

# Improving Minimax Group Fairness in Sequential Recommendation

Krishna Acharya\*  
Georgia Institute of Technology  
krishna.acharya@gatech.edu

David Wardrope  
Amazon  
dwardrop@amazon.co.uk

Timos Korres  
Amazon  
korres@amazon.co.uk

Aleksandr Petrov†  
University of Glasgow  
a.petrov.1@research.gla.ac.uk

Anders Uhreholt  
Amazon  
akuhren@amazon.co.uk

## Abstract

Training sequential recommenders such as SASRec with uniform sample weights achieves good overall performance but can fall short on specific user groups. One such example is popularity bias, where mainstream users receive better recommendations than niche content viewers. To improve recommendation quality across diverse user groups, we explore three Distributionally Robust Optimization (DRO) methods: Group DRO, Streaming DRO, and Conditional Value at Risk (CVaR) DRO. While Group and Streaming DRO rely on group annotations and struggle with users belonging to multiple groups, CVaR does not require such annotations and can naturally handle overlapping groups. In experiments on two real-world datasets, we show that the DRO methods outperform standard training, with CVaR delivering the best results. Additionally, we find that Group and Streaming DRO are sensitive to the choice of group used for loss computation. Our contributions include (i) a novel application of CVaR to recommenders, (ii) showing that the DRO methods improve group metrics as well as overall performance, and (iii) demonstrating CVaR’s effectiveness in the practical scenario of intersecting user groups. Our code is available at <https://github.com/krishnacharya/sequentialrec-fairness>

## 1 Introduction

Recommender systems play a crucial role in user-facing services, from ecommerce Smith and Linden [2017], Chen et al. [2019b] to videoCovington et al. [2016], music streamingHansen et al. [2020], Bendada et al. [2023] and social networks Eksombatchai et al. [2018], Backstrom and Leskovec [2011]. Among recommender systems, sequential recommenders specialize in next-item prediction, i.e., deciding what to recommend next based on the sequence of items a user has previously interacted with. Transformer-based models Kang and McAuley [2018], Petrov and Macdonald [2023], Sun et al. [2019] excel here and have been deployed at scaleHansen et al. [2020], Bendada et al. [2023]. These model are usually trained using empirical risk minimization (ERM), i.e., by minimizing the loss uniformly for all examples in the training set.

While standard training (ERM) performs well overall, it can be suboptimal for specific user groups Boratto et al. [2022], Li et al. [2021], Rahmani et al. [2022], as models often learn correlations that apply broadly but not within certain groups Beutel et al. [2019], Li et al. [2023]. A common example is popularity bias, where mainstream users receive better recommendations than minorities. In this work, we focus on minimax group fairness Diana et al. [2021], Sagawa et al. [2020], which minimizes the maximum loss across user groups, with the goal of improving metrics for all groups. This differs from traditional group fairness, which aims to equalize metrics across groups Li et al. [2022], Zeng et al. [2024]. We consider two types of user groups: popularity-based groups and sequence length groups. These choices are motivated by prior

\*Lead author. Work done while the author was an intern at Amazon.

†Work done while the author was an intern at Amazon.

findings that recommendation quality differs based on the popularity of items a user interacts with Kowald et al. [2021, 2020], Abdollahpouri et al. [2017], and across users with different sequence lengths Volkovs et al. [2017], Lee et al. [2019]. Importantly, these user groups are chosen because they can be extracted for all datasets, and we expect the findings to generalize to other such attributes.

Previous work Wen et al. [2022] improves minimax group fairness for a non-sequential retrieval model using Group DRO (GDRO) Sagawa et al. [2020] and Streaming DRO (SDRO) Wen et al. [2022]. However, both methods require predefined group annotations and are also unable to handle users belonging to multiple groups. Additionally, attributes such as age, ethnicity are protected under GDPR<sup>1</sup> and not disclosed by users Grosso et al. [2020]. As a result, GDRO and SDRO are inherently limited in achieving fairness for such groups; in fact, focusing only on known attributes can amplify bias Chen et al. [2019a].

In contrast, Conditional Value at Risk DRO (CVaR) Levy et al. [2020] trains models to be robust to a wide range of shifts by identifying subsets of each training mini-batch with the highest losses and updating the model to minimize those. This approach does not require group annotations and dynamically up-weights the highest-loss examples in each mini-batch. In an orthogonal study, Singh et al. Singh et al. investigate safe reinforcement learning and apply a CVaR objective to optimize worst-case reward trajectories. In contrast, we focus on improving minimax fairness across user groups in the standard setting of training a sequential recommender (SASRec Kang and McAuley [2018]). In this paper, we evaluate the effectiveness of CVaR DRO alongside Group DRO and Streaming DRO. Our contributions are:

1. We demonstrate the effectiveness of Conditional Value at Risk DRO for SASRec, a transformer-based sequential recommender. To the best of our knowledge, this is the first application of CVaR DRO to recommender systems for addressing minimax group fairness.
2. Through an in-depth evaluation across group sizes, we show that the DRO methods improve NDCG scores on user groups while also achieving higher overall NDCG compared to standard training. Notably, CVaR, which requires no group information outperforms both group and streaming DRO.
3. We evaluate these methods in a practical scenario with users belonging to intersecting groups, a previously unexplored setup. Here too, the DRO methods outperform standard training, with CVaR often surpassing group and streaming DRO. We also highlight the sensitivity of the GDRO and SDRO results to the group selected for loss computation.

Our experimental evidence suggests that practitioners should prioritize group-agnostic methods like CVaR over group-dependent approaches such as GDRO or SDRO. CVaR not only scales easily to multiple groups but also performs better.

The paper is organised as follows: Section 2 provides background on sequential recommenders and the training methods; Section 3.2 contains experiments with varying group sizes on the Retailrocket dataset, Section 3.3 then evaluates the methods with intersecting user groups; Section 4 concludes the paper; Appendix A repeats the experiments in Section 3 on the MovieLens-1M dataset.

## 2 Sequential recommenders and training methods

In this section, we provide the necessary background on sequential recommenders and the training methods: standard training (empirical risk minimization); the class balanced (CB), inverse propensity weighted (IPW) baselines; and the three distributionally robust methods: Conditional Value at Risk (CVaR) DRO, group DRO and streaming DRO.

### 2.1 Sequential recommenders

In sequential recommendation we want to predict the next item for each user  $u$  given  $H_u = (h_u^1, \dots, h_u^n)$ , a chronological sequence of  $n$  items with which a user has interacted. Each  $h_u^j$  is an item from the item

---

<sup>1</sup>General Data Protection Regulation <https://gdpr-info.eu/>

catalogue  $I$ . There have been several approaches based on Recurrent Neural Networks Hidasi et al. [2016], Yue et al. [2024] and Transformers Kang and McAuley [2018], Sun et al. [2019] towards this, but the main idea is similar: 1) Transform the user sequence  $H_u$  to an embedding  $\in \mathbb{R}^d$ , 2) Multiply this embedding by a matrix of learnt item embeddings  $W_I^\theta \in \mathbb{R}^{|I| \times d}$  to obtain *scores* for each item, 3) The items with the highest scores are sent as candidates to a downstream ranker.

In the context of SASRec, a transformer decoder model, a user sequence  $H_u$  is first padded/truncated to a length  $L$  of most recent items, combined with positional embeddings and then passed through the transformer. For detailed information about the SASRec architecture we refer the reader to Kang and McAuley [2018]. The  $j^{th}$  item in the sequence  $h_u^j$  serves as a target for the previous  $j - 1$  items. Denoting the transformer by  $f^\theta(h) \rightarrow \mathbb{R}^d$  a function from item history to embeddings, the logits and cross entropy loss for user  $u$  at position  $j$  are

$$Logits_{u,j}^\theta = W_I^\theta \times f^\theta([h_u^1, \dots, h_u^{j-1}]) \in \mathbb{R}^{|I|} \quad \ell_{u,j}^\theta = \text{CE}(\text{SoftMax}(Logits_{u,j}^\theta), h_u^j)$$

$\ell_u^\theta = \frac{1}{L} \sum_{j=1}^L \ell_{u,j}^\theta$  is defined as the average cross entropy loss for user  $u$ . Note that while the original SASRec Kang and McAuley [2018] is setup as binary classification with the binary cross entropy loss, recent papers Petrov and Macdonald [2023], Klenitskiy and Vasilev [2023], Zhai et al. [2023] demonstrate that applying softmax over all items followed by the cross entropy loss achieves state-of-the-art accuracy. We adopt this approach in our work.

## 2.2 Standard training (ERM)

Standard training, formally known as Empirical Risk minimization(ERM), minimizes the average loss across users. The loss given a mini-batch of  $B$  users:

$$Loss_{ERM}^\theta = \frac{1}{B} \sum_{u=1}^B \ell_u^\theta = \frac{1}{B} \sum_{u=1}^B \frac{1}{L} \sum_{j=1}^L \ell_{u,j}^\theta \quad (1)$$

## 2.3 Cost-sensitive losses

As simple baselines, we use importance-weighted losses based on item frequency (CB) and group frequency (IPW) Cortes et al. [2010], Han et al. [2024]. These methods have previously been used as effective baselines for improving group metrics in two-tower recommenders Wen et al. [2022]. We also propose a log-weight heuristic as a form of soft-clipping, which often outperforms raw weights in our experiments; these methods are referred to as CBlog and IPWlog, respectively.

### 2.3.1 Class balanced (CB) loss:

Here we weight the loss  $\ell_{u,j}^\theta$  inversely proportional to the frequency of the target item  $h_u^j$ :

$$Loss_{CB}^\theta = \frac{1}{B} \sum_{u=1}^B \frac{1}{L} \sum_{j=1}^L w(h_u^j) \cdot \ell_{u,j}^\theta \quad (2)$$

where the weight  $w(h_u^j) = \frac{\sum_{i \in I} f(i)}{f(h_u^j)}$ ,  $f(h_u^j)$  is the number of times item  $h_u^j$ , appears in the training sequences. CBlog, the variant with log weights replaces  $w(h_u^j)$  by  $\log(w(h_u^j))$ .

### 2.3.2 Inverse propensity weighted (IPW) loss:

Here we weight the loss for user  $u$  inversely proportional to the size of the group( $g_u$ ) it belongs to:

$$Loss_{IPW}^\theta = \frac{1}{B} \sum_{u=1}^B w(g_u) \cdot \ell_u^\theta \quad (3)$$

where the weight  $w(g_u) = \frac{\sum_{g \in G} f(g)}{f(g_u)}$ ,  $f(g_u)$  is the number of users belonging to group  $g_u$  in the training data. Note that for the IPW loss (3) each user must belong to a single group, and it cannot handle intersecting groups.

### 2.3.3 Distributionally robust optimization (DRO):

The main idea for DRO methods is to minimize the expected loss over the worst case distribution which lies within some distance of the empirical training distribution. In this paper, we consider three different DRO training methods. The following contains a brief description, for further details we point the reader to Levy et al. [2020] for Conditional Value at Risk DRO, and Sagawa et al. [2020], Wen et al. [2022] for Group and Streaming DRO respectively.

### 2.3.4 Conditional Value at Risk (CVaR) DRO:

The CVaR loss Levy et al. [2020], Zhai et al. [2021] at level  $\alpha \in (0, 1]$  for a batch of  $B$  training examples

$$Loss_{CVaR}^{\theta}(\alpha) = \sup_{q \in \Delta^B} \left\{ \sum_{u=1}^B q_u \cdot \ell_u^{\theta} \mid \|q\|_{\infty} \leq \frac{1}{\alpha B} \right\}, \quad (4)$$

where  $\Delta^B$  is the probability simplex <sup>2</sup> in  $\mathbb{R}^B$ . The CVaR loss measures how well a model performs over the worst  $\alpha$  fraction of the batch. For e.g., if  $m = \alpha B$  is an integer then the CVaR loss is the average loss over the  $m$  samples that incur the highest losses. In practice, we treat  $\alpha$  as a hyperparameter. A key observation is that the loss (4) does not use group memberships. Thus it does not require groups to be defined upfront, and can directly handle users belonging to multiple groups.

### 2.3.5 Group and Streaming DRO:

Both Group and Streaming DRO build uncertainty sets using group annotations and aim to minimize the maximum loss for a user-group. They maintain a discrete distribution  $\omega$  over non-intersecting user groups, with  $\sum_{g \in G} \omega_g = 1$ , this distribution is first updated using exponentiated gradient ascent with a step-size  $\eta$  Sagawa et al. [2020], which we treat as a hyperparameter in practice. The model parameters  $\theta$  are then updated using the loss in (5) using a first order optimizer. For streaming DRO, given  $L_g^t$  the batch loss for group  $g$ , a streaming estimate <sup>3</sup>  $\tilde{L}_g^t$  is computed for updating the distribution  $\omega$  over the groups Wen et al. [2022]. Group DRO directly uses  $L_g^t$  to update  $\omega$ . The loss is given by

$$Loss_{g/sdro}^{\theta} = \sum_{g \in G} \omega_g^t \cdot L_g^t. \quad (5)$$

### 2.3.6 Loss with intersecting groups:

Note that ERM, CB, and CVaR do not require group memberships and thus generalize to intersecting groups. In contrast, GDRO, SDRO and IPW use group annotations in the loss and are limited to single-group membership per user; we discuss this further in Section 3.3

## 3 Experiments

In this section, we evaluate the training methods from Section 2 in two scenarios:

1. *Single group*: Each user belongs to a single group and we evaluate the training methods across a range group sizes, from balanced to imbalanced.

<sup>2</sup>The probability simplex  $\Delta^B = \{x \in \mathbb{R}^B \mid \sum_{k=1}^B x_k = 1, x_k \geq 0 \forall k\}$

<sup>3</sup>The streaming estimate  $\tilde{L}_g^t = (1 - \beta)L_g^{t-1} + \beta L_g^t$

2. *Intersecting groups*: Each user belongs to intersecting groups, one popularity based and another based on sequence length. We compare the training methods and analyse the sensitivity of the results for group specific methods.

In Section 3.1 we cover the experimental setup while Section 3.2 and 3.3 contain the training method comparisons with single and intersecting groups respectively.

### 3.1 Experimental setup

**Datasets and preprocessing:** We experiment with (i) Retailrocket Ret, an ecommerce dataset from which we use the user views data, and (ii) Movielens-1M Harper and Konstan [2015], a popular movie dataset. We apply an iterative core-5 filtering, ensuring that each user and item has at least 5 interactions. Table 1b summarises the total number of users, items, and interactions.

**User groups:** We define two subgroups,  $G_{\text{pop}} = \{\text{niche, diverse, popular}\}$  and  $G_{\text{seq}} = \{\text{short, medium, long}\}$ . Users belongs to one group from each set.

1.  $G_{\text{pop}}$ : Users with extensive interactions tend to consume more long-tail items Abdollahpouri et al. [2017], so we define groups based on preferences for these items as in Wen et al. [2022]. Specifically, users are grouped by  $r_u$ , the ratio of popular<sup>4</sup> items in their sequence and annotated as niche, diverse, or popular if  $r_u$  falls in the bottom, middle, or top quantiles.
2.  $G_{\text{seq}}$ : Sequence length groups are inspired by cold-start literature Volkovs et al. [2017], Lee et al. [2019], where users with different interaction lengths receive recommendations of varying quality. Users are categorized as short, medium, or long based on whether their sequence length falls in the bottom, middle, or top quantiles.

We compare training methods across different group sizes, using three quantile splits that result in balanced (33% each), semi-balanced (20%, 60%, 20%), and imbalanced (10%, 80%, 10%) groups. Consider the first cell in Table 1a : if the bottom, middle and top quantile sizes for  $r_u$  are 40,16 and 44% respectively then the training data has an equal proportion of niche, diverse, and popular users.

**Backbone:** We use SASRec, a transformer decoder model, but in contrast to the original SASRec Kang and McAuley [2018] which is trained as a binary classification task with negative sampling, we perform multi-class classification with a softmax over the full item corpus followed by the cross entropy loss. Many recent papers Petrov and Macdonald [2023], Klenitskiy and Vasilev [2023], Zhai et al. [2023] show that SASRec trained this way achieves state-of-the-art accuracy.

**Evaluation:** As is typical in sequential recommenders Kang and McAuley [2018], Sun et al. [2019] we evaluate using a leave-one-out data split: we hold out the last item in each user’s item views sequence for testing while the second to last item is used for validation. We perform a grid search to identify the best architectural parameters for both datasets; further details are provided in Appendix B. Following the recommendations in Krichene and Rendle [2020], Cañameres and Castells [2020], we avoid negative sampling when measuring model metrics.

**Training:** After identifying the best architectural parameters and convergence epochs, we train SASRec using this fixed epoch budget across the training methods. Only the DRO-specific<sup>5</sup> parameters are tuned, as in Wen et al. [2022], and model selection is based on the highest overall NDCG@20 on the validation set. Using a fixed epoch budget and architecture ensures a fair comparison across methods.

<sup>4</sup>Popular items are those that fall in the top 0.2 quantile of user interactions Abdollahpouri et al. [2017]

<sup>5</sup>exponentiated gradient step size  $\eta$  for GDRO/SDRO and CVaR level  $\alpha$

### 3.2 Users in a single group

Previous work Wen et al. [2022] defined user groups using a single choice of quantiles then evaluated training methods. However, the performance of the training methods could vary with group size. In this study, we perform a detailed evaluation of the training methods across different group sizes (described in Sec 3.1, *User groups*).

| Dataset split | Popularity groups |               |               | Sequence length groups |               |               |
|---------------|-------------------|---------------|---------------|------------------------|---------------|---------------|
|               | $G_{pop33}$       | $G_{pop2060}$ | $G_{pop1080}$ | $G_{seq33}$            | $G_{seq2060}$ | $G_{seq1080}$ |
| RR dsplit     | (33,33,33)        | (20,60,20)    | (10,80,10)    | (33,33,33)             | (20,60,20)    | (10,80,10)    |
| RR usplit     | (40,16,44)        | (27,45,28)    | (15,70,15)    | (75,23,3)              | (55,44,0.43)  | (34,66,0.09)  |
| ML dsplit     | (33,33,33)        | (20,60,20)    | (10,80,10)    | (33,33,33)             | (20,60,20)    | (10,80,10)    |
| ML usplit     | (18,28,54)        | (10,52,38)    | (5,71,23)     | (73,19,8)              | (59,37,4)     | (42,57,2)     |

(a) Group sizes in the training data(dsplit) and among the users(usplit) in the Retailrocket views and Movielens-1M datasets. For popularity groups the tuple represents (niche,diverse,popular), for sequence length groups it denotes (short,medium,long)

| Dataset | Users | Items | Interactions |
|---------|-------|-------|--------------|
| RR      | 22178 | 17803 | 364943       |
| ML1M    | 6040  | 3416  | 999611       |

(b) Dataset statistics

Table 1: (a) We construct three different group sizes subscribed by (33, 2060, 1080) for both popularity and sequence length groups denoting balanced, semi-balanced and imbalanced sizes. (b) Number of users, items and interactions

| Method | $G_{pop33}$   |               |               |               | $G_{pop2060}$ |               |               |               | $G_{pop1080}$ |               |              |               |
|--------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|--------------|---------------|
|        | Niche         | Diverse       | Popular       | Overall       | Niche         | Diverse       | Popular       | Overall       | Niche         | Diverse       | Popular      | Overall       |
| ERM    | 0.214         | 0.210         | 0.240         | 0.225         | 0.216         | 0.206         | 0.262         | 0.224         | 0.232         | 0.218         | <u>0.268</u> | 0.227         |
| CB     | 0.208*        | 0.205         | 0.229*        | 0.217*        | 0.211*        | 0.202*        | 0.243*        | 0.216*        | 0.231         | 0.211*        | 0.242*       | 0.218*        |
| CBlog  | 0.213         | 0.211         | 0.242         | 0.225         | 0.218         | 0.209         | 0.263         | 0.226         | 0.233         | 0.219         | 0.265        | 0.228         |
| IPW    | 0.214         | 0.210         | 0.244*        | 0.226         | <u>0.220*</u> | 0.201*        | 0.266         | 0.224         | <u>0.238*</u> | 0.217         | 0.264        | 0.227         |
| IPWlog | 0.214         | <u>0.216*</u> | 0.245*        | 0.228*        | 0.217         | 0.204         | 0.260         | 0.223         | 0.233         | 0.215         | 0.255*       | 0.224*        |
| GDRO   | <u>0.215</u>  | 0.213         | 0.248*        | 0.230*        | 0.217         | 0.210*        | <u>0.267*</u> | <u>0.228*</u> | 0.234         | <u>0.220*</u> | 0.264        | <u>0.229</u>  |
| SDRO   | 0.214         | 0.215*        | <u>0.250*</u> | <u>0.230*</u> | 0.219         | <u>0.211*</u> | 0.259         | 0.227*        | 0.233         | 0.220         | 0.266        | 0.228         |
| CVaR   | <b>0.225*</b> | <b>0.228*</b> | <b>0.256*</b> | <b>0.239*</b> | <b>0.232*</b> | <b>0.225*</b> | <b>0.269*</b> | <b>0.239*</b> | <b>0.244*</b> | <b>0.229*</b> | <b>0.268</b> | <b>0.237*</b> |

Table 2: Group and overall NDCG@20 across popularity-based groups on the Retailrocket dataset. \* denotes statistically significant difference to ERM ( $p < 0.05$ , paired T-test). Best in bold, second best is underlined.

We first consider the case in which each user only belongs to either the niche, diverse or popular group. Table 2 records the NDCG@20 on the Retailrocket views dataset. We observe that CVaR achieves the highest NDCG@20 both overall and across user groups, regardless of group size— $G_{pop33}$ ,  $G_{pop2060}$  and  $G_{pop1080}$  (going from balanced to imbalanced).

We now consider the case where each user belongs to the short, medium, or long group. Table 3 shows the NDCG@20 for the Retailrocket views dataset across three group sizes:  $G_{seq33}$ ,  $G_{seq2060}$ , and  $G_{seq1080}$ . CVaR continues to demonstrate strong performance. It is important to note that there are only 20 users with long sequence lengths (0.09% of all users) in the  $G_{seq1080}$  split, which explains why the NDCG@20 values in the penultimate column of Table 3 are not statistically significant. We repeat these experiments on the Movielens-1M dataset in Appendix A.1 with similar insights.

Overall, the summary of this section is that the DRO methods obtain higher NDCGs <sup>6</sup> than standard training across group sizes. CVaR often surpasses GDRO and SDRO, despite not using group information.

### 3.3 Users in intersecting groups

In this section, each user belongs to one of three popularity-based groups and one of three sequence-length groups. We use the  $G_{pop33}$  and  $G_{seq33}$  splits for popularity and sequence length, respectively, ensuring

<sup>6</sup>In a few cases the GDRO and SDRO methods are statistically similar compared to standard training (T-test pvalue > 0.05), barring these they outperform ERM

| Method | $G_{seq33}$    |                |                |                | $G_{seq2060}$  |                |                |                | $G_{seq1080}$  |                |              |                |
|--------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|--------------|----------------|
|        | Short          | Medium         | Long           | Overall        | Short          | Medium         | Long           | Overall        | Short          | Medium         | Long         | Overall        |
| ERM    | 0.191          | 0.337          | 0.229          | 0.225          | 0.159          | 0.309          | 0.088          | 0.225          | 0.127          | <u>0.282</u>   | 0.013        | <u>0.229</u>   |
| CB     | 0.188*         | 0.323*         | 0.218          | 0.219*         | 0.154*         | 0.292*         | 0.075          | 0.215*         | 0.125          | 0.263*         | 0.048        | 0.216*         |
| CBlog  | 0.190          | 0.332*         | 0.228          | 0.224          | 0.157          | 0.307          | 0.090          | 0.223          | 0.129          | 0.277*         | 0.000        | 0.227*         |
| IPW    | 0.194*         | 0.340          | 0.230          | 0.228*         | 0.159          | 0.307          | 0.135*         | 0.225          | 0.120*         | 0.272*         | <b>0.155</b> | 0.221*         |
| IPWlog | 0.193*         | 0.345*         | 0.237          | 0.229*         | 0.157          | 0.309          | 0.131          | 0.224          | 0.119*         | 0.267*         | <u>0.143</u> | 0.217*         |
| GDRO   | <u>0.201</u> * | <u>0.347</u> * | <u>0.277</u> * | <u>0.237</u> * | <u>0.168</u> * | <u>0.315</u> * | 0.064          | <u>0.233</u> * | 0.129          | 0.280          | 0.000        | 0.229          |
| SDRO   | 0.201*         | <b>0.348</b> * | <b>0.279</b> * | <u>0.237</u> * | 0.165*         | 0.313*         | <b>0.152</b> * | 0.231*         | <u>0.130</u> * | 0.279*         | 0.011        | 0.229          |
| CVaR   | <b>0.207</b> * | 0.344*         | 0.271*         | <b>0.240</b> * | <b>0.172</b> * | <b>0.320</b> * | <u>0.136</u> * | <b>0.237</b> * | <b>0.137</b> * | <b>0.289</b> * | 0.083        | <b>0.237</b> * |

Table 3: Group and overall NDCG@20 across sequence-length groups on the Retailrocket dataset. \* denotes statistically significant difference to ERM ( $p < 0.05$ , paired T-test). Best in bold, second best is underlined.

a balanced training set with 33% niche, diverse, popular, and 33% short, medium, long sequence-length users. Table 4 shows the NDCG@20 for the Retailrocket views dataset, where CVaR achieves the highest NDCG@20, except for long-sequence users, where it ranks second.

Recall that the GDRO, SDRO methods require each user to belong to a single group. To apply these methods, we limit the loss computation to either sequence length or popularity groups. The methods are then referred to with the corresponding subscript such as  $SDRO_{seq}$  and  $SDRO_{pop}$ . A key observation for these methods is their sensitivity to the choice of groups. As seen in Figure 1 using popularity groups in the loss calculation yields worse relative improvements than using sequence length groups –  $GDRO_{pop}$  and  $SDRO_{pop}$  perform worse than  $GDRO_{seq}$  and  $SDRO_{seq}$  respectively. We repeat this experiment on the Movielens-1M dataset in Appendix A.2, observing similar results.

The conclusion from this section is that training methods reliant on group annotations, like GDRO and SDRO, are sensitive and do not scale to multiple groups. It is impossible to know beforehand which subgroup in the loss will yield the best NDCG scores, and training multiple variants is impractical. While we defined two intersecting subgroups in this paper, real-world scenarios often involve numerous subgroups. As a result, training multiple models with  $SDRO_{pop}$ ,  $SDRO_{seq}$ ,  $SDRO_{subgroup-n}$  is impractical and even impossible with unknown subgroups. In contrast, group-agnostic methods like CVaR easily handle multiple subgroups, outperform standard training and even group and streaming DRO.

| Method                | Niche           | Diverse         | Popular         | Short           | Medium          | Long            | Overall         |
|-----------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| ERM                   | 0.2122          | 0.2080          | 0.2381          | 0.1859          | 0.3418          | 0.2364          | 0.2230          |
| CB                    | 0.2115          | 0.2036          | 0.2280*         | 0.1861          | 0.3195*         | 0.2161*         | 0.2175*         |
| CBlog                 | 0.2154*         | 0.2126          | 0.2424*         | 0.1927*         | 0.3378          | 0.2258          | 0.2269*         |
| CVaR                  | <b>0.2278</b> * | <b>0.2286</b> * | <b>0.2574</b> * | <b>0.2068</b> * | <b>0.3475</b> * | <u>0.2787</u> * | <b>0.2409</b> * |
| IPW <sub>pop</sub>    | 0.2137          | 0.2102          | 0.2432*         | 0.1905*         | 0.3415          | 0.2327          | 0.2262*         |
| IPWlog <sub>pop</sub> | 0.2139          | 0.2093          | 0.2401          | 0.1873          | 0.3444          | 0.2400          | 0.2247          |
| GDRO <sub>pop</sub>   | 0.2163*         | 0.2139*         | 0.2473*         | 0.1940*         | 0.3441          | 0.2402          | 0.2296*         |
| SDRO <sub>pop</sub>   | 0.2181*         | 0.2135*         | 0.2459*         | 0.1938*         | 0.3445          | 0.2446          | 0.2297*         |
| IPW <sub>seq</sub>    | 0.2132          | 0.2098          | 0.2445*         | 0.1912*         | 0.3401          | 0.2352          | 0.2264*         |
| IPWlog <sub>seq</sub> | 0.2157*         | 0.2115          | 0.2455*         | 0.1929*         | 0.3423          | 0.2312          | 0.2282*         |
| GDRO <sub>seq</sub>   | 0.2242*         | 0.2245*         | 0.2508*         | 0.2009*         | 0.3465*         | 0.2654*         | 0.2360*         |
| SDRO <sub>seq</sub>   | <u>0.2249</u> * | <u>0.2264</u> * | <u>0.2528</u> * | <u>0.2021</u> * | <u>0.3474</u> * | <b>0.2801</b> * | <u>0.2374</u> * |

Table 4: Group and overall NDCG@20 with users in intersecting groups; Retailrocket dataset, \* denotes statistically significant difference to ERM ( $p < 0.05$ , paired T-test). Best in bold, second best is underlined.

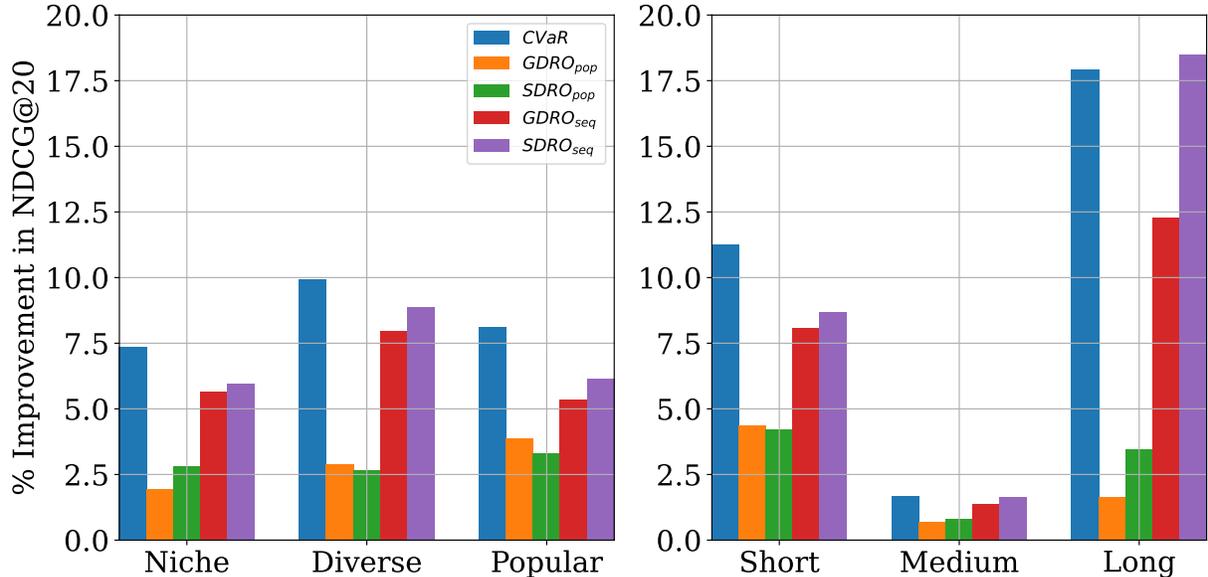


Figure 1: Percentage increase in NDCG@20 using DRO methods relative to standard training on the Retailrocket dataset: (i) increase for niche, diverse and popular groups (ii) increase for short, medium, and long sequence users.

## 4 Conclusion

In this paper, we demonstrated the effectiveness of Conditional Value at Risk DRO in improving both group and overall NDCG scores in sequential recommenders like SASRec. CVaR consistently outperforms Group DRO and Streaming DRO across various group sizes and in scenarios with intersecting user groups. Based on our experiments, we suggest that practitioners prioritize group-agnostic methods like CVaR over group-dependent approaches, as CVaR not only scales easily to multiple groups but also performs better. Although we have restricted attention to group attributes that can be inferred from interaction history, such as mainstream inclination and sequence lengths, we note that important user attributes such as age and ethnicity will commonly be protected and unknown. This effectively renders group-dependent methods inapplicable and strengthens our argument that practitioners should prioritise methods like CVaR.

## References

- Retailrocket recommender system dataset — kaggle.com. <https://www.kaggle.com/datasets/retailrocket/ecommerce-dataset>.
- Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. Controlling popularity bias in learning-to-rank recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, page 42–46, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450346528. doi: 10.1145/3109859.3109912. URL <https://doi.org/10.1145/3109859.3109912>.
- Lars Backstrom and Jure Leskovec. Supervised random walks: predicting and recommending links in social networks. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 635–644, 2011.
- Walid Bendada, Théo Bontempelli, Mathieu Morlon, Benjamin Chapus, Thibault Cador, Thomas Bouabça,

- and Guillaume Salha-Galvan. Track mix generation on music streaming services using transformers. In *Proceedings of the 17th ACM Conference on Recommender Systems*, RecSys '23, page 112–115, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 978400702419. doi: 10.1145/3604915.3608869. URL <https://doi.org/10.1145/3604915.3608869>.
- Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H. Chi, and Cristos Goodrow. Fairness in recommendation ranking through pairwise comparisons. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 2212–2220, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330745. URL <https://doi.org/10.1145/3292500.3330745>.
- Ludovico Boratto, Gianni Fenu, Mirko Marras, and Giacomo Medda. Consumer fairness in recommender systems: Contextualizing definitions and mitigations. In Matthias Hagen, Suzan Verberne, Craig Macdonald, Christin Seifert, Krisztian Balog, Kjetil Nørnvåg, and Vinay Setty, editors, *Advances in Information Retrieval*, pages 552–566, Cham, 2022. Springer International Publishing. ISBN 978-3-030-99736-6.
- Rocío Cañamares and Pablo Castells. On target item sampling in offline recommendersystem;evaluation. In *Proceedings of the 14th ACM Conference on Recommender Systems*, RecSys '20, page 259–268, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450375832. doi: 10.1145/3383313.3412259. URL <https://doi.org/10.1145/3383313.3412259>.
- Jiahao Chen, Nathan Kallus, Xiaojie Mao, Geoffrey Svacha, and Madeleine Udell. Fairness under unawareness: Assessing disparity when protected class is unobserved. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 339–348, 2019a.
- Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, and Wenwu Ou. Behavior sequence transformer for e-commerce recommendation in alibaba. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*, DLP-KDD '19, New York, NY, USA, 2019b. Association for Computing Machinery. ISBN 9781450367837. doi: 10.1145/3326937.3341261. URL <https://doi.org/10.1145/3326937.3341261>.
- Corinna Cortes, Yishay Mansour, and Mehryar Mohri. Learning bounds for importance weighting. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010. URL [https://proceedings.neurips.cc/paper\\_files/paper/2010/file/59c33016884a62116be975a9bb8257e3-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2010/file/59c33016884a62116be975a9bb8257e3-Paper.pdf).
- Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, page 191–198, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450340359. doi: 10.1145/2959100.2959190. URL <https://doi.org/10.1145/2959100.2959190>.
- Emily Diana, Wesley Gill, Michael Kearns, Krishnamurthy Suresh, and Aaron Roth. Minimax group fairness: Algorithms and experiments. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '21, page 66–76, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384735. doi: 10.1145/3461702.3462523. URL <https://doi.org/10.1145/3461702.3462523>.
- Chantat Eksombatchai, Pranav Jindal, Jerry Zitao Liu, Yuchen Liu, Rahul Sharma, Charles Sugnet, Mark Ulrich, and Jure Leskovec. Pixie: A system for recommending 3+ billion items to 200+ million users in real-time. In *Proceedings of the 2018 world wide web conference*, pages 1775–1784, 2018.
- Monica Grosso, Sandro Castaldo, Hua (Ariel) Li, and Bart Larivière. What information do shoppers share? the effect of personnel-, retailer-, and country-trust on willingness to share information. *Journal of Retailing*, 96(4):524–547, 2020. ISSN 0022-4359. doi: <https://doi.org/10.1016/j.jretai.2020.08.002>. URL <https://www.sciencedirect.com/science/article/pii/S0022435920300440>.

- Zongbo Han, Zhipeng Liang, Fan Yang, Liu Liu, Lanqing Li, Yatao Bian, Peilin Zhao, Bingzhe Wu, Changqing Zhang, and Jianhua Yao. U MIX: improving importance weighting for subpopulation shift via uncertainty-aware mixup. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9781713871088.
- Casper Hansen, Christian Hansen, Lucas Maystre, Rishabh Mehrotra, Brian Brost, Federico Tomasi, and Mounia Lalmas. Contextual and sequential user embeddings for large-scale music recommendation. In *Proceedings of the 14th ACM Conference on Recommender Systems*, RecSys '20, page 53–62, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450375832. doi: 10.1145/3383313.3412248. URL <https://doi.org/10.1145/3383313.3412248>.
- F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4), dec 2015. ISSN 2160-6455. doi: 10.1145/2827872. URL <https://doi.org/10.1145/2827872>.
- Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL <http://arxiv.org/abs/1511.06939>.
- Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 197–206, 2018. doi: 10.1109/ICDM.2018.00035.
- Anton Klenitskiy and Alexey Vasilev. Turning dross into gold loss: is bert4rec really better than sasrec? In *Proceedings of the 17th ACM Conference on Recommender Systems*, RecSys '23, page 1120–1125, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400702419. doi: 10.1145/3604915.3610644. URL <https://doi.org/10.1145/3604915.3610644>.
- Dominik Kowald, Markus Schedl, and Elisabeth Lex. The unfairness of popularity bias in music recommendation: A reproducibility study. In *Advances in Information Retrieval*, pages 35–42, Cham, 2020. Springer International Publishing. ISBN 978-3-030-45442-5.
- Dominik Kowald, Peter Muellner, Eva Zangerle, Christine Bauer, Markus Schedl, and Elisabeth Lex. Support the underground: characteristics of beyond-mainstream music listeners. *EPJ Data Science*, 10(1):14, 2021.
- Walid Krichene and Steffen Rendle. On sampled metrics for item recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 1748–1757, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450379984. doi: 10.1145/3394486.3403226. URL <https://doi.org/10.1145/3394486.3403226>.
- Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. Melu: Meta-learned user preference estimator for cold-start recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 1073–1082, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330859. URL <https://doi.org/10.1145/3292500.3330859>.
- Daniel Levy, Yair Carmon, John C Duchi, and Aaron Sidford. Large-scale methods for distributionally robust optimization. *Advances in Neural Information Processing Systems*, 33:8847–8860, 2020.
- Cheng-Te Li, Cheng Hsu, and Yang Zhang. FairSR: Fairness-aware Sequential Recommendation through Multi-Task Learning with Preference Graph Embeddings. *ACM Trans. Intell. Syst. Technol.*, 13(1), February 2022. ISSN 2157-6904. doi: 10.1145/3495163. URL <https://doi.org/10.1145/3495163>.
- Yunqi Li, Hanxiong Chen, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. User-oriented fairness in recommendation. In *Proceedings of the Web Conference 2021*, WWW '21, page 624–632, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383127. doi: 10.1145/3442381.3449866. URL <https://doi.org/10.1145/3442381.3449866>.

- Yunqi Li, Hanxiong Chen, Shuyuan Xu, Yingqiang Ge, Juntao Tan, Shuchang Liu, and Yongfeng Zhang. Fairness in recommendation: Foundations, methods, and applications. *ACM Trans. Intell. Syst. Technol.*, 14(5), October 2023. ISSN 2157-6904. doi: 10.1145/3610302. URL <https://doi.org/10.1145/3610302>.
- Aleksandr V. Petrov and Craig Macdonald. Recj pq: Training large-catalogue sequential recommenders. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining, WSDM '24*, page 538–547, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400703713. doi: 10.1145/3616855.3635821. URL <https://doi.org/10.1145/3616855.3635821>.
- Aleksandr Vladimirovich Petrov and Craig Macdonald. gSASRec: Reducing Overconfidence in Sequential Recommendation Trained with Negative Sampling. In *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys '23*, page 116–128, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400702419. doi: 10.1145/3604915.3608783. URL <https://doi.org/10.1145/3604915.3608783>.
- Hossein A. Rahmani, Mohammadmehdi Naghiaei, Mahdi Dehghan, and Mohammad Aliannejadi. Experiments on generalizability of user-oriented fairness in recommender systems. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '22*, page 2755–2764, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450387323. doi: 10.1145/3477495.3531718. URL <https://doi.org/10.1145/3477495.3531718>.
- Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2020. URL <https://openreview.net/forum?id=ryxGuJrFvS>.
- Ashudeep Singh, Yoni Halpern, Nithum Thain, Konstantina Christakopoulou, Jilin Chen, and Alex Beutel. Building healthy recommendation sequences for everyone: A safe reinforcement learning approach.
- Brent Smith and Greg Linden. Two decades of recommender systems at amazon.com. *IEEE Internet Computing*, 2017. URL <https://www.amazon.science/publications/two-decades-of-recommender-systems-at-amazon-com>.
- Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. *CIKM '19*, page 1441–1450, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450369763. doi: 10.1145/3357384.3357895. URL <https://doi.org/10.1145/3357384.3357895>.
- Maksims Volkovs, Guangwei Yu, and Tomi Poutanen. Dropoutnet: Addressing cold start in recommender systems. *Advances in neural information processing systems*, 30, 2017.
- Hongyi Wen, Xinyang Yi, Tiansheng Yao, Jiayi Tang, Lichan Hong, and Ed H. Chi. Distributionally-robust recommendations for improving worst-case user experience. In *Proceedings of the ACM Web Conference 2022, WWW '22*, page 3606–3610, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450390965. doi: 10.1145/3485447.3512255. URL <https://doi.org/10.1145/3485447.3512255>.
- Zhenrui Yue, Yueqi Wang, Zhankui He, Huimin Zeng, Julian McAuley, and Dong Wang. Linear recurrent units for sequential recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining, WSDM '24*, page 930–938, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400703713. doi: 10.1145/3616855.3635760. URL <https://doi.org/10.1145/3616855.3635760>.
- Huimin Zeng, Zhankui He, Zhenrui Yue, Julian McAuley, and Dong Wang. Fair sequential recommendation without user demographics. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '24*, page 395–404, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704314. doi: 10.1145/3626772.3657703. URL <https://doi.org/10.1145/3626772.3657703>.

Jiaqi Zhai, Zhaojie Gong, Yueming Wang, Xiao Sun, Zheng Yan, Fu Li, and Xing Liu. Revisiting neural retrieval on accelerators. KDD '23, page 5520–5531, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701030. doi: 10.1145/3580305.3599897. URL <https://doi.org/10.1145/3580305.3599897>.

Runtian Zhai, Chen Dan, Arun Suggala, J Zico Kolter, and Pradeep Ravikumar. Boosted cvar classification. *Advances in Neural Information Processing Systems*, 34:21860–21871, 2021.

## A Experiments on the Movielens-1M dataset

### A.1 Users in a single group (Movielens-1M)

We repeat the experiments for users in a single group (Section 3.2) on the Movielens-1M dataset. Table 5 and Table 6 record the NDCG@20 for users in popularity-based groups and sequence length groups respectively. Here too we observe that the DRO methods obtain higher NDCGs than standard training across group sizes, with CVaR often surpassing GDRO and SDRO.

| Method  | $G_{pop33}$  |                |                |                | $G_{pop2060}$ |                |              |                | $G_{pop1080}$ |                |              |                |
|---------|--------------|----------------|----------------|----------------|---------------|----------------|--------------|----------------|---------------|----------------|--------------|----------------|
|         | Niche        | Diverse        | Popular        | Overall        | Niche         | Diverse        | Popular      | Overall        | Niche         | Diverse        | Popular      | Overall        |
| ERM     | 0.189        | 0.179          | 0.208          | 0.196          | <b>0.217</b>  | 0.183          | 0.213        | <u>0.198</u>   | 0.229         | 0.190          | 0.205        | 0.196          |
| CB      | 0.123*       | 0.109*         | 0.126*         | 0.121*         | 0.137*        | 0.113*         | 0.127*       | 0.121*         | 0.152*        | 0.112*         | 0.126*       | 0.117*         |
| CB log  | 0.187        | 0.178          | 0.204          | 0.194          | 0.209*        | 0.181          | 0.208*       | 0.194*         | 0.234         | 0.188          | 0.201        | 0.194          |
| IPW     | 0.188        | 0.180          | 0.208          | 0.196          | 0.200*        | 0.170*         | 0.206*       | 0.187*         | 0.190*        | 0.162*         | 0.182*       | 0.168*         |
| IPW log | 0.188        | 0.180          | 0.208          | 0.197          | 0.201*        | 0.171*         | 0.207*       | 0.188*         | 0.177*        | 0.157*         | 0.183*       | 0.164*         |
| GDRO    | <u>0.194</u> | <b>0.184</b> * | 0.211          | <u>0.201</u> * | 0.210         | <u>0.184</u>   | 0.212        | 0.197          | 0.226         | 0.194*         | 0.204        | 0.198          |
| SDRO    | 0.191        | <u>0.184</u>   | <b>0.212</b> * | 0.201*         | 0.214         | 0.182          | <u>0.213</u> | 0.197          | <u>0.237</u>  | <u>0.196</u> * | <b>0.209</b> | <u>0.201</u> * |
| CVaR    | <b>0.196</b> | 0.183          | <u>0.212</u> * | <b>0.201</b> * | <u>0.215</u>  | <b>0.188</b> * | <b>0.214</b> | <b>0.201</b> * | <b>0.238</b>  | <b>0.196</b> * | <u>0.209</u> | <b>0.201</b> * |

Table 5: Group and overall NDCG@20 across popularity-based groups on the Movielens-1M dataset. \* denotes statistically significant difference to ERM ( $p < 0.05$ , paired T-test). Best in bold, second best is underlined.

| Method | $G_{seq33}$  |                |              |              | $G_{seq2060}$  |                |              |                | $G_{seq1080}$  |                |              |                |
|--------|--------------|----------------|--------------|--------------|----------------|----------------|--------------|----------------|----------------|----------------|--------------|----------------|
|        | Short        | Medium         | Long         | Overall      | Short          | Medium         | Long         | Overall        | Short          | Medium         | Long         | Overall        |
| ERM    | 0.220        | 0.139          | 0.132        | 0.198        | 0.227          | 0.156          | 0.131        | 0.197          | 0.238          | 0.167          | 0.138        | 0.196          |
| CB     | 0.136*       | 0.078*         | 0.081*       | 0.121*       | 0.138*         | 0.090*         | 0.079*       | 0.118*         | 0.149*         | 0.102*         | 0.081*       | 0.121*         |
| CBlog  | 0.216*       | 0.136          | 0.127        | 0.194*       | 0.224          | 0.153          | 0.126        | 0.194*         | 0.237          | 0.164*         | 0.123*       | 0.194*         |
| IPW    | 0.218*       | 0.135          | 0.132        | 0.196*       | 0.213*         | 0.143*         | 0.103*       | 0.183*         | 0.211*         | 0.128*         | 0.091*       | 0.162*         |
| IPWlog | 0.219        | 0.140          | 0.137        | 0.197        | 0.205*         | 0.141*         | 0.105*       | 0.177*         | 0.207*         | 0.126*         | 0.081*       | 0.159*         |
| GDRO   | <u>0.222</u> | <u>0.148</u>   | <u>0.143</u> | <b>0.202</b> | <u>0.235</u> * | <b>0.167</b> * | <u>0.147</u> | <b>0.206</b> * | <u>0.252</u> * | <b>0.176</b> * | <u>0.151</u> | <b>0.207</b> * |
| SDRO   | 0.221        | <b>0.149</b> * | <b>0.145</b> | <u>0.202</u> | <b>0.235</b> * | <u>0.164</u> * | <b>0.148</b> | <u>0.205</u> * | <b>0.253</b> * | <u>0.175</u> * | 0.145        | <u>0.207</u> * |
| CVaR   | <b>0.222</b> | 0.143          | 0.139        | 0.201        | 0.230          | 0.161*         | 0.137        | 0.201*         | 0.249*         | 0.174*         | <b>0.155</b> | 0.205*         |

Table 6: Group and overall NDCG@20 across sequence-length groups on the Movielens-1M dataset. \* denotes statistically significant difference to ERM ( $p < 0.05$ , paired T-test). Best in bold, second best is underlined.

### A.2 Users in intersecting groups (Movielens-1M)

We repeat the experiments for users in intersecting groups (Section 3.3) on the Movielens-1M dataset. Table 7 presents the NDCG@20 scores, showing that CVaR achieves the highest NDCG@20 overall and across

most user groups. Figure 2 again illustrates the sensitivity of the GDRO and SDRO results to the choice of groups in the loss computation.

| Method                           | Niche          | Diverse        | Popular        | Short          | Medium         | Long          | Overall        |
|----------------------------------|----------------|----------------|----------------|----------------|----------------|---------------|----------------|
| ERM                              | 0.1859         | 0.1793         | 0.2092         | 0.2194         | 0.1350         | 0.1314        | 0.1966         |
| CB                               | 0.1252*        | 0.1099*        | 0.1254*        | 0.1357*        | 0.0792*        | 0.0822*       | 0.1209*        |
| CBlog                            | 0.1870         | 0.1781         | 0.2035*        | 0.2155*        | 0.1354         | 0.1257        | 0.1934*        |
| CVaR                             | <b>0.1994*</b> | <b>0.1861*</b> | <b>0.2142*</b> | <b>0.2252*</b> | 0.1449*        | <u>0.1433</u> | <b>0.2037*</b> |
| IPW <sub>pop</sub>               | 0.1875         | 0.1777         | 0.2064*        | 0.2171         | 0.1368         | 0.1276        | 0.1949         |
| IPW <sub>log<sub>pop</sub></sub> | 0.1913*        | 0.1796         | 0.2074         | 0.2197         | 0.1338         | 0.1327        | 0.1967         |
| GDRO <sub>pop</sub>              | 0.1927         | 0.1812         | 0.2115         | 0.2220         | 0.1414*        | 0.1307        | 0.1996*        |
| SDRO <sub>pop</sub>              | 0.1901         | <u>0.1842</u>  | 0.2124         | 0.2215         | 0.1455*        | 0.1365        | 0.2005*        |
| IPW <sub>seq</sub>               | 0.1915*        | 0.1787         | 0.2088         | 0.2190         | 0.1385         | 0.1351        | 0.1972         |
| IPW <sub>log<sub>seq</sub></sub> | 0.1921*        | 0.1793         | 0.2097         | 0.2200         | 0.1399*        | 0.1319        | 0.1980         |
| GDRO <sub>seq</sub>              | <u>0.1989*</u> | 0.1821         | <u>0.2142</u>  | <u>0.2226</u>  | <u>0.1484*</u> | <b>0.1449</b> | <u>0.2025*</u> |
| SDRO <sub>seq</sub>              | 0.1977*        | 0.1812         | 0.2132         | 0.2213         | <b>0.1491*</b> | 0.1429        | 0.2015*        |

Table 7: Group and overall NDCG@20 with users in intersecting groups; MovieLens-1M dataset, \* denotes statistically significant difference to ERM ( $p < 0.05$ , paired T-test). Best in bold, second best is underlined.

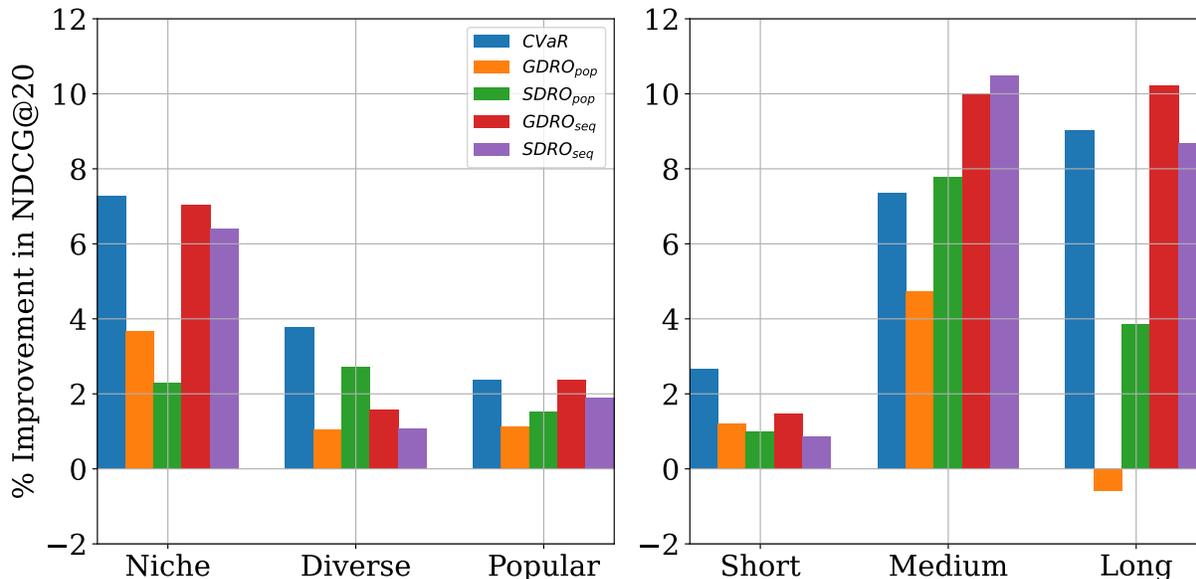


Figure 2: Percentage increase in NDCG@20 using DRO methods relative to standard training on the MovieLens-1M dataset: (i) increase for niche, diverse, and popular groups, and (ii) increase for short, medium, and long sequence users.

## B Hyperparameters

**Backbone search:** We conduct a grid search to determine the best architectural parameters for each dataset. Following Klenitskiy and Vasilev [2023], Petrov and Macdonald [2024], Kang and McAuley [2018],

we truncate or pad sequences to  $L = 200$  most recent interactions, vary embedding dimensions in  $\{128, 256\}$ , feed-forward dimension in  $\{128, 256\}$ , number of transformer blocks in  $\{1, 2, 3\}$ , and the number of attention heads in  $\{1, 2, 4\}$ . We also test dropout values in  $\{0.1, 0.2, 0.5\}$ , use the Adam optimizer with a learning rate of 0.001, and vary batch size  $B$  in  $\{128, 256\}$ . For each configuration, we run standard training and early stop if validation NDCG@20 does not improve for 200 epochs, each epoch consists of 128 mini-batches. The best backbone configurations are reported in Table 8.

| Dataset | emb-dim | ff-dim | #blocks | #heads | dropout | B   |
|---------|---------|--------|---------|--------|---------|-----|
| RR      | 256     | 256    | 3       | 1      | 0.2     | 128 |
| ML1M    | 256     | 256    | 3       | 1      | 0.5     | 256 |

Table 8: Best SASRec parameters for the Retailrocket, Movielens-1M datasets

**Computational resources:** We launch training jobs on Amazon SageMaker using ml.p3.2xlarge instances, each with a Tesla V100 GPU (16GB memory). A single training job takes approximately 8 hours for the Retailrocket dataset and 6 hours for the MovieLens-1M dataset. For GDRO and SDRO, we run five jobs with  $\eta$  values from  $\{1e^{-3}, 5e^{-3}, 1e^{-2}, 5e^{-2}, 0.1\}$ , and for CVaR, ten jobs with  $\alpha$  levels from  $\{0.1, 0.2 \dots 0.9, 0.95\}$ . In Section 3.3, for users in two intersecting groups, we train two variants for group-specific methods (e.g.,  $\text{GDRO}_{\text{pop}}$  and  $\text{GDRO}_{\text{seq}}$ ), launching 10 jobs each for GDRO and SDRO.