature representations. We further propose an adaptive perturbation based on feature biases, which re-scales adversarial perturbation strengths and adversarial training weights.

2.1 Preliminaries of Feature Fairness

Problem Formulation. The recommendation task aims to predict the probability of unobserved user–item interactions $\hat{y}(\mathbf{x})$ given the user and item features $\mathbf{x}$ [16]. We represent one sample, the input as the combination of these features, denoted as $\mathbf{x} = \{x_1, x_2, ..., x_n\}$. Here, x_i represents the i^{th} feature domain, encompassing user features (e.g., user occupation) and item features (e.g., item color). Concerning the k^{th} sample $\mathbf{x}^{(k)} \in \mathcal{X}$, $x_i^{(k)}$ indicates its specific feature value (e.g., *student* or *red*) in feature domain x_i. In our work, the feature domains include user/item ID, and the categorical attributes of user/item. Concerning specific feature value v in domain x_i, we denote its corresponding samples of subset data as $\mathcal{X}_{x_i:v} = \{\mathbf{x}^{(k)} | x_i^{(k)} = v\}$. The overall prediction error of the subset data is denoted as $\mathcal{E}_{x_i:v} = \sum_{\mathbf{x}\in\mathcal{X}_{x_i:v}} \mathcal{E}(\hat{y}(\mathbf{x}), y)$. Here, $\mathcal{E}$ indicates the metric (e.g., Logloss) measuring errors between the prediction $\hat{y}$ and ground-truth y, where $y \in \{0, 1\}$.

To achieve feature fairness, we expect a smaller difference between errors $\mathcal{E}_{x_i:v_1}$ and $\mathcal{E}_{x_i:v_2}$ with respect to each feature domain x_i and each value pair (v_1, v_2) within x_i. In neural models, the precise representation of each value is vital, as it directly affects errors in corresponding samples. The quality of feature value representation depends on the statistical bias (e.g., popularity bias [24]) of feature values in the data.

Two Forms of Feature Biases. Feature values in the data distribution have the following statistical properties. To aid understanding, we show an example of feature value v in the feature domain x_i.

- *Frequency* α_v indicates the occurrence rate of the value v concerning its feature domain.
- *Combination variety* β_v indicates the number of diverse samples where value v co-occurs with other features in combination.

α_v can be used to measure how many times this feature value has been seen by the model, while β_v better reflects the degree of isolation of this feature-based data group. The more isolated the groups are, the more likely they are sensitive to model perturbation.

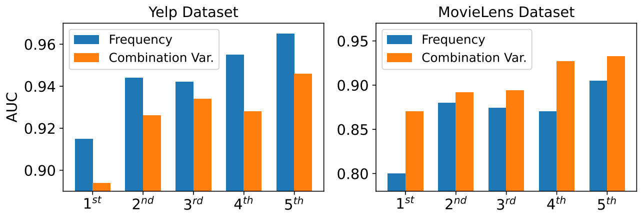

Figure 2: Unbalanced results regarding two forms of feature biases. x-axis indicates the indices of sample groups sorted by the overall feature frequency/combination variety. The results are from FM applied to the Yelp/Movielens dataset.

In normal distributions, combination variety and frequency can be viewed as equivalent, where the frequency increase, the combination variety increase as well. But in real-world cases, this may not hold true as feature values may not always follow a strict joint probability dependence. Take the feature domain gender as an example. Given a situation where *female* has fewer combinations with *occupation* than *male*, this does not mean that the feature value *female*’s frequency is necessarily less than *male*. In the results depicted in Figure 2, data samples were grouped into 5 bins based on the multiplied value of frequency or combination variety across all feature domains (*user features+item features*). While both biases contribute significantly to performance imbalances, they are not aligned, highlighting the interdependence between features in real-world data. Therefore, we consider them as separate statistical biases for utilization.

2.2 Adversarial Factorization Machine (AdvFM)

2.2.1 *Base Model.*

Our framework consists of three stages (Figure), characterized by stages for Embedding. Representation learning and Prediction.

(a) Embedding Initialization. To improve the representative ability of features, we first map each original discrete feature value of x_i into d-dimensional continuous vectors $e_i = \mathcal{M}(x_i|\Theta)$ through the embedding layer $\mathcal{M}$. Here, the concatenated feature embeddings are denoted as $\mathbf{e} = cat[e_1; ...; e_n]$.

(b) Representation Learning. Our key insight is that the interdependencies among low-level feature groups play a critical role in robustness and fairness. For this reason, we use Factorization Machines (FM) [25] as the backbone for our methodology. FM takes a set of vector inputs, each consisting of n feature values and performs recommendations through their cross-product. An FM model of degree 2 estimates the rating behavior $\hat{y}$ as:

$$f(\mathbf{e}) = \sum_{i=1}^{n} \langle w_i, e_i \rangle + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle v_i, v_j \rangle \, e_i e_j, \tag{1}$$

where parameter $w_i \in \mathcal{R}^{1\times d}$ models the linear, first-order interactions, and $v_i \in \mathbb{R}^{1\times d}$ models second-order interactions for each low dimensional vector e_i. $\langle \cdot, \cdot \rangle$ indicates the dot product operation and $e_i e_j$ indicates element-wise product between them. To be concise, we use the notation $\hat{y}(\mathbf{x}|\Theta) = f(\mathbf{e})$ to represent the model’s processing of input $\mathbf{x}$ with the embedding parameter Θ.

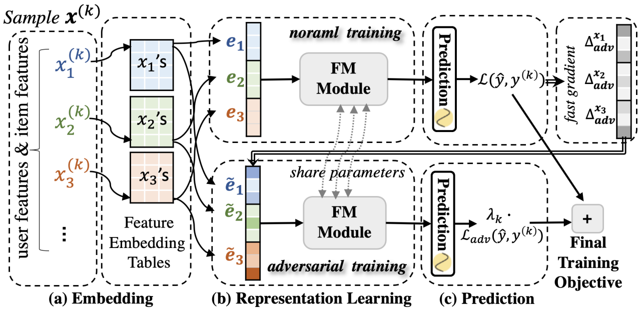

Figure 3: The training process of adversarial factorization machine on sample x$^{(k)}$.

(c) Prediction & Model Training. The training objective function is defined as:

$$\mathcal{L}(\hat{y}, y) = \sum_{(\mathbf{x},y)} y \log(\hat{y}(\mathbf{x}|\Theta) + (1-y)\log(1-\hat{y}(\mathbf{x}|\Theta)) \tag{2}$$

where $\mathcal{L}$ indicates the cross entropy loss [10], the difference between the predicted and true values.

2.2.2 *Adversarial Perturbation.*

Inspired by previous work [28] which observed that users with rare interactions would benefit more from robustness, we adopt gradient-based adversarial noise [11] as the perturbation mechanism to improve balanced robustness from the feature perspective.

As shown in Fig 3, the normal representation learning of FM module utilizes the original embedding $\mathbf{e}$. The adversarial training adds noise to each feature’s embedding by perturbing FM’s parameters:

$$\tilde{e}_i = \mathcal{M}(x_i|\Theta + \Delta_{adv}^{e_i}) \tag{3}$$

where $\Delta_{adv}^{e_i}$ is the parameter noise providing the maximum perturbation on the embedding layer. $\Delta_{adv} = \{\Delta_{adv}^{e_1}, ...\Delta_{adv}^{e_n}\}$ denotes the overall perturbations on embedding layer.

To efficiently perturb normal training, we estimate the optimal adversarial perturbation by maximizing the loss incurred during training:

$$\Delta_{adv} = \arg \max_{\|\delta\| \leq \epsilon} \mathcal{L}(\hat{y}(\mathbf{x}|\Theta + \delta), y), \tag{4}$$

where the hyper-parameter ϵ controls the strength level of perturbations, and $\|\cdot\|$ denotes the $l2$ norm. Our adversarial noise uses the backward propagated fast gradient [11] of each feature’s embedding parameters as their most effective perturbing direction. Specifically, to perturb the embedding e_i, we calculate the partial derivative of the normal training loss:

$$\Delta_{adv}^{e_i} = \epsilon \cdot \frac{\partial \mathcal{L}(\hat{y}(\mathbf{e}|\Theta), y)/\partial e_i}{\|\partial \mathcal{L}(\hat{y}(\mathbf{e}|\Theta), y)/\partial e_i\|}, \tag{5}$$

where the right-hand side’s normalized term is the sign of the fast gradient direction of the feature x_i’s embedding parameters.

Training objective. In each epoch, we conduct training as normal first, then introduce the adversarial perturbations in another following training session, round by round. We define the final optimization objective for AdvFM as a min–max game:

$$\arg \min_{\Theta} \{\arg \max_{\Delta_{adv}} [\mathcal{L}(\hat{y}(\mathbf{x}|\Theta), y) + \lambda \cdot \mathcal{L}(\hat{y}(\mathbf{x}|\Theta + \Delta_{adv}), y)]\} \tag{6}$$

where Δ_{adv} provides the maximum perturbation and Θ is trained to provide a robust defense to minimize the overall loss. Here, λ is a hyper-parameter to control the adversarial training weights.

2.3 Automatic Adaptation on AdvFM

The approach described so far has a key drawback: It introduces a single, uniform perturbation strength level ϵ overall features, and uniform adversary weights λ over all samples. This makes the method inflexible, and unable to model nuanced weighting. To further balance and improve the accuracy, we further propose an Adaptive version of AdvFM (AAFM). It auto-strengthens the adversarial perturbations on the feature embedding parameters, and re-weights the samples in adversarial training. Our adaptive version leverages the two forms of feature biases previously introduced (Fig 3, right).

- *Auto-Strengthening.* Considering each feature domain x_i with the corresponding value v_i, a smaller combination variety β_{v_i} indicates a higher degree of sensitivity representation. Thus, it needs to be trained with stronger perturbations on its embedding parameters to improve its robustness. We estimate the feature-specific ϵ_{v_i} based on an inversely proportional basis:
$$\epsilon_{v_i} = \psi\left(\omega_i \times (\beta_{v_i})^{-1}\right), \tag{7}$$
where ω_i is a learnable parameter with respect to the feature domain x_i. We adopt SoftPlus activation function for ψ, as it does not change the sign of the gradient, and the SoftPlus unit has a non-zero gradient over all real inputs.
- *Re-Weighting.* Unlike previous work [15] conducting fixed adversarial training weight λ for all samples, we conduct sample-specific ones. Specifically, given a sample $\mathbf{x}^{(k)}$, the sample-specific adversary weight λ_k is defined as:
$$\lambda_k = \Phi(-\prod_{x \in \mathbf{x}^{(k)}} \alpha_x, t), \tag{8}$$
For the sample $\mathbf{x}^{(k)}$ with a low overall feature frequency $\prod_{x \in \mathbf{x}^{(k)}} \alpha_x$ in training, we increase the weight of its adversarial loss by increasing its associated λ value. The function $\Phi(\cdot, t)$ is used to scale the values between 1 and t. If we use the previous design of trainable parameter ω to scale, λ is easily eliminated by the overarching optimization goal (Equation 6); hence we apply manually-controlled scaling via t.

Optimization of Decaying Adversarial Perturbation. When the model adaptively adjusts the adversarial perturbation (noise) level ϵ, we observe that optimization may simply set ϵ to zero, which best meets the normal training objectives by achieving a local optimum. However, this thwarts the benefit of introducing adversarial perturbation; canceling it prematurely.

To mitigate this, we envision a slow decline in the effect of adversarial perturbation, proportional to the time already trained.

To this end, we design a regulation term for ω by defining an additional loss $\mathcal{L}_{decay} = \alpha(\tau \cdot \|\omega\|)^{-1}$, where τ represents the trained epoch number, and α is an annealing hyper-parameter controlling regulation strength. As such, the change of ω is more marked during early training, where a small ω would make $\mathcal{L}_{decay}$ large. As the training proceeds and the model stabilizes, the sensitivity of ω gradually decays, as τ increases.

3 EXPERIMENTS

Datasets. We experiment on three public datasets to examine our model's debiasing effect on both user and item groups. User feature enriched recommendation datasets include movie dataset *MovieLens-100K*[1] (user gender, occupation, and zip code), and image dataset *Pinterest*[2] (user preference categories). Item feature enriched recommendation datasets include movie dataset *MovieLens-100K*[1] (movie category, and release timestamp), and business dataset *Yelp*[3] (business city, star). Following the previous work [13] to reduce the excessive cost, we filtered out the user with more than 20 interactions in Yelp, and randomly selected 6,000 users to construct our Pinterest dataset. We convert all continuous feature values into categorical values (e.g., by binning user age into appropriate brackets), and consider the user and item IDs as additional features.

Baselines. We choose our comparison baseline with respect to models achieving strong recommendation accuracy and debias effects. *Accuracy Baselines* include matrix factorization-based method ONCF [14] and FM-families — FM [25], NFM [13], DeepFM [12], CFM [35]. *Debiasing Baselines* include regularization-based approach M-Match [17], classical inverse propensity scoring approach IPS [26], MACR [31] incorporating user/item's effect in the loss, DecRS [30] investigating the causal representation of users.

Evaluation Protocols. For the train–test data split, we employ standard leave-one-out [15]. To evaluate the *accuracy*, we adopt *AUC* (Area Under Curve) and *Logloss* (cross-entropy). To assess the *fairness* concerning imbalanced features, we split data into buckets for evaluation, following previous work [23, 31]. We first rank the data samples $\mathbf{x}^{(k)}$ by joint feature statistics $\prod_{x \in \mathbf{x}^{(k)}} (\alpha_x \cdot \beta_x)$, and divide the ranked samples into 6 buckets. We propose two quantitative metrics as follows.

- *EFGD* (extreme feature-based groups difference). Following the previous practice [23] that term the difference between the two extreme data groups as the indicator, we take EFGD as the AUC difference between the first 10% samples and the last 10%.
- *STD* (overall groups' standard deviation). STD is used to measure more fine-grained fairness (as [31]). And STD stands for the AUC standard deviation of the buckets.

3.1 Recommendation Accuracy Comparison

3.1.1 *Superior Accuracy Against Baselines.*

We present the overall results in Table 1. Regarding both user and item feature-enriched datasets, our AAFM consistently outperforms other FM-based baselines. Among the baselines, DeepFM achieves the best performance in three datasets, as indicated by both Logloss and AUC metrics. This highlights its effectiveness in mapping sparse features to dense vectors using the neural embedding layer. CFM, employing 3D CNN, outperforms ONCF, which uses 2D CNN, indicating the superiority of 3D CNN in extracting feature interactions.

3.1.2 *Ablation Study.*

To further investigate where the performance improvement of AAFM originates from, we present the ablation study in the right-hand columns of Table 1. We can see that compared to AdvFM (without any adaptive optimization), the

[1] https://grouplens.org/datasets/movielens/

[2] https://sites.google.com/site/xueatalphabeta/academic-projects

[3] https://www.yelp.com/dataset

Scenarios	*Dataset*	*Metrics*	FM	NFM	CFM	DeepFM	ONCF	AdvFM	AAFM$^\lambda$	AAFM$^\epsilon$	AAFM	D-AAFM
item-centric	ML^i	LL	0.4093	0.3688	0.3635	0.3597	0.3641	0.3730	0.3391	0.3673	0.3352	**0.3248**
		AUC	0.9154	0.9203	0.9257	0.9381	0.9243	0.9337	0.9406	0.9308	0.9408	**0.9431**
	Yelp	LL	0.1934	0.1895	0.0963	0.1584	0.1527	0.1692	0.0878	0.1751	**0.0731**	0.0742
		AUC	0.9474	0.9569	0.9732	0.9665	0.9668	0.9653	0.9790	0.9619	**0.9813**	0.9795
user-centric	ML^u	LL	0.4493	0.4297	0.3876	0.3109	0.3721	0.4325	0.3182	0.4323	0.3072	**0.2996**
		AUC	0.8796	0.8908	0.9172	0.9319	0.9012	0.8810	0.9249	0.8808	0.9323	**0.9357**
	Pinterest	LL	0.5647	0.3865	0.3577	0.3541	0.4026	0.3859	0.3573	0.3914	0.3447	**0.3042**
		AUC	0.5700	0.7430	0.7356	0.7580	0.7251	0.7432	0.7695	0.7408	0.7756	**0.8031**

Table 1: Overall accuracy performance comparison. Smaller LL (Logloss) or larger AUC indicates better accuracy. ML^i or ML^u indicate the partial MovieLens dataset with only item or item features. AAFM$^\lambda$ and AAFM$^\epsilon$ only adaptively adjust λ (with fixed $\epsilon = 0.5$) and ϵ (with fixed $\lambda = 1$) respectively. D-AAFM indicates AAFM incorporating decaying perturbation regularization.

Scenarios	*Dataset*	*Metrics*	FM	IPS	M-match	MACR	DecRS	AdvFM	AAFM$^\lambda$	AAFM$^\epsilon$	AAFM	D-AAFM
item-centric	ML^i	EFGD	0.0713	0.0336	0.0381	0.0282	0.0589	0.0401	0.0232	0.0241	0.0110	**0.0105**
		STD	0.0257	0.0237	0.0236	0.0171	0.0246	0.0235	0.0139	0.0161	0.0091	**0.0069**
	Yelp	EFGD	0.0440	0.0243	0.0301	0.0177	0.0272	0.0230	0.0166	0.0228	**0.0144**	0.0181
		STD	0.0131	0.0114	0.0153	0.0082	0.0122	0.0086	0.0082	0.0079	**0.0064**	0.0068
user-centric	ML^u	EFGD	0.0415	0.0289	0.0337	0.0294	0.0374	0.0340	0.0323	0.0377	**0.0281**	0.0368
		STD	0.0280	0.0198	0.0208	0.0199	0.0230	0.0225	0.0219	0.0259	**0.0195**	0.0220
	Pin.	EFGD	0.1068	0.0682	0.0726	0.0558	0.0545	0.0853	0.0289	0.0580	0.0193	**0.0132**
		STD	0.0307	0.0277	0.0299	0.0275	0.0265	0.0300	0.0213	0.0296	0.0195	**0.0178**

Table 2: Feature fairness effect comparison. The smaller the STD or EFGD, the fairer the results. The abbreviations are the same as in Table 1. The upper/lower two datasets correspond to item-centric/user-centric fairness.

introduction of adaptive λ significantly enhances the overall performance. This indicates that our proposed adversarial training reweighting is promising and can optimize well, instead of locking the fairness model within performance-compromising constraints.

However, introducing only adaptive ϵ worsens the overall performance on several datasets. By considering both aspects together, synthesizing them into AAFM, and adding decaying perturbation regularization loss, we get D-AAFM. Either of them performs best across all datasets. In most cases, D-AAFM performs better, demonstrating that persistent adversarial perturbations can severely impact model accuracy.

3.2 Feature Fairness Results

3.2.1 Superior Fairness Against Baselines. Feature fairness is another aspect of concern in our study. As depicted in Table 2, all fairness baselines show improvement over FM in terms of metrics measuring the reduction in bias (EFGD and STD). We observe that the phenomenon of feature unfairness does exist, and that current fairness models do alleviate this issue. Among the baselines, MACR performs the best; it considers the popularity bias of both users and items, taking into account the impact of skewed occurrences of user or item IDs. Our AdvFM also provides more fair results, compared to FM. However, it is not as good as the aforementioned debiasing baselines. This corroborates that though adversarial training has shown promise in promoting fairness recently, it necessitates further detailed investigation. Through careful design of adversarial perturbations, our AAFM and D-AAFM achieve better fairness, concerning either user features or item features.

3.2.2 Ablation Study. To figure out how the effects of adversaries improve fairness, we conduct an additional ablation study, shown in the right columns of Table 2. Compared to AdvFM, the inclusion of adaptive λ and adaptive ϵ both significantly contribute to improving fairness. When both are utilized (i.e., AAFM), the effect on feature fairness is further enhanced. This demonstrates that both proposed automatic adaptations are complementary and indispensable. Features with smaller combination variety require a larger ϵ to improve generalization ability. Even though we encourage it by using the reciprocal of its bias, it is still very easy to reduce ϵ during training (thereby reverting back to normal training). In order to forcefully encourage adversarial training, it is necessary for samples with less frequent features to have more adversarial training weight, thus enabling the adversaries to truly play their role. Similar to the finding from the accuracy comparison, D-AAFM and AAFM alternately become the best models, suggesting different dataset sensitivities to long-term perturbations.

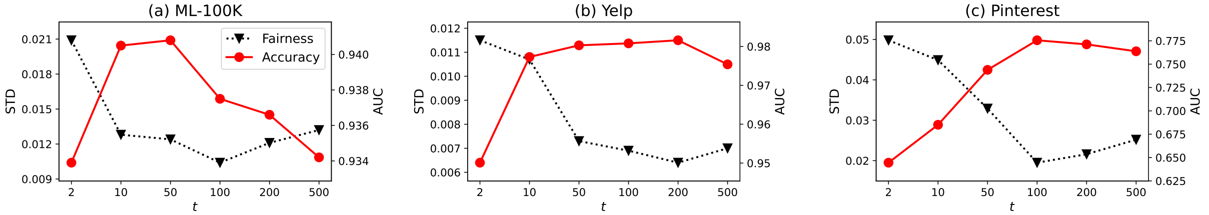

Figure 4: Trade-off between accuracy and user group fairness via control of the re-weighting parameter t. Smaller STD (↓) indicates better fairness, and larger AUC (↑) indicates better accuracy.

Dataset	ML-100K			Yelp			Pinterest		
Noise	0.5	1.0	2.0	0.5	1.0	2.0	0.5	1.0	2.0
FM	-4.67	-9.58	-18.9	-6.14	-12.7	-24.3	-2.74	-3.40	-4.95
AdvFM	-2.32	-4.75	-9.60	-3.38	-6.79	-13.3	-1.48	-1.53	-1.64
AAFM	-0.64	-0.76	-1.00	-1.41	-3.03	-6.37	-0.29	-0.30	-0.32

Table 3: Performance drop ratio (%) in AUC of models in the presence of external adversarial perturbation.

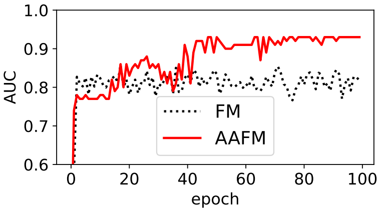

Figure 5: Validation accuracy (AUC) on the small group *(male entertainment)* in MovieLens dataset. The case study.

3.3 Robustness of AdvFM

Driven by the premise that adversarial training enhances robustness for perturbed parameters, we delve into understanding this improvement. In order to probe the robustness of groups under feature representation perturbations, we adopt the methodology from [15], which infuses external noise into the model parameters at levels spanning 0.5 to 2.0. As shown in Table 3, we observe that AdvFM exhibits less sensitivity to adversarial perturbations compared to FM. For instance, on the Yelp dataset, a noise level of 0.5 results in a decrease of 6.14% for FM, whereas AdvFM only experiences a decrease of 3.38%. Moreover, AAFM demonstrates even greater stability with a decrease of only 1.41%. From the perspective of these improvements in robustness, we see the model's ability to generalize to unseen inputs, giving indicative evidence for why rare features are handled well by our proposed methods.

Case Study. The benefits of such robustness improvement are particularly pronounced for small groups characterized by less frequent features and unstable performance during training. To illustrate this, we select the *male entertainment* group, which accounts for only 0.2% of the total users, as a case study (Figure 4). The figure demonstrates that normal FM training exhibits significant fluctuations, indicating the sensitivity of the data group to model updates. In contrast by incorporating annealing adaptive noise in AAFM, performance gradually converges while improving overall AUC in the later stages of training. This notable improvement in stability further confirms the enhanced robustness in small groups.

3.4 Trade-off Between Fairness and Accuracy

Fairness and accuracy often involve a trade-off, and sometimes their objectives can even be contradictory [31]. However, we argue that fairness and accuracy can find common ground with appropriate adaptive adversarial weights. We adjust the hyperparameter t to control the scale of λ_k in Figure 4. As t increases, we observe that fairness achieves the best results when t takes on the values of 100, 200, and 100 for MovieLens, Yelp, and Pinterest, respectively. On the other hand, accuracy reaches its peak when t is set to 50, 200, and 100. Notably, these two objectives are mostly aligned, suggesting that the improvement in fairness mainly stems from the enhanced accuracy of small groups rather than compromising the performance of larger groups (which could significantly reduce overall accuracy). The exception occurs in the MovieLens dataset, where there is a trade-off between the best accuracy ($t = 50$) and the best fairness $t = 100$. MovieLens contains more feature domains compared to the other two datasets. This implies a finer feature granularity and more similar joint feature statistics for samples. Larger t will magnify the differences in adversarial weights of samples that were originally similar. This will lead to a rapidly increasing amount of samples with low training weights, resulting in a more prominent overall performance drop.

4 RELATED WORK

Fairness in recommendation is a nascent but growing topic of interest [4], but hardly has a single, unique definition. The concept has been extended to cover multiple stakeholders[1, 29] and implies different trade-offs in utility. From a stakeholder perspective, fairness can be considered from both item and user aspects. *User fairness* [9, 19] expects equal recommendation quality for individual users or demographic groups, and *item fairness* [21, 41] indicates fair exposure among specific items or item groups. From an architectural perspective, there are mainly two approaches to address fairness: One method is to post-process model predictions (i.e., re-ranking) to alleviate unfairness [9, 19, 29]. The other unbiased learning method is to directly debias in the training process. Such latter methods come from two origins. Causal Embedding [5] is one way to control the embedding learning from the bias-free uniform data (e.g., by re-sampling [9]). Re-weighting [6, 38] is another method to balance the impact of unevenly distributed data during training, where the Inverse Propensity Scoring [27, 40] is a common means to measure the difference between actual and expected distributions. In this work, we generalize the problem to solve both user and item groups' unfairness, proposing an unbiased learning technique at the feature-level.

Adversarial training in recommendation helps models pursue robustness by introducing adversarial samples. One of the most effective techniques is to perturb adversarial samples by gradient-based noise (e.g., FGSM [11], PGD [22], and C&W [7]). Previous work found such noise is effective in improving recommendation accuracy, such as applying fixed FGSM on matrix factorization [15] and multiple adversarial strengths [39]. Current adversarial perturbation in recommendation systems mostly focuses on representing individual users [3, 20, 28] or items [3, 18] properly.

Adversarial training is increasingly discussed in unbiased learning approaches [33]. Recent work [28] also found adversarial perturbation could benefit under-served users. Yu et al. [37] found a positive correlation between the node representation uniformity and the debias ability, and added adversarial noise to each node in contrastive graph learning. However, they lack systematic comparison with fair recommendation baselines and overlook the flexibility of selected features. While there have been discussions in computer vision on connecting fairness and model robustness [32, 36], there is a lack of studies addressing the bridging between model robustness and the co-improvement of accuracy and fairness in recommendation tasks.

5 CONCLUSION AND FUTURE WORK

In this work, we propose a feature-oriented fairness approach, employing feature-unbiased learning for simultaneous improvement of fairness and accuracy. We address imbalanced performance among feature-based groups by identifying its root causes in feature frequency and combination variety. Our proposed Adaptive Adversarial Factorization Machine (AAFM) uses adversarial perturbation to mitigate this imbalance during training, applying varied perturbation levels to different features and adversarial training weights to different samples. This adaptive approach effectively enhances the generalizability of feature representation. Our experimental results show that AAFM outperforms in fairness, accuracy, and robustness, highlighting its potential as an effective approach for further study in this field.

While AAFM introduces adversarial training to unbiased learning, there are still many possible refinements. For example, AAFM defaults to using random negative sampling, which biases toward the majority of users/items features. How to balance the impact of such biased negative sampling in different groups deserves future study. It will also be valuable to further investigate the effectiveness of different adversaries (e.g., PGD [22], or C&W [7]) on more complex neural recommendation backbones.

A DERIVATION OF ADVERSARIAL PERTURBATION

We present the mathematical derivation of the adversarial perturbation for feature embedding e_i, and explain the reasoning behind utilizing combination variety as the bias parameter to achieve balance.

By applying the Chain Rule, we express the adversarial feature perturbation $\Delta_{adv}^{e_i}$ in the following manner:

$$\begin{aligned}\Delta_{adv}^{x_i} &= \epsilon \cdot \frac{\partial \mathcal{L}(\hat{y}, y)/\partial e_i}{\|\partial \mathcal{L}(\hat{y}, y)/\partial e_i\|} \\ &= \epsilon \cdot \frac{\left(-\frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}}\right) \cdot \partial \hat{y}/\partial e_i}{\|\left(-\frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}}\right) \cdot \partial \hat{y}/\partial e_i\|}\end{aligned} \tag{9}$$

y can take on values of either 0 or 1, hence we can simplify the above expression as:

$$\Delta_{adv}^{e_i} = -\epsilon \cdot \frac{\partial \hat{y}/\partial e_i}{\| \partial \hat{y}/\partial e_i\|} \tag{10}$$

Given that we have chosen FM as our prediction model, we can calculate the partial derivative of $\hat{y}$ with respect to the feature embedding e_i as follows:

$$\begin{aligned}\frac{\partial \hat{y}}{\partial e_i} &= w_i + \frac{\partial}{\partial e_i}\left[\frac{1}{2}\sum_{f=1}^{d}\left(\sum_{i=1}^{n} v_{i,f} e_i\right)^2 - \frac{1}{2}\sum_{f=1}^{d}\left(\sum_{i=1}^{n} v_{i,f}^2 e_i^2\right)\right] \\ &= w_i + \frac{1}{2}\left[\frac{\partial}{\partial e_i}\sum_{f=1}^{d}\left(v_{i,f}^2 e_i^2 + 2\sum_{j=1}^{n} v_{i,f} v_{j,f} e_i e_j\right) - \frac{\partial}{\partial e_i}\sum_{f=1}^{d}\left(\sum_{i=1}^{n} v_{i,f}^2 e_i^2\right)\right] \\ &= w_i + \sum_{f=1}^{d}\sum_{\substack{j=1 \\ j\neq i}}^{n} v_{i,f} v_{j,f} e_j \\ &= w_i + \sum_{\substack{j=1 \\ j\neq i}}^{n} < v_i, v_j > e_j\end{aligned} \tag{11}$$

where $\frac{\partial}{\partial e_i}\sum_{f=1}^{d}\left(\sum_{i=1}^{n} v_{i,f}^2 e_i^2\right)$ can be reduced and vector multiplication involved is performed element-wise. Substituting $\partial \hat{y}/\partial e_i$ into $\Delta_{adv}^{e_i}$, we thus have:

$$\Delta_{adv}^{e_i} = -\epsilon \cdot \frac{w_i + \sum_{\substack{j=i \\ j\neq i}}^{n} \left\langle v_i, v_j \right\rangle e_j}{\left\| w_i + \sum_{\substack{j=1 \\ j\neq i}}^{n} \left\langle v_i, v_j \right\rangle e_j \right\|} \tag{12}$$

The addition of this adversarial perturbation to the original embedding e_i utilizes the interacted feature embeddings e_j weighted by the pair-wise interaction weights $\left\langle v_i, v_j \right\rangle$ to enhance the representation of embedding e_i.

Hence, we can find the perturbation on e_i is controlled by the strength ϵ, and the perturbation direction is influenced by w_i and v. There exists a direct relationship between w_i and the perturbation direction. As for $\left\langle v_i, v_j \right\rangle$, being the second-order interaction parameters, their pairwise combinations determine the impact of other e_j values on the perturbation direction. When w_i is held constant, a larger variety of feature combinations results in a more diverse range of perturbation directions. Consequently, in our work, we assign a smaller perturbation strength to balance between the influence of adversaries and their impact.

REFERENCES

[1] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. 2020. The connection between popularity bias, calibration, and fairness in recommendation. In *Fourteenth ACM Conference on Recommender Systems*. 726–731.

[2] Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W Bruce Croft. 2018. Unbiased learning to rank with unbiased propensity estimation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 385–394.

[3] Ghazaleh Beigi, Ahmadreza Mosallanezhad, Ruocheng Guo, Hamidreza Alvari, Alexander Nou, and Huan Liu. 2020. Privacy-aware recommendation with private-attribute protection using adversarial learning. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 34–42.

[4] Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H Chi, et al. 2019. Fairness in recommendation ranking through pairwise comparisons. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2212–2220.

[5] Stephen Bonner and Flavian Vasile. 2018. Causal embeddings for recommendation. In *Proceedings of the 12th ACM conference on recommender systems*. 104–112.

[6] Robin Burke, Nasim Sonboli, Masoud Mansoury, and Aldo Ordoñez-Gauger. 2017. Balanced neighborhoods for fairness-aware collaborative recommendation. (2017).

[7] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*. IEEE, 39–57.

[8] Huiyuan Chen and Jing Li. 2019. Adversarial tensor factorization for context-aware recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 363–367.

[9] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. 2019. Fairness-aware ranking in search & recommendation systems with application to linkedin talent search. In *Proceedings of the 25th acm sigkdd international conference on knowledge discovery & data mining*. 2221–2231.

[10] Irving John Good. 1952. Rational decisions. *Journal of the Royal Statistical Society: Series B (Methodological)* 14, 1 (1952), 107–114.

[11] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[12] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).

[13] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 355–364.

[14] Xiangnan He, Xiaoyu Du, Xiang Wang, Feng Tian, Jinhui Tang, and Tat-Seng Chua. 2018. Outer product-based neural collaborative filtering. *arXiv preprint arXiv:1808.03912* (2018).

[15] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. 2018. Adversarial personalized ranking for recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 355–364.

[16] Hengchang Hu, Wei Guo, Yong Liu, and Min-Yen Kan. 2023. Adaptive Multi-Modalities Fusion in Sequential Recommendation Systems. In *Proceedings of the 32nd ACM International Conference on Information & Knowledge Management*.

[17] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2013. Efficiency Improvement of Neutrality-Enhanced Recommendation.. In *Decisions@ RecSys*. Citeseer, 1–8.

[18] Adit Krishnan, Ashish Sharma, Aravind Sankar, and Hari Sundaram. 2018. An adversarial approach to improve long-tail performance in neural collaborative filtering. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 1491–1494.

[19] Yunqi Li, Hanxiong Chen, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2021. User-oriented Fairness in Recommendation. In *Proceedings of the Web Conference 2021*. 624–632.

[20] 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*. 1054–1063.

[21] Yunqi Li, Yingqiang Ge, and Yongfeng Zhang. 2021. Tutorial on Fairness of Machine Learning in Recommender Systems. SIGIR.

[22] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083* (2017).

[23] Dino Pedreschi, Salvatore Ruggieri, and Franco Turini. 2009. Measuring discrimination in socially-sensitive decision records. In *Proceedings of the 2009 SIAM international conference on data mining*. SIAM, 581–592.

[24] Yiding Ran, Hengchang Hu, and Min-Yen Kan. 2022. PM K-LightGCN: Optimizing for Accuracy and Popularity Match in Course Recommendation. In *Workshop of Multi-Objective Recommender Systems (MORS'22), in conjunction with the 16th ACM Conference on Recommender Systems, RecSys*, Vol. 22. 2022.

[25] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International conference on data mining*. IEEE, 995–1000.

[26] Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. 2020. Unbiased recommender learning from missing-not-at-random implicit feedback. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 501–509.

[27] Masahiro Sato, Sho Takemori, Janmajay Singh, and Tomoko Ohkuma. 2020. Unbiased learning for the causal effect of recommendation. In *Fourteenth ACM Conference on Recommender Systems*. 378–387.

[28] Pannaga Shivaswamy and Dario Garcia-Garcia. 2022. Adversary or Friend? An adversarial Approach to Improving Recommender Systems. In *Proceedings of the 16th ACM Conference on Recommender Systems*. 369–377.

[29] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of exposure in rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2219–2228.

[30] Wenjie Wang, Fuli Feng, Xiangnan He, Xiang Wang, and Tat-Seng Chua. 2021. Deconfounded recommendation for alleviating bias amplification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1717–1725.

[31] Tianxin Wei, Fuli Feng, Jiawei Chen, Ziwei Wu, Jinfeng Yi, and Xiangnan He. 2021. Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender system. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1791–1800.

[32] Tong Wei, Jiang-Xin Shi, Yu-Feng Li, and Min-Ling Zhang. 2022. Prototypical Classifier for Robust Class-Imbalanced Learning. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 44–57.

[33] Chuhan Wu, Fangzhao Wu, Xiting Wang, Yongfeng Huang, and Xing Xie. 2021. Fairrec: fairness-aware news recommendation with decomposed adversarial learning. AAAI.

[34] 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*. 2657–2662.

[35] Xin Xin, Bo Chen, Xiangnan He, Dong Wang, Yue Ding, and Joemon Jose. 2019. CFM: Convolutional Factorization Machines for Context-Aware Recommendation.. In *IJCAI*, Vol. 19. 3926–3932.

[36] Han Xu, Xiaorui Liu, Yaxin Li, Anil Jain, and Jiliang Tang. 2021. To be robust or to be fair: Towards fairness in adversarial training. In *International Conference on Machine Learning*. PMLR, 11492–11501.

[37] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and QVH Nguyen. 2022. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *SIGIR*.

[38] Jiangxing Yu, Hong Zhu, Chih-Yao Chang, Xinhua Feng, Bowen Yuan, Xiuqiang He, and Zhenhua Dong. 2020. Influence function for unbiased recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1929–1932.

[39] Feng Yuan, Lina Yao, and Boualem Benatallah. 2019. Adversarial collaborative neural network for robust recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1065–1068.

[40] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. Causal intervention for leveraging popularity bias in recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 11–20.

[41] Ziwei Zhu, Jianling Wang, and James Caverlee. 2021. Fairness-aware Personalized Ranking Recommendation via Adversarial Learning. *arXiv preprint arXiv:2103.07849* (2021).