WSFE: Wasserstein Sub-graph Feature Encoder for Effective User Segmentation in Collaborative Filtering

Yankai Chen
The Chinese University of Hong Kong
ykchen@cse.cuhk.edu.hk

Yifei Zhang
The Chinese University of Hong Kong
yfzhang@cse.cuhk.edu.hk

Menglin Yang
The Chinese University of Hong Kong
mlyang@cse.cuhk.edu.hk

Zixing Song
The Chinese University of Hong Kong
zxsong@cse.cuhk.edu.hk

Chen Ma
City University of Hong Kong
chenma@cityu.edu.hk

Irwin King
The Chinese University of Hong Kong
king@cse.cuhk.edu.hk

Abstract

Maximizing the user-item engagement based on vectorized embeddings is a standard procedure of recent recommender models. Despite the superior performance for item recommendations, these methods however implicitly deprioritize the modeling of *user-wise similarity* in the embedding space; consequently, identifying similar users is underperforming, and additional processing schemes are usually required otherwise. To avoid thorough model re-training, we propose WSFE, a model-agnostic and training-free representation encoder, to be flexibly employed on the fly for effective user segmentation. Underpinned by the optimal transport theory, the encoded representations from WSFE present a matched user-wise similarity/distance measurement between the realistic and embedding space. We incorporate WSFE into six state-of-the-art recommender models and conduct extensive experiments on six real-world datasets. The empirical analyses well demonstrate the superiority and generality of WSFE to fuel multiple downstream tasks with diverse underlying targets in recommendation.

1 Introduction

Collaborative filtering (CF), as one effective strategy to perform personalized modeling and prediction, has been widely deployed for recommendation. One prevalent learning paradigm of CF models [4, 13, 25, 45, 47, 50] is to parameterize users and items as vectorized embeddings and learn to reconstruct users' historical interactions. As such, the learned embeddings are convenient to interpret target users' diverse preferences and predict their future behaviors.
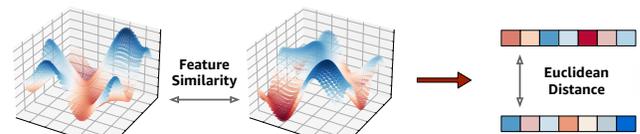
Figure 1: Illustration of WSFE in encoding the similarity of empirical feature distribution (left) with the corresponding embedding distance (right).

In addition to reflecting preferences on items, another desirable property of learned user embeddings is to explicitly capture the *user-wise similarity*; this provides an intuitive recognition of similar user interests and affinities, which lays the foundation and is particularly beneficial for *user-centric* analyses and applications such as group recommendation and advertising [37, 39]. However, this property is usually deprioritized and neglected by recent models [13, 22, 29, 44, 46, 48]. To address the unsatisfactory performance in similar user identification, thorough model re-training may thus be required. To tackle this issue, we are motivated to encode high-quality embeddings in collaborative filtering, such that they can efficiently and seamlessly serve the task of user segmentation.

In this work, we propose Wasserstein Sub-graph Feature Encoder (WSFE), to explicitly model the user behaviors in the form of user-item interaction graph, and measure the user-wise similarity by exploiting their high-order sub-graph patterns. We notice that users with similar interaction behaviors naturally share overlapping sub-graph patterns. Based on this observation, one straightforward solution would be to exhaustively calculate similarities for all the nodes in underlying sub-graphs; this however may be intractable in practice mainly because of the exponential node scale in graph exploration. On the contrary, our proposed WSFE captures user similarity by directly encoding their sub-graph latent features, enabling it model-agnostic and flexible for a variety of graph-based recommender models. Specifically, as shown in Figure 1, we assume the user preference follows an unknown high-dimensional probability distribution; this unique preference distribution is *empirically observed and represented by the latent features that are well-learned in the item recommendation task*. Then WSFE explicitly captures the distribution distances with Wasserstein metrics from the optimal transport theory [19, 32, 35, 42]. Consequently, the encoded user representations can effectively reflect their realistic item-interaction similarity, producing a matched Euclidean distance measurement for ease of user segmentation in the embedding space.

To summarize, our contributions are highlighted as follows:

- To the best of our knowledge, we are the first to focus on improving the embedding quality for effective user segmentation in collaborative filtering, while not jeopardizing the model evaluation for item recommendation.
- We propose WSFE for effective representation encoding via capturing the feature similarity of high-order user-item interaction graph patterns. WSFE is adaptive for any graph-based models and training-free; thus it can be invoked on the fly as long as the backbone models are well-trained.
- We conduct extensive experiments by fusing WSFE into six state-of-the-art models on six real-world datasets. Not only do we present its performance superiority in empirical evaluation, but we also provide technical discussion for future investigation.

2 WSFE Methodology

2.1 Preliminaries

Graph-based Collaborative Filtering. In view of user-item interaction graphs, the general idea of graph-based approaches is to capture CF signals in high-hop neighbors. In this work, we study the *Graph Convolutional Networks (GCNs)* to learn node representations by smoothing the latent features via topology [3, 18, 40, 41]. It iteratively propagates neighborhood information to the target node, e.g., user u, which can be abstracted:

$$v_{ngh \to u}^{(l)} = Prop\Big(\{v_i^{(l-1)} : i \in \mathcal{N}(u)\}\Big), \quad (1)$$

where $v_{ngh \to u}^{(l)}$ is the representation after l layers of propagation from interacted items in u's neighboring set $\mathcal{N}(u)$. With the propagated information, node embeddings are iteratively updated by aggregating features of the center and neighbor nodes [11, 12, 53].

Optimal Transport and Wasserstein Metrics. Optimal transport is the general problem of moving one distribution of mass, e.g., P, to another, e.g., Q, as efficiently as possible. The derived minimum L_2 cost can be referred as their distribution distance:

$$W_2(P, Q) = \Big(\inf_{f \in \mathcal{F}(P,Q)} \int \|x - f(x)\|^2 dP(x)\Big)^{\frac{1}{2}}, \quad (2)$$

where the infimum is over all transport plans in $\mathcal{F}$ between P and Q. For *one-dimensional* distributions, there is a closed-form solution to compute such *optimal transport map* f^* as $f^*(x) := F_P^{-1}(F_Q(x))$; F is the cumulative distribution function (CDF) associated with P.

For the *high-dimensional* case, the metric of *sliced-Wasserstein distance* [2, 8, 35] is formally defined as follows:

$$SW_2(P, Q) = \Big(\int_{\mathbb{S}^{d-1}} W_2^2(P^\theta, Q^\theta) d\theta\Big)^{\frac{1}{2}}, \quad (3)$$

where P^θ is projected by function $g^\theta : \mathbb{R}^d \to \mathbb{R}$ as $P^\theta := g^\theta(P)$ and $g^\theta(x) = \theta^\top x$. θ is a unit vector in $\mathbb{R}^d$ and $\mathbb{S}^{d-1}$ is the unit d-dimensional hypersphere. Due to holding *positive-definiteness*, *symmetry*, and *triangle inequality* [20, 21, 32, 36, 55], we employ it as the distance measurement for high-dimensional subgraph feature distributions.

2.2 Sub-graph Feature Encoding

2.2.1 Formulating Sub-graph Feature Distributions. As illustrated in Figure 2(A), if two users are considered to be similar in terms of historical preferences, they should share similar behaviors with overlapping interaction graph patterns. Based on this intuition, consider that each user's preference follows an unknown,
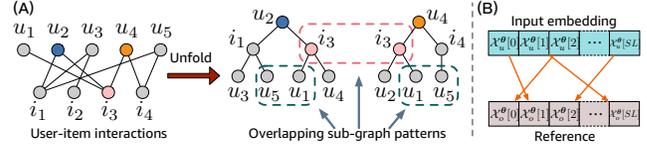

Figure 2: (A) Illustration of the similar sub-graph patterns. (B) Mapping processing between the input and reference.

independent, and d-dimensional probability measure, e.g., P_u. We assume that the interaction pattern of u observed so far is sampled from the underlying distribution P_u. Thus the *empirical (discrete) distribution* $\widehat{P}_u$ with its empirical CDF can be formulated as:

$$F_{\widehat{P}_u}(x) = \frac{1}{L+1} \sum_{l=0}^{L} \delta(x - v_{ngh \to u}^{(l)}). \quad (4)$$

Notice that we initialize $v_{ngh \to u}^{(0)} := v_u^{(0)}$. Here $\delta(\cdot)$ returns 1 if the input is zero and 0 otherwise (discretizing from continuous case $\int \delta(x)dx = 1$). Without loss of generality, these empirical distributions are representative, i.e., $\widehat{P}_u \approx P_u$; thus we would refer P_u to $\widehat{P}_u$ hereafter to avoid notation abuse.

2.2.2 Implementing f^*. We first set a d-dimensional *reference distribution* P_o that functions as the "origin" in the embedding space to measure the distance toward any inputs. P_o is associated with random feature embeddings, e.g., $v_o^{(l)}$, as: $F_{P_o}(x) = \frac{1}{L+1} \sum_{n=0}^{L} \delta(x - v_o^{(l)})$. To implement the optimal transport map f^* for such discrete and d-dimensional case, we have the following procedure.

We first conduct distribution slicing to P_o and P_u by projection function g^θ. For each pair of distribution slices P_u^θ and P_o^θ, let $\mathcal{X}_u^\theta$ collect their projected sub-graph features as $\mathcal{X}_u^\theta = \{\theta^\top v_{ngh \to u}^{(l)}\}_{l=0}^{L}$ (so does for $\mathcal{X}_o^\theta = \{\theta^\top v_o^{(l)}\}_{l=0}^{L}$). Then the corresponding optimal transport map $f^*(x) := F_{P_u^\theta}^{-1}(F_{P_o^\theta}(x))$ can be quantitatively intepreted:

$$f^*(x|\mathcal{X}_u^\theta) = \operatorname{argmin}_{x' \in \mathcal{X}_u^\theta} \big(F_{P_o^\theta}(x') = r\big), \text{ where } r = F_{P_o^\theta}(x). \quad (5)$$

Furthermore, let $\tau(x'|\mathcal{X}_u^\theta)$ denote the ranking of each input x' in the *ascending sorting* of $\mathcal{X}_u^\theta$. We can replace the term $F_{P_u^\theta}$ and have:

$$f^*(x|\mathcal{X}_u^\theta) = \operatorname{argmin}_{x' \in \mathcal{X}_u^\theta} \big(\tau(x'|\mathcal{X}_u^\theta) = \tau(x|\mathcal{X}_o^\theta)\big). \quad (6)$$

As shown in Figure 2(B), Eqn.(6) essentially permutes different layers of sub-graph embeddings of $\mathcal{X}_u^\theta$ in encoding, such that the distance to the reference of $\mathcal{X}_o^\theta$ can be subsequently captured and embedded. Please notice that the distance is in the L_2-norm form as shown in Eqn.(2), *a.k.a.* the Euclidean distance, which is favorable to scenarios for recalling vectorized objects that requires a reasonable distance measurement in the embedding space.

2.2.3 Implementing WSFE. For each pair of distribution slices, based on the algorithmic implementation of Eqn.(6), we proceed to encode their representations as follows:

$$E_u^\theta := \frac{1}{L+1} \Big\|_{l=0}^{L} f^*\big(\theta^\top v_o^{(l)} | \mathcal{X}_u^\theta\big) - E_o^\theta \text{ and } E_o^\theta := \frac{1}{L+1} \Big\|_{l=0}^{L} \theta^\top v_o^{(l)}, \quad (7)$$

where $||$ denotes the concatenation operation. According to the theory in Eqn.(3), the next step is to draw infinite projections for distance integral, which, however, may be computationally expensive and infeasible in practice. In this work, we implement it with Monte-Carlo approximation with S times of uniform sampling from $\mathbb{S}^{d-1}$. Consequently, this leads to a cumulative sliced-Wasserstein

distance (i.e., approximating Eqn.(3)) between reference P_o and the original input feature distribution P_u as:

$$SW_2(P_o, P_u) \approx \left(\frac{1}{S} \sum_{s=1}^{S} W_2^2 \left(P_o^{\theta_s}, P_u^{\theta_s} \right) \right)^{\frac{1}{2}}. \tag{8}$$

Regularized by the distance cumulation in Eqn.(8), our *Wasserstein Sub-graph Feature Encoder (WSFE)* is finally defined:

$$E_u := \frac{1}{S} \Big\|_{s=1}^{S} E_u^{\theta_s} \quad \text{and} \quad E_o := \frac{1}{S} \Big\|_{s=1}^{S} E_o^{\theta_s}, \tag{9}$$

where $E_u, E_o \in \mathbb{R}^{S(L+1)}$. Notice that in practice, the number of graph convolutions $L \leq 4$ [11, 13, 18] is a common setting mainly to avoid the *over-smoothing problem* [23]. Moreover, our empirical observations in § 3.2.2 reveal that setting $S = 64$ already achieves satisfactory model performances with an acceptable computational cost.

2.3 Theoretical Analysis

One major expectation of encoded representations is that they can reflect the similarity/distance of their sub-graph feature distributions. We illustrate this in Figure 3 with the theorem as follows:

THEOREM 1. *For any input sub-graph features of users u_i and u_j with distributions P_{u_i} and P_{u_j}, their encoded representations hold:*

① $\|E_{u_i}\|_2 \approx SW_2(P_{u_i}, P_0)$.
② $\|E_{u_i} - E_{u_j}\|_2 \approx SW_2(P_{u_i}, P_{u_j})$.

PROOF. The proof is twofold.
For property ②, we have:

$$\|E_{u_i} - E_{u_j}\|_2$$
$$= \left\| \Big\|_{s=1}^{S} (E_{u_i}^{\theta_s} - E_{u_j}^{\theta_s}) \right\|_2$$
$$= \left\| \sum_{s=1}^{S} (E_{u_i}^{\theta_s} - E_{u_j}^{\theta_s})^2 \right\|_2$$
$$= \left\| \frac{1}{S} \sum_{s=1}^{S} \frac{1}{L+1} \sum_{l=0}^{L} \left(f^*(\theta_s^{\mathsf{T}} v_o^{(l)} | \mathcal{X}_{u_i}^{\theta_s}) - f^*(\theta_s^{\mathsf{T}} v_o^{(l)} | \mathcal{X}_{u_j}^{\theta_s}) \right)^2 \right\|_2$$
$$\approx \left\| \int_{\mathbb{S}^{d-1}} \int_{\mathbb{R}} \left(F_{P_{u_i}^{\theta}}^{-1} \left(F_{P_o^{\theta}}(t) \right) - F_{P_{u_j}^{\theta}}^{-1} \left(F_{P_o^{\theta}}(t) \right) \right)^2 dP_o^{\theta}(t) d\theta \right\|_2$$
$$= \left\| \int_{\mathbb{S}^{d-1}} \int_0^1 \left(F_{P_{u_i}^{\theta}}^{-1}(r) - F_{P_{u_j}^{\theta}}^{-1}(r) \right)^2 dr d\theta \right\|_2. \tag{10}$$

Let $F_{P_{u_i}^{\theta}}^{-1}(r) = x$, meaning that $r = F_{P_{u_i}^{\theta}}$. We have:

$$\|E_{u_i} - E_{u_j}\|_2 = \left\| \int_{\mathbb{S}^{d-1}} \int_0^1 \left(x - F_{P_{u_j}^{\theta}}^{-1}(F_{P_{u_i}^{\theta}}(x)) \right)^2 dP_{u_j}^{\theta}(x) d\theta \right\|_2$$
$$= \left\| \int_{\mathbb{S}^{d-1}} W_2^2(P_{u_j}^{\theta}, P_{u_i}^{\theta}) d\theta \right\|_2 = SW_2(P_{u_j}, P_{u_i}). \tag{11}$$

With symmetry, we have $SW_2(P_{u_i}, P_{u_j}) = SW_2(P_{u_j}, P_{u_i})$. Then for the reference P_o, its encoded representation is straightforward to have $E_o = 0$. Thus we complete the proof as follows:

$$\|E_{u_i}\|_2 = \|E_{u_i} - E_o\|_2 \approx SW_2(P_{u_i}, P_o), \tag{12}$$

∎

Complexity Analysis. WSFE is *training-free* that can be utilized on the fly right after the backbone model is trained. Thus, the complexity of WSFE is $O(S(L+1)Md \log d)$, where the cost $O(d \log d)$ is for implementing $\tau(\cdot)$. Fortunately, it is linear to the input data size, indicating that the encoding can be done at the input scale.

Figure 3: Wasserstein to Euclidean geodesics.

Table 1: The statistics of datasets.

	MovieLens	Gowalla	Pinterest	Yelp2018	Kindle	AMZ-Book
#Users	6,040	29,858	55,186	31,668	115,652	52,643
#Items	3,952	40,981	9,916	38,048	98,729	91,599
#Avg. Interactions	165.60	34.31	26.52	49.31	15.85	56.69
#All Interactions	1,000,209	1,027,370	1,463,556	1,561,406	1,833,068	2,984,108

3 Experimental Results

We evaluate WSFE with the aim of answering the following research questions: **RQ1.** How does WSFE boost the user segmentation performance of state-of-the-art recommender models? **RQ2.** How do different model settings affect WSFE performance?

3.1 Experimental Setups

Datasets. We collect six widely-evaluated public datasets (including their original training/test data splits) from: MovieLens [5, 6, 31, 53], Gowalla [7, 10], Pinterest [9, 33], Yelp [49], Kindle [17, 52], and Amazon-Book [1, 13]. Dataset statistics are reported in Table 1.

Evaluation Protocol. The fundamental property required by user segmentation is *user-wise similarity measurement*. Thus, given a query user, we treat this task as ranking towards candidates of similar users, based on the encoded user representations. In this work, we sort out similar users based on the number of overlapping items they have interacted with; then we compared these ranking lists with Top-K results inferred from the learning models. Recall@K and NDCG@K are the evaluation metrics.

Baselines. To demonstrate the effectiveness of WSFE, we incorporate it into the following state-of-the-art models.

(1) **LightGCN** [13] is one state-of-the-art GCN-based recommender model with a more concise and powerful structure.
(2) **SGL** [46] is one representative graph-based model with contrastive learning to tackle the data sparsity issue.
(3) **SimGCL** [51] is the state-of-the-art contrastive-learning-based recommender model that conducts the simplified augmentation directly in the feature space.
(4) **NCL** [24] is one of the state-of-the-art graph-based models with contrastive neighborhood information enrichment.
(5) **BUIR** [22] is one state-of-the-art model that bootstraps user and item representations for collaborative filtering.
(6) **DirectAU** [43] is the latest model that improves the representation quality from the perspective of alignment and uniformity.

3.2 Empirical Analyses and Discussions

3.2.1 *Overall Performance (RQ1).* From Table 2, we notice that

- After integrating WSFE, recent recommender models improve their segmentation capability across all datasets. Not only does this show our method's effectiveness, but more importantly, this also validates its generality and flexibility to the variety of graph-based models as well as different datasets.
- We notice that the model improvements on MovieLens dataset are larger than those on other datasets. One major explanation is that users of MovieLens have more average interactions, i.e., 165.60 as shown in Table 1, leading to more complicated user preference distributions whereas our WSFE can well utilize such rich information to encode the user-wise similarity.

Table 2: Experimental results before and after implementing WSFE into the underlying models (best view in color).

Dataset	Model	Recall@5	NDCG@5	Recall@20	NDCG@20	Recall@50	NDCG@50	Recall@100	NDCG@100
Movie	LightGCN	0.10→0.18 (+80.00%)	0.36→0.61 (+69.44%)	0.37→0.69 (+86.49%)	0.36→0.65 (+80.56%)	0.85→1.68 (+97.65%)	0.62→1.20 (+93.55%)	1.63→3.04 (+86.50%)	0.97→1.83 (+88.66%)
	SGL	0.08→0.13 (+62.50%)	0.32→0.47 (+46.88%)	0.24→0.37 (+54.17%)	0.25→0.39 (+56.00%)	0.42→0.73 (+73.81%)	0.36→0.59 (+63.89%)	0.60→1.22 (+103.33%)	0.47→0.81 (+72.34%)
	SimGCL	0.90→1.02 (+13.33%)	3.40→3.04 (+13.82%)	2.13→3.77 (+77.00%)	2.30→3.60 (+56.52%)	3.10→8.56 (+176.13%)	2.92→5.99 (+105.14%)	4.45→13.71 (+208.09%)	3.35→8.58 (+156.12%)
	NCL	0.25→0.38 (+52.00%)	0.84→1.30 (+54.76%)	0.88→1.47 (+67.05%)	0.84→1.38 (+64.29%)	2.01→3.42 (+70.15%)	1.47→2.47 (+68.03%)	3.60→6.19 (+71.94%)	2.20→3.74 (+70.00%)
	BUIR	0.22→0.24 (+9.09%)	0.74→0.82 (+10.82%)	0.83→0.96 (+15.66%)	0.79→0.92 (+16.46%)	1.81→2.19 (+20.99%)	1.34→1.60 (+19.40%)	3.21→3.95 (+23.05%)	1.98→2.40 (+21.21%)
	DirectAU	0.09→0.10 (+11.11%)	0.32→0.36 (+12.50%)	0.25→0.31 (+24.00%)	0.26→0.32 (+23.08%)	0.53→0.66 (+24.53%)	0.41→0.51 (+24.39%)	0.98→1.23 (+25.51%)	0.62→0.77 (+24.19%)
Gowalla	LightGCN	4.41→4.56 (+3.40%)	6.59→6.52 (-1.06%)	8.75→9.26 (+5.83%)	6.64→6.80 (+2.41%)	13.30→14.05 (+5.64%)	8.46→8.71 (+2.96%)	17.78→18.86 (+6.07%)	9.94→10.29 (+3.52%)
	SGL	4.96→5.30 (+6.85%)	7.58→7.62 (+0.53%)	9.85→10.66 (+8.22%)	7.53→7.86 (+4.38%)	15.13→16.68 (+10.24%)	9.66→10.23 (+5.90%)	20.70→22.72 (+9.76%)	11.49→12.19 (+6.09%)
	SimGCL	5.25→6.94 (+32.19%)	8.55→10.32 (+20.70%)	10.14→13.99 (+37.97%)	8.11→10.52 (+29.72%)	14.73→20.23 (+37.34%)	9.99→13.02 (+30.33%)	17.88→24.15 (+35.07%)	11.12→14.34 (+28.96%)
	NCL	4.65→5.01 (+7.53%)	8.31→8.71 (+4.81%)	9.41→10.20 (+8.40%)	7.81→8.34 (+6.79%)	13.78→15.11 (+9.65%)	9.71→10.43 (+7.42%)	18.16→20.02 (+10.24%)	11.22→12.11 (+7.93%)
	BUIR	2.94→3.07 (+4.42%)	5.64→5.74 (+1.77%)	6.65→6.92 (+4.06%)	5.52→5.66 (+2.54%)	10.74→11.00 (+2.42%)	7.30→7.42 (+1.64%)	15.09→15.36 (+1.79%)	8.78→8.91 (+1.48%)
	DirectAU	5.03→5.41 (+7.55%)	7.99→8.22 (+2.88%)	10.24→10.98 (+7.23%)	7.90→8.31 (+5.19%)	15.71→16.94 (+7.83%)	10.16→10.74 (+5.71%)	21.20→22.86 (+7.83%)	12.00→12.73 (+6.08%)
Pinterest	LightGCN	2.24→2.38 (+6.25%)	4.64→4.94 (+6.47%)	7.24→7.68 (+6.08%)	5.51→5.87 (+6.53%)	13.67→14.67 (+7.32%)	8.37→8.94 (+6.81%)	21.11→22.65 (+7.30%)	11.04→11.81 (+6.97%)
	SGL	3.93→3.92 (-0.25%)	7.58→7.58 (0%)	11.21→11.35 (+1.25%)	8.57→8.63 (+0.70%)	19.62→19.86 (+1.22%)	12.23→12.35 (+0.98%)	28.16→28.44 (+0.99%)	15.31→15.44 (+0.85%)
	SimGCL	5.04→8.49 (+68.45%)	9.28→14.86 (+60.13%)	13.56→21.31 (+57.15%)	10.33→16.20 (+56.82%)	22.70→32.82 (+44.58%)	14.31→21.21 (+48.22%)	31.68→41.79 (+31.91%)	17.54→24.37 (+38.94%)
	NCL	4.10→4.63 (+12.93%)	8.21→9.11 (+10.84%)	11.48→13.02 (+13.41%)	8.98→10.08 (+12.25%)	19.55→22.32 (+14.17%)	12.59→14.18 (+12.63%)	27.85→31.76 (+14.04%)	15.60→17.58 (+12.69%)
	BUIR	1.16→1.22 (+5.17%)	2.55→2.64 (+3.53%)	4.09→4.31 (+5.38%)	3.16→3.28 (+3.80%)	8.29→8.77 (+5.79%)	5.05→5.26 (+4.16%)	13.34→14.16 (+6.15%)	6.92→7.23 (+4.48%)
	DirectAU	8.03→9.12 (+13.57%)	13.34→14.84 (+11.24%)	21.32→24.11 (+13.09%)	15.40→17.26 (+12.08%)	34.53→38.47 (+11.47%)	20.93→23.30 (+11.32%)	46.73→51.57 (+10.36%)	25.21→27.88 (+10.59%)
Yelp	LightGCN	2.13→2.37 (+11.27%)	1.31→1.64 (+25.19%)	2.16→2.72 (+25.93%)	1.94→2.39 (+23.20%)	4.07→5.23 (+28.50%)	2.88→3.62 (+25.69%)	6.47→8.37 (+29.37%)	3.87→4.88 (+26.10%)
	SGL	0.92→0.97 (+5.43%)	2.43→2.50 (+2.88%)	2.28→2.57 (+12.72%)	2.14→2.33 (+8.88%)	4.06→4.67 (+15.02%)	3.02→3.36 (+11.26%)	6.21→7.18 (+15.62%)	3.91→4.38 (+12.02%)
	SimGCL	1.01→2.00 (+98.02%)	2.72→5.24 (+92.65%)	2.30→5.31 (+130.87%)	2.22→4.83 (+117.57%)	3.67→8.68 (+136.51%)	2.92→6.52 (+123.29%)	4.96→11.71 (+136.09%)	3.46→7.78 (+124.86%)
	NCL	1.16→1.41 (+21.55%)	3.43→4.06 (+18.37%)	2.91→3.65 (+25.43%)	2.87→3.52 (+22.65%)	4.76→6.13 (+28.78%)	3.83→4.79 (+25.07%)	6.74→8.88 (+31.75%)	4.67→5.91 (+26.55%)
	BUIR	0.53→0.53 (+0%)	1.61→1.62 (+0.62%)	1.55→1.58 (+1.94%)	1.49→1.51 (+1.34%)	2.76→2.84 (+2.90%)	2.13→2.17 (+1.88%)	4.19→4.33 (+3.34%)	2.74→2.79 (+1.82%)
	DirectAU	1.39→1.66 (+19.42%)	3.54→4.12 (+16.38%)	3.36→4.15 (+23.51%)	3.11→3.74 (+20.26%)	5.66→7.05 (+24.56%)	4.24→5.15 (+21.46%)	8.28→10.27 (+24.03%)	5.31→6.46 (+21.66%)
Kindle	LightGCN	7.21→7.26 (+0.69%)	8.81→8.56 (-2.84%)	14.88→15.06 (+1.21%)	10.26→10.21 (-0.49%)	19.76→20.23 (+2.38%)	12.18→12.19 (+0.08%)	23.20→24.01 (+3.49%)	13.28→13.37 (+0.68%)
	SGL	7.90→7.94 (+0.51%)	9.63→9.33 (-3.12%)	16.93→17.58 (+3.84%)	11.53→11.61 (+0.69%)	23.32→24.37 (+4.50%)	13.99→14.19 (+1.43%)	27.87→29.43 (+5.60%)	15.41→15.74 (+2.14%)
	SimGCL	8.54→8.80 (+3.04%)	11.55→11.25 (-2.60%)	17.27→18.57 (+7.53%)	12.62→12.98 (+2.85%)	22.79→24.71 (+8.42%)	14.82→15.38 (+3.78%)	25.95→28.20 (+8.67%)	16.87→17.50 (+3.73%)
	NCL	9.26→9.66 (+4.32%)	12.55→12.85 (+2.39%)	18.25→19.42 (+6.41%)	13.50→14.10 (+4.44%)	23.89→25.43 (+6.45%)	15.83→16.58 (+4.74%)	27.62→29.60 (+7.17%)	17.08→17.96 (+5.15%)
	BUIR	7.30→7.43 (+1.78%)	10.01→10.01 (0%)	15.50→15.67 (+1.10%)	11.28→11.34 (+0.53%)	20.64→20.98 (+1.65%)	13.46→13.58 (+0.89%)	24.20→24.72 (+2.15%)	14.68→14.85 (+1.16%)
	DirectAU	8.00→8.26 (+3.25%)	10.24→10.23 (-0.10%)	17.69→18.66 (+5.48%)	12.23→12.60 (+3.03%)	24.89→26.57 (+6.75%)	15.07→15.67 (+3.98%)	30.18→32.61 (+8.05%)	16.79→17.60 (+4.82%)
AMZ-Book	LightGCN	2.02→2.14 (+5.94%)	4.62→4.77 (+3.25%)	5.15→5.65 (+9.71%)	4.48→4.82 (+7.59%)	8.13→9.26 (+13.90%)	5.92→6.49 (+9.63%)	11.05→12.78 (+15.66%)	7.06→7.85 (+11.19%)
	SGL	2.47→2.59 (+4.86%)	5.51→5.62 (+2.00%)	6.06→6.64 (+9.57%)	5.30→5.65 (+6.60%)	9.46→10.66 (+12.68%)	6.94→7.52 (+8.36%)	12.64→14.39 (+13.84%)	8.18→8.95 (+9.41%)
	SimGCL	2.72→3.09 (+13.60%)	6.40→7.02 (+9.69%)	6.13→7.14 (+16.48%)	5.64→6.42 (+13.83%)	8.69→10.35 (+19.10%)	6.19→7.99 (+29.08%)	10.55→12.76 (+20.95%)	7.68→8.95 (+16.54%)
	NCL	2.63→2.97 (+7.53%)	6.49→7.18 (+4.81%)	6.30→7.32 (+8.40%)	5.81→6.61 (+6.79%)	9.53→11.35 (+9.65%)	7.44→8.58 (+7.42%)	12.44→15.10 (+10.24%)	8.58→10.07 (+7.93%)
	BUIR	1.43→1.46 (+2.10%)	3.78→3.82 (+1.06%)	3.95→4.00 (+1.27%)	3.60→3.63 (+0.83%)	6.30→6.40 (+1.59%)	4.79→4.83 (+0.84%)	8.45→8.55 (+1.18%)	5.66→5.70 (+0.71%)
	DirectAU	2.88→3.13 (+8.68%)	6.55→7.00 (+6.87%)	6.88→7.63 (+10.90%)	6.12→6.68 (+9.15%)	10.48→11.80 (+12.60%)	7.89→8.70 (+10.27%)	13.68→15.54 (+13.60%)	9.16→10.17 (+11.03%)
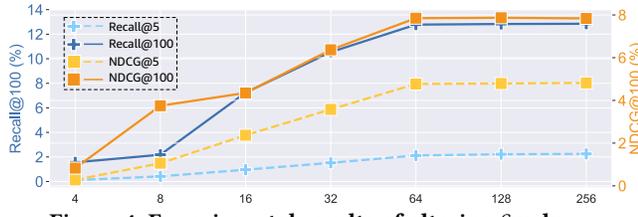
Figure 4: Experimental results of altering S values.

- Furthermore, equipped with WSFE, contrastive-learning-based models, e.g., SGL [46], SimGCL [51], and NCL [24], generally have larger model improvements. This is because augmentation techniques (either to original data or to the latent features) subsequently provide the embedding enrichment for WSFE to exert.

3.2.2 Effect of Slicing Number S (RQ2A).
Due to the renowned and stable performance of LightGCN [13], we utilize it as the backbone on AMZ-Book dataset to exemplify the model analysis. We alternatively change the value of S and plot the results in Figure 4. We notice that, altering S from 4 to 64 is more influential to the model performance, which is intuitive as this produces a more accurate and fine-grained cumulative approximation. However, on the other hand, consistently increasing S will also put more computation and memory strains. Thus, setting S as 64 is the balanced spot with positive momentum that presents a practical trade-off between model performance and resource consumption.

3.2.3 Dimension Reduction (RQ2B).
During evaluation, we notice that some models encounter the "out-of-memory" problem. To address this issue, we approach to aggregate layer-wise embeddings in Eqn.(7) to reduce the total dimensionality from $S(L+1)$ to S.

Table 3: Aggregation Selection.

Aggregator	Concat		Sum		Max	
Metric	K=5	K=100	K=5	K=100	K=5	K=100
Recall@K	**2.14**	12.78	2.12	**12.81**	1.77	11.27
NDCG@K	4.77	**7.85**	**4.78**	7.77	4.31	7.01

From Table 3, we notice that *Sum* surprisingly presents a competitive performance with the original *Concat* operation. This indicates that, while *Concat* has a more complete representation encoding with theoretical supports, *Sum* is suitable for dimension reduction in scenarios with limited computational resources.

4 Conclusion and Future Extension

In this work, we propose WSFE to encode representations for effective user segmentation in collaborative filtering. The extensive experiments demonstrate the effectiveness of our proposed method and its generality to a variety of model deployments. As for future work, we point out three major directions as follows:

(1) In light of the empirical findings in § 3.2.1, it is interesting to explore *contrastive learning* techniques [28, 30, 54, 56] in the sub-graph feature domain for further model improvement.
(2) It is worth investigating adapting our training-free model to other scenarios of information retrieval and autonomous database management [14–16, 26, 27, 34, 38].
(3) We plan to design unsupervised regularization mechanisms such that WSFE and the backbone model can be *jointly optimized* or even *mutually enhanced* for multi-task learning.

Acknowledgments

References

[1] Amazon. 2015. Dataset. https://github.com/gusye1234/LightGCN-PyTorch/tree/master/data/amazon-book.

[2] Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. 2015. Sliced and radon wasserstein barycenters of measures. *JMIV* 51, 1 (2015), 22–45.

[3] Yankai Chen, Yixiang Fang, Yifei Zhang, and Irwin King. 2023. Bipartite Graph Convolutional Hashing for Effective and Efficient Top-N Search in Hamming Space. *WebConf.*

[4] Yankai Chen, Huifeng Guo, Yingxue Zhang, Chen Ma, Ruiming Tang, Jingjie Li, and Irwin King. 2022. Learning binarized graph representations with multi-faceted quantization reinforcement for top-k recommendation. In *SIGKDD*.

[5] Yankai Chen, Menglin Yang, Yingxue Zhang, Mengchen Zhao, Ziqiao Meng, Jianye Hao, and Irwin King. 2022. Modeling scale-free graphs with hyperbolic geometry for knowledge-aware recommendation. In *WSDM*. 94–102.

[6] Yankai Chen, Yaming Yang, Yujing Wang, Jing Bai, Xiangchen Song, and Irwin King. 2022. Attentive knowledge-aware graph convolutional networks with collaborative guidance for personalized recommendation. In *ICDE*.

[7] Eunjoon Cho, Seth A Myers, and Jure Leskovec. 2011. Friendship and mobility: user movement in location-based social networks. In *SIGKDD*. 1082–1090.

[8] Ishan Deshpande, Yuan-Ting Hu, Ruoyu Sun, Ayis Pyrros, Nasir Siddiqui, Sanmi Koyejo, Zhizhen Zhao, David Forsyth, and Alexander G Schwing. 2019. Max-sliced wasserstein distance and its use for gans. In *CVPR*. 10648–10656.

[9] Xue Geng, Hanwang Zhang, Jingwen Bian, and Tat-Seng Chua. 2015. Learning image and user features for recommendation in social networks. In *ICCV*. 4274–4282.

[10] Gowalla. 2011. Dataset. https://github.com/gusye1234/LightGCN-PyTorch/tree/master/data/gowalla.

[11] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NeurIPS*. 1025–1035.

[12] Bowei He, Xu He, Yingxue Zhang, Ruiming Tang, and Chen Ma. 2023. Dynamically Expandable Graph Convolution for Streaming Recommendation. *WebConf.*

[13] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*. 639–648.

[14] Xuming Hu, Fukun Ma, Chenyao Liu, Chenwei Zhang, Lijie Wen, and Philip S Yu. 2021. Semi-supervised Relation Extraction via Incremental Meta Self-Training. In *EMNLP: Findings*.

[15] Xuming Hu, Lijie Wen, Yusong Xu, Chenwei Zhang, and Philip S. Yu. 2020. Self-ORE: Self-supervised Relational Feature Learning for Open Relation Extraction. In *EMNLP*. 3673–3682.

[16] Xuming Hu, Chenwei Zhang, Yawen Yang, Xiaohe Li, Li Lin, Lijie Wen, and Philip S. Yu. 2021. Gradient Imitation Reinforcement Learning for Low Resource Relation Extraction. In *EMNLP*. 2737–2746.

[17] Amazon Kindle. 2015. Dataset. https://github.com/Coder-Yu/SELFRec/tree/main/dataset/amazon-kindle.

[18] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. *ICLR*.

[19] Soheil Kolouri, Navid Naderializadeh, Gustavo K Rohde, and Heiko Hoffmann. 2020. Wasserstein embedding for graph learning. *ICLR*.

[20] Soheil Kolouri, Kimia Nadjahi, Umut Simsekli, Roland Badeau, and Gustavo Rohde. 2019. Generalized sliced wasserstein distances. *NeurIPS* 32.

[21] Soheil Kolouri, Yang Zou, and Gustavo K Rohde. 2016. Sliced Wasserstein kernels for probability distributions. In *CVPR*. 5258–5267.

[22] Dongha Lee, SeongKu Kang, Hyunjun Ju, Chanyoung Park, and Hwanjo Yu. 2021. Bootstrapping user and item representations for one-class collaborative filtering. In *SIGIR*. 317–326.

[23] Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. 2019. Deepgcns: Can gcns go as deep as cnns?. In *CVPR*. 9267–9276.

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

[25] Guang Ling, Haiqin Yang, Irwin King, and Michael R Lyu. 2012. Online learning for collaborative filtering. In *IJCNN*. IEEE, 1–8.

[26] Aiwei Liu, Xuming Hu, Li Lin, and Lijie Wen. 2022. Semantic Enhanced Text-to-SQL Parsing via Iteratively Learning Schema Linking Graph. In *SIGKDD*. 1021–1030.

[27] Aiwei Liu, Xuming Hu, Lijie Wen, and Philip S Yu. 2023. A comprehensive evaluation of ChatGPT's zero-shot Text-to-SQL capability. *arXiv preprint arXiv:2303.13547* (2023).

[28] Shuliang Liu, Xuming Hu, Chenwei Zhang, Shu'ang Li, Lijie Wen, and Philip S. Yu. 2022. HiURE: Hierarchical Exemplar Contrastive Learning for Unsupervised Relation Extraction. In *NAACL-HLT*. 5970–5980.

[29] Fangyuan Luo, Jun Wu, and Tao Wang. 2022. Discrete Listwise Personalized Ranking for Fast Top-N Recommendation with Implicit Feedback. (2022).

[30] Yueen Ma, Zixing Song, Xuming Hu, Jingjing Li, Yifei Zhang, and Irwin King. 2023. Graph Component Contrastive Learning for Concept Relatedness Estimation. In *AAAI*.

[31] MovieLens. 2003. Dataset. https://grouplens.org/datasets/movielens/1m/.

[32] Navid Naderializadeh, Joseph F Comer, Reed Andrews, Heiko Hoffmann, and Soheil Kolouri. 2021. Pooling by sliced-Wasserstein embedding. *NeurIPS* 34, 3389–3400.

[33] Pinterest. 2015. Dataset. https://sites.google.com/site/xueatalphabeta/dataset-1/pinterest_iccv.

[34] Zexuan Qiu, Qinliang Su, Jianxing Yu, and Shijing Si. 2022. Efficient Document Retrieval by End-to-End Refining and Quantizing BERT Embedding with Contrastive Product Quantization. *EMNLP* (2022).

[35] Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. 2011. Wasserstein barycenter and its application to texture mixing. In *SSVM*. Springer, 435–446.

[36] Xin Shen, Kyungdon Joo, and Jean Oh. 2023. FishRecGAN: An End to End GAN Based Network for Fisheye Rectification and Calibration. *arXiv* (2023).

[37] Xin Shen, Jiaying Shi, Sungro Yoon, Jon Katzur, Hanbo Wang, Jim Chan, and Jin Li. 2023. Learning to Personalize Recommendations based on Customers' Shopping Intents. *arXiv* (2023).

[38] Xin Shen, Xiaonan Zhao, and Rui Luo. 2023. Semantic Embedded Deep Neural Network: A Generic Approach to Boost Multi-Label Image Classification Performance. *arXiv* (2023).

[39] Xin Shen, Yan Zhao, Sujan Perera, Yujia Liu, Jinyun Yan, and Mitchell Goodman. 2023. Learning Personalized Page Content Ranking Using Customer Representation. *arXiv* (2023).

[40] Zixing Song, Ziqiao Meng, Yifei Zhang, and Irwin King. 2021. Semi-supervised Multi-label Learning for Graph-structured Data. In *CIKM*.

[41] Zixing Song, Yifei Zhang, and Irwin King. 2022. Towards an Optimal Asymmetric Graph Structure for Robust Semi-supervised Node Classification. In *SIGKDD*. 1656–1665.

[42] Cédric Villani. 2009. Optimal transport: old and new, Vol. 338. Springer.

[43] Chenyang Wang, Yuanqing Yu, Weizhi Ma, Min Zhang, Chong Chen, Yiqun Liu, and Shaoping Ma. 2022. Towards Representation Alignment and Uniformity in Collaborative Filtering. In *SIGKDD*. 1816–1825.

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

[45] Jun Wu, Fangyuan Luo, Yujia Zhang, and Haishuai Wang. 2020. Semi-discrete matrix factorization. *IEEE Intelligent Systems* 35, 5 (2020), 73–83.

[46] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *SIGIR*. 726–735.

[47] Menglin Yang, Zhihao Li, Min Zhou, Jiahong Liu, and Irwin King. 2022. HICF: Hyperbolic informative collaborative filtering. In *SIGKDD*. 2212–2221.

[48] Menglin Yang, Min Zhou, Jiahong Liu, Defu Lian, and Irwin King. 2022. HRCF: Enhancing collaborative filtering via hyperbolic geometric regularization. In *WebConf*. 2462–2471.

[49] Yelp2018. 2018. Dataset. https://github.com/gusye1234/LightGCN-PyTorch/tree/master/data/yelp2018.

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

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

[52] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Jundong Li, and Zi Huang. 2022. Self-Supervised Learning for Recommender Systems: A Survey. *arXiv* (2022).

[53] Xinni Zhang, Yankai Chen, Cuiyun Gao, Qing Liao, Shenglin Zhao, and Irwin King. 2022. Knowledge-aware Neural Networks with Personalized Feature Referencing for Cold-start Recommendation. *arXiv* (2022).

[54] Yifei Zhang, Yankai Chen, Zixing Song, and Irwin King. 2023. Contrastive Cross-scale Graph Knowledge Synergy. *arXiv* (2023).

[55] Yifei Zhang and Hao Zhu. 2020. Discrete wasserstein autoencoders for document retrieval. In *ICASSP*. IEEE, 8159–8163.

[56] Yifei Zhang, Hao Zhu, Zixing Song, Piotr Koniusz, and Irwin King. 2022. COSTA: Covariance-Preserving Feature Augmentation for Graph Contrastive Learning. In *SIGKDD*. 2524–2534.