BSL: Understanding and Improving Softmax Loss for Recommendation

Junkang Wu[1], Jiawei Chen[2*], Jiancan Wu[1*], Wentao Shi[1], Jizhi Zhang[1], Xiang Wang[1†]

[1] *MoE Key Laboratory of Brain-inspired Intelligent Perception and Cognition,*
University of Science and Technology of China, Hefei, China
[2] *The State Key Laboratory of Blockchain and Data Security, Zhejiang Unversity*
{jkwu0909, wujcan, xiangwang1223}@gmail.com, sleepyhunt@zju.edu.cn,
{shiwentao123, cdzhangjizhi}@mail.ustc.edu.cn

Abstract—**Loss functions steer the optimization direction of recommendation models and are critical to model performance, but have received relatively little attention in recent recommendation research. Among various losses, we find Softmax loss (SL) stands out for not only achieving remarkable accuracy but also better robustness and fairness. Nevertheless, the current literature lacks a comprehensive explanation for the efficacy of SL.**

Toward addressing this research gap, we conduct theoretical analyses on SL and uncover three insights: 1) Optimizing SL is equivalent to performing Distributionally Robust Optimization (DRO) on the negative data, thereby learning against perturbations on the negative distribution and yielding robustness to noisy negatives. 2) Comparing with other loss functions, SL implicitly penalizes the prediction variance, resulting in a smaller gap between predicted values and and thus producing fairer results.

Building on these insights, we further propose a novel loss function Bilateral SoftMax Loss (BSL) that extends the advantage of SL to both positive and negative sides. BSL augments SL by applying the same Log-Expectation-Exp structure to positive examples as is used for negatives, making the model robust to the noisy positives as well. Remarkably, BSL is simple and easy-to-implement — requiring just one additional line of code compared to SL. Experiments on four real-world datasets and three representative backbones demonstrate the effectiveness of our proposal. The code is available at https://github.com/junkangwu/BSL.

Index Terms—**Recommendation System, Robustness, Distributionally Robust Optimization**

I. INTRODUCTION

Recommender Systems (RS) have become a leading technology addressing information overload. Owing to the easy availability of user behaviors (*e.g.,* click and purchase), collaborative filtering (CF) with implicit feedback emerges as an effective solution. Recent years have witnessed the flourishing publications on designing model architecture, spanning matrix factorization [1], autoencoders[2; 3; 4], and advanced graph neural networks [5; 6; 7]. In contrast, relatively less work focuses on loss functions, which play critical roles in steering CF.
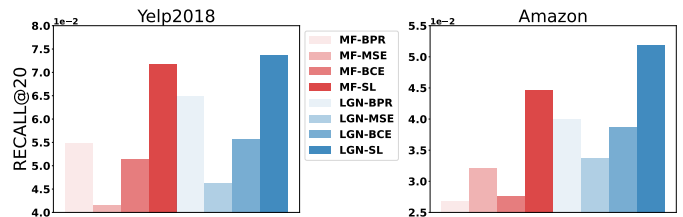
Fig. 1: Performance comparison of different loss functions based on MF (red bars) and LightGCN (blue bars) on the Yelp2018 and Amazon datasets. SL consistently achieves superior performance over other losses by a significant margin ($> 15\%$ gain) across models and datasets. Additional results on other datasets and architectures (presented in Table I) exhibit similar patterns, highlighting the effectiveness of SL for collaborative filtering.

Existing loss functions for CF with implicit feedback can be mainly categorized into three types: 1) Pointwise loss (*e.g.,* BCE [8], MSE [9]) casts the problem into a classification or regression, promoting model predictions close to the labels; 2) Pairwise loss (*e.g.,* BPR [10]) encourages a higher score for positive item compared to its negative counterpart; 3) Softmax loss (SL) [11; 12] normalizes model predictions over items into a multinomial distribution, optimizing the probability of positive instances versus negative ones.

Through empirical evaluation of several losses paired with two representative recommenders (*i.e.,* - MF[13] and LightGCN[5]) on real-world datasets, we find SL consistently and significantly outperforms other losses, with over 15% higher accuracy, as is shown in Figure 1. Beyond accuracy, we find SL also exhibits better performance in terms of robustness and fairness (*cf.* , Figure 4). This impressive result reveals the superiority of SL, which motivates us to explore a fundamental question — **what are the underlying reasons behind SL's effectiveness?** Prior work [11; 14; 15] has explored the properties of SL in recommendations but with several limitations. 1) The connections between SL and other losses, as well as how SL improves upon them, remain unrevealed; 2) The fairness of SL was attributed to the popularity-based negative sampling strategy, rather than the loss itself, whereas we find uniform sampling also yields fairness.

In this work, we strive to conduct in-depth analysis of

SL through the lens of *Distributionally Robust Optimization* (DRO) [16; 17; 18]. DRO is an advanced stochastic optimization framework that learns models robust to perturbations in the data distribution. Remarkably, our rigorous theoretical analyses reveal the following key insights:

- **Optimizing SL is equivalent to performing DRO on the vanilla pointwise loss**, where certain perturbations are added to the negative item distribution. In fact, noisy (*i.e.,* false) negative instances are common in RS due to factors, such as user unawareness, low ranking position or popularity of items [19; 20]. Such noisy negative data inevitably causes distribution shift between sampled data and the true one, misleading learning procedure and deteriorating performance. In contrast, the DRO nature of SL imparts robustness to negative noise, and hence yielding better performance.

- **SL introduces an implicit regularizer that controls the prediction variance on negative instances.** Due to the long tail distribution exhibited in recommendation data, models tend to overly favor popular items while neglecting others, resulting in notorious popularity bias. By penalizing the variance of model predictions, SL would reduce the prediction discrepancy between popular and unpopular items, leading to fairer recommendation results.

The above analyses not only explain the strengths of SL, but inspire us an enhanced loss design. Intuitively, robustness to only negative noise is insufficient, as positive instances also contain unreliability due to the factors like clickbait [21] or user conformity [22]. In light of this, we propose a novel loss function named ***Bilateral Softmax Loss*** (BSL), which extends the advantage of SL to both positive and negative sides. Specifically, we augment the positive loss of SL to a Log-Expectation-Exp structure mirroring the negative part. This bilateral architecture provides robustness to noise on both positives and negatives. Remarkably, our BSL is easy to implement, requiring only one line of code amended as compared with SL.

Contributions. In summary, our contributions are:

- We prove that optimizing SL is equivalent to performing DRO, which provides novel theoretical insights into SL's strengths on both robustness and fairness.
- We propose a novel Bilateral Softmax Loss, which is easily implemented while achieving bilateral robustness to both positive and negative noise.
- We conduct extensive experiments on four real-world datasets and three representative backbones to demonstrate the effectiveness of BSL.

The remainder of this paper is structured as follows. Section II provides background on recommender systems and existing loss functions, followed by a brief introduction on distributionally robust optimization. Our analysis on SL through theoretical proofs and empirical experiments is presented in Section III. Section IV identifies SL's weaknesses and proposes our positive denoising loss — Bilateral Softmax Loss. The effectiveness of our proposal is evaluated through extensive experiments in Section V. Finally, Section VI and VII review related work on robust recommendation and draw conclusion.

II. PRELIMINARIES

In this section, we present the background of Recommender System and Distributionally Robust Optimization.

A. Recommender System

Suppose we have a recommender system (RS) with a user set $\mathcal{U}$ and an item set $\mathcal{I}$. The historical user-item interactions can be expressed by a matrix $\mathbf{R} \in \{0,1\}^{|\mathcal{U}| \times |\mathcal{I}|}$, whose element r_{ui} represents whether a user u has interacted (*e.g.,* click) with an item i. For convenience, we define the positive (or negative) items for user u as $\mathcal{S}_u^+ \equiv \{i \in \mathcal{I} | r_{ui} = 1\}$ (or $\mathcal{S}_u^- \equiv \{i \in \mathcal{I} | r_{ui} = 0\}$). The goal of a RS is to recommend new items for each user that he may be interested in.

Regarding model optimization, existing work typically frames the task as stardard supervised learning. Let P_u^+ and P_u^- denote the distribution of positive and negative training instances respectively. P_u^+, is usually set to a uniform distribution over $\mathcal{S}_u^+$, while P_u^- can be configured either as a uniform distribution over $\mathcal{S}_u^-$ or using population-based sampling from $\mathcal{S}_u^-$. There are mainly three types of loss functions in RS [23]:

- Pointwise loss frames the problem of recommendation from implicit feedback as a classification or regression problem, encouraging model predictions close to the corresponding labels:

$$\mathcal{L}_{Pointwise}(u) = -\mathbb{E}_{i \sim P_u^+}[\phi(f(u,i))] + c\mathbb{E}_{j \sim P_u^-}[\psi(f(u,j))] \quad (1)$$

where $f(u,i)$ is the predicted score, c balances positive and negative contributions, and $\phi(\cdot)$ and $\psi(\cdot)$ are functions adapted for different loss choices. For example, in terms of BCE and MSE, we have:

$$\begin{cases} \phi(f(u,i))_{BCE} = \log \sigma(f(u,i)) \\ \psi(f(u,j))_{BCE} = -\log(1 - \sigma(f(u,j))) \end{cases}$$
$$\begin{cases} \phi(f(u,i))_{MSE} = -||f(u,i) - 1||^2 \\ \psi(f(u,j))_{MSE} = ||f(u,j) - 0||^2 \end{cases} \quad (2)$$

- Pairwise loss enforces higher scores for positive items versus the negative counterparts:

$$\mathcal{L}_{Pairwise}(u) = -\mathbb{E}_{i \sim P_u^+, \ j \sim P_u^-}[\varphi(f(u,i) - f(u,j))] \quad (3)$$

For the classical BPR loss, $\varphi(.)$ can is set to log-Sigmoid function.

- Softmax Loss normalizes model predictions into a multinomial distribution and optimizes the probability of positive instances over negative ones[1]:

$$\mathcal{L}_{SL}(u) = -\mathbb{E}_{i \sim P_u^+}\left[\log \frac{\exp\left(f(u,i)/\tau\right)}{N^- \mathbb{E}_{j \sim P_u^-}\left[\exp\left(f(u,j)/\tau\right)\right]}\right] \tag{4}$$

where N^- denotes the number of sampled negative instances. Note that SL (*cf.* Eq.(4)) can be expressed by a combination of positive and negative parts:

$$\mathcal{L}_{SL}(u) = \underbrace{-\mathbb{E}_{i \sim P_u^+}[f(u,i)]}_{\text{Positive Part}} + \underbrace{\tau \log \mathbb{E}_{j \sim P_u^-}[\exp(f(u,j)/\tau)]}_{\text{Negative Part}} \tag{5}$$

Upon comparison between the $\mathcal{L}_{SL}$ and the point-wise loss, we discern a significant distinction primarily within the negative part — SL utilizes a specific Log-Expectation-Exp structure (*i.e.,* $\log \mathbb{E}[\exp(.)]$) in the negative part. In the forthcoming section, we will elucidate the advantages inherent to this structure. Notably, an additional hyperparmeter τ is always introduced in SL. The role of this hyperparameter within the context of SL will also be explored.

B. Distributionally Robust Optimization

The success of machine learning relies on the assumption that the test distribution matches the training distribution (*aka.* , iid assumption). However, this assumption fails to hold in many real-world applications, leading to sub-optimal performance. Distributionally Robust Optimization (DRO) [16; 17; 18] addresses this by demanding models to perform well not only on the observed data distribution, but over a range of distribution with perturbations. Formally, DRO plays a min-max game: it first identifies the most difficult distribution (*aka.* worst-case distribution) from the uncertainty set $\mathbb{P}$, and then optimizes the model under that distribution:

$$\hat{\theta} = \underset{\theta}{\arg\min}\{\max_{P \in \mathbb{P}} \mathbb{E}_{x \sim P}[\mathcal{L}(x;\theta)]\}$$
$$\mathbb{P} = \{P \in \mathbb{D} : D(P, P_o) \leq \eta\} \tag{6}$$

where $\mathbb{D}$ denotes the set of all distributions, $D(.,.)$ measures the distance between the original distribution P_o and the perturbed distribution P; η denotes the distance threshold. The uncertainty set $\mathbb{P}$ includes the possible distributions to optimize well over. Hence, the choice of D and η are critical as they control the uncertainty set $\mathbb{P}$.

III. UNDERSTANDING SL FROM DRO

In this section, we conduct thorough analyses on SL with both theoretical proofs and empirical experiments.

---

[1]Note that here we simply remove the positive term in the denominator as it makes negligible contribution when N^- takes a relatively large value. Also, recent work suggests that this treatment could boost the embedding uniformity and empirically yield slightly better performance [24].

A. Theoretical Analyses

Lemma 1. *Optimizing Softmax loss (*cf. *Equation 4) is equivalent to performing Distributionally Robust Optimization over the original point-wise loss,* i.e., *optimizing:*

$$\mathcal{L}_{DRO}(u) = -\mathbb{E}_{i \sim P_u^+}[f(u,i)] + \max_{P \in \mathbb{P}} \mathbb{E}_{j \sim P}[f(u,j)]$$
$$\mathbb{P} = \{P \in \mathbb{D} : D_{KL}(P, P_u^-) \leq \eta\} \tag{7}$$

where $\mathbb{P}$ denotes the uncertainty set of the negative data distribution; D_{KL} denotes the KL-divergence from distribution P to the original P_u^-; η denotes the robustness radius controlling how P deviates from P_u^-.

Proof. By comparing the Softmax loss (Equation (4)) with $\mathcal{L}_{DRO}(u)$ (Equation (7)), we can find the difference mainly lies on the negative part. In fact, the equivalence of the Log-Expectation-Exp structure and KL-based DRO objective has been revealed by recent work on other areas [18] [25]. To ensure the completeness of the article, here we also give the complete proof by referring to these work.

Let $L(j) = P(j)/P_u^-(j)$. Note that the KL-divergence between P and $P_u^-(j)$ is constrained, and thus $L(.)$ is fine definition. For brevity, we usually short $L(j)$ as L. Also, we define a specific convex function $g(x) = x \log x$. Then the KL divergence $D_{KL}(P, P_u^-)$ can be written as $\mathbb{E}_{j \sim P_u^-}[g(L)]$. Equation (7) can be reformulated as follow:

$$-\mathbb{E}_{i \sim P_u^+}[f(u,i)] + \max_{L} \mathbb{E}_{j \sim P_u^-}[f(u,j)L]$$
$$\text{subject to } \mathbb{E}_{j \sim P_u^-}[g(L)] \leq \eta, \quad \mathbb{E}_{P_u^-}[L] = 1 \tag{8}$$

Considering the convex of $\mathbb{E}_{P_u^-}g(L)$ and $\mathbb{E}_{P_u^-}[f(u,j)L]$ on L, implying it is a convex optimization problem, we use the Lagrangian function to solve it:

$$-\mathbb{E}_{i \sim P_u^+}[f(u,i)] + \min_{\alpha \geq 0, \beta} \max_{L}\{\mathbb{E}_{j \sim P_u^-}[f(u,j)L -$$
$$\alpha(\mathbb{E}_{j \sim P_u^-}[g(L)] - \eta) + \beta(\mathbb{E}_{j \sim P_u^-}[L] - 1)\} \tag{9}$$

where α and β are Lagrangian multipliers. Now we focus on eliminate L and β. In fact, we have:

$$\min_{\beta} \max_{L}\{\mathbb{E}_{j \sim P_u^-}[f(u,j)L -$$
$$\alpha(\mathbb{E}_{j \sim P_u^-}[\phi(L)] - \eta) + \beta(\mathbb{E}_{P_u^-}[L] - 1)\}$$
$$= \min_{\beta}\left\{\alpha\eta - \beta + \alpha \max_{L}\{\mathbb{E}_{P_u^-}[\frac{f(u,j)+\beta}{\alpha}L] - \mathbb{E}_{P_u^-}[g(L)]\}\right\}$$
$$= \min_{\beta}\left\{\alpha\eta - \beta + \alpha\mathbb{E}_{P_u^-}[\max_{L}\{\frac{f(u,j)+\beta}{\alpha}L - g(L)\}]\right\}$$
$$= \min_{\beta}\left\{\alpha\eta - \beta + \alpha\mathbb{E}_{P_u^-}[g^*(\frac{f(u,j)+\beta}{\alpha})]\right\} \tag{10}$$

The last equality held since the definition of convex conjugate $g^*(y) = \max_x\{xy - g(x)\}$. Given $g(x) = x\log x$, we have $g^*(x) = e^{x-1}$:

$$\min_\beta \left\{ \alpha\eta - \beta + \alpha\mathbb{E}_{P_u^-}[g^*(\frac{f(u,j)+\beta}{\alpha})]\right\}$$
$$= \min_\beta \left\{ \alpha\eta - \beta + \alpha\mathbb{E}_{P_u^-}[e^{\frac{f(u,j)+\beta}{\alpha}-1}]\right\} \quad (11)$$
$$= C + \alpha\log\mathbb{E}_{P_u^-}[e^{\frac{f(u,j)}{\alpha}}]$$

where $C = \alpha\eta$ denotes a constant. The Last equality holds due to the optimal solution found for β. By plugging Eq. (11) into problem (9), we get the overall optimal result as follows:

$$-\mathbb{E}_{P_u^+}[f(u,i)] + \min_{\alpha\geq 0,\beta}\max_L\{\mathbb{E}_{P_u^-}[f(u,j)L-$$
$$(\mathbb{E}_{P_u^-}[\phi(L)] - \eta) + \beta(\mathbb{E}_{P_u^-}[L] - 1)\}$$
$$= \min_{\alpha\geq 0}\left\{-\mathbb{E}_{P_u^+}[f(u,i)] + C + \alpha\log\mathbb{E}_{P_u^-}[\exp(f(u,j)/\alpha)]\right\}$$
$$= -\mathbb{E}_{P_u^+}[f(u,i)] + \alpha^*\log\mathbb{E}_{P_u^-}[\exp(f(u,j)/\alpha^*)] + C$$
$$(12)$$

Where the optimal α^* is determined by the hyperparameter η. We may reversely treat α^* as a surrogate hyperparameter and consider η as a function of α^*. In this way, by comparing Eq.(12) with SL and removing constant terms that is irrelevant with $f(u,i)$, we can conclude that optimizing SL is equivalence to optimizing $\mathcal{L}_{DRO}(u)$ (Eq.(7)) and α is identical to τ.

$\square$

Remark 1 (Capability of Robustness) This lemma points out the essence and superiority of SL compared to other loss functions. In RS, the presence of noise inevitably leads to a significant gap between the distribution of sampled items and the true negative distribution (*aka.* distributional shifts). Due to the intrinsic connection to DRO, SL explicitly considers the uncertainty of distributions to seek a model that performs well against distribution perturbation. Specifically, if the uncertainty set $\mathcal{P}$ is appropriately selected and the true negative data distribution is contained in the built uncertainty set, the robustness of the model could be guaranteed under distributional shifts, and performance superiority would begin to emerge.

Lemma 2 (Fairness). *With the Lemma 1, we can approximate the Softmax loss as follow:*

$$\mathcal{L}_{SL}(u) = -\mathbb{E}_{i\sim P_u^+}[f(u,i)] + \max_{P\in\mathbb{P}}\mathbb{E}_{j\sim P}[f(u,j)]$$
$$\approx -\mathbb{E}_{i\sim P_u^+}[f(u,i)] + \mathbb{E}_{j\sim P_u^-}[f(u,j)] \quad (13)$$
$$+\frac{\mathbb{V}[f(u,j)]}{2\tau} + \tau\eta + o_\infty(1/\tau)$$

where $\mathbb{V}[f(u,j)]$ denotes the variance of score $f(u,j)$ under the distribution P_u^-.

Proof. As the optimal value of $\max_{P\in\mathbb{P}}\mathbb{E}_{j\sim P}[f(u,j)]$ can be written as $\tau\log\mathbb{E}_{j\sim P_u^-}[e^{f(u,j)/\tau}] + \tau\eta$. To ease notations, we denote $\Omega(\gamma) = \log\mathbb{E}_{j\sim P_u^-}[e^{f(u,j)\gamma}]$ and $\gamma = 1/\tau$. By

leveraging a second-order Taylor expansion around 0 on $\Omega(\gamma)$, we have:

$$\Omega(\gamma) = \gamma\mathbb{E}_{j\sim P_u^-}[f(u,j)] + \frac{\gamma^2}{2}\mathbb{V}[f(u,j)] + o_0(\gamma^2) \quad (14)$$

Thereafter, we plug $\tau = 1/\gamma$ into $\Omega(\gamma)$, which finishes the proof:

$$\max_{P\in\mathbb{P}}\mathbb{E}_{j\sim P}[f(u,j)] = \tau\log\mathbb{E}_{j\sim P_u^-}[e^{f(u,j)/\tau}] + \tau\eta$$
$$\approx \mathbb{E}_{j\sim P_u^-}[f(u,j)] + \frac{\mathbb{V}[f(u,j)]}{2\tau} + o_\infty(1/\tau) + \tau\eta$$
$$(15)$$

$\square$

Remark 2 (Capability of Fairness) This lemma provides the insight that SL pursues the fairness of the recommendation system through regularization on the variance of negative samples $f(u,j)$. Recent research [26] has indicated that the variance penalty corresponds to fairness. From Equation (13), we can conclude that SL implicitly introduces a regularizer that penalizes the variance of the model predictions on the negative instances. It is important to note that in a typical recommender systems, recommendation model is prone to give extensive higher scores on popular items than unpopular ones, incurring popularity bias [27]. SL could mitigate this effect to a certain degree, as it inherently reduces the variance of model predictions, which implies that the predictions are likely to be more uniform. Consequently, the discrepancy in model predictions between popular and unpopular items is diminished, leading to more fair recommendation results. Unpopular items get more opportunity in recommendation, and thus we observe better NDCG@20 for those unpopular groups.

Corollary III.1 (The optimal α^* - Lemma 5 of [28]). *The value of the optimal α^* (i.e., temperature τ) can be approximated as follows:*

$$\tau^* \approx \sqrt{\frac{\mathbb{V}[f(u,j)]}{2\eta}}. \quad (16)$$

where $\mathbb{V}[f(u,j)]$ denotes the variance of score $f(u,j)$ under the distribution P_u^-.

Remark 3 (Role of the Temperature τ). From Lemma 1, we can see that τ is not just a heuristic design, but can be interpreted as a Lagrange multiplier in solving DRO optimization problem. In additional, Corollary III.1 indicates that τ is determined by the η and the variance of $f(u,j)$. It means that tuning the hyperparameter τ in SL is essentially equivalent to adjusting the robustness radius.

The above insight also could help us to better understand how temperature affects model performance. When the value of τ is excessive large, it creates a limited uncertainty set which fails to encompass the true negative distribution, resulting in a lack of robustness (*cf.* Figure 2 left). As τ decreases, the robustness radius η grows, the model is exposed to more challenging distributions. Given that the uncertainty set is more prone to encompass the true distribution, there is a potential
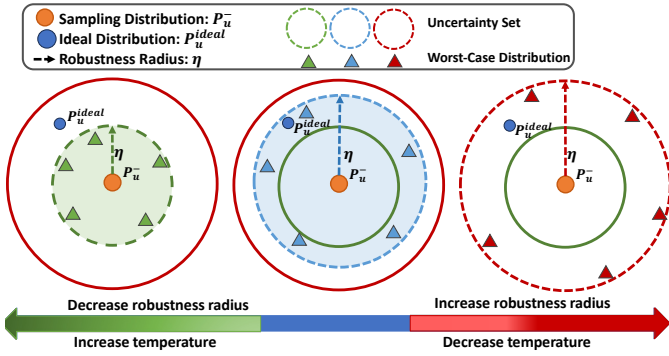
Fig. 2: The relationships between τ, η and uncertainty set

Fig. 3: Robustness Analysis. (a) **Left**: we show the NDCG of SL with different noise ratios across different temperatures τ. (b) **Right**: we show the distribution of robustness radius η with the best choice of τ across different noise ratios.

to enhance both the robustness and accuracy of the model. However, a continual decrease of τ would also increase the risk of implausible worst distribution. Typically, when η is too large, the worst-case distribution in DRO may not be the desired distribution but rather uninformative one hurting model performance (*cf.* Figure 2 right). These two aspects create trade-off of selecting τ. we will empirical verify it in subsection III-B2.

B. Empirical Analysis

In this section, we conduct the following empirical experiments to understand the essence of SL. The detailed experimental setup refers to section V. Here We take Yelp2018 dataset [5] as an example, and could draw similar conclusions from other datasets.

1) Experimental design.: To understand the above theories, here we conduct four experiments. **(1)** We record the performance of SL under different choices of τ and noise level [2] (r_{noise}). (Figure 3a). The higher value of r_{noise}, the more noise in the negative data. It can be implemented during negative sampling, where a certain portion of positive instances are sampled as negatives. **(2)** We grid search the best τ for different scenarios and visualize the distribution of η according to Equation (16) (Figure 3b). **(3)** To verify fairness, we show SL's performance on different item groups with different popularity (Figure 4a). **(4)** We also study the relationship between the prediction scores $f(u, j)$ and weights $P^*(j)$ in DRO. The weights can be understood as the sampling probability of an item in the worst-case P^* (Figure 4b)[3].

2) Exploring robustness **w.r.t.** τ*:* Figure 3a shows the effect of changing the temperature on the model performance. As can be seen, the performance grows with the increase of τ at the beginning, while drops with further increase. This phenomenon is as expected. An excessively high value of τ leads to insufficient perturbations in the model adversarial training, resulting in a lack of robustness and subpar

generalization performance. An overly small τ might also degrade performance due to the increasing risk of exposure to implausible and abnormal worst-case distributions. Therefore, it is important to select an appropriate value of τ to ensure optimal performance of the model.

From Figure 3b, we find that when we search for the best τ at each noise level and record the corresponding distribution of η, η actually rises with the increase of false negative samples. It is consistent with our intuition, more noisy data requires a larger robustness radius. However, as shown in Figure 3a, when the noise rate $(0.0 \rightarrow 3.0)$ enlarges, the best τ for SL gradually increases $(0.10 \rightarrow 0.12)$, which is against our intuition that small τ represents larger η to obtain strong robustness and superior performance. The contradiction is that the optimal τ is also correlated with the variance of $f(u, i)$. Although the robustness radius increases, the cases with more noisy samples would have a larger variance and thus incur a larger value of optimal τ, which is consistent with Corollary III.1.

3) Exploring worst-case distribution.: To verify the advantages of SL that relieves distribution shifts in negative sample distribution, we randomly choose one batch of training data and record the negative sample score and respective worst-case P^* in Equation (7). From Figure 4b, we observe that SL places more emphasis on hard negative samples. Meanwhile, if we decrease the value of τ (*e.g.,* τ=0.09), the distribution of weight exhibits more "extreme", which increases the difficulty in optimization. In contrast, increasing τ (*e.g.,* τ=0.11) represents a smaller robustness radius, and the distribution becomes more "gentle". We remark this phenomenon has also been observed by [11], but here we give explanation from a new perspective of DRO.

4) Exploring fairness in DRO: To verify the fairness of SL compared to other loss functions, we follow [11] and divide items into ten groups *w.r.t.* item popularity (*i.e.,* interaction frequency). The y-axis represents the cumulative score for a single group and the larger GroupID denotes the group where items are more popular. As shown in Figure 4a, SL achieves superior performance on those unpopular items comparing with traditional loss functions (BPR, MSE, BCE). We attribute this phenomenon to the regularization of the variance of negative samples (see Equation (13)). In other words, SL not only pursues accuracy but also fairness in recommendation.

---

[2]In the sampling of negative samples, r_{noise} represents the ratio of the sampling probability of positive samples to that of negative samples. Therefore, a larger value of r_{noise} indicates that more positive samples are mistakenly classified as negative samples.

[3]Here $P^*(j)$ and P^* both represent the worst case distribution on negative part. $P^* = \arg\max_{P \in \mathbb{P}} \mathbb{E}_{j \sim P}[f(u, j)]$
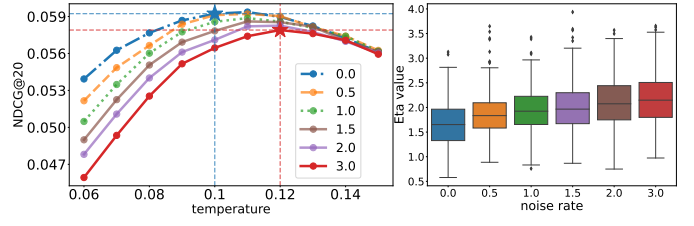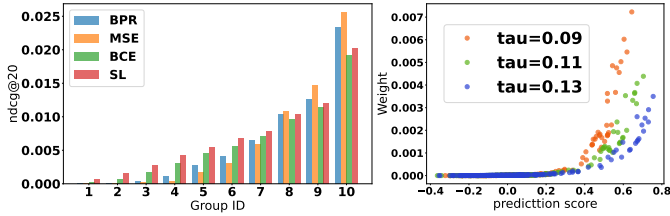
Fig. 4: (a) Left: Performance comparison over different item groups among different loss functions. (b) Right: We visualize the weight of each negative sample in DRO concerning the prediction score for different τ.
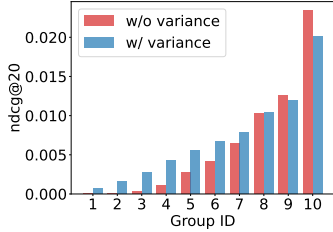

Fig. 5: Ablation study on variance terms. The item groups with larger ID suggest these items have larger popularity.

To establish a clear contrast, we conducted ablation study by removing the variance penalty term and, compared the performance of the standard SL loss with the revised loss. The results are presented in Figure 5. As can be seen, the exclusion of the variance term significantly exacerbates the unfairness of recommendation — *i.e.,* better performance on popular groups (*e.g.,* groups 8-10) while much worse performance on unpopular groups (*e.g.,* groups 1-5).

IV. PROPOSED LOSS FUNCTION

In this section, we first analyze the necessity of positive denoising and then propose a novel loss function through the lens of DRO.

A. Necessity of Positive Denoising

In the former section, we examined the superiority of SL, particularly its robustness in negative sampling. However, it is worth noting that noise can also be present in positive samples. This is particularly relevant in recommendation systems, where clicks can be influenced by various factors, such as attractive titles or high popularity, leading to a gap between the observed positive data and the user's true preference [29; 30]. Training a model directly on such noisy data may result in poor performance.

To investigate this potential yet critical issue in recommendation systems, we conducted experiments where we randomly treated a certain proportion of negative items (in accordance with the interaction frequency per user) as fake positive instances and evaluated the performance of the model using SL. As illustrated in Figure 6, a rise in the noise ratio corresponded to a decline in performance. We suspect that this phenomenon may be even more pronounced in real-world
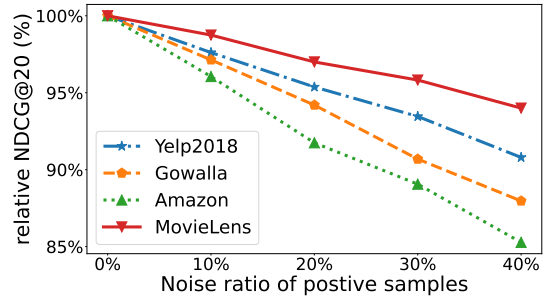

Fig. 6: The performance comparison with adding a certain ratio of noisy data into the positive data.

datasets. Therefore, we are inclined to leverage the benefits of DRO to tackle the aforementioned critical issues.

B. Bilateral SoftMax Loss

In order to enhance the robustness of the model to both noisy positive and negative data, we propose the Bilateral SoftMax Loss (BSL), which extends the benefits of SL to both the positive and negative sides. It is worth noting that SL (as seen in Eq.(4)) can be expressed as follows, comprising of two components:

$$\mathcal{L}_{SL}(u) = \underbrace{-\mathbb{E}_{i \sim P_u^+}[f(u,i)]}_{\text{Positive Part}} + \underbrace{\tau \log \mathbb{E}_{j \sim P_u^-}[\exp(f(u,j)/\tau)]}_{\text{Negative Part}} \quad (17)$$

The effectiveness of SL is attributed to its negative loss structure, which involves the Log-Expectation-Exp structure (i.e., $\log \mathbb{E}[\exp(.)]$), equivalent to conducting distributional robust optimization over the vanilla point-wise loss (as illustrated in Lemma 1). Therefore, it is reasonable to incorporate the positive loss into this structure, allowing it to benefit from the robust property. Based on this intuition, the Bilateral SoftMax Loss can be formulated as follows:

$$\mathcal{L}_{BSL}(u) = \underbrace{-\tau_1 \log \mathbb{E}_{i \sim P_u^+}[\exp(f(u,i)/\tau_1)]}_{\text{Positive Part}} + \underbrace{\tau_2 \log \mathbb{E}_{j \sim P_u^-}[\exp(f(u,j)/\tau_2)]}_{\text{Negative Part}} \quad (18)$$

It is important to note that in BSL, distinct temperatures are utilized for the positive (τ_1) and negative (τ_2) components. This is mainly due to the belief that in real-world datasets, the positive and negative data may exhibit different levels of noise.

Overall, we remark that BSL satisfies the following desirable properties:

- **Bilateral Robustness.** Thanks to the application of Log-Expectation-Exp structure on both positive and negative sides, BSL possesses the capability to maintain robustness on both sides. Unlike previous research that primarily focuses on positive sample denoising [29; 30] or negative sample debiasing [31; 32; 33], our work elegantly accomplishes

TABLE I: Statistics of the datasets.

Dataset	#Users	#Items	#Interactions	Density
Amazon	192,403	63,001	1,689,188	0.014%
Yelp2018	31,668	38,048	1,561,406	0.130%
Gowalla	29,858	40,981	1,027,370	0.084%
Movielens-1M	6,022	3,043	995,154	5.431%

both objectives. In section V, we also provide empirical validation of its robustness and superior performance.

- **Model-Agnostic.** BSL is model-agnostic and straightforward to implement. Notably, BSL only requires one additional line of code compared to the standard SL (as seen in Algorithm 2). In our experiments, we also apply BSL to common recommendation models such as MF [13], NGCF [34], LightGCN [5], SGL [35], SimSGL [36] and LightGCL [37] to validate its effectiveness.

V. EXPERIMENTS

In this section, we present various experiments to verify the superiority of our model. We aim to answer the following research questions:

- **RQ1:** How do BSL and SL perform compared to other loss functions?
- **RQ2:** Can BSL and SL maintain stability when we increase the number of noisy negative instances?
- **RQ3:** Is BSL more resilient to positive noise than SL?
- **RQ4:** How does BSL perform when hyper-parameters are changed ?

A. Experimental Setup

Datasets and Evaluation Protocol. We utilize four publicly available datasets, namely Yelp2018[5], Amazon [38], Movielens [39], and Gowalla [8], to evaluate the effectiveness of our model. To ensure a fair comparison, we adhere to the traditional settings on collaborative filtering and employ the same data split as they did [5]. Table I showcases the statistics of the aforementioned datasets. Additionally, we use conventional metrics, Recall@20 and NDCG@20, to assess the recommendation performance.

Baselines. As the suggested loss function is model-agnostic, we select three fundamental and efficient models (MF[13], NGCF[34], and LightGCN[5]) as backbones and substitute their loss functions with our proposed losses. Additionally, we implement other loss functions for comparison purposes. Moreover, to further verify the efficacy of our proposal, we compare it with various models from three different categories:

(1) *MF-based method*: ENMF[40] leverages a simple Neural Matrix Factorization architecture without sampling.

(2) *metric learning-based method*: CML[41], which combines metric learning and collaborative filtering, has the advantage of learning user-user and item-item similarity.

(3) *GCN-based methods*:

- NGCF[34] was the pioneering work that integrated the bipartite graph structure (GCN) into the embedding process.

- NIA-GCN[42] explicitly considered the relational information in the neighborhood and employed a novel neighbor-aware graph attention layer.
- LR-GCCF[43] revisited GCN based CF models from non-linear activations and proposed a residual network structure to alleviate the oversmoothing problem.
- LightGCN[5] introduced a novel GCN structure that is more concise and suitable for recommendation systems. It is also one of the most commonly used baselines in the field of recommender systems.
- SGL[35] explored self-supervised learning on the user-item graph and proposed a new learning paradigm based on LightGCN.
- Ultra-GCN[44] proposed an ultra-simplified formulation of GCNs, which skips infinite layers of message passing for efficient recommendation.
- SimpleX[45] demonstrated the significance of the choice of loss function and incorporated the cosine contrastive loss into a simple unified CF model.
- SimSGL[36] revealed that graph augmentations play a trivial role and instead added uniform noises to the embedding space to create contrastive views.
- NCL[46] focused on the influence of data sparsity in real scenarios and proposed a novel contrastive learning approach that incorporates potential neighbors into contrastive pairs.
- DGCF[47] devised a new model to disentangle the finer granularity of user intents, offering the advantages of disentangling user intents and interpretability of representations.

Parameter Settings. To ensure a fair comparison, we set the embedding size to 64 for all compared models, and the initialization is unified using Xavier [48]. We perform a grid search to determine the optimal parameter settings for each model. Specifically, we tune the learning rate among $\{1e^{-3}, 5e^{-3}, 1e^{-4}\}$, and search for the coefficient of the L_2 regularization term in $\{1e^{-9}, 1e^{-8}, ..., 1e^{-1}\}$. Regarding the backbone of NGCF and LightGCN, we tune the number of layers among $\{1,2,3\}$, with or without using dropout to prevent over-fitting. With regards to SL and BSL, we search for temperatures with an interval of 0.10 in the range $[0.05, 1.0]$. We also vary the number of negative samplings among 200, 400, 800, 1500. For the methods under comparison, we meticulously adhere to the hyperparameter settings delineated in the original paper, while conducting a comprehensive grid search to identify the optimal configuration in our experiments. Please refer to the Appendix for more details.

B. Performance Comparison (RQ1)

In this subsection, we begin to analyze the superiority of our loss, as compared with other baselines.

- As demonstrated in Table II, SL and BSL consistently outperform other loss functions by a significant margin on all four datasets and three backbone models. Moreover, even when applied to basic models such as MF and LightGCN, SL and BSL can outperform state-of-the-art methods. We

TABLE II: Overall performance comparison. (+X) denotes the loss functions. The best result is bolded and the runner-up is underlined.

Model		Amazon		Yelp2018		Gowalla		Movielens-1M	
		Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
WWW'2017	CML	0.0362	0.0165	0.0622	<u>0.0536</u>	0.1670	0.1292	0.1730	0.1563
SIGIR'2019	ENMF	0.0405	0.0182	0.0624	0.0515	0.1523	0.1315	0.2315	0.2069
SIGIR'2020	NIA-GCN	-	-	0.0599	0.0491	0.1359	0.1106	0.2359	0.2242
AAAI'2020	LR-GCCF	0.0326	0.0158	0.0561	0.0343	0.1519	0.1285	0.2231	0.2124
SIGIR'2020	DGCF	0.0365	0.0164	0.0654	0.0534	0.1842	0.1561	0.2640	0.2504
SIGIR'2021	SGL	0.0454	0.0209	0.0690	0.0570	0.1798	0.1524	0.2528	0.2401
CIKM'2021	SimpleX	-	-	0.0701	0.0575	0.1872	0.1557	0.2802	0.2670
CIKM'2021	UltraGCN	0.0421	0.0195	0.0683	0.0561	0.1862	0.1580	0.2787	0.2642
SIGIR'2022	SimSGL	0.0486	0.0225	0.0721	0.0601	0.1834	0.1550	0.2689	0.2520
WWW'2022	NCL	0.0426	0.0195	0.0664	0.0552	0.1760	0.1506	0.2623	0.2496
MF UAI'2009	+ BPR	0.0268	0.0115	0.0549	0.0445	0.1616	0.1366	0.2153	0.2175
	+ BCE	0.0276	0.0120	0.0513	0.0415	0.1492	0.1251	0.2329	0.2244
	+ MSE	0.0321	0.0135	0.0415	0.0337	0.1160	0.1005	0.2216	0.2023
	+ SL	0.0446	0.0195	0.0718	0.0585	0.1760	0.1399	0.2786	0.2633
	+ BSL	0.0476	0.0217	0.0725	0.0599	<u>0.1882</u>	<u>0.1584</u>	<u>0.2804</u>	**0.2681**
NGCF SIGIR'2019	+ BPR	0.0235	0.0076	0.0579	0.0477	0.1570	0.1327	0.2513	0.2511
	+ BCE	0.0266	0.0107	0.0498	0.0401	0.1410	0.1169	0.2421	0.2292
	+ MSE	0.0234	0.0097	0.0391	0.0319	0.0837	0.0720	0.2376	0.2135
	+ SL	0.0468	0.0218	0.0711	0.0589	0.1674	0.1399	0.2789	0.2632
	+ BSL	0.0491	0.0232	0.0732	0.0610	0.1829	0.1513	**0.2808**	<u>0.2676</u>
LGN SIGIR'2020	+BPR	0.0399	0.0180	0.0649	0.0530	0.1830	0.1554	0.2576	0.2427
	+BCE	0.0386	0.0177	0.0557	0.0457	0.1470	0.1271	0.2484	0.2412
	+MSE	0.0336	0.0150	0.0463	0.0366	0.1347	0.1080	0.2488	0.2291
	+SL	<u>0.0519</u>	<u>0.0248</u>	<u>0.0736</u>	<u>0.0612</u>	0.1878	0.1577	0.2792	0.2627
	+BSL	**0.0521**	**0.0249**	**0.0744**	**0.0617**	**0.1893**	**0.1591**	0.2799	0.2627

TABLE III: Overall performance comparison. (+X) denotes the loss functions. The best result is bolded and the runner-up is underlined.

Model		Amazon		Yelp2018		Gowalla		Movielens-1M		Avg.
		Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20	
SGL (Wu et al., 2021)	-	0.0454	0.0209	0.0690	0.0570	0.1798	0.1524	0.2528	0.2401	
	+SL	$0.0513^{+13.14\%}$	$0.0241^{+15.45\%}$	$0.0736^{+6.68\%}$	$0.0612^{+7.35\%}$	$0.1872^{+4.12\%}$	$0.1568^{+2.89\%}$	$0.2766^{+9.41\%}$	$0.2579^{+7.41\%}$	$+8.31\%$
	+BSL	$0.0513^{+13.16\%}$	$0.0242^{+15.60\%}$	$\mathbf{0.0741}^{+7.35\%}$	$\mathbf{0.0615}^{+7.86\%}$	$\mathbf{0.1890}^{+5.12\%}$	$\mathbf{0.1592}^{+4.46\%}$	$\underline{0.2787}^{+10.25\%}$	$\underline{0.2621}^{+9.16\%}$	$+9.12\%$
SimSGL (Yu et al., 2022)	-	0.0486	0.0225	0.0721	0.0601	0.1834	0.1550	0.2689	0.2520	
	+SL	$0.0515^{+5.84\%}$	$0.0239^{+6.35\%}$	$0.0733^{+0.89\%}$	$0.0608^{+1.64\%}$	$0.1835^{+0.05\%}$	$0.1552^{+0.13\%}$	$0.2752^{+2.34\%}$	$0.2565^{+1.79\%}$	$+2.38\%$
	+BSL	$0.0514^{+5.78\%}$	$0.0239^{+6.35\%}$	$0.0736^{+1.32\%}$	$0.0611^{+2.17\%}$	$0.1834^{+0.00\%}$	$0.1558^{+0.52\%}$	$\mathbf{0.2796}^{+3.98\%}$	$\mathbf{0.2676}^{+6.19\%}$	$+3.29\%$
LightGCL (Cai et al., 2023)	-	0.0489	0.0224	0.0697	0.0573	0.1867	0.1562	0.2316	0.2287	
	+SL	$\underline{0.0530}^{+8.49\%}$	$\underline{0.0246}^{+9.86\%}$	$0.0698^{+0.11\%}$	$0.0581^{+1.38\%}$	$0.1805^{-1.58\%}$	$0.1539^{+0.26\%}$	$0.2742^{+18.39\%}$	$0.2562^{+12.02\%}$	$+6.12\%$
	+BSL	$\mathbf{0.0533}^{+8.94\%}$	$\mathbf{0.0247}^{+10.22\%}$	$0.0714^{+2.39\%}$	$0.0594^{+3.70\%}$	$0.1837^{+0.16\%}$	$0.1553^{+1.17\%}$	$0.2767^{+19.47\%}$	$0.2597^{+13.55\%}$	$+7.45\%$

attribute this to the efficacy of SL and BSL in mitigating negative distribution shifts.

- Our proposed BSL consistently outperforms SL in all conditions, empirically verifying the necessity of denoising over positive interactions. By upgrading the positive component of SL to an advanced Log-Expectation-Exp structure, BSL exhibits better robustness against positive noise.

- Notably, due to the variations among datasets and baseline models, the improvements may exhibit slight disparities. For instance, in the case of Gowalla, while MF-SL performs comparably with MF-BPR, MF-BSL improves the performance by 16.0% and 16.5%, in terms of Recall@20 and NDCG@20 respectively. We suspect this phenomenon is due to the presence of more positive noise in the Gowalla dataset, which hinders the effectiveness of SL.

- We incorporated additional experiments on more SOTA models including SGL (SIGIR'21) [35], SimSGL (SIGIR'22) [36] and LightGCL (ICLR'23) [37]). The results are presented in Table III. Here We applied both SL and BSL losses to those SOTA methods and observed that both

losses enhance model performance, with BSL yielding more significant improvements. These results are in alignment with our analyses and demonstrate the effectiveness of the proposed loss.

- As shown in Figure 7, our analysis confirms that, across the metrics of NDCG@5, NDCG@10, and NDCG@15, the integration of BSL consistently enhances model performance. This enhancement is substantial enough to allow basic models (*e.g.,* MF, lightGCN) to surpass SOTA models (*e.g.,* SimSGL, NCL).

C. Performance with Noisy Negative Data (RQ2)

In this section, we verify the resilience of SL and BSL against noise on negative data. We conduct two sets of experiments: 1) Fixing the number of negative instances and allowing the negative sampler to draw a certain ratio of positive data. 2) Increasing the number of uniformly sampled instances using the conventional method. Strategy 1 may lead to more false negatives, while strategy 2 increases the quantity
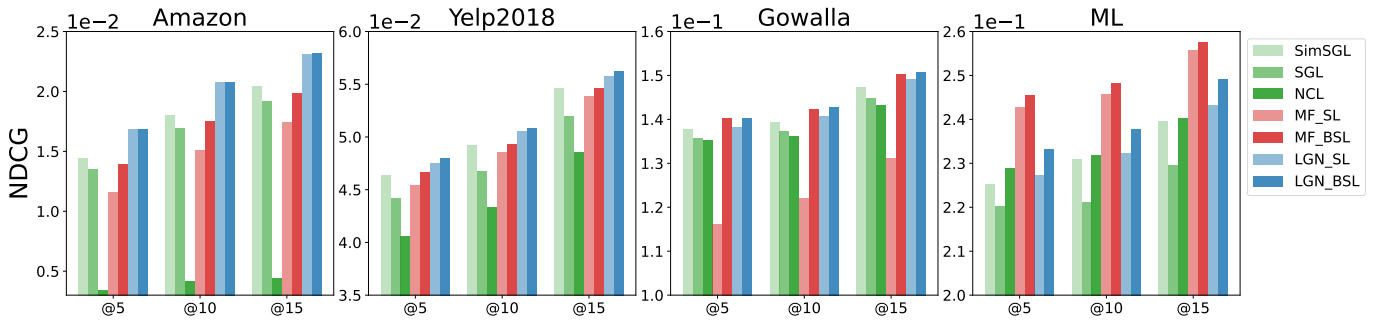
Fig. 7: Performance comparison *w.r.t.* different metric setting.
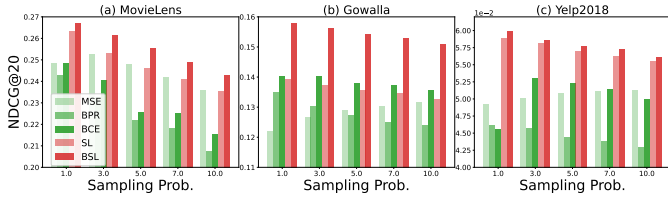

Fig. 8: Performance comparison *w.r.t.* the ratio of false negative samples. A grid search is conducted to confirm the optimal parameter setting for each model.
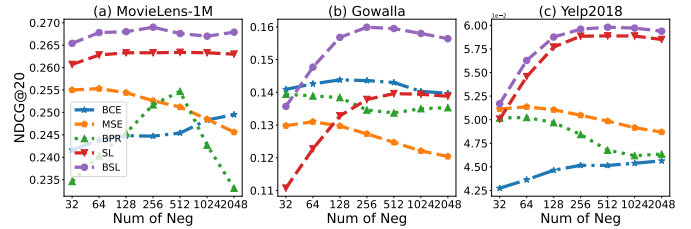

Fig. 9: Performance comparison *w.r.t.* the number of negative samples. A grid search is conducted to confirm the optimal parameter setting for each model.

of noisy negative instances. We perform these experiments on MF and observe similar results on other backbone models.

1) Impact of noisy ratio in negative samples:

- Figure 8 shows that noisy data negatively affects loss function performance. However, SL and BSL can handle noise in negative sampling by adjusting the parameter τ, leading to better performance compared to other loss functions. Moreover, as the proportion of noisy negatives increases, the optimal τ also increases, aligning with the analysis in Section III-B.

- Furthermore, an interesting finding is that increasing the noise ratio on Yelp2018 can unexpectedly boost the performance of MSE and BCE. This anomaly arises from the inherent instability and varying performance of point-wise loss functions across datasets and scenarios. However, SL and BSL consistently outperform other loss functions and demonstrate greater stability.

2) Impact of sampling number:

- Firstly, as depicted in Figure 9, traditional loss functions such as BCE, BPR, and MSE can be negatively impacted by the number of negative samples. Specifically, on the MovieLens dataset, MSE continues to decline while BCE experiences fluctuations with the addition of negative samples. We attribute this to the fact that MovieLens is a relatively small and dense dataset, with only 3k items. Excess sampling numbers (greater than 512) inevitably increase the number of false negatives. However, in terms of distributionally robust optimization, SL and BSL exhibit extremely stable performance.

- Secondly, we have observed that BSL consistently outperforms SL, which validates the importance of leverag-

ing the "worst-case" optimization over positive samples. When examining the Gowalla dataset, although SL displays a rising and gradually stabilizing trend, it is unable to achieve superior performance against other baseline models. Instead, BSL exhibits the same trend as SL but effectively addresses the issue of false positive samples on Gowalla, enabling it to demonstrate superiority. As a result, we argue that our proposed simple yet effective BSL is crucial for recommendation systems.

D. Performance with Noisy Positive data (RQ3)

In this section, we aim to validate the resilience of BSL to positive noise. To this end, we contaminate the positive instances by introducing a certain proportion of randomly sampled negative items, ranging from 10% to 40%, while keeping the test set unchanged. Additionally, we utilize t-SNE to visualize the item embeddings and analyze the effect of false positive data. The MF backbone is employed, and our observations are as follows:

1) Performance Comparison:

- Undoubtedly, the introduction of noise into the positive data can have a detrimental impact on model performance. However, we contend that BSL possesses a significant ability to withstand such noise. As demonstrated in Table IV, even when the ratio of noise training data reaches 40%, the decrease in performance is not excessively large, ranging from 5% to 15%. We attribute this to two factors: 1) The Log-Expectation-Exp structure in the positive loss enables the model to optimize under various distributions with perturbations, resulting in better robustness to positive distribution shifts. 2) The separate temperatures allow the

| ratio	Model	Amazon		Yelp2018		Gowalla		Movielens-1M	
		Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
10%	MF-SL	0.0465	0.0203	0.0704	0.0576	0.1719	0.1356	0.2755	0.2600
	MF-BSL	**0.0482**	**0.0217**	**0.0711**	**0.0589**	**0.1844**	**0.1566**	**0.2796**	**0.2659**
	%Improv.	+3.66%	+6.90%	+1.00%	+2.26%	+7.27%	+15.49%	+1.49%	+2.27%
20%	MF-SL	0.0442	0.0191	0.0686	0.0563	0.1667	0.1315	0.2716	0.2554
	MF-BSL	**0.0465**	**0.0209**	**0.0700**	**0.0577**	**0.1815**	**0.1542**	**0.2772**	**0.2635**
	%Improv.	+5.20%	+9.42%	+2.04%	+2.49%	+8.88%	+17.26%	+2.06%	+3.17%
30%	MF-SL	0.0421	0.0182	0.0675	0.0551	0.1611	0.1266	0.2681	0.2523
	MF-BSL	**0.0446**	**0.0199**	**0.0687**	**0.0570**	**0.1786**	**0.1517**	**0.2742**	**0.2622**
	%Improv.	+5.94%	+9.34%	+1.78%	+3.45%	+10.86%	+19.83%	+2.28%	+3.51%
40%	MF-SL	0.0403	0.0172	0.0657	0.0536	0.1571	0.1228	0.2645	0.2475
	MF-BSL	**0.0428**	**0.0192**	**0.0681**	**0.0560**	**0.1758**	**0.1495**	**0.2717**	**0.2578**
	%Improv.	+6.20%	+11.63%	+3.65%	+4.48%	+11.90%	+21.74%	+2.72%	+4.16%
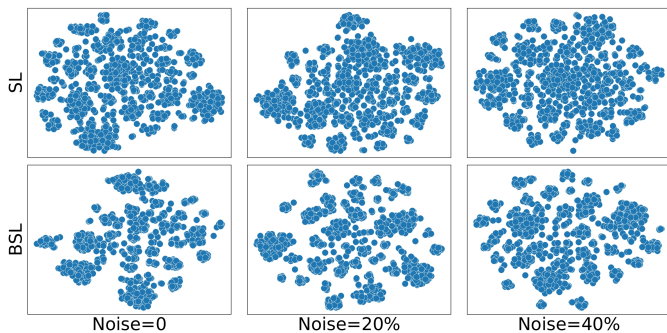
Fig. 10: t-SNE Visualization on Gowalla with additional noise data. BSL leads to better group-wise separation than SL in each noise case.
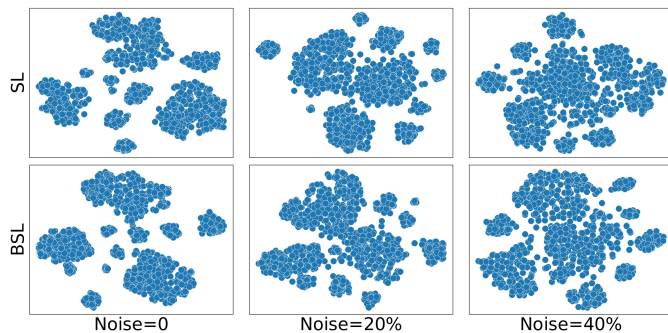

Fig. 11: t-SNE Visualization on Yelp2018 with additional noise data. BSL leads to better group-wise separation than SL in each noise case.

model to adapt to different levels of positive and negative noise.

- The performance degradation of our proposed BSL is consistently lower than that of SL, which validates the effectiveness of denoising. For instance, when contaminated with 40% noisy interactions on Gowalla, SL experiences a significant drop in performance (from 0.1399 to 0.1228). In contrast, the robustness of BSL is enhanced by increasing the ratio $\frac{\tau_1}{\tau_2}$ and achieving a score of 0.1496 (from 0.1584 to 0.1496). Additionally, BSL demonstrates greater improvement over SL in scenarios with more noise. Specifically, with regards to Gowalla, the performance of BSL with 40% additional noisy interactions still surpasses that of SL with a noise-free dataset. This outcome highlights the necessity of denoising positive samples.

2) Visual Analysis: Due to spatial constraints, we conducted visual experiments on Gowalla and Yelp2018. As illustrated in Figure 10 and 11, as the ratio of noise data increases, the item embeddings via SL begin to entangle. When the ratio reaches 40%, the representation appears to resemble a uniform distribution, which confuses the relationships among items. In contrast, the embeddings via our proposed BSL still achieve decent group-wise separation. We believe that this is a superior and more meaningful embedding that reflects the characteristics of collaborative filtering, where similar items

are grouped in the embedding space.

E. Sensitivity Study (RQ4)

In this section, we conducted sensitivity analysis experiments on parameters in SL/BSL, thereby assisting everyone in providing some valuable insights for parameter configuration in the new dataset. Specifically, we initially conducted a validation of the model's dimensions. In the previous experiments, we fixed the representation dimension at 64. However, in this section, we attempted dimensions such as 128, 256, and 512. Additionally, we also examined the performance variations based on the τ_1/τ_2. The details are as follows:

1) Impact of embedding dimension: Our analysis confirms that (*cf.* Figure 12), even with increased embedding sizes (*e.g.,* 512), the integration of SL/BSL consistently improves model performance. Notably, MF equipped with SL/BSL still achieves comparable performance with SOTA models under the setting of large embedding sizes. Additionally, despite the marginal performance gains of SL/BSL as the dimensions increase, it showcases its robustness towards the model and feature dimensions. In particular, the high accuracy at lower dimensions also makes it suitable for practical scenarios.

2) Impact of temperature: We conducted additional experiments to investigate the model's performance with varying ratios of τ_1/τ_2 by adjusting the hyper-parameter τ_1. The results are illustrated in Figure 13. When the value of τ_1 is excessively
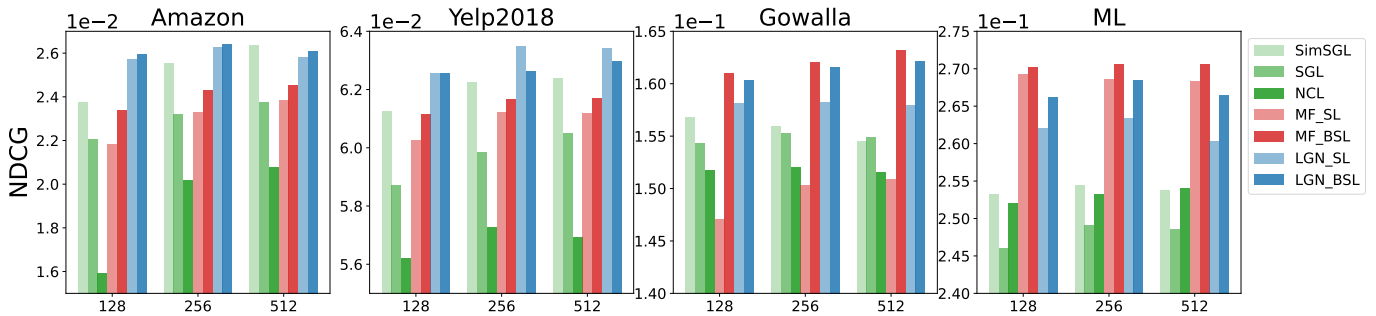
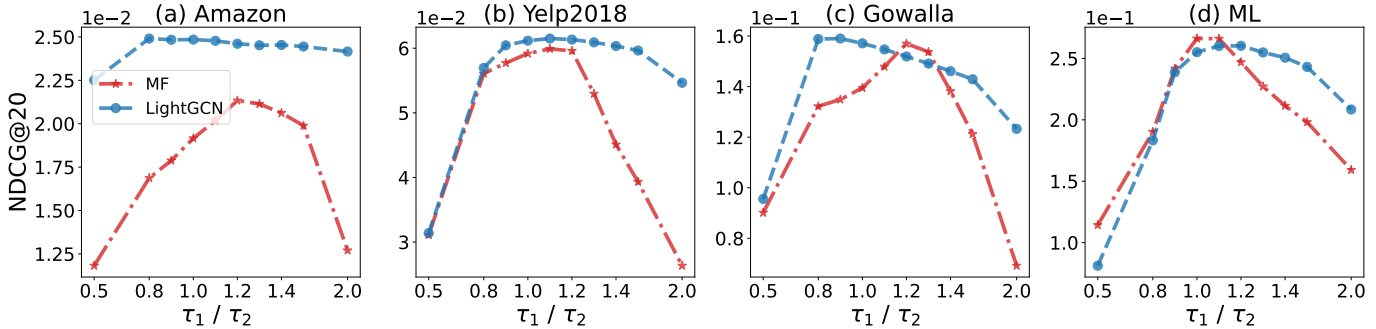**Fig. 12:** Performance comparison *w.r.t.* the different embedding dimension.

Fig. 13: Performance comparison *w.r.t.* the ratio of τ_1/τ_2.

large (*e.g.*, $\tau_1/\tau_2 = 2$), which implies an overly small robust radius according to our Corollary III.1, the model has poor robustness against positive noise. With τ_1 decreasing, leading to increased robustness radius, both the model's robustness and performance boost. However, a further reduction of τ_1 can be detrimental to performance, since an excessively large robust radius might amplify the risk of encountering an implausible worst-case distribution in DRO.

VI. RELATED WORK

In this section, we review the most related work from the following perspectives.

A. Recommendation System

The primary goal of the recommendation system is to predict potential interactions. Recent research has primarily focused on developing advanced model architectures, ranging from matrix factorization (MF-based) methods [49], VAE-based methods [2; 3; 4], to GNN-based methods [5; 6; 7]. For instance, MF-based models project the ID of users (or items) into embedding vectors to reconstruct historical interactions, while GNN-based models aggregate neighborhood information to enhance user/item representations. For further details, readers can refer to the survey [50].

B. Robustness in Recommender System

There has been increasing attention to enhancing the robustness of recommendation systems. In the early stages, Michael et al. [51] improved robustness by incorporating neighbor selection into MF via an adversarial network [52]. To combine with user-item graph, Chen et al. [53] and Feng et al. [54] designed several denoising strategies or modules, such as attention mechanism [53] or stochastic binary masks

[54]. More recently, Wang et al. [30] adaptively pruned noisy interactions during training by monitoring the training process. Furthermore, Lin et al. [55] introduced AutoDenoise, serving as an agent in DRL to dynamically select noise-free and predictive data instances. Ge et al. [56] learned to automatically assign the most appropriate weights to each implicit data sample. In general, the existing methods incur additional time and computational cost to achieve robustness, or rely on access to prior knowledge to guide the denoising process. In contrast, our work tackles the challenge of denoising from the perspective of innovating the loss function architecture.

C. Loss functions in Recommender System

The loss function is critical in recommendation systems. Various loss functions have been proposed to address different challenges in collaborative filtering (CF). [45] proposed the cosine contrastive loss inspired by contrastive learning. To combat the negative impact of popularity bias in CF models, [57] integrated bias-aware margins into contrastive loss. Auto-ML technique has been employed by [58] to search for the optimal loss design. They developed a novel controller network that can dynamically adjust the loss probabilities in a differentiable manner. Although [45] and [57] can be seen as improvements to SL, they lack theoretical analysis of their advantages or suffer from the efficiency issues caused by additional optimization.

To the best of our knowledge, there is limited work studying the theoretical properties of SL in RS. For example, [15] related SL to NDCG and proved SL is bounded by NDCG under multiple relaxations. However, their analysis struggles to explain the robustness, role of τ, and fairer recommendation. [11] attributed the fairness arising from SL to the popularity-

aware sampling. Nevertheless, we observe that SL maintains fairness and strong performance even with uniform sampling.

By examining the Softmax loss from other areas, we identified a recent study [25] that analyzes contrastive loss from DRO perspective, which is closely related to our research. The distinctions between our work with [25] are threefold: 1) Our theoretical and empirical analyses of SL are specifically tailored to the recommendation domain; 2) We demonstrate both the robustness and fairness that are inherent in the SL when applied to recommendation systems; 3) We have developed a novel loss function BSL, which is specifically designed for recommender systems.

D. Temperature in Softmax Loss

Limited work has analyzed temperature in SL, primarily from the community of contrastive learning (CL). For instance, Wang et al. [11; 59] attributed the success of CL to its inherent hardness-awareness, while Zhang et al. [60] focused on the decorrelation ability. Additionally, Zhang et al. [61] proposed dual temperature in the vector and scalar components, which is different from our Log-Expectation-Exp structure. Furthermore, [62] demonstrated that a simple cosine schedule for achieving dynamic τ can significantly improve long-tail performance. However, these studies regard temperature as a heuristic design. In contrast, our work provides novel theoretical insights into the importance of temperature for robustness in SL.

VII. CONCLUSION AND FUTURE WORK

In this work, we focus on explaining the essence of softmax loss in recommendation system. We prove that: 1) Optimizing SL is equivalent to performing DRO on the negative parts in vanilla pointwise loss, giving it the ability to alleviate negative sample noise; 2) Comparing with other loss functions, SL implicitly penalizes the prediction variance, yielding fairer results; Based on the above analysis, we propose BSL that mirrors the advantageous structure of the negative loss part to the positive part such that BSL enjoys the merit of both positive and negative denoising. In our experiments, we evaluate BSL on numerous backbones to validate the effectiveness of our model and demonstrate its superiority.

This work represents the first step towards bridging the gap between SL and its superiority and robustness. We believe that the proposed DRO method, which is generally used to deal with noisy data in RS, holds promising potential for extending to other fields of denoising. Additionally, exploring the role of different loss functions in fairness from our perspective is a potential direction for future research.

APPENDIX

In this section, we present the implementation details of our BSL in different backbones. During training, we use cosine similarity and BSL to calculate the training loss. During testing, we use cosine similarity for prediction score in MF and inner product in GCN-based models (NGCF, LightGCN). In "Negative Sampling," we use efficient uniform sampling techniques, while in "In Batch," we treat other positive items in a batch as negative samples.

TABLE V: The difference between MF, NGCF and LightGCN.

Backbone	Training	Testing	Sampling Method
MF	cosine similarity	cosine similarity	Negative Sampling
NGCF	cosine similarity	inner product	In Batch
LightGCN	cosine similarity	inner product	In Batch

Algorithm 1 BSL Pseudocode (Negative Sampling)

```
# f: user and item embedding table
# t1: temperature scaling for positive samples
# t2: temperature scaling for negative samples
for (u,i,j) in loader:
# load a minibatch (u,i) with m negative samples
# dimension u: [B]; i:[B]; j:[B,m]
  emb_u, emb_i, emb_j = f(u), f(i), f(j)
  # dimension u: [B, D]; i:[B, D]; j:[B, m, D]
  L = loss_fn(emb_u, emb_i, emb_j)
  L.backward() # back-propagate
  update(f) # Adam update

def loss_fn(emb_u, emb_i, emb_j):
  emb_u = normalize(emb_u, dim=1) # l2-normalize
  emb_i = normalize(emb_u, dim=1) # l2-normalize
  emb_j = normalize(emb_u, dim=1) # l2-normalize
  pos_score=(emb_u*emb_i).sum(dim=1) # dimension: [B]
  neg_score=(emb_u.unsqueeze(1)*emb_j).sum(dim=2)
  # dimension: [B, m]
  L(BSL) = -((pos_score/t1).exp() / (neg_score/t2).exp
    ().sum().pow(t1/t2) ).log()
  return L(BSL)
```

Algorithm 2 BSL Pseudocode (In Batch), PyTorch-like

```
# f: user and item embedding table
# t1: temperature scaling for positive samples
# t2: temperature scaling for negative samples
for (u,i) in loader:
# load a minibatch (u,i)
# dimension u: [B]; i:[B]
  emb_u, emb_i = f(u), f(i)
  # dimension u: [B, D]; i:[B, D]; j:[B, m, D]
  L = loss_fn(emb_u, emb_i)
  L.backward() # back-propagate
  update(f) # Adam update

def loss_fn(emb_u, emb_i):
  emb_u = normalize(emb_u, dim=1) # l2-normalize
  emb_i = normalize(emb_u, dim=1) # l2-normalize

  s=emb_u.matmul(emb_i.t()) # pairwise Similarity

  B = emb_u.shape(0)
  mask_index = mask_index(B)
  pos_score=s[~mask_index] # dimension: [B]
  neg_score=s[mask_index].view(B, -1)

  L(BSL) = -((pos_score/t1).exp() / (neg_score/t2).
    exp().sum().pow(t1/t2) ).log()
  return L(BSL)

def mask_index(batch_size):
  negative_mask = ones((batch_size, batch_size))
  for i in range(batch_size):
    negative_mask[i, i] = 0
  return negative_mask
```

REFERENCES

[1] Rong Pan, Yunhong Zhou, Bin Cao, Nathan Nan Liu, Rajan M. Lukose, Martin Scholz, and Qiang Yang. One-class collaborative filtering. In *ICDM*, pages 502–511. IEEE Computer Society, 2008.

[2] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *WWW*, pages 689–698. ACM, 2018.

[3] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. Learning disentangled representations for recommendation. In *NeurIPS*, pages 5712–5723, 2019.

[4] Harald Steck. Embarrassingly shallow autoencoders for sparse data. In *WWW*, pages 3251–3257. ACM, 2019.

[5] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*, pages 639–648. ACM, 2020.

[6] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *KDD*, pages 974–983. ACM, 2018.

[7] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *SIGIR*, pages 165–174. ACM, 2019.

[8] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *WWW*, pages 173–182. ACM, 2017.

[9] Xiangnan He and Tat-Seng Chua. Neural factorization machines for sparse predictive analytics. In *SIGIR*, pages 355–364. ACM, 2017.

[10] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461. AUAI Press, 2009.

[11] Jiancan Wu, Xiang Wang, Xingyu Gao, Jiawei Chen, Hongcheng Fu, Tianyu Qiu, and Xiangnan He. On the effectiveness of sampled softmax loss for item recommendation. *CoRR*, abs/2201.02327, 2022.

[12] Yoshua Bengio and Jean-Sébastien Senecal. Quick training of probabilistic neural nets by importance sampling. In *AISTATS*. Society for Artificial Intelligence and Statistics, 2003.

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

[14] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *RecSys*, pages 191–198. ACM, 2016.

[15] Sebastian Bruch, Xuanhui Wang, Michael Bendersky, and Marc Najork. An analysis of the softmax cross entropy loss for learning-to-rank with binary relevance. In *ICTIR*, pages 75–78. ACM, 2019.

[16] Aharon Ben-Tal, Dick den Hertog, Anja De Waegenaere, Bertrand Melenberg, and Gijs Rennen. Robust solutions of optimization problems affected by uncertain probabil-ities. *Manag. Sci.*, 59(2):341–357, 2013.

[17] John C. Duchi, Peter W. Glynn, and Hongseok Namkoong. Statistics of robust optimization: A generalized empirical likelihood approach. *Math. Oper. Res.*, 46(3):946–969, 2021.

[18] Zhaolin Hu and L Jeff Hong. Kullback-leibler divergence constrained distributionally robust optimization. *Available at Optimization Online*, pages 1695–1724, 2013.

[19] Honglei Zhuang, Zhen Qin, Xuanhui Wang, Michael Bendersky, Xinyu Qian, Po Hu, and Dan Chary Chen. Cross-positional attention for debiasing clicks. In *WWW*, pages 788–797. ACM / IW3C2, 2021.

[20] Rocío Cañamares and Pablo Castells. Should I follow the crowd?: A probabilistic analysis of the effectiveness of popularity in recommender systems. In *SIGIR*, pages 415–424. ACM, 2018.

[21] Rolf Jagerman, Harrie Oosterhuis, and Maarten de Rijke. To model or to intervene: A comparison of counterfactual and online learning to rank from user interactions. In *SIGIR*, pages 15–24. ACM, 2019.

[22] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. Causal intervention for leveraging popularity bias in recommendation. In *SIGIR*, pages 11–20. ACM, 2021.

[23] Steffen Rendle. Item recommendation from implicit feedback. In *Recommender Systems Handbook*, pages 143–171. Springer US, 2022.

[24] Chun-Hsiao Yeh, Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu, Yubei Chen, and Yann LeCun. Decoupled contrastive learning. In *ECCV (26)*, volume 13686 of *Lecture Notes in Computer Science*, pages 668–684. Springer, 2022.

[25] Junkang Wu, Jiawei Chen, Jiancan Wu, Wentao Shi, Xiang Wang, and Xiangnan He. Understanding contrastive learning via distributionally robust optimization. In *NeurIPS*, 2023.

[26] Robert C. Williamson and Aditya Krishna Menon. Fairness risk measures. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 6786–6797. PMLR, 2019.

[27] Jiawei Chen, Hande Dong, Yang Qiu, Xiangnan He, Xin Xin, Liang Chen, Guli Lin, and Keping Yang. Autodebias: Learning to debias for recommendation. In *SIGIR*, pages 21–30. ACM, 2021.

[28] Louis Faury, Ugo Tanielian, Elvis Dohmatob, Elena Smirnova, and Flavian Vasile. Distributionally robust counterfactual risk minimization. In *AAAI*, pages 3850–3857. AAAI Press, 2020.

[29] Wenjie Wang, Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. Clicks can be cheating: Counterfactual recommendation for mitigating clickbait issue. In *SIGIR*, pages 1288–1297. ACM, 2021.

[30] Wenjie Wang, Fuli Feng, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. Denoising implicit feedback for recommendation. In *WSDM*, pages 373–381. ACM, 2021.

[31] Tinglin Huang, Yuxiao Dong, Ming Ding, Zhen Yang, Wenzheng Feng, Xinyu Wang, and Jie Tang. Mixgcf: An improved training method for graph neural network-based recommender systems. In *KDD*, 2021.

[32] Weinan Zhang, Tianqi Chen, Jun Wang, and Yong Yu. Optimizing top-n collaborative filtering via dynamic negative item sampling. In *SIGIR*, pages 785–788. ACM, 2013.

[33] Zhen Yang, Ming Ding, Chang Zhou, Hongxia Yang, Jingren Zhou, and Jie Tang. Understanding negative sampling in graph representation learning. In *KDD*, pages 1666–1676. ACM, 2020.

[34] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 165–174, 2019.

[35] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. Self-supervised graph learning for recommendation. In *SIGIR*, 2021.

[36] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. Are graph augmentations necessary?: Simple graph contrastive learning for recommendation. In *SIGIR*, pages 1294–1303. ACM, 2022.

[37] Xuheng Cai, Chao Huang, Lianghao Xia, and Xubin Ren. Lightgcl: Simple yet effective graph contrastive learning for recommendation. In *ICLR*. OpenReview.net, 2023.

[38] Tinglin Huang, Yuxiao Dong, Ming Ding, Zhen Yang, Wenzheng Feng, Xinyu Wang, and Jie Tang. Mixgcf: An improved training method for graph neural network-based recommender systems. In *KDD*, pages 665–674. ACM, 2021.

[39] Wenhui Yu and Zheng Qin. Graph convolutional network for recommendation with low-pass collaborative filters. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 10936–10945. PMLR, 2020.

[40] Chong Chen, Min Zhang, Yongfeng Zhang, Yiqun Liu, and Shaoping Ma. Efficient neural matrix factorization without sampling for recommendation. *ACM Trans. Inf. Syst.*, 38(2):14:1–14:28, 2020.

[41] Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge J. Belongie, and Deborah Estrin. Collaborative metric learning. In *WWW*, pages 193–201. ACM, 2017.

[42] Jianing Sun, Yingxue Zhang, Wei Guo, Huifeng Guo, Ruiming Tang, Xiuqiang He, Chen Ma, and Mark Coates. Neighbor interaction aware graph convolution networks for recommendation. In *SIGIR*, pages 1289–1298. ACM, 2020.

[43] Lei Chen, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. In *AAAI*, pages 27–34. AAAI Press, 2020.

[44] Kelong Mao, Jieming Zhu, Xi Xiao, Biao Lu, Zhaowei Wang, and Xiuqiang He. Ultragcn: Ultra simplification of graph convolutional networks for recommendation. In *CIKM*, pages 1253–1262. ACM, 2021.

[45] Kelong Mao, Jieming Zhu, Jinpeng Wang, Quanyu Dai, Zhenhua Dong, Xi Xiao, and Xiuqiang He. Simplex: A simple and strong baseline for collaborative filtering. In *CIKM*, pages 1243–1252. ACM, 2021.

[46] Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In *WWW*, pages 2320–2329. ACM, 2022.

[47] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. Disentangled graph collaborative filtering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 1001–1010, 2020.

[48] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, volume 9 of *JMLR Proceedings*, pages 249–256. JMLR.org, 2010.

[49] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, pages 263–272. IEEE Computer Society, 2008.

[50] Le Wu, Xiangnan He, Xiang Wang, Kun Zhang, and Meng Wang. A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation. *IEEE Trans. Knowl. Data Eng.*, 35(5):4425–4445, 2023.

[51] Michael P. O'Mahony, Neil J. Hurley, and Guenole C. M. Silvestre. Efficient and secure collaborative filtering through intelligent neighbour selection. In *ECAI*, pages 383–387. IOS Press, 2004.

[52] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. Adversarial personalized ranking for recommendation. In *SIGIR*, pages 355–364. ACM, 2018.

[53] Huiyuan Chen, Lan Wang, Yusan Lin, Chin-Chia Michael Yeh, Fei Wang, and Hao Yang. Structured graph convolutional networks with stochastic masks for recommender systems. In *SIGIR*, pages 614–623. ACM, 2021.

[54] Ping Feng, Yang Qian, Xiaohan Liu, Guoliang Li, and Jian Zhao. Robust graph collaborative filtering algorithm based on hierarchical attention. In *WISA*, volume 12999 of *Lecture Notes in Computer Science*, pages 625–632. Springer, 2021.

[55] Weilin Lin, Xiangyu Zhao, Yejing Wang, Yuanshao Zhu, and Wanyu Wang. Autodenoise: Automatic data instance denoising for recommendations. In *WWW*, pages 1003–1011. ACM, 2023.

[56] Yingqiang Ge, Mostafa Rahmani, Athirai A. Irissappane, Jose Sepulveda, James Caverlee, and Fei Wang. Automated data denoising for recommendation. *CoRR*, abs/2305.07070, 2023.

[57] An Zhang, Wenchang Ma, Xiang Wang, and Tat-Seng Chua. Incorporating bias-aware margins into contrastive loss for collaborative filtering. In *NeurIPS*, 2022.

[58] Xiangyu Zhao, Haochen Liu, Wenqi Fan, Hui Liu, Jiliang Tang, and Chong Wang. Autoloss: Automated loss

function search in recommendations. In *KDD*, pages 3959–3967. ACM, 2021.

[59] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *CVPR*, pages 2495–2504. Computer Vision Foundation / IEEE, 2021.

[60] Chaoning Zhang, Kang Zhang, Chenshuang Zhang, Trung X. Pham, Chang D. Yoo, and In So Kweon. How does simsiam avoid collapse without negative samples? A unified understanding with self-supervised contrastive learning. In *ICLR*. OpenReview.net, 2022.

[61] Chaoning Zhang, Kang Zhang, Trung X. Pham, Axi Niu, Zhinan Qiao, Chang D. Yoo, and In So Kweon. Dual temperature helps contrastive learning without many negative samples: Towards understanding and simplifying moco. In *CVPR*, pages 14421–14430. IEEE, 2022.

[62] Anna Kukleva, Moritz Böhle, Bernt Schiele, Hilde Kuehne, and Christian Rupprecht. Temperature schedules for self-supervised contrastive methods on long-tail data. In *ICLR*. OpenReview.net, 2023.

We have made substantial revisions to the paper based on the comments provided by the meta-reviewer and three reviewers. The major changes are as follows:

1) We have considered more SOTA models including SGL (SIGIR'21) [35], SimSGL (SIGIR'22) [36] and Light-GCL (ICLR'23) [37] and conducted experiments on more metrics including NDCG@5, NDCG@10, NDCG@15. The results are presented in Table III and Figure 7.

2) We have conducted additional sensitivity experiments to investigate the model's performance with varying ratios of τ_1/τ_2 and embedding dimension. The results are illustrated in Figure 13 and Figure 12.

3) We have added some statements to emphasize the significance of fairness stemming from DRO. Furthermore, additional analyses and experiments have been conducted in Section III to demonstrate the fairness of SL.

4) We have carefully checked the entire paper, polishing the language and figures, as well as typos, confusing expressions, and other minor issues in the revised version.

We thank all reviewers for the careful reading and valuable feedback. We will address their concerns as follows.

We greatly appreciate your acknowledgement of our contributions and your insightful comments. In what follows, we provide responses to the concerns you have raised:

Comment 1: *The author should survey more SOTA models and evaluate the results are true.*

Response 1: Thank you for the insightful suggestion. We incorporated additional experiments on more SOTA models including SGL (SIGIR'21) [35], SimSGL (SIGIR'22) [36] and LightGCL (ICLR'23) [37]. The results are presented in Table III. Here we applied both SL and BSL losses to those SOTA methods and observed that both losses enhance model performance, with BSL yielding more significant improvements. These results are in alignment with our analyses and demonstrate the effectiveness of the proposed loss.

Comment 2: *The proposed BSL is a slightly enhanced SL. The novelty part of this paper is somewhat not sufficient. Though the claim of SL/BSL is good, more justifications and recommendation models should be added.*

Response 2: Thank you for raising these concerns. First and foremost, we wish to highlight the novelty of our work, which lies not only in proposing a novel loss function but also in conducting a comprehensive analysis of the SL loss's properties. This includes theoretically demonstrating the robustness and fairness of SL and elucidating the role of temperature. We contend that such analyses are crucial, as they provide insights that could enable researchers in this field to deepen their understanding of loss properties and potentially inspire further research along this line. The introduction of our BSL is, in fact, a direct outcome of these analyses. Although BSL is a simple revision of SL, it has been shown to yield significant enhancements in performance across a variety of backbone models and metrics.

Furthermore, we are very grateful for the reviewers' constructive suggestions and have accordingly incorporated additional state-of-the-art recommendation models, metrics, and visualization analysis into our experiments. For further details, the reviewer is directed to our Response 1, 3, 4.

Comment 3: *How about other metric settings? For example, NDCG@5, NDCG@10, NDCG@15.*

Response 3: Thank you for the valuable suggestion. We have included additional experiments with various metrics (*cf.* Figure 7). Our analysis confirms that, across the metrics of NDCG@5, NDCG@10, and NDCG@15, the integration of BSL consistently enhances model performance. This enhancement is substantial enough to allow basic models (*e.g.,* MF, lightGCN) to surpass SOTA models (*e.g.,* SimSGL, NCL).

Comment 4: *Figure 8 shows the visualization of Gowalla by adding some noise data. How about other datasets ?*

Response 4: Thank you for the valuable suggestion. We have conducted visualization experiments on other three datasets and drawn similar conclusions as on Gowalla. However, given the constraints of paper length, we simply report the visualization results from two representative datasets, as illustrated in Figures 10 and 11 .

We sincerely appreciate your recognition of our work and deeply regret any confusion caused by some presentation issues within our paper. Your detailed comments are highly valued. In the revised version, we have refined the paper in accordance with your feedback, rectifying any typographical errors or presentation issues. In the following, we provide responses to the questions you have raised:

Comment 1: *Do the findings still hold with a larger embedding size?*

Response 1: Thank you for the valuable suggestion. In the revised manuscript, we have incorporated further experiments with diverse embedding dimensions, as detailed in Figure 12. Our analysis confirms that, even with increased embedding sizes (*e.g.,* 512), the integration of SL/BSL consistently improves model performance. Notably, MF equipped with SL/BSL still achieves comparable performance with SOTA models under the setting of large embedding sizes.

Comment 2: *How about the sensitivity of BSL* w.r.t. *the hyper-parameter τ_1?*

Response 2: Thanks for the insightful suggestion. In the revised manuscript, we have conducted additional experiments to investigate the model's performance with varying ratios of τ_1/τ_2 by adjusting the hyper-parameter τ_1. The results are illustrated in Figure 13. When the value of τ_1 is excessively large (*e.g.,* $\tau_1/\tau_2 = 2$), which implies an overly small robust radius according to our Corollary III.1, the model has poor robustness against positive noise. As τ_1 decreases (*i.e.,* increasing robustness radius), both the model's robustness and performance are boosted. However, a further reduction of τ_1 can be detrimental to performance, since an excessively

large robust radius might amplify the risk of encountering an implausible worst-case distribution in DRO.

Comment 3: *Why Table 1 does not show the results of SimpleX and NIA-GCN on the Amazon dataset ?*

Response 3: We apologize for the inconvenience, but we encountered difficulties in replicating the results of SimpleX and NIA-GCN on the Amazon dataset. On the one hand, their original papers do not disclose performance on the Amazon dataset. On the other hand, despite our diligent efforts to fine-tune a substantial number of hyperparameters, we were unable to obtain decent results of these two methods.

—————————— **Response to Reviewer #14** ——————————

We appreciate your recognition of our contribution to the connection between SL and DRO. We also express our gratitude for your insightful inquiries regarding BSL. Below, we present responses to your comments:

Comment 1: *The fairness proportion is also relatively small in the paper. Besides, there is still a minor lack of clarity on how the fairness theory (Equation* (13)*) can be used to support empirical results (Figure 4a).*

Response 1: Thank you for the constructive comments to enhance our paper. In the revised manuscript, we have placed greater emphasis on the aspect of fairness within both the Contributions and Conclusions. Furthermore, we provide more analyses on the fairness of SL in Section III.

Meanwhile, the connection between Equation (13) **and Figure 4a stems from the fundamental role of the negative sample variance penalty.** Recent research [26] has indicated that the variance penalty corresponds to fairness. From Equation (13), we can conclude that SL implicitly introduces a regularizer that penalizes the variance of the model predictions on the negative instances. It is important to note that in typical recommender systems, recommendation model is prone to give extensive higher scores on popular items than unpopular ones, incurring popularity bias [27]. SL could mitigate this effect to a certain degree, as it inherently reduces the variance of model predictions, which implies that the predictions are likely to be closer. Consequently, the discrepancy in model predictions between popular and unpopular items is diminished, leading to more fair recommendation results. Unpopular items get more opportunity in recommendation, and thus we observe better NDCG@20 for those unpopular groups (*e.g.,* groups 1-5).

To establish a clear contrast, we conducted ablation studies by removing the variance penalty term and comparing the performance of the standard SL loss with the revised loss. The results are presented in Figure 14. As can be seen, the exclusion of the variance term significantly exacerbates the unfairness of recommendation — *i.e.,* better performance on popular groups (*e.g.,* groups 8-10) while much worse performance on unpopular groups (*e.g.,* groups 1-5).

Comment 2: *1. P* in Figure 2 refers to true distribution but in paragraph it refers to worst case distribution. 2. The term "potential distribution" in Figure 2 does not appear anywhere else, does it mean the worst distribution mentioned before?*
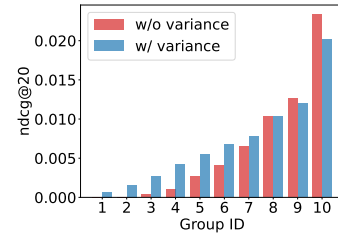

Fig. 14: Performance comparison *w.r.t.* different item groups. The item groups with larger ID suggest these items have larger popularity.

Response 2: Thank you for identifying these problems. In the revised manuscript, we polished Figure 2 and denoted the ideal distribution as p^{ideal} and revised the term "potential distribution" to the "worst-case distribution".

Comment 3: *Why SL performs much worse than BPR, MSE in Figure 4a ?*

Response 3: We apologize for the confusion arising from the previously unclear description. In fact, Figure 4a demonstrate SL exhibits better fairness compared with other losses. Here we classify items into 10 groups according to item popularity and test recommendation performance for each item group [22]. A higher Group ID indicates greater item popularity. As depicted in Figure 4a, the SL loss significantly improves model performance for less popular item groups (groups 1-7), albeit at the cost of reduced performance for more popular item (groups 8-10). This result demonstrates SL achieves more fair recommendation, where the performance gap between popular items and unpopular items has been narrowed. Furthermore, SL achieves overall better performance (as shown in Table II) due to the improvements over the majority of groups. In the revised manuscript, we have provided more detailed explanations in Section III to facilitate a clearer understanding for readers.

Comment 4: *In figures 8 and 9, SL keeps performing bad in Gowalla dataset. The author should explain more than just doubting there are positive noise.*

Response 4: We appreciate your insightful comments. We surmise that the observed phenomenon could be attributed to the characteristics of the Gowalla dataset, which may contain less noisy negative instances than other datasets. Consequently, the benefits of the SL method are less pronounced. Moreover, considering more complex function of SL than other losses, SL might be relatively more hard to optimize. This could explain the suboptimal performance of SL with a very small number of negative instances (*e.g.,* 32) in Figure 9. However, as the number of negative instances increases and the risk of sampling noisy data increases, SL can achieve comparable performance with the best compared losses (*e.g.,* SL performs similar to BCE with 2048 negative instances).

Comment 5: *Can elaborate more on "the fundamental reasons for SL's effectiveness remain poorly explored" mentioned in Abstract.*

Response 5: Thank you for the valuable suggestion. In the revised manuscript, we give better descriptions of recent work on SL in Abstract.