

---

# How Does Topology Bias Distort Message Passing in Graph Recommender? A Dirichlet Energy Perspective

---

Yanbiao Ji<sup>1</sup>   Yue Ding<sup>1\*</sup>   Dan Luo<sup>2</sup>   Chang Liu<sup>1</sup>  
Yuxiang Lu<sup>1</sup>   Xin Xin<sup>3</sup>   Hongtao Lu<sup>1</sup>

<sup>1</sup>Shanghai Jiao Tong University   <sup>2</sup>Lehigh University   <sup>3</sup>Shandong University

{jiyanbiao, dingyue, isonomialiu, luyuxiang\_2018, htlu}@sjtu.edu.cn  
danluo.ir@gmail.com   xinxin@sdu.edu.cn

## Abstract

Graph-based recommender systems have achieved remarkable effectiveness by modeling high-order interactions between users and items. However, such approaches are significantly undermined by popularity bias, which distorts the interaction graph’s structure—referred to as *topology bias*. This leads to overrepresentation of popular items, thereby reinforcing biases and fairness issues through the user-system feedback loop. Despite attempts to study this effect, most prior work focuses on the embedding or gradient level bias, overlooking how topology bias fundamentally distorts the message passing process itself. We bridge this gap by providing an empirical and theoretical analysis from a Dirichlet energy perspective, revealing that graph message passing inherently amplifies topology bias and consistently benefits highly connected nodes. To address these limitations, we propose **Test-time Simplicial Propagation** (TSP), which extends message passing to higher-order simplicial complexes. By incorporating richer structures beyond pairwise connections, TSP mitigates harmful topology bias and substantially improves the representation and recommendation of long-tail items during inference. Extensive experiments across five real-world datasets demonstrate the superiority of our approach in mitigating topology bias and enhancing recommendation quality. The implementation code is available at <https://github.com/sotaagi/TSP>.

## 1 Introduction

Recommender systems (RS) are fundamental components of modern online platforms, playing a crucial role in connecting users with relevant items [16, 64, 8]. Recently, graph-based methods, such as LightGCN [21], have gained prominence in RS, as they effectively capture collaborative filtering signals by modeling high-order connectivity between users and items [34, 66, 63, 72]. Despite their effectiveness, graph-based methods suffer from popularity bias, where popular items are recommended disproportionately more often than less popular items [1, 3]. This notorious effect not only undermines the accuracy and fairness of recommendation [74, 35, 55], but also exacerbates the Matthew Effect and the filter bubble [38, 45] through the user-system feedback loop.

In graph-based recommender systems, popularity bias mainly distorts the topology of the interaction graph, producing a highly biased degree distribution as shown in Figure 1(a), which we refer to as *topology bias*. While research has shown that the topology bias gets further amplified [56, 76] in graph-based methods, the root cause of this effect remains rather unexplored. Although some recent studies have attempted to understand this issue, they exhibit significant limitations. Their

---

\*Corresponding Author

analyses focus mainly on the embedding landscape [28, 26] or gradient bias [10, 71, 31] *after message passing*. The question of how topology bias distorts the message passing process remains unsolved.

To bridge this research gap, we investigate the message passing process in graph-based RS from the Dirichlet energy perspective. Our empirical and theoretical analysis reveals the following results: (1) The graph message passing mechanism is equivalent to minimizing the global graph Dirichlet energy. (2) Popular nodes with high degrees concentrate embedding norms. (3) The update of the norm of a node’s embedding is upper bounded by local Dirichlet energy.

Based on the above studies, we propose **Test-time Simplicial Propagation (TSP)**, which performs message passing over simplicial complexes with intra- and inter-simplex smoothing to avoid the dominance of certain nodes or substructures. By incorporating simplices instead of mere edges, our proposed TSP mitigates harmful topology bias during inference, improving the performance of tail nodes without requiring RS retraining.

We summarize our contributions below:

- We provide comprehensive theoretical analysis of how topology bias distorts message passing in graph-based recommender systems. We establish formal connections between graph message passing, Dirichlet energy optimization, and node embeddings, offering a mathematical framework to understand why popular nodes are systematically favored.
- We propose a principled approach to overcome the limitations of pairwise edge message passing by introducing TSP, which leverages higher-order simplicial complexes for information propagation, enabling more balanced representation learning across nodes of varying popularity.
- We develop TSP as a plug-and-play solution that operates solely during inference, making it compatible with existing recommender systems without requiring costly retraining.
- Through extensive experiments across five diverse real-world datasets, we demonstrate that TSP can effectively mitigate topology bias without impairing the overall recommendation quality. Our results show consistent performance gains in tail nodes as well as competitive or better results in overall recommendation compared to state-of-the-art baselines.

## 2 Preliminaries

### 2.1 Graph-based Recommendation

Let  $\mathcal{U} = \{u_1, u_2, \dots, u_{|\mathcal{U}|}\}$  denote the set of users and  $\mathcal{I} = \{i_1, i_2, \dots, i_{|\mathcal{I}|}\}$  denote the set of items. User-item interactions are represented as an implicit feedback matrix  $\mathbf{R} \in \{0, 1\}^{|\mathcal{U}| \times |\mathcal{I}|}$ , where  $\mathbf{R}_{ui} = 1$  if user  $u$  interacted with item  $i$ , and 0 otherwise. These interactions can be naturally modeled as a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , specifically a user-item bipartite graph where the node set  $\mathcal{V} = \mathcal{U} \cup \mathcal{I}$  and the edge set  $\mathcal{E}$  contains edges  $(u, i)$  if and only if  $\mathbf{R}_{ui} = 1$ . Let  $\mathbf{A} \in \{0, 1\}^{|\mathcal{V}| \times |\mathcal{V}|}$  be the adjacency matrix of the graph and  $\mathbf{D}$  be the diagonal degree matrix where  $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$ . The normalized adjacency matrix is defined as  $\tilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}$ . In the widely adopted LightGCN [21], the message passing mechanism in layer  $l$  can be formally written as:

$$\mathbf{X}^{(l+1)} = \tilde{\mathbf{A}} \mathbf{X}^{(l)}, \quad (1)$$

where  $\mathbf{X}^{(l)} \in \mathbb{R}^{|\mathcal{V}| \times d}$  is the node embedding matrix at layer  $l$ , with  $d$  representing the embedding dimension. The recommendation score for user  $u$  and item  $i$  is then computed as the inner product between their respective embeddings:  $y_{ui} = \mathbf{x}_u^T \mathbf{x}_i$ .

### 2.2 Graph Laplacian and Dirichlet Energy

The graph Laplacian is a fundamental operator that captures the structural properties of a graph. The symmetrically normalized Laplacian [13, 59] is defined as:

$$\tilde{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}. \quad (2)$$

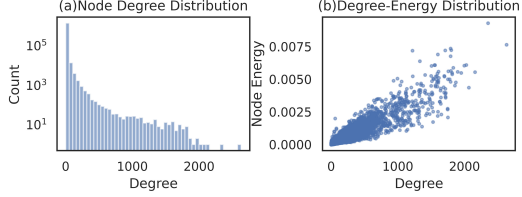


Figure 1: The degree and energy distribution of LightGCN embeddings on Gowalla dataset.

Given node embeddings  $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$ , the Dirichlet energy on the graph measures the smoothness of node embeddings with respect to the graph structure:

**Definition 1 (Graph Dirichlet Energy [13])** *The Dirichlet energy of node embeddings  $\mathbf{X}$  with respect to graph  $\mathcal{G}$  is defined as:*

$$E_{\mathcal{G}}(\mathbf{X}) := \text{Tr}(\mathbf{X}^T \tilde{\mathbf{L}} \mathbf{X}) = \frac{1}{2} \sum_{(i,j) \in \mathcal{E}} \left\| \mathbf{x}_i / \sqrt{d_i} - \mathbf{x}_j / \sqrt{d_j} \right\|_2^2, \quad (3)$$

where  $\text{Tr}(\cdot)$  denotes the matrix trace,  $\mathbf{x}_i \in \mathbb{R}^d$  is the embedding of node  $i$ , and  $d_i$  is its degree.

Building on this global measure, we define a localized node Dirichlet energy:

**Definition 2 (Local Dirichlet Energy)** *The local Dirichlet energy of a node  $v$  is defined as:*

$$E_v(\mathbf{x}_v) := \frac{1}{2} \sum_{j \in \mathcal{N}(v)} \left\| \mathbf{x}_v / \sqrt{d_v} - \mathbf{x}_j / \sqrt{d_j} \right\|_2^2, \quad (4)$$

where  $\mathcal{N}(v)$  denotes the set of neighbors of node  $v$ .

### 2.3 Simplicial Complexes

Simplicial complexes (SC) provide a powerful framework to generalize graphs to higher-order relationships [77]. These structures are combinations of simplices, which we define as follows:

**Definition 3 ( $k$ -simplex)** *A  $k$ -simplex  $\sigma^k$  is the convex hull of  $k + 1$  affinely independent points  $S = \{v_0, v_1, \dots, v_k\}$ , which constitute the vertices of the simplex. The dimension (or order) of the simplex is  $k$ . A  $j$ -simplex ( $j < k$ ) formed by a subset of  $S$  is called a  $j$ -face of  $\sigma^k$ .*

A  $k$ -simplex represents connections among  $k + 1$  nodes: a 0-simplex is a node, a 1-simplex is an edge, a 2-simplex is a triangle, etc. These simplices are organized into simplicial complexes:

**Definition 4 (Simplicial Complex)** *A simplicial complex  $\mathcal{X}$  is a finite collection of simplices satisfying the following properties: (1) Every face of a simplex in  $\mathcal{X}$  is also in  $\mathcal{X}$ . (2) The non-empty intersection of any two simplices in  $\mathcal{X}$  is a face of both.*

*The dimension (or order) of the simplicial complex  $\mathcal{X}$  is the maximum dimension of any simplex in  $\mathcal{X}$ .*

The structure of a simplicial complex can be algebraically represented using boundary matrices:

**Definition 5 (Boundary Matrix)** *The boundary matrix  $\mathbf{B}_k$  maps  $k$ -simplices to  $(k - 1)$ -simplices by encoding the incidence relations between them. Specifically,  $\mathbf{B}_k(i, j) = \pm 1$  if the  $(k - 1)$ -simplex  $\sigma_i^{k-1}$  is a face of the  $k$ -simplex  $\sigma_j^k$ , and 0 otherwise. The sign is determined by the orientation of simplices. Detailed discussion of orientations and examples can be found in Appendix D.*

Now we can define the Hodge Laplacians, generalizing the graph Laplacian to higher-order structures:

**Definition 6 (Hodge Laplacian [22])** *For a  $K$ -dimensional simplicial complex, the  $k$ -th Hodge Laplacian  $\mathbf{L}_k$  is defined as:*

$$\mathbf{L}_0 = \mathbf{B}_1 \mathbf{B}_1^T, \quad \mathbf{L}_K = \mathbf{B}_K^T \mathbf{B}_K, \quad (5)$$

$$\mathbf{L}_k = \mathbf{B}_k^T \mathbf{B}_k + \mathbf{B}_{k+1} \mathbf{B}_{k+1}^T \quad (1 \leq k < K). \quad (6)$$

## 3 Topology Bias from Dirichlet Energy Perspective

In this section, we provide a theoretical analysis of how topology bias shapes the message passing process in graph-based recommender systems. Our analysis uncovers the mechanism by which popular nodes are systematically favored, a phenomenon rooted in an energy minimization dynamic.

### 3.1 Message Passing as Dirichlet Energy Optimization

We begin by establishing a fundamental equivalence between graph message passing and a Dirichlet energy minimization problem. This equivalence allows us to understand how the underlying graph topology directly governs information propagation.

**Lemma 3.1** *The message passing mechanism is equivalent to optimizing the following energy function with Tikhonov regularization [46] and initial embedding matrix  $\mathbf{X}^{(0)}$ :*

$$\min_{\mathbf{X}} \mathcal{J}(\mathbf{X}) = \frac{1}{2} \text{Tr}(\mathbf{X}^T \tilde{\mathbf{L}} \mathbf{X}) + \frac{\mu}{2} \|\mathbf{X} - \mathbf{X}^{(0)}\|_2^2, \quad \mu > 0, \quad (7)$$

where  $\tilde{\mathbf{L}}$  is the normalized graph Laplacian, and  $\mu$  is a regularization parameter.

□ *Proof in Appendix A.1.*

Lemma 3.1 reveals that the message passing operation implicitly solves a global smoothness objective. Topology bias directly alters this energy landscape, thus shaping the resulting embeddings in a degree-dependent manner, as shown by the positive correlation in Figure 1(b).

### 3.2 Effect of Topology Bias on Embedding Norms

Next, we characterize how embeddings after message passing reflect the topology bias. Specifically, we show that nodes with higher degrees unavoidably accumulate larger embedding norms, a dynamic we refer to as *norm concentration*.

**Corollary 3.1 (Norm Concentration)** *Let  $\mathbf{x}_v^*$  denote the embedding of node  $v$  in the minimizer  $\mathbf{X}^*$  of Equation 7. Its squared norm is lower bounded by its degree  $d_v$ :*

$$\|\mathbf{x}_v^*\|_2^2 \geq d_v \cdot C(\tilde{\mathbf{L}}, \mathbf{X}^{(0)}), \quad (8)$$

where  $C$  depends on the graph Laplacian  $\tilde{\mathbf{L}}$  and initial embedding  $\mathbf{X}^{(0)}$ .

□ *Proof in Appendix A.2.*

Corollary 3.1 formalizes that popular nodes (with larger  $d_v$ ) necessarily acquire larger norms during message passing, often resulting in higher recommendation scores. This effect is also noted by Kim et al. [26] in their empirical observations.

### 3.3 Topology Bias in Embedding Update Dynamics

We further show how topology bias persists and is amplified at every message passing step by bounding each node’s embedding update in terms of its local graph energy.

**Lemma 3.2** *The squared norm of the embedding update for node  $v$  in a single message passing layer is upper bounded as follows:*

$$\|\Delta \mathbf{x}_v\|_2^2 \leq E(v), \quad (9)$$

where  $\Delta \mathbf{x}_v$  denotes the change in embedding for node  $v$  after message passing, and  $E(v)$  is the local Dirichlet energy of  $v$ , as defined in Definition 2.

□ *Proof in Appendix A.3.*

Lemma 3.2 exposes a self-reinforcing cycle: since the Dirichlet energy  $E(v)$  itself is greater for higher-degree (popular) nodes as shown in Figure 1(b), these nodes receive larger embedding updates at each propagation step and potentially aggregate more information.

## 4 Methodology

### 4.1 Hodge Decomposition and Simplicial Dirichlet Energy

**Theorem 1 (Hodge Decomposition [22])** *The  $k$ -simplicial space  $\mathbb{R}^{N_k}$  admits an orthogonal direct sum decomposition*

$$\mathbb{R}^{N_k} = \text{im}(\mathbf{B}_k^T) \oplus \ker(\mathbf{L}_k) \oplus \text{im}(\mathbf{B}_{k+1}), \quad (10)$$

where  $\text{im}(\cdot)$  and  $\ker(\cdot)$  are shorthand for the image and kernel spaces of the matrices,  $\oplus$  represents the union of orthogonal subspaces,  $N_k$  is the cardinality of the space of  $k$ -simplex signals.

Theorem 1 shows that any  $k$ -order simplicial signal  $\mathbf{x}_{\sigma^k}$  can be decomposed [42]:

$$\mathbf{x}_{\sigma^k} = \mathbf{x}_{\sigma^k, G} + \mathbf{x}_{\sigma^k, H} + \mathbf{x}_{\sigma^k, C}, \quad (11)$$

where the gradient component  $\mathbf{x}_{\sigma^k, G} = \mathbf{B}_k^T \mathbf{x}_{\sigma^{k-1}}$  (corresponding to Equation 20), curl component  $\mathbf{x}_{\sigma^k, C} = \mathbf{B}_{k+1} \mathbf{x}_{\sigma^{k+1}}$  (corresponding to Equation 22), and the harmonic component  $\mathbf{x}_{\sigma^k, H}$  satisfies

$\mathbf{L}_k \mathbf{x}_{\sigma^k, H} = \mathbf{0}$ . This decomposition provides theoretical guidance for the interaction and conversion between simplicial signals of different orders.

**Definition 7 (Simplicial Dirichlet Energy [61])** *The Dirichlet energy of simplices of order  $k$  can be defined as:*

$$E_{\sigma^k}(\mathbf{X}_{\sigma_i^k}) := \|\mathbf{B}_k \mathbf{X}_{\sigma^k}\|_2^2 + \|\mathbf{B}_{k+1}^T \mathbf{X}_{\sigma^k}\|_2^2 \quad (12)$$

$$= \underbrace{\sum_{\sigma_i^k \cap \sigma_j^k \neq \emptyset} \|\mathbf{x}_{\sigma_i^k} - \mathbf{x}_{\sigma_j^k}\|_2^2}_{\text{shared lower simplices}} + \underbrace{\sum_{\sigma_i^k, \sigma_j^k \subset \sigma^{k+1}} \|\mathbf{x}_{\sigma_i^k} - \mathbf{x}_{\sigma_j^k}\|_2^2}_{\text{shared upper simplices}}. \quad (13)$$

Different from the Dirichlet energy defined on a graph, the simplicial Dirichlet energy consists of two terms: energy between simplices sharing the same lower order component simplices (lower simplices) and energy between simplices consisting of the same higher order simplices (higher simplices).

Consider optimizing the simplicial Dirichlet energy similar to Equation 7:

$$\min_{\mathbf{X}_{\sigma^k}} \frac{1}{2} E_{\sigma^k}(\mathbf{X}_{\sigma_i^k}) + \frac{\mu}{2} \|\mathbf{X}_{\sigma^k} - \mathbf{X}_{\sigma^k}^{(0)}\|_2^2. \quad (14)$$

The gradient with respect to  $\mathbf{X}_{\sigma^k}$  is:

$$\nabla_{\mathbf{X}_{\sigma^k}} = \mathbf{B}_k^T \mathbf{B}_k \mathbf{X}_{\sigma^k} + \mathbf{B}_{k+1} \mathbf{B}_{k+1}^T \mathbf{X}_{\sigma^k} + \mu(\mathbf{X}_{\sigma^k} - \mathbf{X}_{\sigma^k}^{(0)}) = \mathbf{L}_k \mathbf{X}_{\sigma^k} + \mu(\mathbf{X}_{\sigma^k} - \mathbf{X}_{\sigma^k}^{(0)}). \quad (15)$$

One gradient descent step with learning rate  $\eta$  can be written as:

$$\mathbf{X}_{\sigma^k}^{(1)} = \mathbf{X}_{\sigma^k}^{(0)} - \eta \nabla_{\mathbf{X}_{\sigma^k}} = \mathbf{X}_{\sigma^k}^{(0)} - \eta \mathbf{L}_k \mathbf{X}_{\sigma^k}^{(0)} = (\mathbf{I} - \eta \mathbf{L}_k) \mathbf{X}_{\sigma^k}^{(0)}. \quad (16)$$

Notably, Equation 16 is the simplicial version of graph message passing. The orthonormal eigen-decomposition of  $\mathbf{L}_k$  can be written as:

$$\mathbf{L}_k = \sum_{i=1}^N \lambda_i \mathbf{u}_i \mathbf{u}_i^T, \quad (17)$$

where  $0 = \lambda_1 = \dots = \lambda_r < \lambda_{r+1} < \dots \leq \lambda_N$  are the eigenvalues and  $\mathbf{u}_i$  are the corresponding eigenvectors. Similarly, we can define a resolvent to this optimization problem as in Equation 31:

$$\mathbf{R}_k = \mu(\mathbf{L}_k + \mu \mathbf{I})^{-1}. \quad (18)$$

For every initial embedding vector  $\mathbf{v}$ ,  $\mathbf{R}_k \mathbf{v}$  yields the corresponding energy minimizer. Now we show how simplicial message passing can alleviate the topology bias.

**Lemma 4.1** *For any  $\mathbf{v} \in \ker(\mathbf{B}_{k+1})$  we have:  $\|\mathbf{R}_k \mathbf{v}\|_2^2 < \|\mathbf{v}\|_2^2$ .*

□ *Proof of Lemma 4.1 is in Appendix A.4.*

Lemma 4.1 reveals that in simplicial message passing, for any embedding in the kernel of the upper adjacency operator ( $\ker(\mathbf{B}_{k+1})$ ), the norm is strictly reduced by the resolvent operator  $\mathbf{R}_k$ . This means that, unlike traditional graph-based propagation which tends to inflate the norms of embeddings for popular (high-degree) nodes, simplicial message passing can contract such dominant embeddings. By shrinking oversized embedding norms where bias accumulates, the simplicial approach helps regularize the embedding space, leading to a more balanced embedding landscape.

## 4.2 The Proposed TSP

Guided by our theoretical analysis, we propose **Test-time Simplicial Propagation (TSP)** based on simplicial Dirichlet energy optimization, extending the message passing procedure to simplicial complexes. TSP comprises the following key modules: **Semantic Graph Construction**, **Intra-Simplex Smoothing**, **Inter-Simplex Propagation**, and **Multi-Order Fusion**. An overview of the whole pipeline is in Figure 2. We also provide an algorithm description in pseudo code in Appendix E.

**Semantic Graph Construction** The objective of this module is to furnish a topology that is more reflective of *semantic* relationships between nodes, rather than relying solely on raw interaction patterns. Formally, given pretrained node embedding  $\mathbf{X}$  and hyperparameter  $\theta$  controlling the

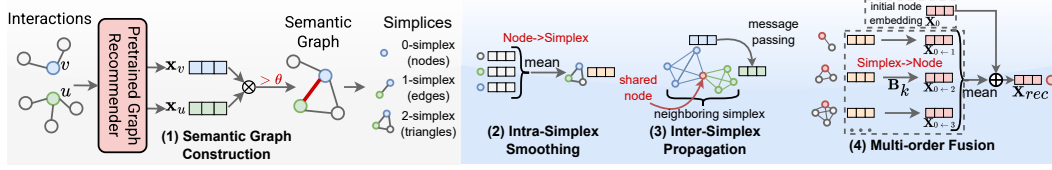


Figure 2: The pipeline of our proposed TSP, consisting of four main steps: (1) Semantic Graph Construction, which builds a semantic graph based on node embeddings; (2) Intra-Simplex Smoothing, which aggregates information within each simplex; (3) Inter-Simplex Propagation, which performs message passing between simplices; and (4) Multi-Order Fusion, which combines embeddings from different orders of simplices for the final recommendation.

similarity threshold, we can construct the adjacency matrix of the semantic graph  $\mathbf{A}_S$ :

$$\mathbf{A}_S[i, j] = \begin{cases} 1, & \mathbf{x}_i^T \mathbf{x}_j \geq \theta \\ 0, & \text{otherwise} \end{cases}, \quad (19)$$

where  $\mathbf{x}_i$  is the embedding of node  $i$ ,  $\mathbf{A}_S[i, j]$  means the element at the  $i$ -th row,  $j$ -th column in  $\mathbf{A}_S$ . This process naturally densifies the tail nodes by giving low-degree nodes additional connections, thus partially alleviating the topology bias [4, 69]. Then we lift this semantic graph into simplicial complex, as described in Appendix I.

**Intra-Simplex Smoothing** Once we have  $\mathbf{A}_S$ , we lift it into a simplicial complex by incorporating nodes, edges, and higher-order simplices (triangles, tetrahedrons, etc.) that capture multi-node connectivity. Each  $k$ -simplex  $\sigma^k$  consists of  $k + 1$  nodes joined by semantic adjacency. Let  $\mathbf{B}_k$  be the boundary matrix of  $k$ -simplices. These matrices enable us to convert node embeddings to high-order simplices signals as in Equation 11. Formally, we can aggregate the information for  $\sigma^k$  from its component nodes as follows:

$$(\mathbf{B}_k^T \mathbf{B}_{k-1}^T \dots \mathbf{B}_1^T) \mathbf{X} \quad (20)$$

where  $\mathbf{X}_{\sigma^k}$  is the embedding matrix of  $k$ -simplices. This process performs smoothing inside the simplex, and also provides initial embeddings for each simplex without the need for embedding learning. The factor  $\prod_{i=1}^k \mathbf{B}_i$  converts the embeddings of component nodes into simplex embeddings.

**Inter-Simplex Propagation** Although intra-simplex averaging helps to smooth embeddings within each simplex structure, it alone does not capture the relations *between* simplices. To address this, we introduce *inter-simplex propagation*. More specifically, we utilize the Hodge Laplacians in Equation 6 to define the message passing between simplices within each order  $k$ :

$$\mathbf{X}_{\sigma^k}^{(l+1)} = (\mathbf{I} - \beta \mathbf{L}_k) \mathbf{X}_{\sigma^k}^{(l)}, \quad (21)$$

where  $\mathbf{L}_k$  is the Hodge Laplacian of order  $k$ ,  $l$  is the message passing layer, and  $\beta$  controls the signal filtering on simplices [62]. This message passing process further refine the representation based on higher-order simplicial structure, incorporating simplex interactions.

**Multi-Order Fusion** To obtain the final recommendation scores, we need to fuse the embeddings from different orders of simplices. We can perform the reverse process of Equation 20 to fuse the embeddings of all the simplices into node embeddings:

$$\mathbf{X}_{0 \leftarrow k} = (\mathbf{B}_1 \mathbf{B}_2 \dots \mathbf{B}_k) \mathbf{X}_{\sigma^k}^{(L)} \quad (22)$$

where  $\mathbf{X}_{0 \leftarrow k}$  represents the node embeddings converted from  $k$ -simplices. We then produce the final representation for recommendation by performing mean pooling of multi-order representations:

$$\mathbf{X}_{\text{rec}} = \mathbf{X}_0 + \text{Mean}(\mathbf{X}_{0 \leftarrow 1}, \dots, \mathbf{X}_{0 \leftarrow K}), \quad (23)$$

where  $\mathbf{X}_0$  is the original pretrained embedding, and  $\mathbf{X}_{0 \leftarrow k}$  are the embeddings from  $k$ -order simplices after message passing. This multi-order fusion ensures that our final embeddings incorporate both interaction information and higher-order semantic relations.

## 5 Experiments

In this section, we empirically evaluate the effectiveness of our proposed method TSP. We aim to answer the following research questions:

- **RQ1:** How does TSP perform compared to existing state-of-the-art debiasing methods?
- **RQ2:** Can TSP improve recommendation fairness and diversity?
- **RQ3:** Can TSP produce fairer item embeddings, as suggested by the theoretical analysis?
- **RQ4:** What is the contribution of each component in TSP to model performance?
- **RQ5:** How much computational overhead is introduced by TSP?

### 5.1 Experimental Setup

**Datasets** We conduct experiments on five real-world datasets representing diverse domains: Yelp2018 (business reviews), Gowalla (location check-ins), Adressa (news), Globo (news), and ML10M (movie ratings). User ratings in ML10M are converted to implicit feedback (1 if rated, 0 otherwise). Details of these dataset statistics are in Appendix F.

**Evaluation Protocols** We split the datasets into 80%, 10%, 10% as train, validation, and test sets following the unbiased sampling method proposed by Wei et al. [56]. We report Recall@20 (R@20) and NDCG@20 (N@20) averaged across all users. Specifically, we also report performance of the 20% least popular items (tail items) beside overall performance. Evaluation details are in Appendix B.

**Backbone Models** We demonstrate TSP’s effectiveness by applying it to three strong GNN-based backbone models: **LightGCN** [21] is a simplified GCN model widely used for collaborative filtering. **SimGCL** [65] is a graph contrastive learning method using noise injection for augmentation. **LightGCL** [9] is the state-of-the-art contrastive learning model using SVD for graph view construction.

**Debiasing Baselines** We compare TSP against several representative debiasing methods: **IPS** [43] uses inverse propensity scoring to reweight interactions based on item popularity. **CauseE** [6] relies on causal inference to learn representations robust to exposure bias. **MACR** [56] employs counterfactual reasoning based on a causal graph to mitigate popularity bias. **SDRO** [58] uses distributionally robust optimization to improve worst-case performance.

**Implementation** GNNs use 3 message passing layers and an embedding dimension of 64. We use the Adam optimizer with learning rate 0.001, weight decay 1e-4, batch size 4096. Training runs for up to 500 epochs with early stopping patience 50. Further details can be found in Appendix G.

### 5.2 Recommendation Performance (RQ1)

Table 1 presents the recommendation performance specifically for tail items, while Figure 3 shows the overall performance across all items on Gowalla dataset. More empirical results of overall recommendation performance can be found in Appendix C.

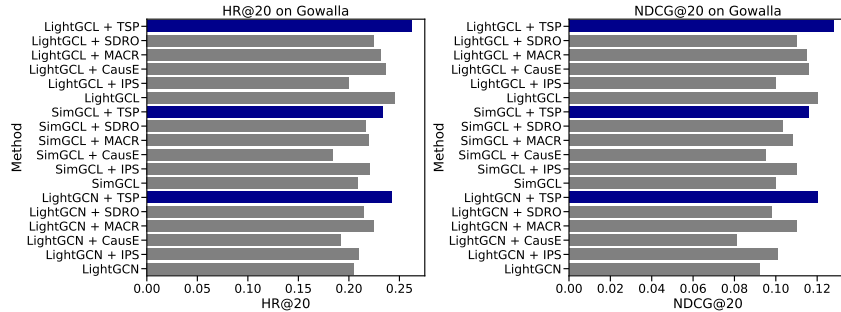


Figure 3: The overall performance of TSP and other baselines on Gowalla dataset.

We have following key observations: (1) TSP consistently achieves substantial improvements in Recall@20 and NDCG@20 for tail items across all datasets and backbone models (Table 1). It frequently outperforms the best baseline debiasing method by a significant margin (%Improve). This validates the strong ability of TSP to improve tail performance. (2) While the primary goal is mitigating topology bias, TSP often maintains or even improves overall performance compared to

Table 1: Performance on tail items (20% least popular items). Best results are in **bold**, and second-best results are underlined. %Improve is TSP’s relative gain over the second-best method. \* indicates the statistically significant improvements (*i.e.*, two-sided t-test with  $p < 0.05$ ) over the best baseline.

| Method       | Adressa      |              | Gowalla      |              | Yelp2018     |              | ML10M        |              | Globo        |              |
|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|              | R@20         | N@20         | R@20         | N@20         | R@20         | N@20         | R@20         | N@20         | R@20         | N@20         |
| LightGCN     | 0.015        | 0.007        | 0.006        | 0.006        | 0.002        | 0.001        | 0.002        | 0.001        | 0.001        | 0.002        |
| + IPS        | <u>0.040</u> | <u>0.020</u> | <u>0.030</u> | <b>0.025</b> | 0.006        | 0.007        | <u>0.011</u> | 0.005        | <u>0.022</u> | <u>0.012</u> |
| + CausE      | 0.023        | 0.011        | 0.020        | 0.013        | <u>0.007</u> | 0.004        | 0.007        | 0.002        | 0.014        | 0.008        |
| + MACR       | 0.016        | 0.007        | 0.014        | 0.008        | 0.005        | 0.004        | 0.007        | 0.002        | 0.006        | 0.005        |
| + SDRO       | 0.014        | 0.007        | 0.012        | 0.007        | 0.005        | 0.003        | 0.007        | 0.002        | 0.003        | 0.004        |
| + TSP (Ours) | <b>0.055</b> | <b>0.024</b> | <b>0.038</b> | <u>0.021</u> | <b>0.009</b> | <b>0.008</b> | <b>0.013</b> | <b>0.008</b> | <b>0.024</b> | <b>0.013</b> |
| %Improve     | 37.50*       | 20.00*       | 26.67*       | -            | 28.57*       | 14.29*       | 18.18*       | 60.00*       | 9.09*        | 8.33*        |
| SimGCL       | 0.016        | 0.007        | 0.008        | 0.007        | 0.002        | 0.003        | 0.003        | 0.002        | 0.002        | 0.001        |
| + IPS        | <u>0.052</u> | <u>0.027</u> | <u>0.021</u> | <u>0.018</u> | <u>0.008</u> | <u>0.006</u> | 0.012        | 0.008        | <b>0.025</b> | <u>0.011</u> |
| + CausE      | 0.018        | 0.008        | <u>0.028</u> | <u>0.018</u> | 0.006        | 0.005        | 0.010        | 0.004        | 0.010        | 0.008        |
| + MACR       | 0.014        | 0.006        | <u>0.017</u> | 0.006        | 0.005        | 0.004        | <u>0.013</u> | <u>0.009</u> | <u>0.012</u> | 0.008        |
| + SDRO       | 0.016        | 0.007        | 0.014        | 0.007        | 0.006        | 0.003        | 0.008        | 0.003        | 0.007        | 0.006        |
| + TSP (Ours) | <b>0.085</b> | <b>0.046</b> | <b>0.055</b> | <b>0.047</b> | <b>0.015</b> | <b>0.014</b> | <b>0.016</b> | <b>0.012</b> | <b>0.025</b> | <b>0.017</b> |
| %Improve     | 63.46*       | 70.37*       | 96.43*       | 161.11*      | 87.50*       | 133.33*      | 23.08*       | 23.33*       | -            | 54.55*       |
| LightGCL     | 0.020        | 0.009        | 0.010        | 0.008        | 0.003        | 0.003        | 0.003        | 0.002        | 0.004        | 0.004        |
| + IPS        | <u>0.045</u> | <u>0.022</u> | <u>0.040</u> | <u>0.030</u> | 0.010        | 0.010        | 0.010        | 0.006        | <u>0.024</u> | <u>0.016</u> |
| + CausE      | 0.022        | 0.008        | 0.026        | 0.019        | 0.002        | 0.002        | 0.003        | 0.001        | 0.015        | 0.011        |
| + MACR       | 0.008        | 0.005        | 0.015        | 0.012        | <u>0.011</u> | <u>0.012</u> | <u>0.011</u> | 0.007        | 0.013        | 0.007        |
| + SDRO       | 0.015        | 0.007        | 0.013        | 0.007        | 0.010        | 0.003        | 0.007        | 0.002        | 0.007        | 0.005        |
| + TSP (Ours) | <b>0.075</b> | <b>0.047</b> | <b>0.050</b> | <b>0.032</b> | <b>0.020</b> | <b>0.019</b> | <b>0.014</b> | <b>0.010</b> | <b>0.032</b> | <b>0.018</b> |
| %Improve     | 66.67*       | 113.64*      | 25.00*       | 6.67*        | 81.82*       | 58.33*       | 27.27*       | 66.67*       | 133.33*      | 12.50*       |

the backbone (Figure 3, Appendix C). In many cases, it also outperforms other debiasing methods on overall metrics, suggesting TSP can effectively balance representations without sacrificing overall recommendation quality. (3) The improvements offered by TSP are particularly pronounced when applied to contrastive learning backbones (SimGCL, LightGCL) compared to LightGCN. This might indicate that contrastive methods can benefit more from higher-order structures.

### 5.3 Recommendation Fairness and Diversity (RQ2)

To assess the effect of TSP on recommendation fairness and diversity, we evaluate two established metrics, Expected Free Discovery (EFD) [47] and Average Percentage of Tail Items (APT) [2], with detailed definitions provided in Appendix B. As reported in Table 2, TSP consistently outperforms competitive baselines on both EFD@20 and APT@20 across all datasets when using the LightGCN backbone, indicating that our method improves the exposure of long-tail content while enhancing overall fairness.

Table 2: Performance on LightGCN backbone with respect to EFD@20 and APT@20. Best results are in **bold**, and second-best results are underlined.

| Method       | Adressa      |              | Gowalla      |              | Yelp2018     |              | ML10M        |              | Globo        |              |
|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|              | EFD@20       | APT@20       | EFD@20       | APT@20       | EFD@20       | APT@20       | EFD@20       | APT@20       | EFD@20       | APT@20       |
| LightGCN     | 0.823        | 0.124        | 0.791        | 0.131        | 0.752        | 0.085        | 0.715        | 0.072        | 0.724        | 0.064        |
| + IPS        | 0.831        | 0.133        | 0.801        | 0.156        | 0.764        | 0.101        | 0.732        | 0.080        | 0.714        | 0.073        |
| + CausE      | 0.846        | <u>0.158</u> | 0.831        | <u>0.157</u> | 0.793        | 0.105        | 0.763        | 0.093        | 0.735        | 0.079        |
| + MACR       | <u>0.901</u> | 0.147        | <u>0.856</u> | 0.150        | <b>0.814</b> | <u>0.118</u> | <u>0.794</u> | <u>0.107</u> | <u>0.756</u> | <u>0.121</u> |
| + SDRO       | 0.832        | 0.136        | 0.781        | 0.138        | 0.735        | 0.110        | 0.705        | 0.097        | 0.721        | 0.112        |
| + TSP (Ours) | <b>0.923</b> | <b>0.217</b> | <b>0.875</b> | <b>0.164</b> | <u>0.813</u> | <b>0.133</b> | <b>0.802</b> | <b>0.116</b> | <b>0.773</b> | <b>0.149</b> |

### 5.4 Distribution of Item Embeddings (RQ3)

To explore the impact of our proposed method on the distribution of learned embeddings, we project the item embeddings learned by LightGCN and LightGCL on the Gowalla dataset into a two-dimensional space using t-SNE visualization. We compare these results with competitive MACR baselines, as shown in Figure 4. Our key observations are as follows: (1) LightGCL exhibits a more balanced distribution of item embeddings compared to LightGCN, aligning with the expectation that contrastive learning methods can learn more uniform representations [51]. (2) The incorporation of simplicial propagation in our method results in a more uniform distribution of item embeddings, validating the effectiveness of simplicial message passing.

Table 3: Ablation study on Gowalla dataset. We report the overall and 20% tail item results of Top 20 performance.

| Variant  | Overall |       | Tail  |       |
|----------|---------|-------|-------|-------|
|          | R@20    | N@20  | R@20  | N@20  |
| TSP-Full | 0.082   | 0.058 | 0.038 | 0.021 |
| TSP-SE   | 0.056   | 0.041 | 0.017 | 0.009 |
| LGN      | 0.047   | 0.038 | 0.006 | 0.006 |

Table 4: TSP preprocessing and inference time costs compared with LightGCN backbone training/inference time.

| Time Cost   | Adressa | Gowalla | Yelp18 | ML10M  | Globo  |
|-------------|---------|---------|--------|--------|--------|
| LGN Train   | 10min   | 35min   | 1h40m  | 9h     | 3h     |
| TSP Preproc | 2.54s   | 15.97s  | 26.32s | 91.34s | 44.29s |
| LGN Infer   | 0.037s  | 0.080s  | 0.134s | 0.575s | 0.501s |
| TSP Infer   | 0.098s  | 0.130s  | 0.200s | 0.650s | 0.606s |

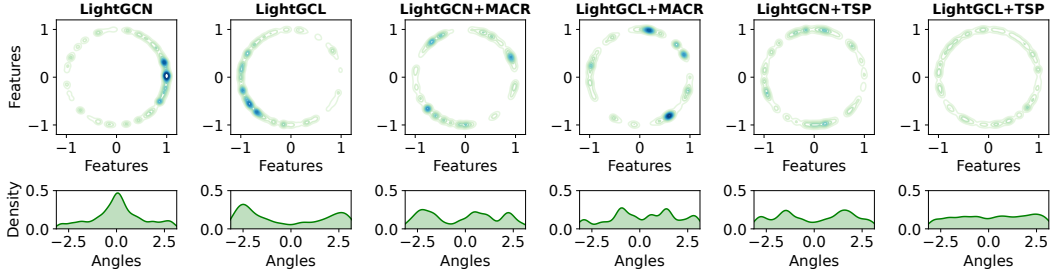


Figure 4: t-SNE visualization of item embeddings learned on Gowalla dataset. The top figures plot the Gaussian KDE of embeddings projected to  $\mathbb{R}^2$ . The darker the color, the more items are in this region. The figures below show the KDE of angles (i.e.  $\arctan2(y, x)$  for each point  $(x, y)$ ).

## 5.5 Ablation Study (RQ4)

To isolate the contribution of simplicial message passing and semantic graph construction to the final performance gains, we conduct an ablation study on Gowalla dataset. The results are in Table 3, with more results in Appendix H. Specifically, we compare following variants of TSP with same LightGCN backbone: (1) **TSP-Full**, the original model, (2) **TSP-SE**, which uses the same semantic graph as TSP, but only performs graph message passing, (3) **LGN**, the LightGCN baseline. The results show that while TSP-SE does provide some improvement, suggesting that denoising the graph structure can improve model performance. The full TSP framework, incorporating message passing on the simplicial complex structure, yields significantly larger gains. This confirms that explicitly modeling and propagating information through higher-order structures is crucial for effectively smoothing representations and achieving fairer recommendations.

## 5.6 Scalability (RQ5)

To assess practical viability (RQ5), we measure the preprocessing time (building the complex and Laplacian) and the additional inference time introduced by TSP compared to the LightGCN backbone. Table 4 shows these times across datasets. The lightweight one-time preprocessing and efficient inference compared to the backbone LightGCN suggests that our approach introduces minimal additional computational overhead. Notably, as the dataset size increases, the relative computational overhead introduced by TSP diminishes in comparison to the backbone model.

## 6 Related Works

**Graph Neural Networks for Recommendation** Graph Neural Networks (GNNs) [15, 48] are widely used for modeling high-order interactions in recommendation tasks. Early works like NGCF [52] utilize graph convolutions to iteratively aggregate collaborative signals. Recent advances focus on improving GNN architectures to enhance efficiency and accuracy, such as LightGCN [21], UltraGCN [37], and SVD-GCN [40]. The integration of contrastive learning [60, 65, 70, 49] has further improved representation robustness, with models like SimGCL [65] demonstrating the benefits of self-supervised learning objectives and lightweight graph augmentations. Emerging research also explores the utilization of large language models (LLMs) enhancing graph recommenders [24, 57, 54], expanding the applicability of graph-based methods.

**Higher-Order and Simplicial Neural Networks** While GNNs effectively capture pairwise relationships, their expressive power is limited by the 1-Weisfeiler-Lehman (1-WL) test [53], impeding their ability to model higher-order structures such as triangles, cliques, and communities [12]. To overcome this, recent studies employ simplicial complexes and higher-order message passing. Models like Simplicial Neural Networks (SNN) [17], BSCNet [11], and HSN [20] extend convolutions beyond simple edges, leveraging Hodge Laplacians for richer structural representations. Simplicial attention mechanisms [19, 18, 29] are also developed to model interactions among higher-order simplices.

**Popularity Bias Mitigation** Popularity bias remains a critical challenge for fair recommendation. Earlier approaches utilize regularization [7, 50] or adversarial frameworks [33, 68, 27] to reduce popularity-driven ranking bias. Causal modeling has emerged as a powerful tool to disentangle popularity effects [56, 75], with counterfactual reasoning being applied to mitigate bias at the inference stage. Newer perspectives focus on employing topological interventions or stochastic graph augmentations [36, 73]. Recent research also investigates debiasing via distributionally robust optimization [58], embedding landscape modeling [32, 28], and reinforcement learning frameworks [44, 30] to achieve a balance between recommendation quality and fairness.

**Oversmoothing in Graph Neural Networks** The performance of GNNs is strongly affected by the depth of the network [5, 14, 39]. As the number of layers increases, the node embeddings contract and converge exponentially toward a constant vector, severely degrading the predictive accuracy [25]. Several metrics have been proposed to quantify oversmoothing and characterize message-passing dynamics, including Dirichlet energy and Mean Average Distance (MAD). Although we also employ Dirichlet energy to analyze message passing, our focus is fundamentally different: we investigate the topology bias instead of network depth. Specifically, we provide the first analysis of how graph topology influences Dirichlet energy and extend this energy-based characterization to higher-order simplicial complexes.

## 7 Conclusion

This paper addresses topology bias in graph-based recommender systems by analyzing message passing through a Dirichlet energy perspective. Based on our theoretical analysis, we propose Test-time Simplicial Propagation (TSP) to address this bias by using higher-order simplicial complex structures for more balanced message passing. Extensive experiments demonstrate that TSP effectively reduces the topology bias and improves recommendation quality.

## Acknowledgments

This paper is supported by NSFC (No. 62176155), Shanghai Municipal Science and Technology Major Project, China, under grant No. 2021SHZDZX0102.

## References

- [1] Himan Abdollahpouri and Masoud Mansoury. Multi-sided exposure bias in recommendation. *arXiv preprint arXiv:2006.15772*, 2020.
- [2] 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>.
- [3] 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.

- [4] Asan Agibetov and Matthias Samwald. Benchmarking neural embeddings for link prediction in knowledge graphs under semantic and structural changes. *Journal of Web Semantics*, 64: 100590, October 2020. ISSN 1570-8268. doi: 10.1016/j.websem.2020.100590.
- [5] Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL <https://openreview.net/forum?id=i800Ph0CVH2>.
- [6] Stephen Bonner and Flavian Vasile. Causal embeddings for recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys '18*, page 104–112, 2018. ISBN 9781450359016.
- [7] Ludovico Boratto, Gianni Fenu, and Mirko Marras. Connecting user and item perspectives in popularity debiasing for collaborative recommendation. *Information Processing and Management*, page 102387, 01 2021.
- [8] Thomas Kleine Buening, Aadirupa Saha, Christos Dimitrakakis, and Haifeng Xu. Bandits meet mechanism design to combat clickbait in online recommendation. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.
- [9] Xuheng Cai, Chao Huang, Lianghao Xia, and Xubin Ren. Lightgcl: Simple yet effective graph contrastive learning for recommendation. In *The Eleventh International Conference on Learning Representations*, 2023.
- [10] Jiajia Chen, Jiancan Wu, Jiawei Chen, Xin Xin, Yong Li, and Xiangnan He. How graph convolutions amplify popularity bias for recommendation? *Frontiers of Computer Science*, 18 (5):185603, 2023. ISSN 2095-2236. doi: 10.1007/s11704-023-2655-2.
- [11] Yuzhou Chen, Yulia R. Gel, and H. Vincent Poor. Bscnets: Block simplicial complex neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(6):6333–6341, Jun. 2022. doi: 10.1609/aaai.v36i6.20583.
- [12] Zhengdao Chen, Lei Chen, Soledad Villar, and Joan Bruna. Can graph neural networks count substructures? In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20*, 2020. ISBN 9781713829546.
- [13] F.R.K. Chung. *Spectral Graph Theory*. Number no. 92 in CBMS Regional Conference Series. Conference Board of the Mathematical Sciences. ISBN 9780821889367.
- [14] Andreea Deac, Marc Lackenby, and Petar Veličković. Expander graph propagation. In Bastian Rieck and Razvan Pascanu, editors, *Proceedings of the First Learning on Graphs Conference*, volume 198 of *Proceedings of Machine Learning Research*, pages 38:1–38:18. PMLR, 09–12 Dec 2022. URL <https://proceedings.mlr.press/v198/deac22a.html>.
- [15] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16*, 2016.
- [16] Yue Ding, Yanbiao Ji, Xun Cai, Xin Xin, Yuxiang Lu, Suizhi Huang, Chang Liu, Xiaofeng Gao, Tsuyoshi Murata, and Hongtao Lu. Towards personalized federated multi-scenario multi-task recommendation. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining, WSDM '25*, page 429–438, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400713293. doi: 10.1145/3701551.3703523.
- [17] Stefania Ebli, Michaël Defferrard, and Gard Spreemann. Simplicial neural networks. In *Topological Data Analysis and Beyond workshop at NeurIPS*, 2020. URL <https://arxiv.org/abs/2010.03633>.
- [18] L. Giusti, C. Battiloro, P. Di Lorenzo, S. Sardellitti, and S. Barbarossa. Simplicial attention neural networks, 2022.
- [19] Christopher Wei Jin Goh, Cristian Bodnar, and Pietro Liò. Simplicial attention networks, 2022.

- [20] Mustafa Hajij, Karthikeyan Natesan Ramamurthy, Aldo Guzmán-Sáenz, and Ghada Za. HIGH SKIP NETWORKS: A HIGHER ORDER GENERALIZATION OF SKIP CONNECTIONS. In *ICLR 2022 Workshop on Geometrical and Topological Representation Learning*, 2022.
- [21] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. 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*, SIGIR '20, page 639–648, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450380164. doi: 10.1145/3397271.3401063.
- [22] W.V.D. Hodge. *The Theory and Applications of Harmonic Integrals*. Cambridge mathematical library. Cambridge University Press, 1989. ISBN 9780521358811.
- [23] Kalervo Järvelin and Jaana Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, page 41–48, New York, NY, USA, 2000. Association for Computing Machinery. ISBN 1581132263. doi: 10.1145/345508.345545. URL <https://doi.org/10.1145/345508.345545>.
- [24] Yanbiao Ji, Yue Ding, Dan Luo, Chang Liu, Jing Tong, Shaokai Wu, and Hongtao Lu. Generating negative samples for multi-modal recommendation, 2025. URL <https://arxiv.org/abs/2501.15183>.
- [25] Yufei Jin and Xingquan Zhu. Oversmoothing alleviation in graph neural networks: a survey and unified view. *Knowledge and Information Systems*, August 2025. ISSN 0219-3116. doi: 10.1007/s10115-025-02548-6. URL <https://doi.org/10.1007/s10115-025-02548-6>.
- [26] Dain Kim, Jinhyeok Park, and Dongwoo Kim. Test-time embedding normalization for popularity bias mitigation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 4023–4027, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701245. doi: 10.1145/3583780.3615281.
- [27] Adit Krishnan, Ashish Sharma, Aravind Sankar, and Hari Sundaram. An adversarial approach to improve long-tail performance in neural collaborative filtering. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 1491–1494, 2018. ISBN 9781450360142.
- [28] Geon Lee, Kyungho Kim, and Kijung Shin. Post-training embedding enhancement for long-tail recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, CIKM '24, page 3857–3861, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704369. doi: 10.1145/3627673.3679978.
- [29] See Hian Lee, Feng Ji, and Wee Peng Tay. Sgat: Simplicial graph attention network. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, 7 2022.
- [30] Allen Lin, Jianling Wang, Ziwei Zhu, and James Caverlee. Quantifying and mitigating popularity bias in conversational recommender systems. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 1238–1247, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392365. doi: 10.1145/3511808.3557423. URL <https://doi.org/10.1145/3511808.3557423>.
- [31] Siyi Lin, Chongming Gao, Jiawei Chen, Sheng Zhou, Binbin Hu, Yan Feng, Chun Chen, and Can Wang. How do recommendation models amplify popularity bias? an analysis from the spectral perspective. In Wolfgang Nejdl, Sören Auer, Meeyoung Cha, Marie-Francine Moens, and Marc Najork, editors, *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining, WSDM 2025, Hannover, Germany, March 10-14, 2025*, pages 659–668. ACM, 2025.
- [32] Siyi Lin, Chongming Gao, Jiawei Chen, Sheng Zhou, Binbin Hu, Yan Feng, Chun Chen, and Can Wang. How do recommendation models amplify popularity bias? an analysis from the spectral perspective. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining, WSDM 2025, Hannover, Germany, March 10-14, 2025*, pages 659–668. ACM, 2025. doi: 10.1145/3701551.3703579.

- [33] Huiting Liu, Lingling Guo, Peipei Li, Peng Zhao, and Xindong Wu. Collaborative filtering with a deep adversarial and attention network for cross-domain recommendation. *Information Sciences*, 565:370–389, 2021. ISSN 0020-0255. doi: <https://doi.org/10.1016/j.ins.2021.02.009>.
- [34] Yuxi Liu, Lianghao Xia, and Chao Huang. Selfggnn: Self-supervised graph neural networks for sequential recommendation. In Grace Hui Yang, Hongning Wang, Sam Han, Claudia Haufl, Guido Zuccon, and Yi Zhang, editors, *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, pages 1609–1618. ACM, 2024.
- [35] Sichun Luo, Chen Ma, Yuanzhang Xiao, and Linqi Song. Improving long-tail item recommendation with graph augmentation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM '23*, page 1707–1716, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701245. doi: 10.1145/3583780.3614929.
- [36] Sichun Luo, Chen Ma, Yuanzhang Xiao, and Linqi Song. Improving long-tail item recommendation with graph augmentation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM '23*, 2023.
- [37] Kelong Mao, Jieming Zhu, Xi Xiao, Biao Lu, Zhaowei Wang, and Xiuqiang He. Ultragn: Ultra simplification of graph convolutional networks for recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management, CIKM '21*, 2021.
- [38] Tien T. Nguyen, Pik-Mai Hui, F. Maxwell Harper, Loren Terveen, and Joseph A. Konstan. Exploring the filter bubble: the effect of using recommender systems on content diversity. In *Proceedings of the 23rd International Conference on World Wide Web, WWW '14*, page 677–686, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450327442. doi: 10.1145/2566486.2568012.
- [39] Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. In *International Conference on Learning Representations*, 2020. URL <https://openreview.net/forum?id=S1ld02EFPr>.
- [40] Shaowen Peng, Kazunari Sugiyama, and Tsunenori Mine. Svd-gcn: A simplified graph convolution paradigm for recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, CIKM '22*, 2022. ISBN 9781450392365.
- [41] PYT-Team. Topox: A suite of python packages for machine learning on topological domains. *arXiv preprint arXiv:2402.02441*, 2024.
- [42] Michael T. Schaub, Yu Zhu, Jean-Baptiste Seby, T. Mitchell Roddenberry, and Santiago Segarra. Signal processing on higher-order networks: Livin’ on the edge... and beyond. *Signal Processing*, 187:108149, October 2021. ISSN 0165-1684. doi: 10.1016/j.sigpro.2021.108149. URL <http://dx.doi.org/10.1016/j.sigpro.2021.108149>.
- [43] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. Recommendations as treatments: debiasing learning and evaluation. In *Proceedings of the 33rd International Conference on Machine Learning - Volume 48, ICML'16*, page 1670–1679, 2016.
- [44] Xiaoyu Shi, Quanliang Liu, Hong Xie, Di Wu, Bo Peng, MingSheng Shang, and Defu Lian. Relieving popularity bias in interactive recommendation: A diversity-novelty-aware reinforcement learning approach. *ACM Trans. Inf. Syst.*, 42(2), November 2023. ISSN 1046-8188. doi: 10.1145/3618107. URL <https://doi.org/10.1145/3618107>.
- [45] Nicholas Sukiennik, Chen Gao, and Nian Li. Uncovering the deep filter bubble: Narrow exposure in short-video recommendation. In *Proceedings of the ACM Web Conference 2024, WWW '24*, page 4727–4735, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400701719. doi: 10.1145/3589334.3648159.
- [46] Andrey N. Tikhonov and Vasiliy Y. Arsenin. *Solutions of ill-posed problems*. V. H. Winston & Sons, Washington, D.C.: John Wiley & Sons, New York, 1977. Translated from the Russian, Preface by translation editor Fritz John, Scripta Series in Mathematics.

- [47] Saúl Vargas and Pablo Castells. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the Fifth ACM Conference on Recommender Systems, RecSys '11*, page 109–116, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450306836. doi: 10.1145/2043932.2043955. URL <https://doi.org/10.1145/2043932.2043955>.
- [48] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.
- [49] Chen Wang, Yueqing Liang, Zhiwei Liu, Tao Zhang, and Philip S. Yu. Pre-training graph neural network for cross domain recommendation. In *2021 IEEE Third International Conference on Cognitive Machine Intelligence (CogMI)*, 2021.
- [50] Chenxu Wang, Aodian Liu, and Tao Qin. Learning-to-rank debias with popularity-weighted negative sampling and popularity regularization. *Neurocomputing*, 587:127681, 2024. ISSN 0925-2312.
- [51] Chenyang Wang, Yuanqing Yu, Weizhi Ma, Min Zhang, Chong Chen, Yiqun Liu, and Shaoping Ma. Towards representation alignment and uniformity in collaborative filtering. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1816–1825, 2022.
- [52] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'19*, 2019. ISBN 9781450361729. doi: 10.1145/3331184.3331267.
- [53] Xiyuan Wang and Muhan Zhang. How powerful are spectral graph neural networks. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 23341–23362. PMLR, 17–23 Jul 2022.
- [54] Yan Wang, Zhixuan Chu, Xin Ouyang, Simeng Wang, Hongyan Hao, Yue Shen, Jinjie Gu, Siqiao Xue, James Zhang, Qing Cui, Longfei Li, Jun Zhou, and Sheng Li. Llmrg: improving recommendations through large language model reasoning graphs. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence, AAAI'24/IAAI'24/EAAI'24*. AAAI Press, 2024. ISBN 978-1-57735-887-9. doi: 10.1609/aaai.v38i17.29887. URL <https://doi.org/10.1609/aaai.v38i17.29887>.
- [55] Chunyu Wei, Jian Liang, Di Liu, Zehui Dai, Mang Li, and Fei Wang. Meta graph learning for long-tail recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, pages 2512–2522. ACM, 2023. doi: 10.1145/3580305.3599428.
- [56] Tianxin Wei, Fuli Feng, Jiawei Chen, Ziwei Wu, Jinfeng Yi, and Xiangnan He. Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender system. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, KDD '21*, page 1791–1800, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383325. doi: 10.1145/3447548.3467289.
- [57] Wei Wei, Xubin Ren, Jiabin Tang, Qinyong Wang, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. Llmrec: Large language models with graph augmentation for recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining, WSDM '24*, page 806–815, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400703713. doi: 10.1145/3616855.3635853. URL <https://doi.org/10.1145/3616855.3635853>.
- [58] 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, 2022. ISBN 9781450390965. doi: 10.1145/3485447.3512255.

- [59] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6861–6871. PMLR, 09–15 Jun 2019.
- [60] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. Self-supervised graph learning for recommendation. *SIGIR '21*, 2021.
- [61] Maosheng Yang and Elvin Isufi. Hodge-aware learning on simplicial complexes, 2023. URL <https://openreview.net/forum?id=QSJKr01Qpy>.
- [62] Maosheng Yang, Elvin Isufi, Michael T. Schaub, and Geert Leus. Simplicial convolutional filters. *IEEE Transactions on Signal Processing*, 70:4633–4648, 2022. ISSN 1941-0476. doi: 10.1109/tsp.2022.3207045.
- [63] Ziqi Yang, Zhaopeng Peng, Zihui Wang, Jianzhong Qi, Chaochao Chen, Weike Pan, Chenglu Wen, Cheng Wang, and Xiaoliang Fan. Federated graph learning for cross-domain recommendation. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.
- [64] Fan Yao, Yiming Liao, Jingzhou Liu, Shaoliang Nie, Qifan Wang, Haifeng Xu, and Hongning Wang. Unveiling user satisfaction and creator productivity trade-offs in recommendation platforms. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.
- [65] Junliang Yu, Xin Xia, Tong Chen, Lizhen Cui, Nguyen Quoc Viet Hung, and Hongzhi Yin. Xsimgcl: Towards extremely simple graph contrastive learning for recommendation. *IEEE Trans. on Knowl. and Data Eng.*, 36(2):913–926, February 2024. ISSN 1041-4347. doi: 10.1109/TKDE.2023.3288135.
- [66] Wenhui Yu and Zheng Qin. Graph convolutional network for recommendation with low-pass collaborative filters. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 10936–10945. PMLR, 2020.
- [67] Wenhui Yu and Zheng Qin. Graph convolutional network for recommendation with low-pass collaborative filters. In *Proceedings of the 37th International Conference on Machine Learning, ICML'20*. JMLR.org, 2020.
- [68] An Zhang, Wenchang Ma, Pengbo Wei, Leheng Sheng, and Xiang Wang. General debiasing for graph-based collaborative filtering via adversarial graph dropout, 2024.
- [69] Jinghao Zhang, Yanqiao Zhu, Qiang Liu, Shu Wu, Shuhui Wang, and Liang Wang. Mining latent structures for multimedia recommendation. In *Proceedings of the 29th ACM International Conference on Multimedia, MM '21*, page 3872–3880. ACM, October 2021. doi: 10.1145/3474085.3475259. URL <http://dx.doi.org/10.1145/3474085.3475259>.
- [70] Junwei Zhang, Min Gao, Junliang Yu, Lei Guo, Jundong Li, and Hongzhi Yin. Double-scale self-supervised hypergraph learning for group recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management, CIKM '21*, 2021.
- [71] Yifei Zhang, Hao Zhu, Yankai Chen, Zixing Song, Piotr Koniusz, and Irwin King. Mitigating the popularity bias of graph collaborative filtering: A dimensional collapse perspective. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- [72] Chu Zhao, Enneng Yang, Yuliang Liang, Pengxiang Lan, Yuting Liu, Jianzhe Zhao, Guibing Guo, and Xingwei Wang. Graph representation learning via causal diffusion for out-of-distribution recommendation. In Guodong Long, Michale Blumstein, Yi Chang, Liane Lewin-Eytan, Zi Helen Huang, and Elad Yom-Tov, editors, *Proceedings of the ACM on Web*

Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025- 2 May 2025, pages 334–346. ACM, 2025.

- [73] Qian Zhao, Zhengwei Wu, Zhiqiang Zhang, and Jun Zhou. Long-tail augmented graph contrastive learning for recommendation. In *Machine Learning and Knowledge Discovery in Databases: Research Track: European Conference, ECML PKDD 2023, Turin, Italy, September 18–22, 2023, Proceedings, Part IV*, 2023.
- [74] Yuying Zhao, Yu Wang, Yunchao Liu, Xueqi Cheng, Charu C. Aggarwal, and Tyler Derr. Fairness and diversity in recommender systems: A survey. *ACM Trans. Intell. Syst. Technol.*, 16(1), January 2025. ISSN 2157-6904. doi: 10.1145/3664928.
- [75] Zihao Zhao, Jiawei Chen, Sheng Zhou, Xiangnan He, Xuezhi Cao, Fuzheng Zhang, and Wei Wu. Popularity bias is not always evil: Disentangling benign and harmful bias for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 35(10):9920–9931, 2023.
- [76] Ziwei Zhu, Yun He, Xing Zhao, Yin Zhang, Jianling Wang, and James Caverlee. Popularity-opportunity bias in collaborative filtering. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, WSDM '21, page 85–93, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450382977.
- [77] Afra Zomorodian. *Topology for Computing*, volume 16. 01 2005. doi: 10.1017/CBO9780511546945.

## NeurIPS Paper Checklist

### 1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

Answer: [\[Yes\]](#)

Justification: See the abstract and introduction part.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

### 2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: See Appendix J.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be

used reliably to provide closed captions for online lectures because it fails to handle technical jargon.

- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

### 3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: We provide all proofs in Appendix A.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

### 4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: We provide implementation details in Appendix G as well as anonymous link to our code implementation.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

- (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
- (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

## 5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [\[Yes\]](#)

Justification: We provide github link to our code implementation and the links of all used datasets.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

## 6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: We provide detailed experimental settings in Section 5 and Appendix B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

## 7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [\[Yes\]](#)

Justification: We report significance results in Table 1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer “Yes” if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

#### 8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [\[Yes\]](#)

Justification: See Appendix G.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

#### 9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [\[Yes\]](#)

Justification: We check the NeurIPS Code of Ethics and our paper conform in every aspect with them.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

#### 10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [\[Yes\]](#)

Justification: See Appendix J.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate

to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

#### 11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

#### 12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have mentioned or cited them properly.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, [paperswithcode.com/datasets](https://paperswithcode.com/datasets) has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

#### 13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: This paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.

- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

**14. Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

**15. Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

**16. Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

# Appendix

## A Proofs

### A.1 Proof of Lemma 3.1

**Lemma 3.1** *The message passing mechanism is equivalent to optimizing the following energy function with Tikhonov regularization [46] and initial embedding matrix  $\mathbf{X}^{(0)}$ :*

$$\min_{\mathbf{X}} \mathcal{J}(\mathbf{X}) = \frac{1}{2} \text{Tr}(\mathbf{X}^T \tilde{\mathbf{L}} \mathbf{X}) + \frac{\mu}{2} \|\mathbf{X} - \mathbf{X}^{(0)}\|_2^2, \quad \mu > 0, \quad (7)$$

where  $\tilde{\mathbf{L}}$  is the normalized graph Laplacian, and  $\mu$  is a regularization parameter.

**Proof** *The gradient of the objective function in Equation (7) with respect to  $\mathbf{X}$  is:*

$$\nabla_{\mathbf{X}} \mathcal{J}(\mathbf{X}) = \tilde{\mathbf{L}} \mathbf{X} + \mu(\mathbf{X} - \mathbf{X}^{(0)}) \quad (24)$$

With a learning rate of  $\eta$ , one step of gradient descent yields:

$$\mathbf{X}^{(1)} = \mathbf{X}^{(0)} - \eta \nabla_{\mathbf{X}} \mathcal{J}(\mathbf{X}^{(0)}) \quad (25)$$

$$= \mathbf{X}^{(0)} - \eta[\tilde{\mathbf{L}} \mathbf{X}^{(0)} + \mu(\mathbf{X}^{(0)} - \mathbf{X}^{(0)})] \quad (26)$$

$$= \mathbf{X}^{(0)} - \eta \tilde{\mathbf{L}} \mathbf{X}^{(0)} \quad (27)$$

$$= (\mathbf{I} - \eta \tilde{\mathbf{L}}) \mathbf{X}^{(0)} \quad (28)$$

When  $\eta = 1$ , Equation (28) becomes  $\mathbf{X}^{(1)} = (\mathbf{I} - \tilde{\mathbf{L}}) \mathbf{X}^{(0)} = \tilde{\mathbf{A}} \mathbf{X}^{(0)}$ , which is exactly the message passing equation described in Equation (1). For  $\eta < 1$ , the operation corresponds to a low-pass filtering on graph signals, consistent with prior work [67].

Alternatively, we can derive the closed-form solution by setting the gradient in Equation (24) to zero:

$$\tilde{\mathbf{L}} \mathbf{X}^* + \mu(\mathbf{X}^* - \mathbf{X}^{(0)}) = 0 \quad (29)$$

$$\Rightarrow (\tilde{\mathbf{L}} + \mu \mathbf{I}) \mathbf{X}^* = \mu \mathbf{X}^{(0)} \quad (30)$$

$$\Rightarrow \mathbf{X}^* = \mu(\tilde{\mathbf{L}} + \mu \mathbf{I})^{-1} \mathbf{X}^{(0)} \quad (31)$$

Let  $\tilde{\mathbf{L}} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^T$  be the eigendecomposition of  $\tilde{\mathbf{L}}$ , where  $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_N)$  is the diagonal matrix of eigenvalues. Then:

$$\mathbf{X}^* = \mu(\mathbf{U} \mathbf{\Lambda} \mathbf{U}^T + \mu \mathbf{I})^{-1} \mathbf{X}^{(0)} \quad (32)$$

$$= \mu \mathbf{U} (\mathbf{\Lambda} + \mu \mathbf{I})^{-1} \mathbf{U}^T \mathbf{X}^{(0)} \quad (33)$$

$$= \mathbf{U} \tilde{\mathbf{\Lambda}} \mathbf{U}^T \mathbf{X}^{(0)} \quad (34)$$

where

$$\tilde{\mathbf{\Lambda}} = \mu(\mathbf{\Lambda} + \mu \mathbf{I})^{-1} = \text{diag} \left( \frac{\mu}{\lambda_k + \mu} \right) \quad \text{for } 1 \leq k \leq N \quad (35)$$

The embedding for node  $v$  can then be expressed as:

$$\mathbf{x}_v^* = \sum_{k=1}^N \left( \frac{\mu}{\lambda_k + \mu} \right) \mathbf{u}_k[v] \mathbf{c}_k, \quad (36)$$

where  $\mathbf{c}_k = \mathbf{u}_k^T \mathbf{X}^{(0)}$  and  $\mathbf{u}_k[v]$  is the  $v$ -th element of eigenvector  $\mathbf{u}_k$ . The spectral filter  $\frac{\mu}{\lambda_k + \mu}$  functions as a low-pass filter on graph signals, attenuating components corresponding to larger eigenvalues.

## A.2 Proof of Corollary 3.1

**Corollary 3.1 (Norm Concentration)** Let  $\mathbf{x}_v^*$  denote the embedding of node  $v$  in the minimizer  $\mathbf{X}^*$  of Equation 7. Its squared norm is lower bounded by its degree  $d_v$ :

$$\|\mathbf{x}_v^*\|_2^2 \geq d_v \cdot C(\tilde{\mathbf{L}}, \mathbf{X}^{(0)}), \quad (8)$$

where  $C$  depends on the graph Laplacian  $\tilde{\mathbf{L}}$  and initial embedding  $\mathbf{X}^{(0)}$ .

**Proof** For the node embedding described in Equation 36, we can obtain a lower bound of norm by only considering the smallest eigenvalue  $\lambda_1 = 0$  corresponding to eigenvector  $\mathbf{u}_1 = (\sqrt{d_1}, \sqrt{d_2}, \dots, \sqrt{d_N})$ , where  $d_i$  is the degree of node  $i$ .

$$\|\mathbf{x}_v^*\|_2^2 = \left\| \sum_{k=1}^N (\mu/(\tilde{\lambda}_k + \mu)) \mathbf{u}_k[v] \mathbf{c}_k^T \right\|_2^2 \quad (37)$$

$$\geq \left\| \mu/(\tilde{\lambda}_1 + \mu) \mathbf{u}_1[v] \mathbf{c}_1^T \right\|_2^2 \quad (38)$$

$$= \|\mathbf{u}_1[v]\|_2^2 \cdot C(\tilde{\mathbf{L}}, \mathbf{X}^{(0)}) = d_v \cdot C(\tilde{\mathbf{L}}, \mathbf{X}^{(0)}) \quad (39)$$

## A.3 Proof of Lemma 3.2

**Lemma 3.2** The squared norm of the embedding update for node  $v$  in a single message passing layer is upper bounded as follows:

$$\|\Delta \mathbf{x}_v\|_2^2 \leq E(v), \quad (9)$$

where  $\Delta \mathbf{x}_v$  denotes the change in embedding for node  $v$  after message passing, and  $E(v)$  is the local Dirichlet energy of  $v$ , as defined in Definition 2.

**Proof** We can write Equation 1 as:

$$\Delta \mathbf{x}_v = \mathbf{x}_v^{(l)} - \mathbf{x}_v^{(l-1)} = \sum_{j \in \mathcal{N}(v)} \frac{1}{\sqrt{d_v d_j}} \mathbf{x}_j^{(l-1)} - \mathbf{x}_v^{(l-1)} \quad (40)$$

$$= \frac{1}{\sqrt{d_v}} \sum_{j \in \mathcal{N}(v)} \left( \frac{1}{\sqrt{d_j}} \mathbf{x}_j^{(l-1)} - \frac{1}{\sqrt{d_v}} \mathbf{x}_v^{(l-1)} \right). \quad (41)$$

$$(42)$$

Then we have:

$$\|\Delta \mathbf{x}_v\|_2^2 = \frac{1}{d_v} \left\| \sum_{j \in \mathcal{N}(v)} \left( \frac{1}{\sqrt{d_j}} \mathbf{x}_j^{(l-1)} - \frac{1}{\sqrt{d_v}} \mathbf{x}_v^{(l-1)} \right) \right\|_2^2 \quad (43)$$

$$\leq \frac{1}{d_v} \cdot d_v \sum_{j \in \mathcal{N}(v)} \left\| \frac{1}{\sqrt{d_j}} \mathbf{x}_j^{(l-1)} - \frac{1}{\sqrt{d_v}} \mathbf{x}_v^{(l-1)} \right\|_2^2 \quad (\text{Cauchy-Schwarz Inequality}) \quad (44)$$

$$= E(v) \quad (45)$$

## A.4 Proof of Lemma 4.1

**Lemma 4.1** For any  $\mathbf{v} \in \ker(\mathbf{B}_{k+1})$  we have:  $\|\mathbf{R}_k \mathbf{v}\|_2^2 < \|\mathbf{v}\|_2^2$ .

**Proof**

$$\|\mathbf{R}_k \mathbf{v}\|_2^2 = \left( \sum_{i=1}^N \frac{\mu}{\lambda_i + \mu} \mathbf{u}_i \mathbf{u}_i^T \mathbf{v} \right)^2 \leq \left( \frac{\mu}{\lambda_{r+1} + \mu} \right)^2 \sum_{i=1}^N \underbrace{\|\mathbf{u}_i\|_2^2}_1 (\mathbf{u}_i^T \mathbf{v})^2 \quad (46)$$

And since  $\mathbf{u}_i$  span the kernel of  $\mathbf{B}_{k+1}$ , we have

$$\sum_{i=1}^N (\mathbf{u}_i^T \mathbf{v})^2 = \|\mathbf{v}\|_2^2. \quad (47)$$

So we get

$$\|\mathbf{R}_k \mathbf{v}\|_2^2 \leq \left( \frac{\mu}{\lambda_{r+1} + \mu} \right)^2 \|\mathbf{v}\|_2^2 < \|\mathbf{v}\|_2^2. \quad (48)$$

## B Evaluation Protocols

### B.1 Unbiased Sampling

To fairly evaluate debiasing performance, standard random train-test splits are inadequate as they inherit the original data’s popularity bias. We adopt the unbiased evaluation protocol from Wei et al. [56]. Specifically, we uniformly sample 10% of interactions for the test set and 10% for validation, ensuring all items have an equal chance of being sampled, regardless of their popularity. The remaining 80% form the training set. This creates unbiased test and validation sets with balanced popularity distributions, allowing for accurate assessment of long-tail performance.

### B.2 Evaluation Metrics

**Recall** Recall@k is the proportion of relevant items found in the top-k recommendations. A higher Recall@k value indicates that the system is better at identifying and ranking relevant items for users. Formally, Recall@k is defined as:

$$\text{Recall@k} = \frac{\text{Number of Relevant Items in Top-k}}{\text{Total Number of Relevant Items}}. \quad (49)$$

**Normalized Discounted Cumulative Gain (NDCG) [23]** NDCG@k is a widely used evaluation metric in recommender systems to measure the ranking quality of suggested items. NDCG@k takes into account both the relevance of recommendations and their positions in the ranked list, assigning higher importance to relevant items appearing earlier. To compute NDCG@k, first calculate the Discounted Cumulative Gain (DCG) up to position k, then normalize it by the ideal DCG (IDCG) obtained from the perfect ranking. Formally, for a list of recommendations  $r_1, r_2, \dots, r_k$ , the DCG@k is defined as:

$$\text{DCG@k} = \sum_{i=1}^k \frac{2^{\text{rel}_i} - 1}{\log_2(i + 1)}, \quad (50)$$

where  $\text{rel}_i$  is the relevance score of each item at position  $i$ . Then NDCG@k is computed as:

$$\text{NDCG@k} = \frac{\text{DCG@k}}{\text{IDCG@k}}, \quad (51)$$

where IDCG@k is the maximum possible DCG@k for the ideal ordering of items.

Here’s a polished version with the APT definition completed and implementation details clarified.

**Expected Free Discovery (EFD) [47]** EFD quantifies how often a system surfaces relevant yet non-obvious items. We define

$$\text{EFD@k} = \frac{1}{k} \sum_{i=1}^k \text{disc}(i) p(\text{rel} \mid i, u) \log_2(p(i \mid \text{seen})), \quad (52)$$

where  $\text{disc}(i)$  discounts lower-rank positions,  $p(\text{rel} \mid i, u)$  is the user-specific relevance probability for the  $i$ -th recommended item, and  $p(i \mid \text{seen})$  denotes the likelihood of encountering item  $i$  given the user’s previously seen items. In our implementation, we set  $\text{disc}(i) = 0.85^{i-1}$ ; we estimate  $p(\text{rel} \mid i, u)$  using the softmax over cosine similarities between item embeddings and the user embedding; and we approximate  $p(i \mid \text{seen})$  via smoothed conditional co-occurrence counts between item  $i$  and the user’s interaction history (Laplace smoothing,  $\alpha = 1$ ) computed on the training set.

**Average Percentage of Tail Items (APT) [2]** APT measures the share of long-tail content in users’ recommendation lists. Let  $\mathcal{U}$  be the evaluation user set,  $r_{u,i}$  the item at rank  $i$  for user  $u$ , and  $\mathcal{T}$  the tail set (defined below). We report APT as:

$$\text{APT@k} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \left( \frac{1}{k} \sum_{i=1}^k \mathbf{1}[r_{u,i} \in \mathcal{T}] \right). \quad (53)$$

Tail items are determined by popularity on the training data: we compute item frequencies  $\{f(j)\}$  and define  $\mathcal{T} = \{j : f(j) \leq \tau\}$ , where  $\tau$  is a threshold chosen to mark the long tail (e.g., bottom 20% by frequency). We use this percentile-based thresholding and average uniformly over users.

## C More Experimental Results

Here we present overall recommendation results on datasets Adressa, Yelp2018, ML10M and Globo extending results in Figure 5-8.

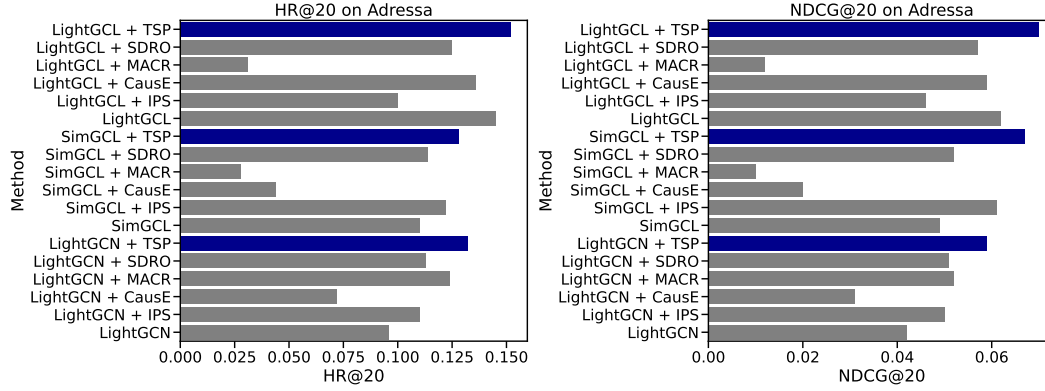


Figure 5: The overall performance of TSP and other baselines on Adressa dataset.

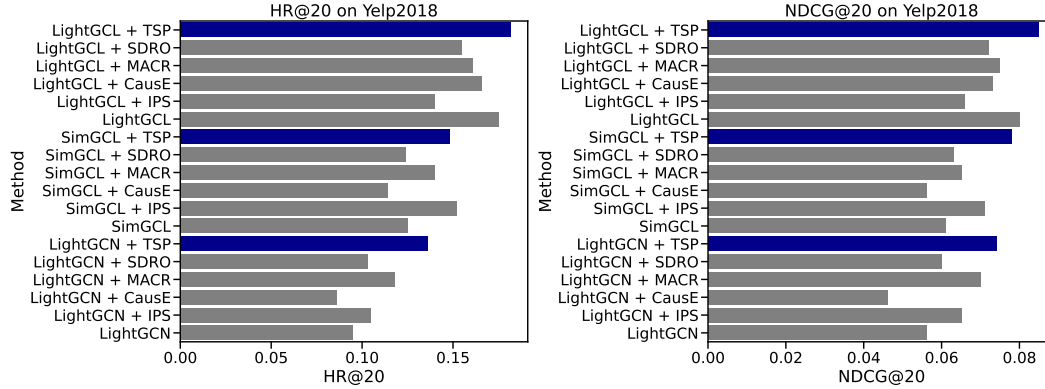


Figure 6: The overall performance of TSP and other baselines on Yelp2018 dataset.

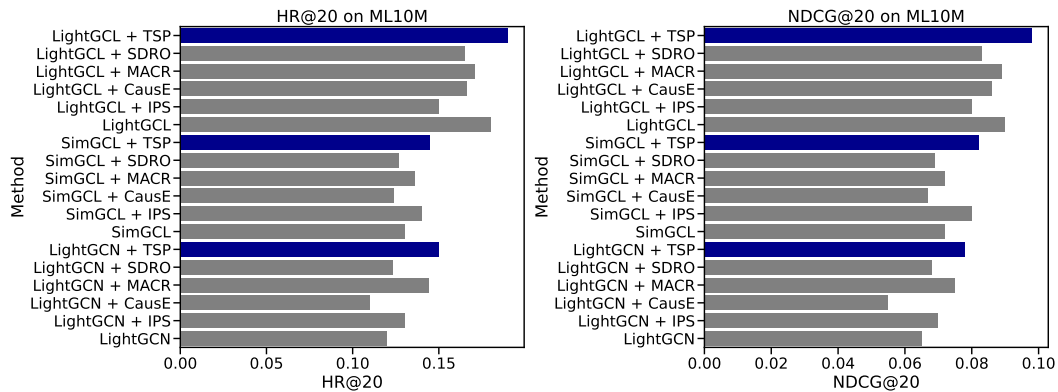


Figure 7: The overall performance of TSP and other baselines on ML10M dataset.

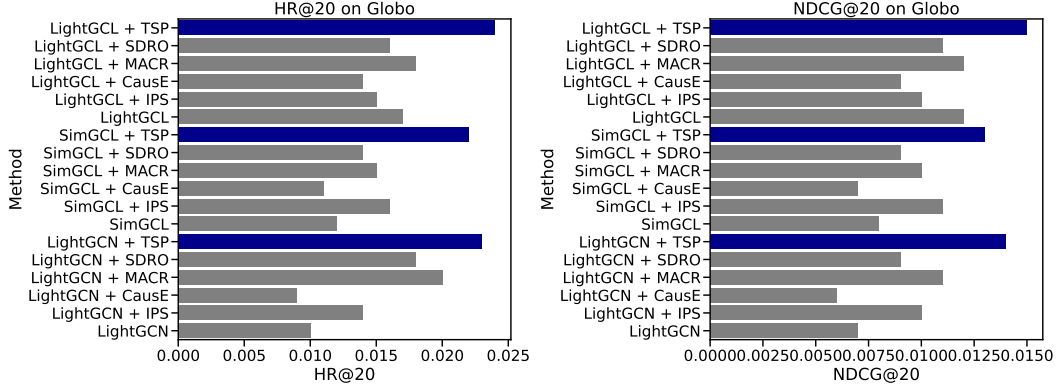


Figure 8: The overall performance of TSP and other baselines on Globo dataset.

## D Examples of Simplicial Complexes

In a simplicial complex, the boundary of a  $k$ -simplex maps to its  $(k - 1)$ -simplices (*i.e.*, the faces of the simplex). For a  $k$ -simplex  $\sigma = \{v_0, v_1, \dots, v_k\}$ , the boundary  $\partial\sigma$  is given by:

$$\partial\sigma = \sum_{i=0}^k (-1)^i \{v_0, v_1, \dots, \hat{v}_i, \dots, v_k\}, \quad (54)$$

where  $\hat{v}_i$  indicates that vertex  $v_i$  is omitted. This rule also determines the orientation of simplices in a SC boundary matrix as  $-1$  or  $1$ , depending on the omitted  $i$  of lower simplices.

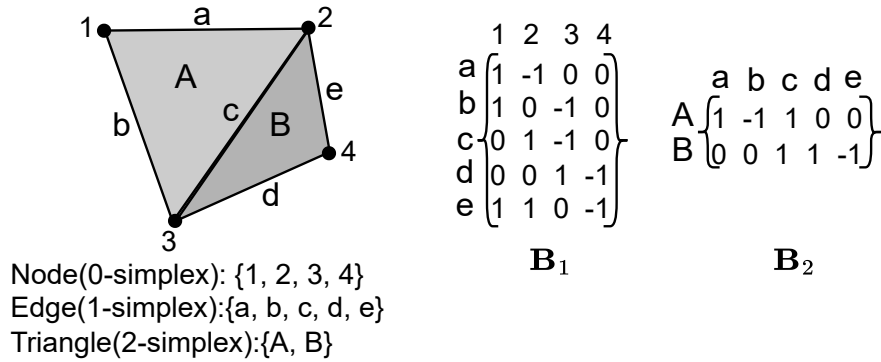


Figure 9: An example of simplices and boundary matrices.

We present an example simplicial complex consisting of 4 vertices in Figure 9. For triangle  $A = \{1, 2, 3\}$ , the edge  $b = \{1, 3\}$  removes the  $v_1$  in Equation 54, so its orientation (sign) in boundary matrix  $B_2$  is  $-1$ .

## E TSP Algorithm

---

**Algorithm 1:** Test-time Simplicial Propagation

---

**Input:** Pretrained node embedding  $\mathbf{X}_0$

Boundary matrices  $\mathbf{B}_k$

Hodge Laplacians  $\mathbf{L}_k$

Number of propagation layers  $L$

Simplicial filter coefficient  $\beta$

Maximum SC order  $K$

**Output:** final node embeddings  $\mathbf{X}_{rec}$

```

1 for  $k = 1$  to  $K$  do
    /* Intra-Simplex Smoothing */
2    $\mathbf{X}_{\sigma^k}^{(0)} \leftarrow (\prod_{i=1}^k \mathbf{B}_i^T) \mathbf{X}_0$ 
3 for  $k = 0$  to  $K$  do
    /* Inter-Simplex Propagation */
4   for  $l = 1$  to  $L$  do
5      $\mathbf{X}_{\sigma^k}^{(l+1)} \leftarrow (\mathbf{I} - \beta \mathbf{L}_k) \mathbf{X}_{\sigma^k}^{(l)}$ 
6 for  $k = 1$  to  $K$  do
    /* Multi-order Fusion */
7    $\mathbf{X}_{0 \leftarrow k} \leftarrow (\prod_{i=1}^k \mathbf{B}_i) \mathbf{X}_{\sigma^k}^{(L)}$ 
8  $\mathbf{X}_{rec} \leftarrow \mathbf{X}_0 + \text{Mean}(\mathbf{X}_{0 \leftarrow 1}, \dots, \mathbf{X}_{0 \leftarrow K})$ 
9 return  $\mathbf{X}_{rec}$ 

```

---

## F Datasets

Five real-world recommendation datasets are used to evaluate the performance of TSP and other baselines with statistics in Table 5. The details of these datasets are as follows:

- **Adressa**<sup>2</sup> is a news dataset that includes news articles (in Norwegian) in connection with anonymized users.
- **Gowalla**<sup>3</sup> is a location-based social networking website where users share their locations by checking-in. The friendship network is undirected and was collected using their public API, and consists of 196,591 nodes and 950,327 edges.
- **Yelp2018**<sup>4</sup> is adopted from the 2018 edition of the yelp challenge, consisting of user interactions with local businesses like restaurants and bars. The 10-core setting (*i.e.*, only users and items with at least 10 interaction records are included) is adopted in order to ensure data quality.
- **Globo**<sup>5</sup> consists of users interactions logs (page views) from a news portal provided by Globo.com, the most popular news portal in Brazil. This dataset also includes rich user contextual attributes such as click\_os, click\_country, click\_region and click\_referrer\_type.
- **ML10M**<sup>6</sup> is part of the movielens datasets with 10 million ratings by users. This dataset collects people’s expressed preferences for movies as 0-5 star ratings. In experiments we follow the common practice of converting ratings to binary interactions.

## G Implementation Details

Here we list the hyperparameter search intervals we use on evaluation sets across different datasets in Table 6. All experiments are conducted on a single Nvidia RTX A6000 Ada GPU with 48GB memory, with 3 repeated runs averaged for final results.

---

<sup>2</sup><https://reclab.idi.ntnu.no/dataset>

<sup>3</sup><https://snap.stanford.edu/data/loc-gowalla.html>

<sup>4</sup><https://www.yelp.com/dataset>

<sup>5</sup><https://www.kaggle.com/gspmoreira/news-portal-user-interactions-by-globocom>

<sup>6</sup><https://grouplens.org/datasets/movielens/10m>

Table 5: Statistics of the datasets.

| Dataset  | Users   | Items  | Interactions | Sparsity |
|----------|---------|--------|--------------|----------|
| Adressa  | 13,485  | 744    | 116,321      | 0.011594 |
| Globo    | 158,323 | 12,005 | 2,520,171    | 0.001326 |
| ML10M    | 69,166  | 8,790  | 5,000,415    | 0.008225 |
| Yelp2018 | 31,668  | 38,048 | 1,561,406    | 0.001300 |
| Gowalla  | 29,858  | 40,981 | 1,027,370    | 0.000840 |

Table 6: Search intervals of hyperparameters for TSP and other baselines.

| Models   | Hyperparameters                                                                                                                                                            |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| LightGCN | reg weight $\alpha$ : [1e-4, 5e-4, 1e-3, 5e-3, 1e-2]                                                                                                                       |
| LightGCL | CL loss weight $\lambda$ : [1e-5, 1e-6, 1e-7, 1e-8], rank of SVD $q$ : [2, 5, 8, 10]                                                                                       |
| SimGCL   | CL loss weight $\lambda$ : [0.01, 0.05, 0.1, 0.2, 0.5, 1], noise magnitude $\epsilon$ : [0, 0.01, 0.05, 0.1, 0.2, 0.5]                                                     |
| IPS      | -                                                                                                                                                                          |
| CausE    | -                                                                                                                                                                          |
| MACR     | intermediate preference blocking $c$ : [10, 20, 30, 40, 50, 60], item loss weight $\alpha$ : [1e-5, 1e-4, 1e-3, 1e-2], user loss weight $\beta$ : [1e-5, 1e-4, 1e-3, 1e-2] |
| SDRO     | -                                                                                                                                                                          |
| TSP      | maximum SC dimension $K$ : [3, 4, 5, 6], simplicial filter $\beta$ : [1e-3, 5e-3, 1e-2, 5e-2, 1e-1, 5e-1, 1], simplicial message passing layer $L$ : [2, 3, 4]             |

## H Ablation Study

We provide full performance of the variants in Section 5.5 across datasets with LightGCN backbone in Table 7.

Table 7: Ablation study across datasets. We report the overall and 20% tail item results of Top 20 performance. Best results are in **bold**, and second-best results are underlined.

| Scope   | Variant  | Adressa      |              | Yelp2018     |              | Gowalla      |              | Globo        |              | ML10M        |              |
|---------|----------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|         |          | R@20         | N@20         | R@20         | N@20         | R@20         | N@20         | R@20         | N@20         | R@20         | N@20         |
| Overall | TSP-Full | <b>0.132</b> | <b>0.059</b> | <b>0.020</b> | <b>0.018</b> | <b>0.082</b> | <b>0.058</b> | <b>0.027</b> | <b>0.019</b> | <b>0.027</b> | <b>0.018</b> |
|         | TSP-SE   | <u>0.108</u> | <u>0.046</u> | <u>0.010</u> | <u>0.006</u> | <u>0.056</u> | <u>0.041</u> | <u>0.016</u> | <u>0.012</u> | <u>0.015</u> | <u>0.011</u> |
|         | LGN      | 0.096        | 0.042        | 0.007        | 0.002        | 0.047        | 0.038        | 0.010        | 0.007        | 0.009        | 0.008        |
| Tail    | TSP-Full | <b>0.055</b> | <b>0.024</b> | <b>0.009</b> | <b>0.008</b> | <b>0.038</b> | <b>0.021</b> | <b>0.024</b> | <b>0.013</b> | <b>0.013</b> | <b>0.008</b> |
|         | TSP-SE   | <u>0.024</u> | <u>0.013</u> | <u>0.006</u> | <u>0.003</u> | <u>0.017</u> | <u>0.009</u> | <u>0.007</u> | <u>0.006</u> | <u>0.006</u> | <u>0.003</u> |
|         | LGN      | 0.015        | 0.007        | 0.004        | 0.002        | 0.006        | 0.006        | 0.001        | 0.002        | 0.002        | 0.001        |

## I Lifting Graphs into Simplicial Complex

The implementation of lifting graphs into (clique) simplicial complexes centers on finding every clique in the graph, where a clique is a subset of vertices with every pair connected by an edge. Efficient algorithms such as Bron–Kerbosch are often employed for this task. Each detected clique of  $k$  vertices is then interpreted as a simplex of dimension  $k - 1$ , so, for example, a triangle in the graph (a 3-clique) becomes a 2-dimensional simplex in the complex. To ensure the resulting structure is a simplicial complex, not only the maximal cliques but also all their subcliques should be included, since every subset of a clique is itself a clique and thus forms a lower-dimensional simplex.

We simplify this complicated process with TopoNetX [41], a suite specially designed to deal with topological data.

## **J Broader Impact and Limitations**

In this paper we focus on addressing the topology bias in recommender systems, ensuring fairer recommendation results. This can directly contribute to the development of more equitable information access and distribution, promoting diversity and inclusivity in user experiences across various platforms. However, our approach is not without its limitations. Our current model solely concentrates on analyzing simplified graph message passing without incorporating weights or non-linear activations. While this simplification is often adopted to simplify the analysis on graph message passing, this may lead to incomplete understanding of how topology bias influences this process. We leave this for future work.