Distributionally Robust Graph-based Recommendation System

Bohao Wang
The State Key Laboratory of
Blockchain and Data Security,
Zhejiang University
Hangzhou, China
bohao.wang@zju.edu.cn

Jiawei Chen
Zhejiang University
Hangzhou, China
sleepyhunt@zju.edu.cn

Changdong Li
Zhejiang University
Hangzhou, China
lichangdongtw@zju.edu.cn

Sheng Zhou
Zhejiang University
Hangzhou, China
zhousheng_zju@zju.edu.cn

Qihao Shi
Zhejiang University
Hangzhou, China
shiqihao321@zju.edu.cn

Yang Gao
Intelligence Indeed
Hangzhou, China
duizhang@i-i.ai

Yan Feng
Zhejiang University
Hangzhou, China
fengyan@zju.edu.cn

Chun Chen
Zhejiang University
Hangzhou, China
chenc@zju.edu.cn

Can Wang
The State Key Laboratory of
Blockchain and Data Security,
Zhejiang University
Hangzhou, China
wcan@zju.edu.cn

ABSTRACT

With the capacity to capture high-order collaborative signals, Graph Neural Networks (GNNs) have emerged as powerful methods in Recommender Systems (RS). However, their efficacy often hinges on the assumption that training and testing data share the same distribution (*a.k.a.* IID assumption), and exhibits significant declines under distribution shifts. Distribution shifts commonly arises in RS, often attributed to the dynamic nature of user preferences or ubiquitous biases during data collection in RS. Despite its significance, researches on GNN-based recommendation against distribution shift are still sparse.

To bridge this gap, we propose **Distributionally Robust GNN (DR-GNN)** that incorporates Distributional Robust Optimization (DRO) into the GNN-based recommendation. DR-GNN addresses two core challenges: 1) To enable DRO to cater to graph data intertwined with GNN, we reinterpret GNN as a graph smoothing regularizer, thereby facilitating the nuanced application of DRO; 2) Given the typically sparse nature of recommendation data, which might impede robust optimization, we introduce slight perturbations in the training distribution to expand its support. Notably, while DR-GNN involves complex optimization, it can be implemented easily and efficiently. Our extensive experiments validate the effectiveness of DR-GNN against three typical distribution shifts.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Graph Recommendation, Out of Distribution, Robust

1 INTRODUCTION

In recent years, Graph Neural Networks (GNNs) [8, 12, 19] have attracted considerable attention in field of recommendation systems (RS) [1, 3, 7, 9, 13, 24, 56]. GNN-based recommendation methods often follow this typical pipeline: 1) constructing a graph based on users' historical interactions in training data; 2) utilizing multi-layered GNNs on the constructed graph to derive user and item embeddings; (3) generating recommendations based on the embedding similarities. Owing to their capability to capture high-order collaborative signals, GNN-based methods have achieved state-of-the-art performance in collaborative recommendation.

However, a pervasive assumption underpinning many GNN-based methods is that testing data holds the same distribution as training data (*a.k.a.* IID assumption). Unfortunately, this assumption often fails in practical, as distribution shifts are ubiquitous in real-world scenarios [16, 17, 43]. Such shifts can arise from: 1) the evolving nature of user preferences [40] — *e.g.,* users may increasingly favor luxury goods as their income increases, sidelining economical alternatives; 2) intrinsic biases within RS [4, 5, 59] — *e.g.,* popular items are often over-represented in the training data due to their elevated visibility. These distributional shifts compromise the

effectiveness of the constructed graph, thereby undermining the efficacy of recommendations. As illustrated in Figure 1, although GNN applications demonstrate significant advancements under IID testing scenarios (40.12% and 6.97% improvements across two datasets), these gains drop dramatically when faced with distribution shift (40.12% → 13.57% and 6.97% → 2.76%). This inspires an important question: *Can GNN-based recommendation methods be refined to better manage distribution shift?*

Despite urgency, existing literature on this problem remains sparse. The majority of relevant research can be broadly classified into two categories:

- **Recommendation Methods against distribution shift**: Some recent efforts have employed invariant learning [43, 54, 57] or causal inference [16, 40] to boost the model robustness against distribution shift. Nonetheless, these methods are not tailored for GNN-based methods. As a result, distributional biases remain entrenched in the constructed graphs, distorting the learned embeddings and recommendations accordingly.

- **Robust GNN-based Recommendation Methods:** Some researchers focus on the robustness of GNN-based recommendation methods, employing graph augmentation [2, 47, 50], graph reconstruction [10], or edge weight adjustments [60]. Nevertheless, these methods primarily address interaction noise or popularity bias, rather than universal distribution shifts. Furthermore, most of these methods are heuristic, *e.g.,* leaning on manual-designed augmentations [10] or edge weights [60]. Such reliance on human intuition not only lacks solid theoretical guarantees but also hinders their real-world applicability, especially when distribution shifts are multifaceted and unpredictable. This leaves practitioners in a quagmire, forced to experimentally tweak strategies without clear guidance.

Given these limitations, there is an imperative need for the design of theoretically-grounded GNN-based methods tailored for against distribution shifts. Inspired by the efficacy of *Distributionally Robust Optimization* (DRO), we are inclined to incorporate DRO into this task. DRO offers a theoretical framework that optimizes a model across not just the observable training distribution, but also across a broader family of distributions, effectively accounting for distribution shift. The pipeline of DRO can be delineated as: 1) identifying the hardest distribution over a given distribution family according to the current optimization context; and 2) optimizing the model based on this identified distribution. However, transferring this appealing technique to GNN-based recommendation poses two primary challenges:

Challenge 1: Current DRO methods are mainly applied in data with Euclidean structures (*e.g.,* images [27, 29, 38, 39], sentences [25]). Adapting them to manage graph-structured data remains an an open problem. More critically, the effect of graph structure is tangled with the complex graph neural networks, complicating the derivation of the hardest distribution within DRO.

Challenge 2: Another challenge arises from the typically sparse nature of recommendation data [15]. With nodes often having limited neighbors, the feasible distribution family for DRO is inherently constrained, which might dampen its robustness and overall performance [32].
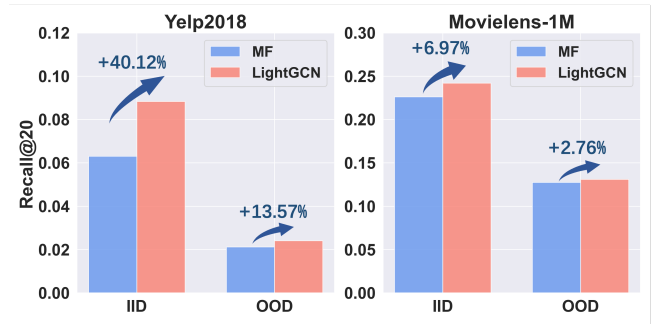
Figure 1: The performance comparison of MF [20] and LightGCN [13] under both in-distribution (IID) and out-of-distribution (OOD) testing scenarios. For the OOD testing, here we introduce popularity shift in Yelp2018 and temporal shift in Movielens-1M. More details about experimental setting refer to section 4.

To bridge these gaps, we propose DR-GNN, a novel method that seamlessly integrates DRO into GNN-based recommendation. To address the first challenge, we harness insights from graph filtering theories [21], recasting the GNN into an equivalent graph smoothing regularizer that penalizes the distance between adjacent nodes' embeddings. Through this perspective, we incorporate DRO into this regularizer, enhancing the GNN's robustness against shifts existing in neighbor distributions. For the second challenge, we propose a strategy to augment the observed neighbor distribution with slight perturbations. Notably, while DR-GNN does introduce intricate nested optimization, meticulous simplification ensures its implementation remains easy and efficient — primarily drawing from similar nodes as new neighbors and adjusting edge weights according to embedding distances. Our rigorous theoretical analysis proves that if the divergence between training and testing distributions is bounded, the robustness of DR-GNN can be guaranteed.

Our contributions are summarized as follows:

- Exploring the less-explored task of GNN-based recommendation with distribution shift, and revealing the limitations and inadequacies of existing methods.
- Proposing a new GNN-based method DR-GNN for OOD recommendation, which seamlessly integrate DRO in GNN-based methods.
- Conducting extensive experiments to validate the superiority of the proposed method against three types of distribution shift (*i.e.,* popularity shift, temporal shift and exposure shift).

2 PRELIMINARY

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

2.1 GNN-based Recommender Systems

Given a data of user-item interactions $\mathcal{D} = \{u, i, r_{u,i} \mid u \in \mathcal{U}, i \in \mathcal{I}\}$, where $\mathcal{U}$ denotes the set of users, $\mathcal{I}$ denotes the set of items, and $r_{u,i} = \{0, 1\}$ indicates whether user u has interacted with item i. We can represent user-item interaction data in the form of a bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \mathcal{U} \cup \mathcal{I}$ denotes the user/item nodes

and $\mathcal{E}$ denotes the edge set representing the interactions between users and items. Let $A \in \mathbb{R}^{(|\mathcal{U}|+|\mathcal{I}|) \times (|\mathcal{U}|+|\mathcal{I}|)}$ denote the adjacency matrix of graph $\mathcal{G}$, where $A_{u,i} = 1$ if $r_{ui} = 1$, and $A_{u,i} = 0$ otherwise. Let $\mathcal{N}(u) = \{i \in \mathcal{I} | A_{u,i} = 1\}$ represents the item set that the user u has interacted with. We also define P_u as the distribution of the neighbors of u, a uniform distribution over neighbors. Let d_u represent the degree of user u. The goal of GNN-based RS is to learn high-quality embeddings from the graph $\mathcal{G}$ and accordingly make accurate recommendations.

LightGCN. As a representative GNN-based recommendation methods, LightGCN learns user/item representations following the general message passing of GNNs. Nevertheless, it removes feature transformation and non-linear activations, as they tend to increase overfitting risks without enhancing performance. let denote the embeddings of users and items as $E \in \mathbb{R}^{(|\mathcal{U}|+|\mathcal{I}|) \times c}$ (c is the dimension of representations). In LightGCN, the final embeddings were obtained via K-th propagation layers, which can be abstracted as:

$$E^{(k)} = \widetilde{A}E^{(k-1)}, \tag{1}$$

where $E^{(k)}$ denotes the embeddings after k-th propagation layers and $\widetilde{A} = \left(D^{-\frac{1}{2}}AD^{-\frac{1}{2}}\right)$ denotes as the normalized adjacency matrix. D is the diagonal node degree matrix. The notation L denote the Laplacian matrix of graph $\mathcal{G}$, i.e., $L = I - \widetilde{A}$.

To facilitate understanding, considering the prominence and widespread application of LightGCN, we simply take it as the backbone for analysis.

2.2 Distributionally Robust Optimization

The machine learning model's success is based on the IID assumption, i.e., training data and testing data are drawn from the same distribution. However, the assumption fails to hold in many real-world applications, leading a sharp drop in performance. Distributionally Robust Optimization (DRO) addresses the issue by considering the uncertainty of the distribution. Specifically, instead of focusing on performance under a single observed data distribution, DRO aims to ensure that the model performs well across a range of potential data distributions. It first identifies the worst distribution(i.e., the loss function is maximized under the expectation of the worst-case distribution) in the family of distributions within a predefined range around the empirical data distribution, and then optimizes the objective function under that distribution, thereby ensuring the robustness of the model under unfavorable conditions. Formally, DRO is solved in a bi-level optimization via playing the min-max game on the objective function $\mathcal{L}(x; \theta)$, in which x represents the data variable and θ represents the model parameter to be optimized. The optimization objective of DRO is as follows:

$$\hat{\theta} = \underset{\theta}{\mathrm{argmin}} \left\{ \max_{P \in \mathbb{P}} \mathbb{E}_{x \sim P}[\mathcal{L}(x; \theta)] \right\}$$
$$\mathbb{P} = \{P \in \mathbb{D} : D(P, P_o) \leq \eta\} \tag{2}$$

where $\mathbb{D}$ denotes the set of all distributions, η denotes robust radius. The *uncertainty set* $\mathbb{P}$ is composed of all distributions within the robust radius distance from P_o, and DRO tried to identify *the worst-case distribution P^** from a family of eligible distributions $\mathbb{P}$ by maximization. The function $D(\cdot, \cdot)$ measures the distance between two distributions *e.g.*, KL-divergence.

(a)LightGCN (b)DR-GNN w/o GEA (c)DR-GNN
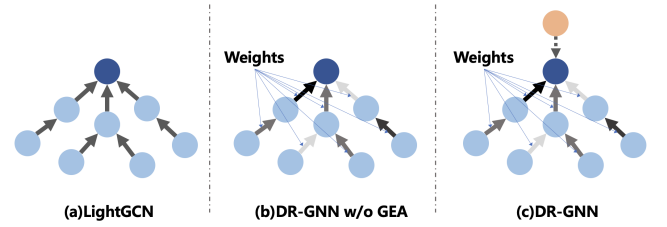
Figure 2: Illustration of how DR-GNN augments LightGCN: it gives edge weights during graph aggregation and introduces new nodes as neighbors.

3 METHODS

In this section, we detail the proposed DR-GNN (Subsection 3.1&3.2), followed by the theoretical analyses to demonstrate its robustness (Subsection 3.3). Finally, we discuss the connections our DR-GNN with existing methods (Subsection 3.4). The schematic diagram of DR-GNN is depicted in Figure 2.

3.1 Distributionally Robust GNN

It is highly challenging to directly apply DRO in GNN-based recommendation methods, as the graph structure is tangled with the complex GNN procedure. To tackle this problem, we draw inspiration from graph filtering theories [21] and recast the GNN into an equivalent graph smoothing regularizer.

LightGCN as a Graph Smoothness Regularizer. By analyzing LightGCN from the graph signal filtering perspective, we have the following important lemma:

LEMMA 1. *Performing graph aggregation in LightGCN is equivalent to optimizing the following graph smoothness regularizer using gradient descent with appropriate learning rate:*

$$\mathcal{L}_{smooth} = \frac{1}{2} \sum_u d_u \cdot \mathbb{E}_{v \sim P_u} \left\| \frac{E_u}{\sqrt{d_u}} - \frac{E_v}{\sqrt{d_v}} \right\|_F^2 \tag{3}$$

$$= \sum_u \mathbb{E}_{v \sim P_u} \left[E_u^T E_u + d_u \cdot g(u, v; \theta) \right] \tag{4}$$

$$g(u, v; \theta) = -\frac{E_u^T E_v}{\sqrt{d_u}\sqrt{d_v}} \tag{5}$$

Here, P_u denotes the distribution of the neighbor nodes of u and θ represents the model parameters. The embeddings of nodes u and v are denoted by E_u and E_v, respectively.

The proof of lemma 1 can be found in Appendix A.1. This lemma clearly elucidates the effect of the graph neural network — graph aggregation tend to draw the degree-normalized embeddings of neighbors closer. Moreover, the impact of distribution shifts on GNNs are highlighted. Taking the popularity shift (*a.k.a.* popularity bias) as an example, user representations may become excessively aligned with popular items, thereby exacerbating the Matthew effect, as demonstrated in [11]. For the convenience of subsequent analysis, we denote $\mathcal{L}_{smooth}(u) = \mathbb{E}_{v \sim P_u} \left[E_u^T E_u + d_u \cdot g(u, v; \theta) \right]$ as the smoothness regularizer on the specific node u.

Leveraging DRO in the Regularizer. Holding the view of GNN as a regularizer, we further introduce DR-GNN, a model that

incorporates DRO to enhance its robustness against distribution shifts. Following the definition of DRO Eq.(2), the objective function of the proposed DR-GNN can be formulated as:

$$\mathcal{L}_{DRO\_smooth}(u) = \max_P \mathbb{E}_{v \sim P} \left[\mathrm{E}_u^T \mathrm{E}_u + d_u \cdot g(u, v; \theta) \right]$$
$$\text{s.t.} \quad D_{KL}(P, P_u) \leq \eta \tag{6}$$

DR-GNN engages in a min-max optimization: 1) it identifies the most difficult distribution over a set of potential distributions. It is defined in the vicinity of the observed neighbor distribution subject to the constraint $D_{KL}(P, P_u) \leq \eta$; 2) Subsequently, the model's optimization is performed on this identified hardest distribution instead of original observed distribution. This strategy intrinsically incorporates potential distribution shifts during training, thereby naturally exhibits better robustness against distribution shifts. We will further validate this point in both theoretical analyses (Subsection 3.3) and empirical experiments (Section 4).

Efficient Implementation. Despite the promise, the objective of DR-GNN involves the complex nested optimization, which may incur heavily computational overhead. Fortunately, it can be largely simplified with the following lemma:

LEMMA 2. *The optimization problem of Eq.(6) can be transformed into optimizing:*

$$\mathcal{L}_{DRO\_smooth}(u) = \mathbb{E}_{v \sim P_u^*} \left[\mathrm{E}_u^T \mathrm{E}_u + d_u \cdot g(u, v; \theta) \right] \tag{7}$$

where α represents the optimal Lagrange coefficient of the constraint $D_{KL}(P, P_u) \leq \eta$, which can be regarded as a surrogate parameter of η. The worst-case distribution P_u^* can be calculated as following

$$P_u^*(v) = P_u(v) \frac{\exp(g(u, v; \theta)/\alpha)}{\mathbb{E}_{w \sim P_u} [\exp(g(u, w; \theta)/\alpha)]} \tag{8}$$

The proof is placed in the appendix A.2. This lemma provides a close-formed expression of the most difficult distribution, greatly simplifying the implementation of DR-GNN. Specifically, modifications to the distribution of neighboring nodes can be simply realized through the alteration of edge weights within the graph. Formally, the normalized adjacency matrix of the graph $\widetilde{\mathrm{A}}$ is transformed into $\widetilde{\mathrm{A}}'$ and

$$\widetilde{\mathrm{A}}'_{ij} = \frac{d_i \cdot \exp(g(i, j; \theta)/\alpha)}{\sum_{k \in \mathcal{N}(i)} \exp(g(i, k; \theta)/\alpha)} \widetilde{\mathrm{A}}_{ij} \tag{9}$$

The adjusted normalized adjacency matrix $\widetilde{\mathrm{A}}'$, characterized by modified weights, is subsequently utilized to execute the aggregation operation pertaining to the graph. Given the equivalence between the GNN and the regularizer, this can be integrated during graph aggregation, necessitating only minor adjustments to the edge weights. In practice, we may suffer numerical instability due to the introduce of $\exp(\cdot)$ in weights. To counteract this, we suggest to introduce the embedding normalization via $L2$ norm in calculating the weights, empirically yielding more stable results.

3.2 Graph Edge-Addition Strategy

Applying DRO in GNN-based recommendation incurs another challenge: Particularly, in DRO, all potential distributions P should share the same support as the original distribution P_u, otherwise the KL-divergence would become infinite. This inherently implies

that the support of P would be restricted to the neighboring nodes, excluding the vast pool of non-neighboring nodes. The situation exacerbates given the typically sparse nature of recommendation data — most users may only have interactions with a handful of items. This significantly limits the flexibility and scope of potential distributions, increasing the risk of missing testing distribution and thereby undermining model robustness.

To address aforementioned issue, we propose a strategy called Graph Edge-Addition(GEA) that introduces slight perturbations P_u^{add} in P_u to expand its support. P_u^{add} is defined over the support of all item set such that those non-neighboring nodes can be utilized for training better and robust embeddings. Formally, The objective of DR-GNN is improved as:

$$\mathcal{L}'_{DRO\_smooth}(u) = \min_{P_u^{new}} \max_P \mathbb{E}_{v \sim P} \left[\mathrm{E}_u^T \mathrm{E}_u + d_u \cdot g(u, v; \theta) \right]$$
$$\text{s.t.} \quad D_{KL}(P, P_u^{new}) \leq \eta \tag{10}$$

where the new distribution P_u^{new} is defined as $P_u^{new} = \gamma P_u + (1 - \gamma) P_u^{add}$, with γ serving as a hyperparameter that controls the magnitude of the perturbations. Remarkably, we adjust P_u towards minimization of regularizer rather than maximization. This is premised on the belief that a non-neighboring item, which exhibits greater similarity to user u, is more likely to be favored by the user. Such items are potentially useful in refining user embeddings.

The introduce of p_u^{add} mitigates the inherent shortcomings of DRO in sparse recommendation datasets. It extends the range of potential distributions and harnesses the abundant information from non-neighboring nodes. Practically, the minimization optimization on P_u^{new} can be formalized as finding the node that minimizes $g(u, v; \theta)$ for each individual node. However, it can be computationally expensive, as it requires traversing over all nodes. To mitigate computational complexity, we employ a strategy of random sampling. Specifically, we select a subset of nodes randomly to form a candidate set, and then confine the traversal operation solely to this subset.

3.3 Theoretical Analyses

In this subsection, we provide a theoretical analysis to demonstrate the robustness of DR-GNN to distribution shift. For any user u, let P_u^{ideal} denotes u's ideal neighbor distribution used for model testing and the corresponding smoothness regularizer for node u can be written as $\mathcal{L}_{ideal}(u; \theta) = \mathbb{E}_{v \sim P_u^{ideal}} \left[\mathrm{E}_u^T \mathrm{E}_u + d_u \cdot g(u, v; \theta) \right]$.

THEOREM 3.1. *[Generalization Bound] Let $\widetilde{\mathcal{L}}_{DRO\_smooth}(u; \theta)$ serve as the estimation for $\mathcal{L}_{DRO\_smooth}(u; \theta)$. Given any finite hypothesis space Θ of models and the ideal neighbor distribution P_u^{ideal} satisfies $D_{KL}(P_u^{ideal}, P_u) \leq \eta$, then we have that with probability at least $1 - \rho$:*

$$\mathcal{L}_{ideal}(u; \theta) \leq \widetilde{\mathcal{L}}_{DRO\_smooth}(u; \theta) + \mathcal{B}(\alpha, d_u, \rho) \tag{11}$$

where $\mathcal{B}(\alpha, d_u, \rho) = \frac{(d_u + \sqrt{d_u}) e^{2\sqrt{d_u}/\alpha}}{d_u - 1 + e^{2\sqrt{d_u}/\alpha}} \sqrt{\frac{1}{2} \log \frac{|\Theta|}{\rho}}$.

The proof is presented in appendix A.3. Theorem 3.1 exhibits that the ideal loss is upper bound by empirical DRO loss. At this point, we provide the theoretical guarantee for the resistance to distribution shift capability of DR-GNN.

3.4 Discussions

The connection with APDA [60]. Interestingly, the edge weights introduced in our DR-GNN is highly similar with APDA. The key distinction lies in our choice to use initial embeddings for weight calculation, while APDA uses embeddings with multi-layer aggregation. This seemingly minor difference leads to a pronounced performance disparity in favor of DR-GNN over APDA in our experiments. The reason behind this is the theoretical grounding of our approach. Specifically, DR-GNN is derived from the theoretical-sound DRO framework, while APDA's design is predominantly heuristic and lacks a solid theoretical base.

Furthermore, our framework offers theoretical insights into several heuristic settings found in APDA:

Our framework also gives a theoretical explanations of many heretical settings used in APDA: 1) Operations such as the exponential function and symmetric normalization, which are adopted heuristically in APDA, can be indeed derived to the DRO objective. 2) APDA also heuristically employs a hyperparamter α to modulate the magnitude of the value in exp() [1]. In the context of DR-GNN, this α finds its theoretical counterpart $-\alpha$ serves as a Lagrange multiplier, acting as a surrogate hyperparameter to control the robust radius. We will discuss in depth the role of α and its relationship with the degree of distribution shift in Subsection 4.3.

The connection with Attention Mechanism [36]. The attention mechanism has been adopted by recent work like Graph Attention Network (GAT) [37] to determine the edge weights. These weights in GAT, similar to ours, are predicated on node embeddings. However, there's a stark difference: while GAT employs a learnable non-linear layer to ascertain the weights, the weights in our DR-GNN are derived directly from DRO.

Despite the success of the attention mechanism across various domains, it underperforms in GNN-based recommendation. The primary reason lies in the sparsity and the lack of rich features of recommendation data. It heavily hinders the effective training of the attention function. This limitation is evident in our experimental results: GAT demonstrates suboptimal performance, while our DR-GNN exhibits effectiveness.

The connection with GraphDA [10] GraphDA is a method that enhances the adjacency matrix of a graph. This method initially pretrains a graph encoder to obtain the user/item embeddings following several iterations of graph convolution. Subsequently, the graph is reconstructed based on the similarity between these embeddings. This process facilitates denoising for active users and augmentating for inactive users. With the enhanced graph adjacency matrix, GraphDA retrains a randomly initialized graph encoder. Contrasting with GraphDA, the edge-adding operation in DR-GNN also reconstructs a new adjacency matrix, but with different motivations. The primary objective of GraphDA's graph enhancement is to equalize the number of neighbors for each node. This is achieved by denoising users with an excess of neighbors in the original data and augmenting users with a paucity of neighbors. Furthermore, GraphDA preemptively introduces edges between user-to-user and item-to-item to enable long-distance message passing. However, the integration of the GEA module aims to broaden the support of
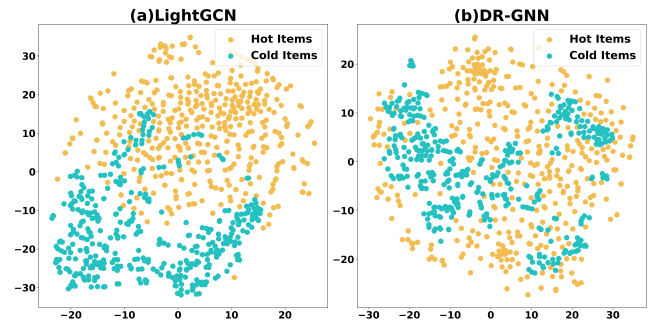
Figure 3: t-SNE Visualization on Douban. DR-GNN ensures that the representations of hot items and cold items are almost distributed in the same space.

neighbor distribution. This expansion subsequently expand the uncertainty set of DRO, thereby endowing the model with enhanced generalization capabilities.

4 EXPERIMENTS

We aim to answer the following research questions:

- **RQ1:** How does DR-GNN perform compared with existing methods under various distribution shifts?
- **RQ2:** What are the impacts of the components (e.g., DRO on neighbour nodes, GEA) on DR-GNN?
- **RQ3:** How does the parameter α impacts DR-GNN?

Datasets. The experiments are conducted under three prevalent distribution shift scenarios: popularity shift, temporal shift, and exposure shift. Thus, eight datasets are employed for testing, namely Gowalla, Douban, AmazonBook, Yelp2018, Movielens-1M, Food, Coat, and Yahoo. Appendix A.4 shows the statistics of each processed dataset and introduces the data processing method.

Baselines. We use the conventional LightGCN as backbone and BPR loss for all the baselines. The methods compared in the study fall into several categories:

- **Methods against distribution shifts in Recommendation System(InvCF [54], BOD [42])** InvCF is the SOTA method on addressing the popularity shift through invariant learning. BOD achieves data denoising through bi-level optimization. Meanwhile, we acknowledge that there are some other methods to address the OOD problem, including CausPref [17], COR [40], HIRL [57], and InvPref [43]. However, these methods require additional information that is not available in our dataset and hence cannot be tested. Certain methods among these necessitate pre-partitioned environmental datasets [57], others demand prior semantic information about users and items [17, 40], and some necessitate the assignment of environmental variables for each interaction, thereby rendering the strategy of random negative sampling inapplicable [43].
- **Graph contrastive learning methods(LightGCL [2], SGL [47]).** The methods use the contrastive learning on the graph and has been proven to alleviate the prevalent popularity bias.

---

[1]It's worth noting that this was not explicitly mentioned in their paper but was clearly implemented in their codes.

Table 1: The performance comparison on popularity shift datasets using LightGCN backbone. The best result is bolded and the runner-up is underlined. OOM stands for out of memory.

	Gowalla			Douban			Amazon-Book			Yelp2018		
	NDCG	Precision	Recall	NDCG	Precision	Recall	NDCG	Precision	Recall	NDCG	Precision	Recall
LightGCN(SIGIR20)	0.0369	0.0170	0.0563	0.0792	0.0510	0.0723	0.0227	0.0129	0.0278	0.0136	0.0060	0.0221
LightGCL(ICLR23)	0.0380	0.0177	0.0593	0.0746	0.0464	0.0721	0.0243	0.0137	0.0305	0.0136	0.0061	0.0227
SGL(SIGIR21)	0.0381	0.0173	0.0595	0.0773	0.0493	0.0716	0.0232	0.0131	0.0296	0.0143	<u>0.0063</u>	0.0242
InvCF(WWW23)	<u>0.0399</u>	<u>0.0181</u>	<u>0.0636</u>	0.0740	0.0459	0.0725	0.0210	0.0118	0.0268	0.0144	0.0062	0.0243
BOD(KDD23)	0.0361	0.0169	0.0556	0.0815	0.0559	0.0686	<u>0.0253</u>	<u>0.0141</u>	<u>0.0315</u>	<u>0.0145</u>	0.0061	<u>0.0253</u>
APDA(SIGIR23)	0.0367	0.0168	0.0579	OOM	OOM	OOM	0.0229	0.0133	0.0286	0.0141	0.0062	0.0228
GAT(ICLR18)	0.0227	0.0104	0.0334	0.0062	0.0050	0.0049	0.0185	0.0107	0.0229	0.0097	0.0044	0.0160
GraphDA(SIGIR23)	0.0341	0.0166	0.0570	<u>0.0892</u>	<u>0.0596</u>	<u>0.0808</u>	0.0199	0.0113	0.0262	0.0140	0.0059	0.0243
DR-GNN	**0.0517**	**0.0238**	**0.0774**	**0.1044**	**0.0704**	**0.0905**	**0.0327**	**0.0183**	**0.0402**	**0.0193**	**0.0086**	**0.0322**

Table 2: The performance comparison on temporal shift datasets using LightGCN backbone. The best result is bolded and the runner-up is underlined.

	MovieLens-1M			Food		
	NDCG	Precision	Recall	NDCG	Precision	Recall
LightGCN	0.2041	0.1741	0.1310	0.0429	0.0168	0.0585
LightGCL	0.2055	0.1750	0.1335	0.0421	0.0165	0.0574
SGL	0.2177	0.1830	0.1407	0.0401	0.0159	0.0553
InvCF	0.2138	0.1759	0.1354	0.0411	0.0161	0.0557
BOD	0.2116	0.1729	0.1343	<u>0.0436</u>	<u>0.0170</u>	<u>0.0592</u>
APDA	**0.2362**	**0.1954**	**0.1518**	0.0400	0.0161	0.0554
GAT	0.1942	0.1713	0.1325	0.0316	0.0131	0.0464
GraphDA	0.2213	0.1866	<u>0.1482</u>	0.0411	0.0161	0.0580
DR-GNN	<u>0.2330</u>	<u>0.1924</u>	0.1452	**0.0440**	**0.0172**	**0.0592**

Table 3: The performance comparison on exposure shift datasets using LightGCN backbone. The best result is bolded and the runner-up is underlined.

	Coat			Yahoo		
	NDCG	Precision	Recall	NDCG	Precision	Recall
LightGCN	0.0802	0.0243	0.1576	0.0736	0.0118	0.1504
LightGCL	0.0839	0.0243	0.1577	0.0734	0.0118	0.1476
SGL	0.0839	<u>0.0245</u>	0.1631	0.0740	0.0122	0.1548
InvCF	0.0842	0.0241	0.1668	0.0742	0.0122	0.1539
BOD	0.0817	0.0219	0.1542	**0.0816**	0.0115	0.1481
APDA	0.0821	0.0243	<u>0.1706</u>	0.0756	<u>0.0124</u>	<u>0.1588</u>
GAT	0.0836	0.0243	0.1633	0.0722	0.0119	0.1500
GraphDA	<u>0.0847</u>	<u>0.0245</u>	0.1641	0.0748	0.0117	0.1528
DR-GNN	**0.0874**	**0.0255**	**0.1744**	<u>0.0774</u>	**0.0126**	**0.1624**

- **Reconstructing the adjacency matrix methods(APDA [60], GAT [37], GraphDA [10])** Such methods reconstruct the adjacency matrix of the graph by adjusting edge weights or reconstructing the edges between nodes according to certain rules. Moreover, APDA can experience memory overflow issues on datasets with a large number of interactions. This is because APDA needs to calculate weights for each layer of aggregation operations.

We use the source code provided in the original papers and search for optimal hyperparameters for all comparison methods according to the instructions in the original papers.

Further experiment details are presented in the appendixA.5.

4.1 Performance Comparison (RQ1 and RQ2)

In this section, we analyze the superior of DR-GNN under different distribution shift setting as compared with other baselines.

4.1.1 Evaluations on Popularity Shift Setting. Table 1 reports the comparison of performance on all the baselines under popularity shift. The majority of comparative methodologies fail to consistently yield satisfactory results across a variety of datasets, including algorithms specifically designed for popularity shift, such as

APDA and InvCF. The improvement of APDA compared to LightGCN is quite limited, indicating that its heuristic dynamic edge weight adjustment algorithm cannot handle popularity shift well. Meanwhile, GAT, which also dynamically adjusts edge weights, performs much worse than LightGCN, suggesting that the attention mechanism may inadvertently intensify the impact of distribution shift. We also notice that GAT fails to converge when trained on the Douban dataset. This may be attributable to the propensity of GATs to overfit when composed of multiple layers. Our proposed method, DR-GNN, consistently and significantly surpasses the state-of-the-art benchmarks across all popularity shift datasets. The robustness of DR-GNN is ascribed to the incorporation of uncertainty inherent in observed data distributions by DRO. This feature enables the model to perform optimally under various environments, rather than solely relying on training data.

Visualization Results. As shown in Figure 3, to better understand how DR-GNN handles distribution shift, we perform t-SNE visualization [35] of the item embeddings on the Douban dataset. According to the ranking results of item popularity, the top 5% most popular items are selected as hot items and the bottom 5% as cold items. It can be observed that in the representations learned by LightGCN, there is a clear gap between hot items and cold items,

Table 4: Ablation Study on Gowalla and Yelp2018

Dataset	Method	NDCG	Precision	Recall
Gowalla	LightGCN	0.0369	0.0170	0.0563
	DR-GNN w/o DRO	0.0391	0.0174	0.0583
	DR-GNN w/o GEA	0.0494	0.0231	0.0761
	DR-GNN	**0.0517**	**0.0238**	**0.0774**
Yelp2018	LightGCN	0.0136	0.0060	0.0221
	DR-GNN w/o DRO	0.0161	0.0070	0.0265
	DR-GNN w/o GEA	0.0182	0.0081	0.0305
	DR-GNN	**0.0193**	**0.0086**	**0.0322**

while both are distributed in the same space in the representations learned by DR-GNN. This suggests that DR-GNN eliminate the impact of distribution shift caused by popularity shift.

4.1.2 Evaluations on Temporal Shift Setting. Temporal bias takes into account changes in user interests over time. It is more complex than the distribution shift caused by popularity, as it encompasses factors beyond popularity alone. We simulate temporal bias by dividing training and test sets for each user according to the timestamp of the interaction.

In the context of the temporal shift setting, we noticed that on the Movielens-1M dataset, although APDA performed better than DR-GNN, the improvement is not significant, and its performance was relatively unstable, evidenced by its inferior performance compared to LightGCN on the Food dataset. Similar observations were made for GraphDA. Compared to most other benchmark methods, DR-GNN continues to consistently deliver good performance.

4.1.3 Evaluations on Exposure Shift Setting. In practical contexts, users are typically exposed to a limited subset of items, thereby neglecting a significant majority. This phenomenon named "exposure bias" suggests that the absence of user interaction with specific items does not unequivocally signify their disinterest. Consequently, in real-world datasets, the patterns of missing user interaction records are not randomly distributed, but rather, they are "missing-not-at-random". Here we conduct experiments on widely used missing-complete-at-random datasets: Yahoo and Coat.

According to table 3, DR-GNN can also handle distribution shift caused by exposure bias well. Besides, we noticed that on the Yahoo dataset, BOD's NDCG metric surpasses other comparison methods, but it is weaker than LightGCN in terms of Precision and Recall metrics. We believe this might be due to BOD's weight adjustment strategy overfitting to certain interactions, causing them to rank higher, which leads the recommendation model to concentrate too much on a few highly relevant results. Furthermore, BOD's performance on the coat dataset is only slightly better than LightGCN, indicating that its performance is not stable.

The experiments under the aforementioned three settings demonstrate that our method can handle different types of distribution shifts.

4.2 Ablation Study (RQ3)

We conducted an ablation study to investigate the effects of different modules in DR-GNN, including the DRO and GEA modules. We
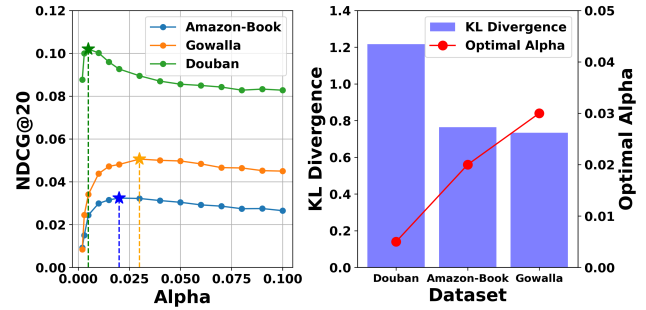
Figure 4: Analysis of the role of α. (a) Left: The performance of DR-GNN in terms of NDCG across different α on three datasets with varying degrees of distribution shift. (b) Right: The relationship between the degree of distribution shift and the optimal α.

compared the DR-GNN with its two variants "DR-GNN w/o DRO" and "DR-GNN w/o GEA" based on whether the DRO and GEA modules were enabled. The results in Table 4 demonstrate that the use of DRO for graph aggregation operations can significantly enhance the model's performance, and GEA further improves the model's effectiveness. Surprisingly, we also found that the simple use of GEA could also yield some gains. This may be attributed to the additional edges introduced by GEA, which have accelerated the convergence of smoothness regularization. The ablation study highlights the fact that all modules in DR-GNN can boost the model's learning.

4.3 Role of the parameter α (RQ4)

The parameter α in Eq.(8) plays an important role in the DRO. Based on the previous derivation, the role of α is to control the size of uncertainty set. A smaller α indicates a larger uncertainty set for optimizing. However, an excessively large search space can easily lead to model overfitting. As α tends towards zero, DRO will excessively amplify the weight of the node with the maximum smooth regularization loss, which can easily lead to overfitting to the meaningless distribution. When α is infinitely large, the worst-case distribution tends towards a uniform distribution. At this point, DR-GNN without GEA is equivalent to LightGCN. Therefore, α is an important hyperparameter that needs to be carefully tuned. Different values need to be set depending on the degree of distribution shift in the dataset. Empirically, we search for α within the range of [0, 1].

Besides, we empirically verified the relationship between the optimal α and the degree of distribution shift on the popularity shift dataset. We utilize the KL-divergence between item frequencies as a metric to measure the degree of distribution shift between the training and testing sets, and then tried to train the model by tuning different α parameters. Figure 4(a) shows the NDCG performance of DR-GNN with different α under varying degrees of popularity shift in different datasets. It can be observed that either excessively large or small α can impact the performance, and the optimal value varies across different datasets. In Figure 4(b), we further quantified the distribution shift using KL divergence. It was found that datasets with a larger degree of shift require a

smaller optimal α, as they necessitate a larger uncertainty set. This corresponds to our previous explanation.

5 RELATED WORK

5.1 GNN-based Recommender System

In recent years, graph-based recommendation systems have attracted extensive attention and research. Such systems leverage the structure of graphs to discover and infer user preferences and interests, thereby providing more personalized and accurate recommendations. Compared to the traditional collaborative filtering method that only use first order information of node, the GNN can capture higher order signal in the interactions by aggregating the information from neighboring nodes. LightGCN [13] simplify the original stucture of GCN [19] by dropping the feature transformation nonlinear activation and self-connection. NIA-GCN [33] improves the aggregation way to model more complex interactions in the graph data.

Contrastive learning has also been extensively applied in graph-based recommendation models. SGL [47] employs various operators to obtain augmented graphs, which are then utilized to construct positive and negative sample pairs for contrastive loss. LightGCL [2] enhances the method of augmenting graphs by utilizing SVD to maximally preserve the intrinsic semantic structures of the augmented graph. SimGCL [51] and XSimGCL [50] replace conventional graph augmentations with a more efficient noise-based embedding strategy to improve contrastive learning.

Some methods are designed to target the OOD problem on GNN-based recommendation models. APDA [60] dynamically adjusts the edge weights of the graph, reducing the impact of popular items while amplifying the influence of unpopular items. GraphDA [10] reconstructs the adjacency matrix of the graph through a pre-trained encoder, thereby achieving signal augmentation and denoising within the graph. RGCF [34] improves graph structure learning by identifying more reliable message-passing interactions, while simultaneously maintaining the diversity of the enhanced data.

Despite these efforts, the aforementioned methods can only address specific types of OOD problems, that is, they only focus on the distribution shift resulting from certain specific factors, such as noise and popularity bias, hence are not generic. Moreover, these methods are mostly based on heuristic rule design and lack theoretical guarantees.

5.2 OOD in Recommender System

When training models, it is a common assumption that test data originates from the same distribution as the training data. However, in real-world scenarios, models may encounter test data that deviates from the distribution of the training data - a phenomenon known as Out-of-Distribution. The presence of OOD data can potentially precipitate a degradation in the model's performance. There are many methods used to solve the OOD problem in recommender systems.

One category of methodologies endeavors to alleviate the impact of distribution shift through the identification of invariant components within embeddings. InvCF [54] effectively mitigates the impact of popularity shifts by incorporating an auxiliary classifier that formulates recommendations predicated on popularity. InvPref [43] and HIRL [57] seperate the dataset into multiple environments by attributing an environment variable to each interaction and then leverage invariant learning to automatically identify elements that remain constant irrespective of the environment. Some of the works employ causal learning in addressing the OOD problem. COR [40] uses causal graph modeling and counterfactual reasoning to address the effect of out-of-date interactions. CausPref [17] utilizes a differentiable causal graph learning approach to obtain the invariant user preferences. Other works, such as BOD [42] applies bi-level optimization to ascertain the weight for each interaction to negate the effects of noisy interactions.

However, the kind of works are not specifically designed for GNN-based recommender system and thus fail to address the impact of distribution shifts on the structure of the graph models themselves.

5.3 Distributionally Robust Optimization

Distributionally Robust Optimization is a method for addressing OOD problems. The primary goal of DRO is to find a solution that is not only optimal for the given data distribution but also robust against variations in the data distribution, i.e. the family of distributions consisting of all distributions within a certain distance from the current data distribution. Many measures of distribution distance are used in DRO, including KL-divergence [18, 45], Wasserstein-distance [30], MMD distance [32]. It has been found that DRO tends to induce model overfitting to the noisy samples [53] and GroupDRO [27] was introduced to address this issue.

DRO has also been applied in the field of RS. S-DRO [44] and Rob-Fair [48] employs DRO to enhance the fairness of user experiences. DROS [49] utilizes DRO to address the issue of data distribution shifts over time within sequence recommendations. BSL [46] incorporates DRO to resolve the Softmax Loss in recommendation systems and to bolster the model's robustness against noisy positives. However, none of these methods are used on GNN-based RS.

Some research has explored the application of DRO on GNN [6, 26, 55]. On one hand, these methods fundamentally aim to address distributional shifts caused by noise in node embeddings, which is distinct from our approach that considers distributional shifts within the graph's topological structure. On the other hand, these methods are not applied in recommendation systems and do not take into account the challenges posed by the unique characteristics of recommendation datasets when applying DRO.

6 CONCLUSION

This paper proposes an method DR-GNN, which introduces DRO into the aggregation operation of GNN, to solve the problem of existing graph-based recommendation systems being easily affected by distribution shift. Based on comparative experiments under multiple datasets and settings, as well as visualization studies on real datasets, DR-GNN has been proven to effectively solve the distribution shift problem, enhancing the robustness of graph models.

A direction worthy of exploration in future work is the application of DRO based on other distance metrics, such as the Wasserstein distance, MMD distance, etc., to address some of the shortcomings of the KL divergence-based DRO discussed in this paper. We believe that DR-GNN could provide a new perspective for future work aimed at enhancing graph-based RS robustness.

ACKNOWLEDGMENTS

REFERENCES

[1] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263* (2017).
[2] Xuheng Cai, Chao Huang, Lianghao Xia, and Xubin Ren. 2023. LightGCL: Simple Yet Effective Graph Contrastive Learning for Recommendation. *arXiv preprint arXiv:2302.08191* (2023).
[3] Chih-Ming Chen, Chuan-Ju Wang, Ming-Feng Tsai, and Yi-Hsuan Yang. 2019. Collaborative similarity embedding for recommender systems. In *The World Wide Web Conference*. 2637–2643.
[4] Jiawei Chen, Hande Dong, Yang Qiu, Xiangnan He, Xin Xin, Liang Chen, Guli Lin, and Keping Yang. 2021. AutoDebias: Learning to debias for recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 21–30.
[5] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2023. Bias and debias in recommender system: A survey and future directions. *ACM Transactions on Information Systems* 41, 3 (2023), 1–39.
[6] Shuyi Chen, Kaize Ding, and Shixiang Zhu. 2023. Uncertainty-Aware Robust Learning on Noisy Graphs. *arXiv preprint arXiv:2306.08210* (2023).
[7] Leyan Deng, Defu Lian, Chenwang Wu, and Enhong Chen. 2022. Graph Convolution Network based Recommender Systems: Learning Guarantee and Item Mixture Powered Strategy. *Advances in Neural Information Processing Systems* 35 (2022), 3900–3912.
[8] Hande Dong, Jiawei Chen, Fuli Feng, Xiangnan He, Shuxian Bi, Zhaolin Ding, and Peng Cui. 2021. On the equivalence of decoupled graph convolution network and label propagation. In *Proceedings of the Web Conference 2021*. 3651–3662.
[9] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *The world wide web conference*. 417–426.
[10] Ziwei Fan, Ke Xu, Zhang Dong, Hao Peng, Jiawei Zhang, and Philip S Yu. 2023. Graph collaborative signals denoising and augmentation for recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2037–2041.
[11] Chongming Gao, Kexin Huang, Jiawei Chen, Yuan Zhang, Biao Li, Peng Jiang, Shiqi Wang, Zhong Zhang, and Xiangnan He. 2023. Alleviating Matthew Effect of Offline Reinforcement Learning in Interactive Recommendation. *arXiv preprint arXiv:2307.04571* (2023).
[12] Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997* (2018).
[13] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.
[14] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.
[15] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 549–558.
[16] Yue He, Zimu Wang, Peng Cui, Hao Zou, Yafeng Zhang, Qiang Cui, and Yong Jiang. 2022. Causpref: Causal preference learning for out-of-distribution recommendation. In *Proceedings of the ACM Web Conference 2022*. 410–421.
[17] Yue He, Zimu Wang, Peng Cui, Hao Zou, Yafeng Zhang, Qiang Cui, and Yong Jiang. 2022. Causpref: Causal preference learning for out-of-distribution recommendation. In *Proceedings of the ACM Web Conference 2022*. 410–421.
[18] Zhaolin Hu and L Jeff Hong. 2013. Kullback-Leibler divergence constrained distributionally robust optimization. *Available at Optimization Online* 1, 2 (2013),

9.

[19] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
[20] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.
[21] Yao Ma, Xiaorui Liu, Tong Zhao, Yozen Liu, Jiliang Tang, and Neil Shah. 2021. A unified view on graph neural networks as graph signal denoising. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1202–1211.
[22] Benjamin M Marlin and Richard S Zemel. 2009. Collaborative prediction and ranking with non-random missing data. In *Proceedings of the third ACM conference on Recommender systems*. 5–12.
[23] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. 2018. *Foundations of machine learning*. MIT press.
[24] Athanasios N Nikolakopoulos and George Karypis. 2019. Recwalk: Nearly uncoupled random walks for top-n recommendation. In *Proceedings of the twelfth ACM international conference on web search and data mining*. 150–158.
[25] Yonatan Oren, Shiori Sagawa, Tatsunori B Hashimoto, and Percy Liang. 2019. Distributionally robust language modeling. *arXiv preprint arXiv:1909.02060* (2019).
[26] Alireza Sadeghi, Meng Ma, Bingcong Li, and Georgios B Giannakis. 2021. Distributionally robust semi-supervised learning over graphs. *arXiv preprint arXiv:2110.10582* (2021).
[27] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. 2019. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731* (2019).
[28] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. In *international conference on machine learning*. PMLR, 1670–1679.
[29] Soroosh Shafieezadeh Abadeh, Peyman M Mohajerin Esfahani, and Daniel Kuhn. 2015. Distributionally robust logistic regression. *Advances in Neural Information Processing Systems* 28 (2015).
[30] Aman Sinha, Hongseok Namkoong, Riccardo Volpi, and John Duchi. 2017. Certifying some distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571* (2017).
[31] Weiping Song, Zhiping Xiao, Yifan Wang, Laurent Charlin, Ming Zhang, and Jian Tang. 2019. Session-based social recommendation via dynamic graph attention networks. In *Proceedings of the Twelfth ACM international conference on web search and data mining*. 555–563.
[32] Matthew Staib and Stefanie Jegelka. 2019. Distributionally robust optimization and generalization in kernel methods. *Advances in Neural Information Processing Systems* 32 (2019).
[33] Jianing Sun, Yingxue Zhang, Wei Guo, Huifeng Guo, Ruiming Tang, Xiuqiang He, Chen Ma, and Mark Coates. 2020. Neighbor interaction aware graph convolution networks for recommendation. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. 1289–1298.
[34] Changxin Tian, Yuexiang Xie, Yaliang Li, Nan Yang, and Wayne Xin Zhao. 2022. Learning to denoise unreliable interactions for graph collaborative filtering. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 122–132.
[35] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).
[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
[37] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
[38] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. 2018. Generalizing to unseen domains via adversarial data augmentation. *Advances in neural information processing systems* 31 (2018).
[39] Serena Wang, Harikrishna Narasimhan, Yichen Zhou, Sara Hooker, Michal Lukasik, and Aditya Krishna Menon. 2022. Robust distillation for worst-class performance. *arXiv preprint arXiv:2206.06479* (2022).
[40] Wenjie Wang, Xinyu Lin, Fuli Feng, Xiangnan He, Min Lin, and Tat-Seng Chua. 2022. Causal representation learning for out-of-distribution recommendation. In *Proceedings of the ACM Web Conference 2022*. 3562–3571.
[41] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 950–958.
[42] Zongwei Wang, Min Gao, Wentao Li, Junliang Yu, Linxin Guo, and Hongzhi Yin. 2023. Efficient bi-level optimization for recommendation denoising. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2502–2511.
[43] Zimu Wang, Yue He, Jiashuo Liu, Wenchao Zou, Philip S Yu, and Peng Cui. 2022. Invariant preference learning for general debiasing in recommendation.

In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1969–1978.

[44] Hongyi Wen, Xinyang Yi, Tiansheng Yao, Jiaxi Tang, Lichan Hong, and Ed H Chi. 2022. Distributionally-robust Recommendations for Improving Worst-case User Experience. In *Proceedings of the ACM Web Conference 2022*. 3606–3610.

[45] Junkang Wu, Jiawei Chen, Jiancan Wu, Wentao Shi, Xiang Wang, and Xiangnan He. 2023. Understanding contrastive learning via distributionally robust optimization. *arXiv preprint arXiv:2310.11048* (2023).

[46] Junkang Wu, Jiawei Chen, Jiancan Wu, Wentao Shi, Jizhi Zhang, and Xiang Wang. 2023. BSL: Understanding and Improving Softmax Loss for Recommendation. *arXiv preprint arXiv:2312.12882* (2023).

[47] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 726–735.

[48] Hao Yang, Zhining Liu, Zeyu Zhang, Chenyi Zhuang, and Xu Chen. 2023. Towards Robust Fairness-aware Recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 211–222.

[49] Zhengyi Yang, Xiangnan He, Jizhi Zhang, Jiancan Wu, Xin Xin, Jiawei Chen, and Xiang Wang. 2023. A Generic Learning Framework for Sequential Recommendation with Distribution Shifts. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[50] Junliang Yu, Xin Xia, Tong Chen, Lizhen Cui, Nguyen Quoc Viet Hung, and Hongzhi Yin. 2023. XSimGCL: Towards extremely simple graph contrastive learning for recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2023).

[51] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. 2022. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*. 1294–1303.

[52] Wenhui Yu and Zheng Qin. 2020. Graph convolutional network for recommendation with low-pass collaborative filters. In *International Conference on Machine Learning*. PMLR, 10936–10945.

[53] Runtian Zhai, Chen Dan, Zico Kolter, and Pradeep Ravikumar. 2021. Doro: Distributional and outlier robust optimization. In *International Conference on Machine Learning*. PMLR, 12345–12355.

[54] An Zhang, Jingnan Zheng, Xiang Wang, Yancheng Yuan, and Tat-Seng Chua. 2023. Invariant Collaborative Filtering to Popularity Distribution Shift. In *Proceedings of the ACM Web Conference 2023*. 1240–1251.

[55] Xiang Zhang, Yinfei Xu, Qinghe Liu, Zhicheng Liu, Jian Lu, and Qiao Wang. 2021. Robust graph learning under Wasserstein uncertainty. *arXiv preprint arXiv:2105.04210* (2021).

[56] Yongfeng Zhang, Xu Chen, et al. 2020. Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval* 14, 1 (2020), 1–101.

[57] Zeyu Zhang, Heyang Gao, Hao Yang, and Xu Chen. 2023. Hierarchical Invariant Learning for Domain Generalization Recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3470–3479.

[58] Wayne Xin Zhao, Yupeng Hou, Xingyu Pan, Chen Yang, Zeyu Zhang, Zihan Lin, Jingsen Zhang, Shuqing Bian, Jiakai Tang, Wenqi Sun, et al. 2022. RecBole 2.0: towards a more up-to-date recommendation library. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 4722–4726.

[59] Zihao Zhao, Jiawei Chen, Sheng Zhou, Xiangnan He, Xuezhi Cao, Fuzheng Zhang, and Wei Wu. 2022. Popularity bias is not always evil: Disentangling benign and harmful bias for recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[60] Huachi Zhou, Hao Chen, Junnan Dong, Daochen Zha, Chuang Zhou, and Xiao Huang. 2023. Adaptive Popularity Debiasing Aggregator for Graph Collaborative Filtering. (2023).

A APPENDICES

A.1 The proof of Lemma 1

PROOF. Eq.(3) can be written as follows,

$$\mathcal{L}_{smooth} = \frac{1}{2} \sum_u d_u \cdot \mathbb{E}_{v \sim P_u} \left\| \frac{\mathrm{E}_u}{\sqrt{d_u}} - \frac{\mathrm{E}_v}{\sqrt{d_v}} \right\|_F^2$$

$$= \frac{1}{2} \sum_u \sum_{v \in \mathcal{N}(u)} \left\| \frac{\mathrm{E}_u}{\sqrt{d_u}} - \frac{\mathrm{E}_v}{\sqrt{d_v}} \right\|_F^2 = \mathrm{tr}\left(\mathrm{E}^\top \mathrm{L} \mathrm{E}\right) \quad (12)$$

We first compute the derivative of the Eq.(12)

$$\frac{\partial \mathcal{L}}{\partial \mathrm{E}} = 2\mathrm{L}\mathrm{E} = 2\mathrm{E} - 2\widetilde{\mathrm{A}}\mathrm{E} \quad (13)$$

In order to minimize Eq.(12) we take multi-step gradient descent. The k-th step is as follows:

$$\mathrm{E}^{(k)} \leftarrow \mathrm{E}^{(k-1)} - b \cdot \frac{\partial \mathcal{L}}{\partial \mathrm{E}}\bigg|_{E=E^{(k-1)}}$$

$$= (1 - 2b)E^{(k-1)} + 2b\widetilde{\mathrm{A}}\mathrm{E}^{(k-1)} \quad (14)$$

When we set the learning rate b as $\frac{1}{2}$, we have the following iterative steps:

$$\mathrm{E}^{(k)} \leftarrow \widetilde{\mathrm{A}}\mathrm{E}^{(k-1)} \quad (15)$$

which is equivalent to one-step aggregation operation in LightGCN. Thus the multilayer aggregation operation is equivalent to minimizing the smoothness regularization with a multi-step gradient descent algorithm.

$$\mathcal{L}_{smooth} = \frac{1}{2} \sum_u \sum_{v \in \mathcal{N}(u)} \left\| \frac{\mathrm{E}_u}{\sqrt{d_u}} - \frac{\mathrm{E}_v}{\sqrt{d_v}} \right\|_F^2$$

$$= \frac{1}{2} \sum_u \sum_{v \in \mathcal{N}(u)} \left(\frac{\mathrm{E}_u^T \mathrm{E}_u}{d_u} + \frac{\mathrm{E}_v^T \mathrm{E}_v}{d_v} - \frac{2\mathrm{E}_u^T \mathrm{E}_v}{\sqrt{d_u}\sqrt{d_v}} \right)$$

$$= \sum_u \left(\mathrm{E}_u^T \mathrm{E}_u - \sum_{v \in \mathcal{N}(u)} \frac{\mathrm{E}_u^T \mathrm{E}_v}{\sqrt{d_u}\sqrt{d_v}} \right)$$

$$= \sum_u \mathbb{E}_{v \sim P_u} \left[\mathrm{E}_u^T \mathrm{E}_u - d_u \cdot \frac{\mathrm{E}_u^T \mathrm{E}_v}{\sqrt{d_u}\sqrt{d_v}} \right]$$

Thus, Lemma 1 is proven.

□

A.2 The proof of Lemma 2

PROOF. In this section we show the derivation of a multi-layer optimization problem in DRO converted to a single-layer optimization problem. The original DRO problem is as follows,

$$\mathcal{L}_{DRO\_smooth}(u) = \max_P \mathbb{E}_{v \sim P} \left[\mathrm{E}_u^T \mathrm{E}_u + d_u \cdot g(u, v; \theta) \right] \quad (16)$$
$$\text{s.t. } D_{KL}(P, P_u) \le \eta$$

Given that d_u and $\mathrm{E}_u^T \mathrm{E}_u$ as constants do not affect the optimization, the aforementioned optimization problem can be simplified to the following form:

$$\max_P \mathbb{E}_{v \sim P} \left[g(u, v; \theta) \right]$$
$$\text{s.t. } D_{KL}(P, P_u) \le \eta \quad (17)$$

We focus on how to eliminate the inner maximisation optimisation problem and the distributional constraint term. Assume $L(j) =$

$P(j)/P_u(j)$ and define a convex function $\phi(x) = x \log x - x + 1$. Then the divergence $D_{KL}(P, P_u)$ can be written as $\mathbb{E}_{P_u}[\phi(x)]$. The inner layer maximization optimization problem can be reformulated as follow:

$$\max_L \mathbb{E}_{v \sim P_u} [g(u, v; \theta)L]$$
$$\text{s.t. } \mathbb{E}_{P_u}[\phi(L)] \leq \eta, \mathbb{E}_{P_u}[L] = 1 \tag{18}$$

As a convex optimization problem, we use the Lagrangian function to solve it:

$$\min_{\alpha \geq 0, \beta} \max_L \mathbb{E}_{v \sim P_u} [g(u, v; \theta)L] - \alpha(\mathbb{E}_{P_u}[\phi(L)] - \eta) + \beta(\mathbb{E}_{P_u}[L] - 1)$$

$$= \min_{\alpha \geq 0, \beta} \left\{ \alpha\eta - \beta + \alpha \max_L \mathbb{E}_{v \sim P_u} \left[\frac{g(u, v; \theta) + \beta}{\alpha} L - \phi(L) \right] \right\}$$

$$= \min_{\alpha \geq 0, \beta} \left\{ \alpha\eta - \beta + \alpha \mathbb{E}_{v \sim P_u} \left[\max_L \left(\frac{g(u, v; \theta) + \beta}{\alpha} L - \phi(L) \right) \right] \right\} \tag{19}$$

Notice that $\max_L \left(\frac{g(u, v; \theta) + \beta}{\alpha} L - \phi(L) \right) = \phi^* (\frac{g(u, v; \theta) + \beta}{\alpha})$ is the convex conjugate function of $\phi(x)$ and we have $\phi^*(x) = e^x - 1$. $L(v) = e^{\frac{g(u, v; \theta) + \beta}{\alpha}}$ when the maximum value is obtained.

$$\min_{\alpha \geq 0, \beta} \left\{ \alpha\eta - \beta + \alpha \mathbb{E}_{v \sim P_u} \left[\max_L \left(\frac{g(u, v; \theta) + \beta}{\alpha} L - \phi(L) \right) \right] \right\}$$

$$= \min_{\alpha \geq 0, \beta} \left\{ \alpha\eta - \beta + \alpha \mathbb{E}_{v \sim P_u} \left[e^{\frac{g(u, v; \theta) + \beta}{\alpha}} - 1 \right] \right\} \tag{20}$$

$$= \min_{\alpha \geq 0} \left\{ \alpha\eta + \alpha \log \mathbb{E}_{v \sim P_u} \left[e^{\frac{g(u, v; \theta)}{\alpha}} \right] \right\}$$

where $\beta = -\alpha \log \mathbb{E}_{v \sim P_u} \left[e^{\frac{g(u, v; \theta)}{\alpha}} \right]$ and $L(v) = \frac{e^{\frac{g(u, v; \theta)}{\alpha}}}{\mathbb{E}_{w \sim P_u} \left[e^{\frac{g(u, w; \theta)}{\alpha}} \right]}$

when the minimum value is obtained. We consider the Lagrange multiplier α as a hyperparameter related to the robustness radius to obtain the final unconstrained single-layer optimization problem,

$$\mathcal{L}_{DRO\_smooth}(u) = \left\{ \alpha\eta + \alpha \log \mathbb{E}_{v \sim P_u} \exp \left(\frac{g(u, v; \theta)}{\alpha} \right) \right\} \tag{21}$$

where the worst-case distribution

$$P_u^*(v) = P_u(v) \frac{\exp(g(u, v; \theta)/\alpha)}{\mathbb{E}_{w \sim P_u}[\exp(g(u, w; \theta)/\alpha)]} \tag{22}$$

Thus lemma 2 is proven. □

A.3 The proof of Theorem 3.1

PROOF. Before detailing the proof process, we first introduce a pertinent theorem:

THEOREM A.1. *(Union Bound). For any two sets A, B we have*

$$\mathbb{P}(A \cup B) \leq \mathbb{P}(A) + \mathbb{P}(B) \tag{23}$$

THEOREM A.2. *(McDiarmid's inequality [23]). Let $X_1, \cdots, X_n$ be independent random variables, where X_i has range X. Let $f : X_1 \times \cdots \times X_n \to \mathbb{R}$ be any function with the $(c_1, \ldots, c_n)$ bounded difference property: for every $i = 1, \ldots, n$ and every $(x_1, \ldots, x_n), (x'_1, \ldots, x'_n) \in X_1 \times \cdots \times X_n$ that differ only in the i-th coordinate $(x_j = x'_j$ for all $j \neq i)$, we have $\left| f(x_1, \ldots, x_n) - f(x'_1, \ldots, x'_n) \right| \leq c_i$. For any $\epsilon > 0$,*

$$\mathbb{P}(f(X_1, \cdots, X_n) - \mathbb{E}[f(X_1, \cdots, X_n)] \leq -\epsilon) \leq \exp\left(\frac{-2\epsilon^2}{\sum_{i=1}^N c_i^2} \right) \tag{24}$$

Now we delve into the proof. As $D_{KL}(P_u^{ideal}, P_u) \leq \eta$, we can bound $\mathcal{L}_{ideal}(u; \theta)$ with:

$$\mathcal{L}_{ideal}(u; \theta) = \mathbb{E}_{v \sim P_u^{ideal}} \left[E_u^T E_u + d_u \cdot g(u, v; \theta) \right]$$

$$\leq \mathbb{E}_{v \sim P_u^*} \left[E_u^T E_u + d_u \cdot g(u, v; \theta) \right] \tag{25}$$

$$= \mathcal{L}_{DRO\_smooth}(u; \theta)$$

where P_u^{ideal}, P_u^* denotes the ideal distribution and the worst-case distribution of the neighbor nodes of u. Assume $h(u, v; \theta) = E_u^T E_u + d_u \cdot g(u, v; \theta)$. Since in DRO we normalize the embeddings by L2 norm then $h(u, v; \theta) \in [1 - \sqrt{d_u}, 1 + \sqrt{d_u}]$. Then we have the following bound:

$$\left| P_u^*(v_i)h(u, v_i; \theta) - P_u^*(v_j)h(u, v_j; \theta) \right| \leq \sup_{v \sim P_u} \left| P_u^*(v)h(u, v; \theta) \right|$$

$$\leq \frac{(1 + \sqrt{d_u})e^{2\sqrt{d_u}/\alpha}}{e^{2\sqrt{d_u}/\alpha} + d_u - 1} \tag{26}$$

The upper bound of $\sup_{v \sim P_u} \left| P_u^*(v)g(u, v; \theta) \right|$ arrives if $h(u, v_i; \theta) = 1 + \sqrt{d_u}$ for the node v_i and $h(u, v_j; \theta) = 1 - \sqrt{d_u}$ for $j \neq i$.

By using McDiarmid's inequality in Theorem A.2, for any ϵ, we have:

$$\mathbb{P}\left[(\mathcal{L}_{DRO\_smooth}(u; \theta) - \widetilde{\mathcal{L}}_{DRO\_smooth}(u; \theta)) \geq \epsilon \right] \leq$$

$$exp\left(\frac{-2\epsilon^2}{d_u} \left(\frac{d_u - 1 + e^{2\sqrt{d_u}/\alpha}}{(1 + \sqrt{d_u})e^{2\sqrt{d_u}/\alpha}} \right)^2 \right) \tag{27}$$

Eq.(27) holds for the unique model θ. For any finite hypothesis space of models $\Theta = \{\theta_1, \theta_2, ..., \theta_{|\Theta|}\}$, we combine this with Theorem A.1 and have the following generalization error bound of the learned model $\widetilde{\theta}$:

$$\mathbb{P}\left[\theta \in \Theta : \left((\mathcal{L}_{DRO\_smooth}(u; \theta) - \widetilde{\mathcal{L}}_{DRO\_smooth}(u; \theta)) \geq \epsilon \right) \right] \leq$$

$$|\Theta| exp\left(\frac{-2\epsilon^2}{d_u} \left(\frac{d_u - 1 + e^{2\sqrt{d_u}/\alpha}}{(1 + \sqrt{d_u})e^{2\sqrt{d_u}/\alpha}} \right)^2 \right) \tag{28}$$

Let $\rho = |\Theta| exp\left(\frac{-2\epsilon^2}{d_u} \left(\frac{d_u - 1 + e^{2\sqrt{d_u}/\alpha}}{(1 + \sqrt{d_u})e^{2\sqrt{d_u}/\alpha}} \right)^2 \right)$ we get

$$\epsilon = \frac{(d_u + \sqrt{d_u})e^{2\sqrt{d_u}/\alpha}}{d_u - 1 + e^{2\sqrt{d_u}/\alpha}} \sqrt{\frac{1}{2} \log \frac{|\Theta|}{\rho}} \tag{29}$$

Thus, for $\forall \rho \in (0, 1)$, we conclude that with probability at least $1 - \rho$:

$$\mathcal{L}_{ideal}(u; \theta) \leq \mathcal{L}_{DRO\_smooth}(u; \theta)$$

$$\leq \widetilde{\mathcal{L}}_{DRO\_smooth}(u; \theta)$$

$$+ \frac{(d_u + \sqrt{d_u})e^{2\sqrt{d_u}/\alpha}}{d_u - 1 + e^{2\sqrt{d_u}/\alpha}} \sqrt{\frac{1}{2} \log \frac{|\Theta|}{\rho}} \tag{30}$$

Table 5: Dataset statistics.

Setting	Dataset	#Users	#Items	#Interactions	Density
Popularity Shift	Gowalla	29858	40705	1024507	0.0008
	Douban	61788	7768	10449897	0.0218
	Amazon-Book	52643	89807	2964807	0.0006
	Yelp2018	77277	41978	2062568	0.0006
Temproal Shift	Movielens-1M	6040	3653	880999	0.0399
	Food	7450	10977	251038	0.0031
Exposure Shift	Coat	290	284	2745	0.0333
	Yahoo	14382	1000	129748	0.0090

□

A.4 Dataset

The statistics of each dataset used in the experiment are shown in Table 5, which lists the number of users, items, interactions, and sparsity of the dataset. The brief introductions of all datasets are as follows:

- **Gowalla [14]**. Gowalla is the check-in dataset obtained from Gowalla[2].
- **Douban [31]**. This dataset is collected from a popular review website Douban[3] in China. We transform explicit data into implicit using the same method as applied in Movielens.
- **Amazon-Book [41]**. The Amazon-Book dataset is a comprehensive collection of data that primarily focuses on book reviews from the Amazon platform[4]. This dataset is often used in research, particularly in the field of recommendation systems.
- **Yelp2018 [13]**. Yelp2018[5] is from the 2018 edition of the Yelp challenge, containing Yelp's bussiness reviews and user data.
- **Movielens-1M [52]**. Movielens is the widely used dataset and is collected from MovieLens[6]. We use the version of 1M. We transform explicit data to implicit feedback by treating all user-item ratings as positive interactions.
- **Food [58]**. Food datasets contain recipe details and reviews from Food.com(formerly GeniusKitchen)[7]. Data includes cooking recipes and review texts.
- **Coat [22] & Yahoo [28]**. These two datasets are obtained from Coat shopping recommendation service and the Yahoo music, respectively. Both datasets contain a training set of biased rating data collecting from the normal user interactions and a test set of unbiased rating data containing user ratings on randomly selected items. The rating data are translated to implicit feedback,*i.e.,* , the rating larger than 3 is regarded as positive.

We processed the original dataset to obtain training and test sets that exhibit three types of distributional shifts, i.e., popularity shift, temporal shift, and exposure shift. Below, we briefly introduce the methods of processing.

- **Popularity shift**. We re-divide the train and test set of the dataset based on item popularity. The test set was designed in such a way that the popularity of all items approximated a uniform distribution, while a long-tail distribution was preserved within the training set. Gowalla, Douban, Amazon-Book, Yelp2018 are used for this type of shift.
- **Temporal shift**. For the datasets under the temporal shift setting, we took the most recent 20% of interaction data from each user as the test set, and the earliest 60% of interaction data as the training set. Movielens-1M, Food are used for this type of shift.
- **Exposure shift**. In the Coat and Yahoo datasets, the training set consists of user ratings collected by the actual system (which are biased), whereas the test set is composed of a random selection of items for each user to rate (which are unbiased), thereby resulting in a distributional shift.

A.5 Experiment Details.

For a fair comparison, the embedding size is fixed to 32 and layer num is set as 3 for all comparison methods. TPE-based Bayesian optimiser is used to search for the optimal hyperparameters. Specifically, as for DR-GNN, the learning rate is tuned from $\{0.1, 0.01\}$ and the coefficient of L_2 regularization term λ is tuned from $\{0, 0.1, 0.01, ..., 1e-6\}$. The α is search from $(0, 1)$. The coefficient γ in GEA is search from $\{0.1, 0.2, 0.3, 0.4, 0.5\}$.

We utilize the all-ranking strategy, in which all items, excluding the positive ones from the training set, are ranked by the recommendation model for each user.

Three commonly used evaluation metrics—Precision@K, Recall@K, and Normalized Discounted Cumulative Gain(NDCG@K) — are used to assess the quality of the recommendations, with K being set by default at 20.

All the experiments were conducted on a server equipped with AMD EPYC 7763 64-Core CPU and RTX4090 (24GB).

---

[2]https://www.gowalla.com/
[3]https://www.douban.com/
[4]https://www.amazon.com/
[5]https://www.yelp.com/dataset
[6]https://movielens.org/
[7]https://cseweb.ucsd.edu/ jmcauley/datasets.html#foodcom