Table 2: The statistics of Yahoo!R3, KuaiRand-Pure, and Coat.

Dataset	#User	#Item	#Biased Data	#Unbiased Data
Yahoo!R3	15,400	1,000	311,704	54,000
KuaiRand-Pure	27,077	7,551	1,375,000	1,177,026
Coat	290	300	6,960	4,640

A Experiment Detail

A.1 Dataset

The statistics of datasets are shown in Table 2.

A.2 Evaluation Metric

We rank the test items for each user u based on DIRL's prediction and denote the corresponding rank of each item i as $R(i|u)$. Accordingly, we adopt the following top-K metrics to evaluate ranking performance in the test data $\mathcal{D}_{Te}$:

- **NDCG@K** measures the quality of recommendation through discounted importance based on rank:

$$\begin{aligned} DCG_u@K &= \sum_{(u,i)\in\mathcal{D}_{Te}} \frac{\delta\left(R\left(i|u\right)\leq K\right)}{log(R\left(i|u\right)+1)} \\ NDCG@K &= \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\frac{DCG_u@K}{IDCG_u@K} \end{aligned} \tag{17}$$

  where $IDCG_u@K$ is the ideal $DCG_u@K$.

- **Recall@K** measures how many items that users like are recommended:

$$\begin{aligned} Recall_u@K &= \frac{\sum_{(u,i)\in\mathcal{D}_{Te}}\delta\left(R\left(i|u\right)\leq K\right)}{|\mathcal{D}^u_{Te}|} \\ Recall@K &= \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}} Recall_u@K \end{aligned} \tag{18}$$

  where $\mathcal{D}^u_{Te}$ is the test data corresponding to the user u.

- **Precision@K** measures how many recommended items are liked by users:

$$\begin{aligned} Precision_u@K &= \frac{\sum_{(u,i)\in\mathcal{D}_{Te}}\delta\left(R\left(i|u\right)\leq K\right)}{K} \\ Precision@K &= \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}} Precision_u@K \end{aligned} \tag{19}$$

A.3 Baselines

We compare DIRL with the following methods.

- **IPS** [Schnabel *et al.*, 2016]: IPS reweights the training data with item popularity as the propensity score.
- **DRJL** [Wang *et al.*, 2019]: DRJL performs propensity-based imputation learning, which combines IPS and the imputation model trained by IPS to train the base model.
- **MRDR** [Guo *et al.*, 2021]: MRDR extends DRJL by reducing the variance of propensity-based imputation learning.
- **CVIB** [Wang *et al.*, 2020]: CVIB employs an information contrastive loss and a prediction confidence penalty to balance the biased and unbiased data.

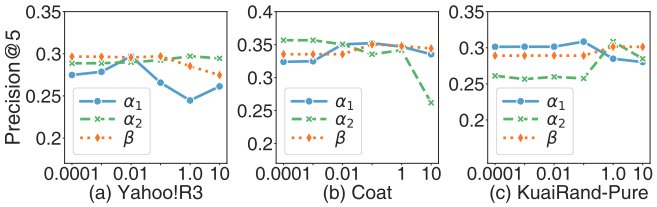

Figure 7: Hyperparameter sensitivity analysis for Precision@5 in Yahoo!R3, Coat, and KuaiRand-Pure.

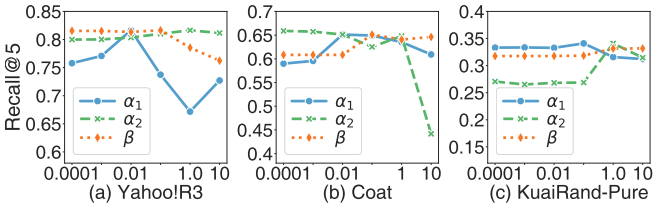

Figure 8: Hyperparameter sensitivity analysis for Recall@5 in Yahoo!R3, Coat, and KuaiRand-Pure.

- **DAMF** [Saito and Nomura, 2022]. DAMF aligns the distributions of predictions on the biased and unbiased data by adversarial learning.
- **InvPref** [Wang *et al.*, 2022]: Invpref iteratively construct heterogeneous environments by clustering in the training data and disentangles variant and invariant representation by adversarial learning.

All compared methods are based on the classic MF model.

A.4 Hyperparameter Setting

We implement DIRL in PyTorch [Paszke *et al.*, 2019]. The embedding size is 64 on Yahoo!R3 and Coat. Due to limited resources, we set the embedding size to 10 on KuaiRand-Pure. We identify the best values for the rest hyperparameters by grid search. In particular, we use a small part of the unbiased data for validation (5% on Yahoo!R3, 5% on KuaiRand-Pure, and 20% on Coat[6]) and the rest for testing. We tune the learning rate in the range of $\{0.001, 0.005, 0.01, 0.05, 0.1\}$ and the weight decay in the range of $\{1e-6, 1e-5, 1e-4, 1e-3\}$. For the three trade-off parameters, α_1 and α_2 are both tuned in the range of $\{0.001, 0.01, 0.1, 1.0\}$ and β is tuned in the range of $\{0.01, 0.1, 1.0, 5.0, 10.0\}$. Furthermore, we re-implement the above baselines in PyTorch [Paszke *et al.*, 2019] following the official codes to match our experiment setting. And we tune the hyperparameters of them as recommended in original papers (Optuna [Akiba *et al.*, 2019] for DAMF and InvPref, grid search for others). All methods are optimized by Adam [Kingma and Ba, 2014].

A.5 Hyperparameter Sensitivity Analysis

Figure 7 and Figure 8 show the results of hyperparameter sensitivity analysis in three different datasets for precision@5 and recall@5 respectively.

[6]Coat is a very small dataset, which only provides 6960 biased data and 4640 unbiased data. In order to provide stable test results, we divide more data from the unbiased data for validation.

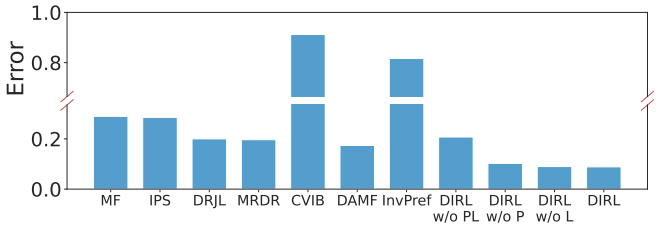

Figure 9: Empirical error on unbiased data in Yahoo!R3.

A.6 In-depth Analysis

Error on unbiased data. In addition to the superior ranking performance in Table 1, we further calculate the classification error of DIRL's representation on the unbiased data. Figure 9 shows that both prior-guided contrasting and label-conditional clustering reduces the error on unbiased data to varying degrees compared to other methods, which is consistent with our conclusion in Section 2. As to the abnormal errors of CVIB and InvPref, we postulate the reason is that we tune hyperparameters by ranking performance but a high ranking performance may not obtain a low error.