Curriculum Co-disentangled Representation Learning across Multiple Environments for Social Recommendation

Xin Wang [1] **Zirui Pan** [1] **Yuwei Zhou** [1] **Hong Chen** [1] **Chendi Ge** [1] **Wenwu Zhu** [1]

Abstract

There exist complex patterns behind the decision-making processes of different individuals across different environments. For instance, in a social recommender system, various user behaviors are driven by highly entangled latent factors from two environments, i.e., consuming environment where users consume items and social environment where users connect with each other. Uncovering the disentanglement of these latent factors for users can benefit in enhanced explainability and controllability for recommendation. However, in literature there has been no work on social recommendation capable of disentangling user representations across consuming and social environments. To solve this problem, we study co-disentangled representation learning across different environments via proposing the curriculum co-disentangled representation learning (CurCoDis) model to disentangle the hidden factors for users across both consuming and social environments. To co-disentangle joint representations for user-item consumption and user-user social graph simultaneously, we partition the social graph into equal-size sub-graphs with minimum number of edges being cut, and design a curriculum weighing strategy for subgraph training through measuring the complexity of subgraphs via Descartes' rule of signs. We further develop the prototype-routing optimization mechanism, which achieves co-disentanglement of user representations across consuming and social environments. Extensive experiments for social recommendation demonstrate that our proposed CurCoDis model can significantly outperform state-of-the-art methods on several real-world datasets.

[1]Department of Computer Science and Technology, BNRist, Tsinghua University. Correspondence to: Xin Wang <xin_wang@tsinghua.edu.cn>, Wenwu Zhu <wwzhu@tsinghua.edu.cn>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

1. Introduction

Human behaviors may demonstrate complex and diverse patterns in different environments. Taking social recommender systems as an example (Fan et al., 2019; Wang et al., 2019; Wu et al., 2019), there exist a consuming environment where users consume items and a social environment where user form social connections with each other. The decision-making processes of each individual across these environments follow complex patterns, driven by highly entangled hidden factors that govern the formations of consuming interactions with items, social connections among users, and their mutual influences. Disentangling and uncovering these entangled latent factors for users when learning representations for social recommendation can bring more explainability and controllability in the representations, thereby boosting the model performance.

However, learning disentangled representation across consuming and social environment simultaneously for social recommendation remains largely unexplored in literature. On the one hand, existing social recommendation approaches learn representations in various manners without disentangling the latent factors across different environments. As a result, these works learn representations in an entangled way, failing to discover the latent explanatory factors hidden in the observed data. On the other hand, existing works on disentangled representation learning for recommendation do not consider the influence from social environment, ignoring the complex relations between consuming interactions and social connections of different individuals. This being the case, the existing literature fails to uncover the mixed explanatory latent factors across consuming environment and social environment.

In this work, we study the problem of co-disentangled representation learning across multiple environments, particularly for social recommendation with consuming environment and social environment. Nevertheless, learning disentangled representations for users across consuming and social environments is fundamentally different from existing settings within a single consuming environment, and thus poses two challenges. First, in addition to the user-item consumption which captures user consuming interactions with items, social recommendation normally employs a massive user-

user graph to model the social connections among users, requiring us to learn joint disentangled representations for users across both the consuming environment and social environment efficiently. Second, people may demonstrate complex and diverse behavioral patterns in consuming and social environments, which makes it challenging to learn adequate disentangled representations capable of capturing consuming interactions as well as social connections simultaneously.

To tackle the challenges, in this paper we propose the Curriculum Co-Disentangled representation learning (CurCoDis) model to disentangle and uncover the hidden explanatory factors for users across consuming and social environments. To solve the first challenge, we present the curriculum subgraph training strategy which helps to co-disentangle the joint representations for user-item consumption and user-user social graph. In concrete, we first partition the social graph into several equal-size sub-graphs with minimum number of edges being cut by resorting to the Kernighan-Lin algorithm. To further boost the representation learning procedure, we then design a curriculum subgraph weighing algorithm based on measuring the complexity of graphs through Descartes' rule of signs, such that the subgraphs can be better utilized from a dynamic easy-to-hard order. To solve the second challenge, we develop a prototype-routing optimization mechanism which achieves co-disentanglement by jointly optimizing the prototype learning process in consuming environment and social dynamic routing process in social environment. In particular, the prototype-routing optimization mechanism identifies explanatory latent factors reflecting user preferences in the consuming environment through a prototype-based concept assigning process with information-theoretic regularization, which initializes the iterative routing process of appropriate neighbor selection in the social environment whose results will in turn be utilized to reconstruct the user-item interactions in the consuming environment.

We theoretically analyze the convergence properties of the prototype-routing optimization mechanism and prove its connection with probabilistic inference under a Gaussian Mixture initialization. Extensive experiments on various real-world datasets demonstrate that our proposed CurCoDis model is able to achieve significant performance gains, up to 18.7%, against state-of-the-art approaches.

The main contributions are summarized as follows:

- We study the problem of co-disentangled representation learning with application to social recommendation to uncover the hidden explanatory factors across consuming and social environments.
- We propose the curriculum subgraph training strategy and prototype-routing optimization mechanism to achieve the co-disentanglement of user representations in an end-to-end manner.
- We theoretically analyze the convergence properties of social dynamic routing optimization mechanism and experimentally show the advantages of co-disentangled representation learning across different environments.

2. Related Work

Disentangled Representation Learning Disentangled representation learning (Wang et al., 2022a), which aims to produce robust, controllable, and explainable representations, has become one of the core problems in machine learning. Variational methods are widely applied for disentangled representation over images and texts (Kingma & Welling, 2013; Higgins et al., 2017; Kim & Mnih, 2018; He et al., 2017; Jain et al., 2018), followed by further improvement through weakly supervised models (Locatello et al., 2019; Kingma et al., 2014; Feng et al., 2018), as well as the recent combination with the diffusion model (Chen et al., 2023). Moreover, with the popularity of graph neural networks (GNN), (Ma et al., 2019a; Li et al., 2021; 2022) apply the idea of disentanglement in training graph convolutional networks. They later learn disentangled representations for users in recommendation (Ma et al., 2019b; 2020; Wang et al., 2022b; Zhang et al., 2023) and handle both textual and visual data for multimodal recommendation (Wang et al., 2021a), which however are only able to handle data from the consuming environment.

Curriculum Learning Curriculum learning (CL) (Bengio et al., 2009; Wang et al., 2021b) is a strategy of training from ease, imitating the procedure of human learning with curricula. The simplest algorithm is named Baby Step (Spitkovsky et al., 2010), which determines the difficulty and input order of data. Later the Self-Paced method (Kumar et al., 2010) is proposed to select data samples automatically according to the training loss. Besides, there are Transfer Teacher (Hacohen & Weinshall, 2019; Weinshall et al., 2018), Reinforcement Learning Teacher (Graves et al., 2017; Zhao et al., 2020), and other automatic CL frameworks based on the specific data, model and task (Castells et al., 2020; Sinha et al., 2020), as well as the combination with disentangled recommendation (Chen et al., 2021), combinational optimization (Zhang et al., 2022), neural architecture (Zhou et al., 2022) and video grounding (Lan et al., 2023). The key parts of CL are a difficulty measurer to judge the difficulty of data samples and a training scheduler to decide the input sequence or weights of data subsets.

Social Recommendation In addition to consuming environment, social recommendation assumes that users are additionally connected within a social environment, resulting in their preferences being determined jointly across consuming and social environments. This motivates research works on

social recommendation (Ma et al., 2009; Jamali & Ester, 2010; Ma et al., 2011; Yang et al., 2011; Ye et al., 2012; Yang et al., 2016; Purushotham et al., 2012; Qian et al., 2014; Zhao et al., 2014; 2016; Wang et al., 2016; 2017a;b; Wu et al., 2018; Gonzalez Camacho & Alves-Souza, 2018; Cui et al., 2018; Chen et al., 2018; Zhang et al., 2018). However, existing literature ignores disentangling the latent factors across consuming and social environments.

3. Curriculum Co-disentangled Representation Learning

We first give a brief introduction on the problem definition, followed by details on the two core components of CurCoDis: i) curriculum subgraph training strategy and ii) prototype-routing optimization mechanism.

3.1. Problem Definition

Given a user behavior dataset $\mathcal{D}$ across consuming and social environments, which consists of the consuming interactions between N users and M items, as well as social connections among these N users. The consuming interaction between user u and item i is denoted by $x_{u,i} \in \{0, 1\}$, where $x_{u,i} = 1$ indicates that user u consumes item i, whereas $x_{u,i} = 0$ means u has not consumed i yet. We denote $\mathbf{x}_u = \{x_{u,i} : x_{u,i} = 1\}$ as the set of items consumed by user u. The social connections between user u and v can be modeled with a graph structure $G = (V, E)$ which contains a set of nodes V and a set of edges E. $(u, v) \in G$, or $(u, v) \in E$ indicates the existence of an edge between node (i.e., user) u and v, where user $u \in V$ is associated with a feature $\mathbf{z}_u$. We denote Θ as the set of trainable parameters for the proposed model. Our goal is to learn representations $\{\mathbf{z}_u\}_{u=1}^N$ for the N users, such that $\{\mathbf{z}_u\}_{u=1}^N$ can achieve co-disentanglement across both consuming and social environments.

3.2. Curriculum Subgraph Training

During the learning process of $\mathbf{z}_u$, calculating social propagations from G is computationally expensive for large social graphs. This requires us to discover a solution capable of handling consuming interactions and social connections in a memory-friendly manner. We begin with the most natural way, i.e., partitioning the social graph into sub-graphs.

Subgraph Partition Partition through randomly sampling nodes from the whole graph to form several disjoint subgraphs seems to be an adequate solution. However, this method may induce a large number of edges that connect different subgraphs to be cut and removed during the partitioning process, losing necessary information from the social environment. Therefore, we assign dynamic weights to the edges and adopt the Kernighan–Lin algorithm (Kernighan & Lin, 1970) to equally and stochastically partition the edge set E of the social graph G, such that every subgraph will be of equal size with minimum sum of weights of edges being cut and removed. Specifically, the weights of edges which has been cut will be gradually increased, making it more favorable to be chosen in future training. So ideally every edge in the social graph will be sufficiently learned.

For each user u in the subgraphs, if the number of her connections, $|\mathcal{N}_u|$ is larger than the preset threshold T, then we randomly select T out of the $|\mathcal{N}_u|$ connections for consideration when calculating her representation $\mathbf{z}_u$. During training, we fit each batch with one subgraph so that every social connection carried in the subgraphs will be explored in each training epoch.

Curriculum Weighing for Subgraph Training The partition of social graph naturally raises a new question, i.e., what is the importance of different subgraphs for the model to achieve the best performance? We propose the curriculum subgraph training strategy, a solution based on curriculum learning (Bengio et al., 2009; Wang et al., 2021b), through measuring the complexities of subgraphs given the intuition that graph complexity may strongly correlate with the difficulty of graph analysis. The assumption from curriculum learning is that easier subgraphs may be more important for the model during early training stage, and those more difficult subgraphs will gradually become important when the model gets well-trained on the easy subgraphs.

Although there have been quite a few approaches to measure the complexities of graphs (Rabinovich & Forschungsgebiet, 2008), most of them involve complicated computations such as tree-width. Our proposed measurement employs an effective approach which measures the difficulty of a graph through utilizing the degrees of each node to form polynomials. We first define:

$$P_G = a_k x^k + \cdots + a_2 x^2 + a_1 x + a_0, \quad P_G^* = \alpha - P_G, \quad (1)$$

where $k = \max \deg(u)$ denotes the maximum node degree, a_k represents the number of nodes with degree k, and α is the parameter used to adjust the zeros of the polynomials. More specific, according to Descartes's Rule of Signs, P_G^* has a unique, positive zero δ if α satisfies the following conditions:

$$P_G^*(0) = \alpha - a_0 > 0, \quad P_G^*(1) = \alpha - \sum_{i=0}^{k} a_i < 0, \quad (2)$$

which constrains $\delta \in (0, 1)$. Given the validated relevance between δ and the edge density $\frac{|E|}{|V|^2 - |V|}$ of G (Dehmer et al., 2019), there exists a positive correlation between δ and the true complexity of G. Based on this conclusion, we define the difficulty of graph G as $\mathcal{D}(G) = \delta$, i.e., larger δ indicates higher level of complexity in G, thus increasing its difficulty.

With the partitioned subgraphs and their difficulties, the proposed curriculum subgraph training strategy is capable of discovering the dynamic importance of different subgraphs in different training epochs, which enables the model

Algorithm 1 Curriculum Subgraph Weighing

Input: $CurriculumEpoch,\ \lambda$;
Subgraphs $\mathcal{SG} = \{SG_1, SG_2, \cdots, SG_g\}$;
Subgraph difficulties $\Delta = \{\delta_{SG_1}, \delta_{SG_2}, \cdots, \delta_{SG_g}\}$;
Output: Subgraph weights $\mathcal{W}$;

function CURRICULUMSCHEDULER($epoch$)
  return $\min\left\{1, \lambda + \frac{(1-\lambda)*epoch}{CurriculumEpoch}\right\}$
$\mathcal{W} \leftarrow \{w_{SG_1}, w_{SG_2}, \cdots, w_{SG_g}\}$
  ▷ w_{SG_j} is the importance weight for subgraph SG_j.
$\Lambda \leftarrow$ CURRICULUMSCHEDULER($epoch$)
for $SG_j \in \mathcal{SG}$ **do**
  if $\delta_{SG_j} \leq \Lambda$ **then**
    $w_{SG_j} \leftarrow 1$
  else
    $w_{SG_j} \leftarrow 1 - \delta_{SG_j}$

to focus more on easier subgraphs during early training stages and then gradually learn from those more difficult subgraphs. Algorithm 1 shows the details of our curriculum subgraph weighing algorithm. During each of the first $CurriculumEpoch$ epochs, we dynamically calculate the importance weight w_{SG_j} of each subgraph SG_j based on its difficulty δ_{SG_j}, and gradually increase the importance weight for more difficult subgraphs as the number of epochs increase. When the number of epochs exceeds $CurriculumEpoch$, all the subgraphs will have equal importance weights.

3.3. Co-disentangled Representation Learning

In this section, we in detail discuss the prototype-routing optimization mechanism capable of learning co-disentangled representations. Figure 1 shows the overall framework.

3.3.1. PROTOTYPE LEARNING AND ENCODING IN CONSUMING ENVIRONMENT

In consuming environment, we have user consuming interactions $\mathbf{x}_u$ for each user u belonging to a subgraph SG_j after the subgraph partition and curriculum subgraph weighing. We achieve disentanglement through learning a factorized representation of user u, i.e., $\mathbf{z}_u = [\mathbf{z}_u^{(1)}; \mathbf{z}_u^{(2)}; \cdots; \mathbf{z}_u^{(k)}; \cdots; \mathbf{z}_u^{(K)}] \in \mathcal{R}^{d \cdot K}$, assuming that there are K prototypes indicating K different concepts. In Figure 1, Prototype with Features represents the feature center or anchor of all items belonging to this prototype. The k^{th} component $\mathbf{z}_u^{(k)} \in \mathcal{R}^d$ is expected to capture user preference over the k^{th} concept. We design one-hot prototype assignment $\mathbf{C} = \{\mathbf{c}_i\}_{i=1}^M$ for all the items, where $\mathbf{c}_i = [c_{i,1}; c_{i,2}; \cdots; c_{i,K}]$. If item i belongs to concept k, then $c_{i,k} = 1$ and $c_{i,k'} = 0$ for any $k' \neq k$. For example, Figure 1 illustrates that item 3, 4 and 8 belong to the *blue* prototype. We learn user representations $\{\mathbf{z}_u\}_{u=1}^N$ and prototypes $\mathbf{C}$ jointly in an unsupervised manner.

For a user u, we assume that her consuming interactions with items $\mathbf{x}_u$ can be generated from the following distribution:

$$
\begin{aligned}
p_\Theta(\mathbf{x}_u) &= \mathbb{E}_{p_\Theta(\mathbf{C})}\left[\int p_\Theta\left(\mathbf{x}_u \mid \mathbf{z}_u, \mathbf{C}\right) p_\Theta(\mathbf{z}_u)\, d\mathbf{z}_u\right], \\
p_\Theta\left(\mathbf{x}_u \mid \mathbf{z}_u, \mathbf{C}\right) &= \prod_{x_{u,i} \in \mathbf{x}_u} p_\Theta(x_{u,i} \mid \mathbf{z}_u, \mathbf{C}), \\
p_\Theta(x_{u,i} \mid \mathbf{z}_u, \mathbf{C}) &= Z_u^{-1} \cdot \sum_{k=1}^{K} c_{i,k} \cdot g_\Theta^{(i)}(\mathbf{z}_u^{(k)}), \\
Z_u &= \sum_{i=1}^{M}\sum_{k=1}^{K} c_{i,k} \cdot \mathcal{M}_\Theta^{(i)}(\mathbf{z}_u^{(k)}), \quad p_\Theta(\mathbf{z}_u) = p_\Theta(\mathbf{z}_u \mid \mathbf{C}),
\end{aligned} \tag{3}
$$

where $g_\Theta : \mathcal{R}^d \to \mathcal{R}^+$ is a shallow neural network that estimates how much a user with given preference is interested in item i, and $\mathcal{M}_\Theta^{(i)}(\mathbf{z}_u^{(k)}) : \mathcal{R}^d \to \mathcal{R}^1$ is a nonlinear mapping function predicting the preference of user u over item i in terms of concept k. To optimize Θ, we follow the VAE literature (Kingma & Welling, 2013; Rezende et al., 2014) and maximize a lower bound of $\sum_u \ln p_\Theta(\mathbf{x}_u)$ based on the following property.

Property 1. $\ln p_\Theta(\mathbf{x}_u)$ *is bounded as follows:*

$$
\begin{aligned}
\ln p_\Theta(\mathbf{x}_u) \geq & \mathbb{E}_{p_\Theta(\mathbf{C})}\Big[\mathbb{E}_{q_\Theta(\mathbf{z}_u \mid \mathbf{x}_u, \mathbf{C})}\big[\ln p_\Theta(\mathbf{x}_u \mid \mathbf{z}_u, \mathbf{C})\big] \\
& - D_{\mathrm{KL}}\big(q_\Theta(\mathbf{z}_u \mid \mathbf{x}_u, \mathbf{C}) \| p_\Theta(\mathbf{z}_u)\big)\Big].
\end{aligned} \tag{4}
$$

See the Appendix for the proof.

Property 1 introduces a variational distribution $q_\Theta(\mathbf{z}_u \mid \mathbf{x}_u, \mathbf{C})$, as well as two expectations, $\mathbb{E}_{p_\Theta(\mathbf{C})}[\cdot]$ and $\mathbb{E}_{q_\Theta(\mathbf{z}_u \mid \mathbf{x}_u, \mathbf{C})}[\cdot]$, which are intractable. Therefore, Gumbel-Softmax (Jang et al., 2017; Maddison et al., 2017) and Gaussian re-parameterization (Kingma & Welling, 2013) are employed during the training process, upon which $q_\Theta(\mathbf{z}_u \mid \mathbf{x}_u, \mathbf{C})$ will be an approximation of the intractable posterior distribution $p_\Theta(\mathbf{z}_u \mid \mathbf{x}_u, \mathbf{C})$.

We further strengthen the disentanglement in $\mathbf{z}_u$ through promoting statistical independence among its dimensions,

$$
\begin{aligned}
q_\Theta(\mathbf{z}_u^{(k)} \mid \mathbf{C}) &\approx \prod_{j=1}^{d} q_\Theta(z_{u,j}^{(k)} \mid \mathbf{C}), \\
q_\Theta(\mathbf{z}_u \mid \mathbf{C}) &= \int q_\Theta(\mathbf{z}_u \mid \mathbf{x}_u, \mathbf{C}) p_{\mathrm{data}}(\mathbf{x}_u)\, d\mathbf{x}_u.
\end{aligned} \tag{5}
$$

Property 2 shows that the Kullback–Leibler (KL) divergence term in Property 1 can encourage the desired independence.

Property 2. *A reformulation of KL term in Eq. (4):*

$$
\begin{aligned}
&\mathbb{E}_{p_{\mathrm{data}}(\mathbf{x}_u)}\left[D_{\mathrm{KL}}(q_\Theta(\mathbf{z}_u \mid \mathbf{x}_u, \mathbf{C}) \| p_\Theta(\mathbf{z}_u))\right] \\
=& I_q(\mathbf{x}_u; \mathbf{z}_u) + D_{\mathrm{KL}}(q_\Theta(\mathbf{z}_u \mid \mathbf{C}) \| p_\Theta(\mathbf{z}_u)).
\end{aligned} \tag{6}
$$

See the Appendix for the proof.

On the one hand, requiring $I_q(\mathbf{x}_u; \mathbf{z}_u)$ in Eq. (6), the mutual information between $\mathbf{x}_u$ and $\mathbf{z}_u$ under $q_\Theta(\mathbf{z}_u \mid \mathbf{x}_u, \mathbf{C}) \cdot$

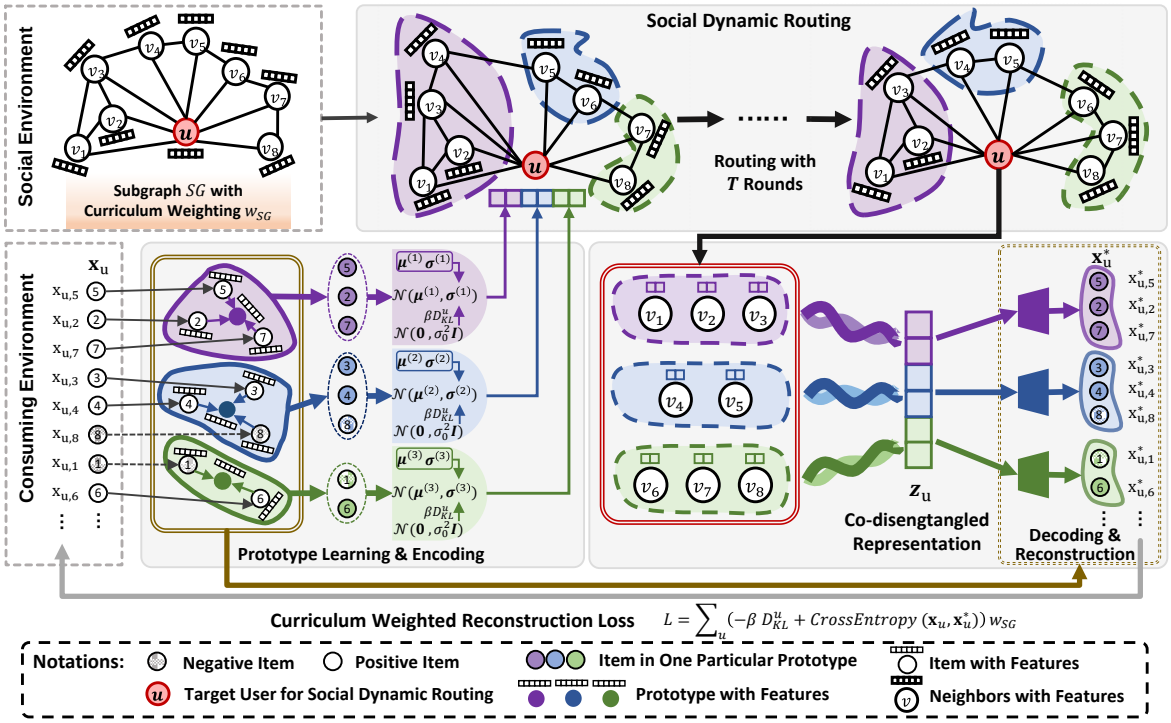


Figure 1. The overall framework of prototype-routing optimization mechanism.

$p_{\text{data}}(\mathbf{x}_u)$, can be regarded as applying the information bottleneck theory (Tishby et al., 2000; Alemi et al., 2015) to forces $\mathbf{z}_u$ maintaining the most important information as much as possible. On the other hand, given a prior satisfying $p_\Theta(\mathbf{z}_u) = \prod_{j=1}^{d'} p_\Theta(z_{u,j})$, it will be possible to encourage independence among the dimensions of $\mathbf{z}_u$ through emphasizing D_{KL} in Eq. (6). Thus we follow the common practice as β-VAE (Higgins et al., 2017) to penalize Eq. (6) by a factor of $\beta \gg 1$, resulting in the following objective:

$$\mathbb{E}_{p_\Theta(\mathbf{C})}\Big[\mathbb{E}_{q_\Theta(\mathbf{z}_u|\mathbf{x}_u,\mathbf{C})}\big[\ln p_\Theta(\mathbf{x}_u \mid \mathbf{z}_u, \mathbf{C})\big] - \beta \cdot D_{\text{KL}}\big(q_\Theta(\mathbf{z}_u \mid \mathbf{x}_u, \mathbf{C}) \| p_\Theta(\mathbf{z}_u)\big)\Big]. \quad (7)$$

3.3.2. Social Dynamic Routing in Social Environment

In the social environment, we will have a set of subgraphs $\mathcal{SG} = \{SG_1, SG_2, \ldots, SG_g\}$ after the curriculum subgraph partition process. The social subgraph $SG_j = (SV_j, SE_j)$ captures the social interactions between user $u \in SV_j$ and $v \in SV_j$, each of which is associated with a representation, i.e., $\mathbf{z}_u$ and $\mathbf{z}_v$ respectively.

In the literature (Shuman et al., 2013), the most popular strategy for calculating node representations within a graph structure is to aggregate information from their neighborhoods. Following this common practice, we next elaborate our social dynamic routing process which co-disentangles user representations within social environment by enriching and enforcing the disentangled representations from consuming environment. The key element of our routing process relies on a nonlinear function $\phi(\cdot)$ that outputs the representation of a user u based on her and her neighbors' representations, i.e., $\mathbf{z}_u = \phi\left(\mathbf{z}_u, \{\mathbf{z}_v : (u, v) \in SV_j\}\right)$, where $\phi(\cdot)$ can also be applied to more general frameworks such as layers in a graph neural network (GNN). From Property 1, we have the mode of $q_\Theta(\mathbf{z}_u \mid \mathbf{x}_u, \mathbf{C})$, i.e., a variational distribution for an approximation to the posterior, as the representation of user u from the consuming environment, which serves as the initialization of $\mathbf{z}_u$ for $\phi(\cdot)$. The nonlinear function $\phi(\cdot)$ is expected to output an updated disentangled representation $\mathbf{r}_u = [\mathbf{r}_u^{(1)}; \mathbf{r}_u^{(2)}; \cdots; \mathbf{r}_u^{(k)}; \cdots; \mathbf{r}_u^{(K)}] \in \mathcal{R}^{d \cdot K}$, composing of K independent components indicating the K different concepts from the consuming environment. The core problem is identifying the subset of neighbors that connect to user u under concept k so that we can characterize the aspect of user u regarding concept k more accurately.

It is natural for $\phi(\cdot)$ to contain K channels which extract different concept features from user $n \in \{u\} \cup \{v : (u, v) \in SV_j\}$, by projecting the input representation $\mathbf{z}_n$ into different subspaces $\mathbf{s}_{n,k} = \sigma(\mathbf{W}_k^\intercal \mathbf{z}_n + \mathbf{b}_k) \Big/ \left\|\sigma(\mathbf{W}_k^\top \mathbf{z}_n + \mathbf{b}_k)\right\|_2$, where $\mathbf{W}_k \in \mathcal{R}^{d \cdot K \times d}$ and $\mathbf{b}_k \in \mathcal{R}^d$ are the parameters of channel k, and $\sigma(\cdot)$ is a nonlinear activation function. L_2 normalization is employed to ensure numerical stability and prevent the neighbors with heavily rich features from distorting the routing process. Given that $\mathbf{z}_n$ is initially generated from a Gaussian Mixture based model, $\mathbf{s}_{n,k}$ is expected to approximately characterize the aspect of user n which are relevant with concept k.

However, the common practice of aggregating information from neighborhood in literature (Shuman et al., 2013)

indicates a valid solution to capture the aspect of user u under concept k more comprehensively. As such, we aggregate information from the neighborhood through constructing a routing center $\mathbf{r}_{u,k}$ based on both $\mathbf{s}_{u,k}$ and $\{\mathbf{s}_{v,k} : (u,v) \in SE_j\}$. To construct $\mathbf{r}_{u,k}$ capable of characterizing user u's aspect related to concept k, it is necessary to dynamically infer a subset of neighbors who are actually connected to user u due to concept k.

Let $l_{v,k} \geq 0, \sum_1^K l_{v,k} = 1$ be the likelihood that concept k is the underlying reason of connection between user u and its neighbor v, then $l_{v,k}$ is also the probability of utilizing neighbor v to construct $\mathbf{r}_{u,k}$. Our social dynamic routing process will infer $l_{v,k}$ and construct $\mathbf{s}_{u,k}$ via iteratively searching for the largest routing center in each subspace under the constraint that each neighbor v approximately belongs to only one routing center: $l_{v,k}(t) = \frac{\exp\left(\eta \cdot \mathbf{s}_{v,k}{}^\intercal \mathbf{r}_{u,k}(t-1)\right)}{\sum_{k'=1}^K \exp\left(\eta \cdot \mathbf{s}_{v,k'}{}^\intercal \mathbf{r}_{u,k'}(t-1)\right)}$ and $\mathbf{r}_{u,k}(t) = \frac{\mathbf{s}_{u,k} + \sum_{v:(u,v)\in SE_j} l_{v,k}(t)\, \mathbf{s}_{v,k}}{1 + \sum_{v:(u,v)\in SE_j} l_{v,k}(t)}$, where $t = 1, \cdots, T$. The output disentangled representation $\mathbf{r}_u$ from social environment can therefore be obtained through $\mathbf{r}_u^{(k)} = \mathbf{r}_{u,k}(T)$ for $k = 1, 2, \cdots, K$. During training, the channels will remain changing because different subsets of the neighborhood will be routed for dynamically aggregating neighbor information in different iterations.

With the Gaussian Mixture initialization from PROTOTYPE LEARNING, we derive the theorem on convergence:

Theorem 1. *The* SOCIALDYNAMICROUTING *procedure is equivalent to an expectation-maximization (EM) algorithm for the mixture model. In particular, it converges to a point estimate of* $\{\mathbf{r}\}_{k=1}^K$ *that maximizes the marginal likelihood* $l\left(\{\mathbf{s}_{v,k} : (u,v) \in E, 1 \leq k \leq K\}; \{\mathbf{r}\}_{k=1}^K\right)$.

See the Appendix for the proof.

3.3.3. DECODING AND RECONSTRUCTION

Given that our prototype-routing optimization mechanism encourages the k concepts to be aligned between $\mathbf{z}_u$ and $\mathbf{r}_u$. The co-disentangled representation across consuming and social environments for user u can then be formulated as $\mathbf{z}_u = \rho \cdot \mathbf{z}_u + \mathbf{r}_u$, which is inspired by the residual block (He et al., 2016) where $\mathbf{r}_u$ can be treated as a disentangled routing of $\mathbf{z}_u$ in social environment and ρ is a parameter controlling the attention over consuming environment.

The decoding process predicts which of the M candidates are most possibly consumed by user u, given her co-disentangled representation $\mathbf{z}_u = [\mathbf{z}_u^{(1)}; \mathbf{z}_u^{(2)}; \ldots; \mathbf{z}_u^{(K)}]$ across consuming and social environments as well as the learned prototype assignment $\mathbf{C} = \{\mathbf{c}_i\}_{i=1}^M$,

$$p_{u,i} = p_\Theta(x_{u,i} \mid \mathbf{z}_u, \mathbf{C}) = \sum_{k=1}^K c_{i,k} \cdot \exp\left(\frac{\mathbf{z}_u^{(k)\intercal} \mathbf{h}_i}{\tau \cdot ||\mathbf{z}_u^{(k)}||_2 \cdot ||\mathbf{h}_i||_2}\right), \tag{8}$$

where $\mathbf{h}_i$ is a learnable latent representation for item i used to derive $\mathbf{c}_i$. Therefore, another training objective is to optimize the model's reconstruction capability through minimizing the cross entropy loss between ground-truth user behaviors $\mathbf{x}_u$ and the reconstructed ones $\mathbf{x}_u^*$. Putting Eq. (7), w_{SG_j} from Algorithm 1 and the cross entropy loss together, we have the following overall training objective:

$$\mathcal{L}_u = \big(-\beta \cdot D_{\mathrm{KL}}^u + \sum_{i:x_{u,i}=1} \ln p_{u,i}\big) w_{SG_j}. \tag{9}$$

Algorithm 2 shows the detailed implementations of the whole algorithm, covering curriculum subgraph training in Sec. 3.2 and prototype-routing optimization in Sec. 3.3.

Algorithm 2 Curriculum Co-disentangled (CurCoDis) Model

Input: $G = (V, E)$, $\mathbf{x}_u = \{x_{u,i} : x_{u,i} = 1\}$ for $u \in V$;
Parameters (Θ): $\mathbf{h}_i \in \mathcal{R}^d, \mathbf{t}_i \in \mathcal{R}^d, \mathbf{c}_i \in \mathcal{R}^K$, $i \in [1, M]$; $\mathbf{m}_k \in \mathcal{R}^d, \mathbf{W}_k \in \mathcal{R}^{d \cdot K \times d}, \mathbf{b}_k \in \mathcal{R}^d$, $k \in [1, K]$; $\mathbf{z}_u \in \mathcal{R}^{d \cdot K}$, $u \in [1, N]$; $\mathbf{f}_{\mathrm{nn}} : \mathcal{R}^d \to \mathcal{R}^{2d}$;
function PROTOTYPELEARNING($\{\mathbf{h}_i\}_{i=1}^M, \{\mathbf{m}_k\}_{k=1}^K, \tau$)
  for $i = 1, 2, \cdots, M$ **do**
    $o_{i,k} \leftarrow \mathbf{h}_i^T \mathbf{m}_k / (\tau \cdot ||\mathbf{h}_i||_2 \cdot ||\mathbf{m}_k||_2), \quad k \in [1, K]$.
    $\mathbf{c}_i \sim$GUMBEL-SOFTMAX($[o_{i,1}; o_{i,2}; \cdots; o_{i,K}]$).
  return $\{\mathbf{c}_i\}_{i=1}^M$
function ENCODING($\mathbf{x}_u, \{\mathbf{c}_i\}_{i=1}^M, \{\mathbf{t}_i\}_{i=1}^M$)
  for $k = 1, 2, \cdots, K$ **do**
    $(\mathbf{a}_k, \mathbf{b}_k) \leftarrow \mathbf{f}_{\mathrm{nn}}\left(\frac{\sum_{i:x_{u,i}=1} c_{i,k} \cdot \mathbf{t}_i}{\sqrt{\sum_{i:x_{u,i}=1} c_{i,k}^2}}\right)$.
    $\boldsymbol{\mu}^{(k)} \leftarrow \mathbf{a}_k / ||\mathbf{a}_k||_2, \quad \boldsymbol{\sigma}^{(k)} \leftarrow \sigma_0 \cdot \exp(-\frac{1}{2}\mathbf{b}_k)$.
  $\boldsymbol{\mu}_u \leftarrow [\boldsymbol{\mu}^{(1)}; \cdots; \boldsymbol{\mu}^{(K)}], \quad \boldsymbol{\sigma}_u \leftarrow [\boldsymbol{\sigma}^{(1)}; \cdots; \boldsymbol{\sigma}^{(K)}]$.
  $\mathbf{z}_u = \boldsymbol{\mu}_u + \boldsymbol{\epsilon} \circ \boldsymbol{\sigma}_u, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$.
    ▷ $\circ$ stands for element-wise multiplication.
  return $\mathbf{z}_u, D_{\mathrm{KL}}^u(\mathcal{N}(\boldsymbol{\mu}_u, diag(\boldsymbol{\sigma}_u)) || \mathcal{N}(0, \boldsymbol{\sigma}_0 \cdot \mathbf{I}))$
function SOCIALDYNAMICROUTING($\mathbf{z}_u, EG, \eta$)
  for $n \in \{u\} \cup \{v : (u,v) \in SE\}$ **do**
    for $k = 1, 2, \cdots, K$ **do**
      $\mathbf{s}_{n,k} \leftarrow \sigma(\mathbf{W}_k^\intercal \mathbf{z}_n + \mathbf{b}_k)$.
      $\mathbf{s}_{n,k} \leftarrow \mathbf{s}_{n,k} / ||\mathbf{s}_{n,k}||_2$.
  $\mathbf{r}_{u,k} \leftarrow \mathbf{s}_{u,k}, \forall k = 1, 2, \cdots, K$.
  for $t = 1, 2, \cdots, T$ **do**
    for $v \in \{v : (u,v) \in SE\}$ **do**
      $l_{v,k} \leftarrow \eta \cdot \mathbf{s}_{v,k}^\intercal \mathbf{r}_{u,k}, \forall k = 1, 2, \cdots, K$.
      $[l_{v,1}; \cdots; l_{v,K}] \leftarrow$SOFTMAX($[l_{v,1}; \cdots; l_{v,K}]$).
    for $k = 1, 2, \cdots, K$ **do**
      $\mathbf{r}_{u,k} \leftarrow \mathbf{s}_{u,k} + \sum_{v:(u,v)\in SE} l_{v,k} \mathbf{s}_{v,k}$.
      $\mathbf{r}_{u,k} \leftarrow \mathbf{r}_{u,k} / \left(1 + \sum_{v:(u,v)\in SE} l_{v,k}\right)$.
  $\mathbf{r}_u \leftarrow [\mathbf{r}_{u,1}; \cdots; \mathbf{r}_{u,K}]$.
  return $\mathbf{r}_u$
function MULTISOCIALROUTING($\mathbf{z}_u, EG, \eta, L$)
  $\mathbf{r}_u \leftarrow \mathbf{z}_u$.
  for $l = 1, 2, \cdots, L$ **do**
    $\mathbf{r}_u \leftarrow$SOCIALDYNAMICROUTING($\mathbf{r}_u, EG, \eta$).
    $\mathbf{r}_u \leftarrow$DROPOUT(RELU($\mathbf{r}_u$)).
  return $\mathbf{r}_u$
function DECODING($\mathbf{z}_u, \{\mathbf{c}_i\}_{i=1}^M, \{\mathbf{h}_i\}_{i=1}^M, \tau$)
  $p_{u,i} \leftarrow \sum_{k=1}^K c_{i,k} \cdot \exp\left(\frac{\mathbf{z}_u^{(k)\intercal} h_i}{\tau \cdot ||\mathbf{z}_u^{(k)}||_2 \cdot ||h_i||_2}\right)$.
  $[p_{u,1}; \cdots; p_{u,M}] \leftarrow$SOFTMAX($\ln p_{u,1}; \cdots; \ln p_{u,M}$).
  return $\{p_{u,i}\}_{i=1}^M$

BEGIN MAIN FUNCTION:
Initialize $\{\mathbf{z}_n\}_{n=1}^{N}, CurriculumEpoch, TotalEpoch, \lambda, \beta, \tau, \eta, \rho, L, epoch \leftarrow 0$.
repeat
  $\mathcal{SG} = \{SG_1, SG_2, \cdots, SG_g\} \leftarrow$ KERNIGHAN–LIN ALGORITHM(G), where $SG_j = (SV_j, SE_j)$ is a subgraph.
  $\Delta \leftarrow \{\delta_{SG_1}, \delta_{SG_2}, \ldots, \delta_{SG_g}\}$ based on Eq.(1) and (2).
  $\mathcal{SG} \leftarrow$ Sort $\mathcal{SG}$ in ascending order based on Δ.
  $\mathcal{W} = \{w_{SG_1}, w_{SG_2}, \ldots, w_{SG_g}\} \leftarrow$ CURRICULUM SUBGRAPH WEIGHING($CurriculumEpoch, \lambda, \mathcal{SG}, \Delta$).
  for $SG_j = (SV_j, SE_j) \in \mathcal{SG}$ **do**
    $\{\mathbf{c}_i\}_{i=1}^{M} \leftarrow$PROTOTYPELEARNING($\{\mathbf{h}_i\}_{i=1}^{M}, \{\mathbf{m}_k\}_{k=1}^{K}, \tau$).
    for $u \in SV_j$ **do**
      $\mathbf{z}_u, D_{\mathrm{KL}}^{u} \leftarrow$ENCODING($\mathbf{x}_u, \{\mathbf{c}_i\}_{i=1}^{M}, \{\mathbf{t}_i\}_{i=1}^{M}$).
      $\mathbf{r}_u \leftarrow$MULTISOCIALROUTING($\mathbf{z}_u, SE_j, \eta, L$).
      $\mathbf{z}_u \leftarrow \rho \cdot \mathbf{z}_u + \mathbf{r}_u$.
      $\{p_{u,i}\}_{i=1}^{M} \leftarrow$DECODING($\mathbf{z}_u, \{\mathbf{c}_i\}_{i=1}^{M}, \{\mathbf{h}_i\}_{i=1}^{M}, \tau$).
    $\mathcal{L} = \sum_u \big(- \beta \cdot D_{\mathrm{KL}}^{u} + \sum_{i:x_{u,i}=1} \ln p_{u,i} \big) w_{SG_j}$.
    $\Theta \leftarrow \mathbf{argmax}_{\Theta} \mathcal{L}$ by $\nabla_{\Theta} \mathcal{L}$.
  $epoch \leftarrow epoch + 1$.
until $epoch$ equals to $TotalEpoch$

4. Experiments

We empirically evaluate the performances of the proposed CurCoDis model over four real-world datasets and conduct several experiments to prove its effectiveness.

4.1. Experimental Setup

Datasets We conduct experiments on four real-world datasets: i) **Lastfm** (Cantador et al., 2011) with 1892 users, 17,632 music artists and 12,717 connections; ii) **Yelp** (Yin et al., 2019) with 34,504 users, 22,611 check-ins and 500,505 connections; iii) **Amazon Instrument** (McAuley et al., 2015) with 4219 consumers, 3933 products and 44,001 connections; iv) **Epinion** (Massa & Avesani, 2007) with 40,163 users, 139,738 items and 381,216 connections. We set $x_{u,i} = 1$ when user u consumes item i.

Baselines We compare our CurCoDis model with the following baselines: 1) **Diffnet** (Wu et al., 2019), a social recommendation model based on graph convolutional network (GCN); 2) **LightGCN** (He et al., 2020), a model employing neighborhood aggregation in GCN for recommendation; 3) **MHCN** (Yu et al., 2021b), a multi-channel hypergraph convolutional network that enhances social recommendation by leveraging high-order user relations via hypergraph convolution; 4) **SEPT** (Yu et al., 2021a), a socially-aware self-supervised learning framework that integrates tri-training; 5) **DISGCN** (Li et al., 2022), a model using the disentangled layer to strengthen social recommendation.

Hyper-parameters The number of subgraphs to be partitioned is set to $4, 64, 8, 128$ for Lastfm, Yelp, Amazon and Epinion respectively, given that their sizes are of different scales. We set d to 200, fixing λ in curriculum weighing to 0.1 and $CurriculumEpoch$ to $0.75 \cdot TotalEpoch$. Then we tune other hyper-parameters using ASHA (Li et al., 2018) implemented by Ray Tune (Liaw et al., 2018). Specifically, we run ray tune with 500 samples under each setting, with the hyper-parameters search space specified as follows: the *learning rate* follows LOG-UNIFORM$[10^{-4}, 10^{-1}]$, *dropout rate* $\in \{0.05, 0.10 \cdots 0.95\}$, $\beta, \tau, \eta \sim$RANDOM$(0, 1)$, $L \in \{1, 2, 3, 4, 5\}$, $\rho \in \{0.5, 1, 2, 4\}$.

Dataset	Method	Metric		
		NDCG@100	Recall@20	Recall@50
Lastfm	Diffnet	0.26318	0.22919	0.34557
	LightGCN	0.28691	0.24333	0.36899
	MHCN	**0.32702**	0.29121	0.41715
	SEPT	0.32216	0.28305	0.41141
	DISGCN	0.28555	0.28092	0.41243
	CurCoDis	0.30714	**0.30172**	**0.43236**
Yelp	Diffnet	0.08594	0.08638	0.15670
	LightGCN	0.09857	0.09656	0.17686
	MHCN	0.11114	0.11384	0.19489
	SEPT	0.10695	0.10995	0.19243
	DISGCN	0.10329	0.11803	0.20128
	CurCoDis	**0.11191**	**0.12846**	**0.21820**
Amazon	Diffnet	0.04745	0.06325	0.10538
	LightGCN	0.07470	0.09335	0.14926
	MHCN	0.07237	0.08289	0.14603
	SEPT	0.04336	0.06047	0.10792
	DISGCN	0.07046	0.09964	0.16245
	CurCoDis	**0.08047**	**0.11665**	**0.19126**
Epinion	Diffnet	0.04334	0.04709	0.08448
	LightGCN	0.05532	0.06199	0.10698
	MHCN	0.06070	0.06612	0.11309
	SEPT	0.06557	0.07502	0.12615
	DISGCN	0.05680	0.06760	0.11839
	CurCoDis	**0.07431**	**0.08908**	**0.14578**

Table 1. Comparisons between our proposed CurCoDis model and baselines on all four datasets, bold font denotes the winner. The full table with deviations will be presented in Appendix.

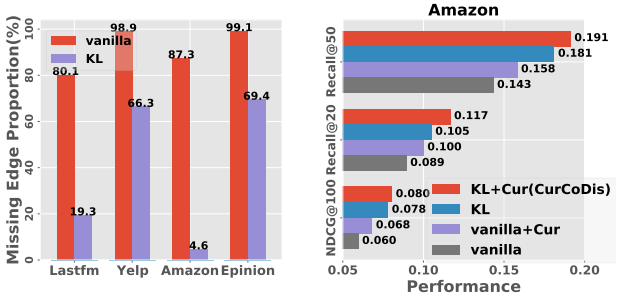


Figure 2. (Left) Proportion of edge lost. (Right) Ablation study.

4.2. Experimental Results

Recommendation Accuracy Table 1 reports the performance comparisons for six models over four datasets in terms of three evaluation metrics. We observe that the proposed CurCoDis model is able to significantly outperform

Metric	$\rho = 1$		$\rho = 2$	
	vanilla	CurCoDis	vanilla	CurCoDis
NDCG@100	**0.30341**	0.30230	**0.30271**	0.30055
Recall@20	0.29750	**0.29889**	**0.30151**	0.29865
Recall@50	0.42085	**0.42435**	0.42108	**0.42443**

Table 2. Comparisons between CurCoDis and vanilla on Lastfm. The vanilla model is trained using the whole social graph. The full table with more values of ρ is presented in Appendix.

Metric	$\rho = 0.5$		$\rho = 2$	
	vanilla	CurCoDis	vanilla	CurCoDis
NDCG@100	0.25863	**0.30230**	0.25534	**0.30055**
Recall@20	0.25074	**0.29889**	0.24840	**0.29865**
Recall@50	0.36875	**0.42435**	0.37040	**0.42443**

Table 3. Comparisons between CurCoDis and vanilla on Lastfm. The vanilla model substitutes our proposed social dynamic routing process with traditional graph convolutional network (GCN). The full table with more values of ρ is presented in Appendix.

all baselines, particularly up to $13.3\%, 18.7\%, 15.6\%$ performance boost over the large-scale Epinion dataset in terms of NDCG@100, Recall@20 and Recall@50 respectively. This demonstrates the benefits of curriculum subgraph weighing design allowing to learn in an easy-to-hard human-like manner and the co-disentangled representation learning capable of discovering disentangled intentions for each individual across consuming and social environments.

Curriculum Subgraph Training with Partition Figure 2 (Left) shows the benefits from adopting Kernighan-Lin (KL) algorithm during subgraph partitioning against the vanilla random algorithm. It is obvious that the number of edges lost with KL algorithm during the partition process is far less than those lost with vanilla algorithm, e.g., roughly 4X less on Lastfm and 19X less on Amazon. This indicates that our employment of Kernighan-Lin algorithm can dramatically prevent information loss in the social environment.

We compare the model performances of training over our partitioned subgraphs and those of training over the whole social graph on the relatively small dataset Lastfm in Table 2. From the results, we observe that the two models perform almost the same under all settings, which illustrates the benefits of training over subgraphs obtained through our proposed partition strategy when the social environment contains large-scale social graphs that can not be fed into the memory as a whole.

We further conduct ablation studies on the effects of different components in curriculum subgraph training strategy. Figure 2 (Right) shows the performances of *vanilla without curriculum* (red), *vanilla with curriculum* (blue), *KL without curriculum* (purple), *KL with curriculum, i.e., CurCoDis* (grey) on Amazon. The results validate the efficacy of both Kernighan-Lin partition and curriculum subgraph weighing strategy in improving the model performances. Similar results hold on other three datasets (more in Appendix).

Consuming Environment v.s. Social Environment We explore the role of parameter ρ within co-disentangled representation learning. Figure 3 shows the model performances with different values of ρ on the four datasets. Particularly for Lastfm, the model generally performs best with $\rho = 2$, i.e., the importance ratio between consuming and social environments reaches 2, being gradually worse when ρ becomes either larger or smaller, indicating the importance of appropriate balancing between different environments.

Disentanglement We additionally measure the degree of the disentanglement achieved based on the independence level of the dimensions within $\mathbf{z}_u$. We quantify the independence level $\mathcal{IL}_u^{(k)}$ of each $\mathbf{z}_u^{(k)}$ as $\mathcal{IL}_u^{(k)} = 1 - \frac{2}{d(d-1)} \sum_{1 \le i,j \le d} |cor_{i,j}|$, where $cor_{i,j}$ is the correlation between i^{th} and j^{th} dimension of $\mathbf{z}_u^{(k)}$. Figure 4 shows the degree of disentanglement (via mean independence level $\frac{1}{N}\frac{1}{K}\sum_{u=1}^{N}\sum_{k=1}^{K}\mathcal{IL}_u^{(k)}$) and the corresponding performances with different training epochs on the four datasets. Particularly for the Amazon dataset, we set $(k, d) = (3, 10)$ and record every 25 epochs for 225 epochs in total. We observe that the proposed CurCoDis model gradually reaches a high degree of disentanglement in the training process and the model performances generally have a positive relevance with the degree of disentanglement, demonstrating the benefits brought by disentanglement.

To further investigate the role of disentanglement within user representation $\mathbf{z_u}$, we additionally conduct an ablation study in which we substitute the proposed social dynamic routing process with classic graph convolutional network (GCN) (Kipf & Welling, 2016). We alter the balance between consuming and social environments, i.e., we change the value of ρ, and fix all other hyper-parameters. The average results on Lastfm dataset are reported in Table 3. The performances corresponding to a non-disentangled user representation drop significantly under all the experimental settings, indicating that the disentanglement within $\mathbf{z_u}$ has a great impact on the prediction accuracy.

Moreover, we visualize the high-dimensional user and item representations learned by CurCoDis on Amazon through t-SNE (Van der Maaten & Hinton, 2008).We treat the k components of each representation as k separate points, i) coloring items based on their ground-truth categories and ii) coloring users with category k if they have the largest $\sum_{i:x_{u,i}>0} c_{i,k}$ for user u, where $c_{i,k}$ is predicted by CurCoDis rather than the ground-truth label. The results are depicted in Figure 5 where the plots from left to right in each subfigure present the visualization at epoch 10, 110 and 210 respectively, demonstrating the capability of CurCoDis

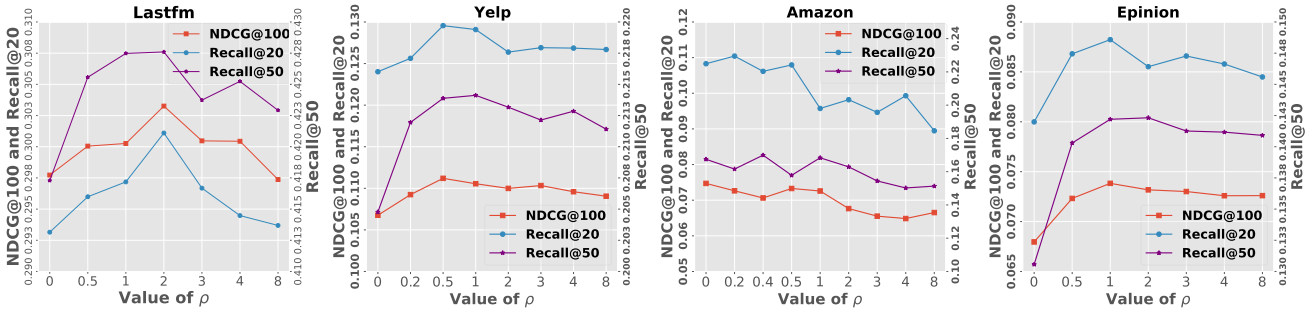


Figure 3. Model performances with different values of ρ for Lastfm, Yelp, Amazon and Epinion.

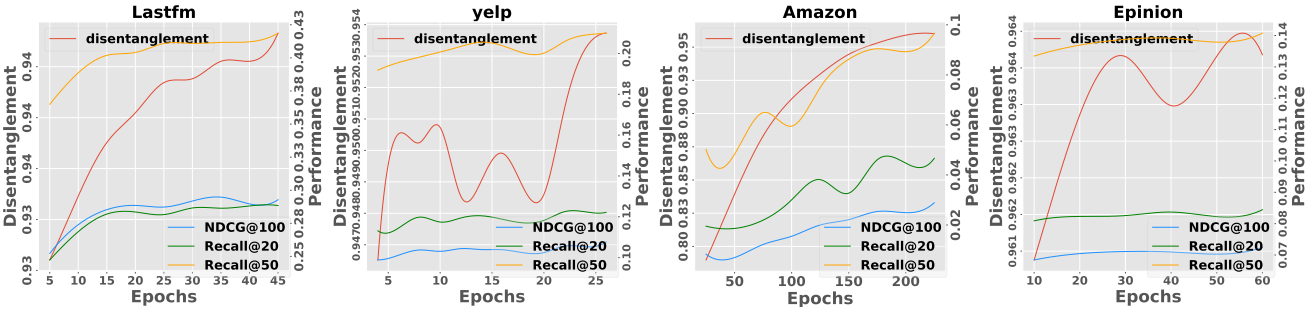
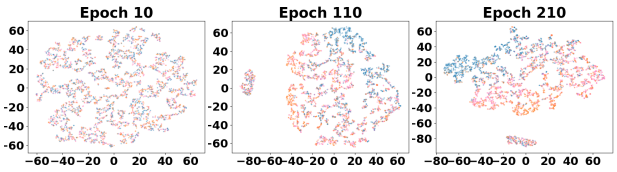
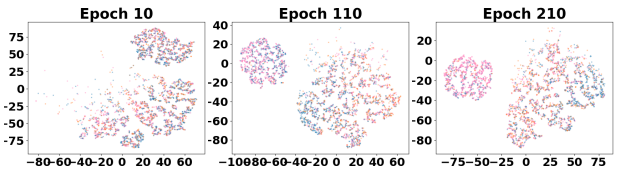


Figure 4. Degree of disentanglement within $\mathbf{z}_u$ and its correlation with the performance for Lastfm, Yelp, Amazon and Epinion.



(a) Items colored based on the ground-truth categories.



(b) Users colored based on the predicted categories.

Figure 5. t-SNE visualizations of representations in Amazon.

in gradually reaching disentanglement upon training.

Explainability We further investigate the explainability of the learned co-disentangled representations. Given a disentangled representation, we gradually alter the values of dimensions representing certain concepts, and retrieve the closest items in the representation space. Figure 6 identifies two concepts with semantic meaning for Amazon, i.e., SIZE and COLOR, and list the closest items when changing the values of the corresponding dimensions. These results show the ability of CurCoDis to learn explainable representations and provide fine-grained controls over particular concepts of the candidate items for explainable recommendation.

(a) SIZE of Microphones & Accessories: *tiny→large*.

(b) COLOR of Amplifiers & Effects: *dark→light*.

Figure 6. List of items with different values of a particular concept.

5. Conclusion

We study curriculum co-disentangled representation learning across different environments for the first time. We believe this work may serve as one step towards conscious aware environment learning, assuming that the human consciousness can be represented in a disentangled manner.

Acknowledgements

This work is supported in part by National Key Research and Development Program of China No.2022ZD0115903, NSFC No.62250008, 62222209, 62102222, BNRist under Grant No.BNR2023RC01003, BNR2023TD03006, and Beijing Key Lab of Networked Multimedia.

References

Alemi, A. A., Fischer, I., Dillon, J. V., and Murphy, K. Deep variational information bottleneck. In *International Conference for Learning Representations*, 2015.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. Curriculum learning. In *International conference on machine learning*, pp. 41–48, 2009.

Cantador, I., Brusilovsky, P., and Kuflik, T. Second workshop on information heterogeneity and fusion in recommender systems (hetrec2011). pp. 387–388, 10 2011.

Castells, T., Weinzaepfel, P., and Revaud, J. Superloss: A generic loss for robust curriculum learning. *Advances in Neural Information Processing Systems*, 33, 2020.

Chen, H., Chen, Y., Wang, X., Xie, R., Wang, R., Xia, F., and Zhu, W. Curriculum disentangled recommendation with noisy multi-feedback. *Advances in Neural Information Processing Systems*, 34:26924–26936, 2021.

Chen, H., Zhang, Y., Wang, X., Duan, X., Zhou, Y., and Zhu, W. Disenbooth: Identity-preserving disentangled tuning for subject-driven text-to-image generation. *arXiv preprint arXiv:2305.03374*, 2023.

Chen, J., Feng, Y., Ester, M., Zhou, S., Chen, C., and Wang, C. Modeling users' exposure with social knowledge influence and consumption influence for recommendation. In *CIKM*, pp. 953–962. ACM, 2018.

Cui, L., Wang, C., Wu, J., Yang, J., and Sheng, M. Individual interest and trust driving collective intelligence awareness for social recommendation. In *IJCNN*, pp. 1–6. IEEE, 2018.

Dehmer, M., Chen, Z., Emmert-Streib, F., Tripathi, S., Mowshowitz, A., Levitchi, A., Feng, L., Shi, Y., and Tao, J. Measuring the complexity of directed graphs: A polynomial-based approach. *Plos one*, 14(11), 2019.

Fan, W., Ma, Y., Li, Q., He, Y., Zhao, E., Tang, J., and Yin, D. Graph neural networks for social recommendation. In *The world wide web conference*, pp. 417–426, 2019.

Feng, Z., Wang, X., Ke, C., Zeng, A.-X., Tao, D., and Song, M. Dual swap disentangling. volume 31, 2018.

Gonzalez Camacho, L. A. and Alves-Souza, S. N. Social network data to alleviate cold-start in recommender system. *Information Processing and Management: an International Journal*, 54(4):529–544, 2018.

Graves, A., Bellemare, M. G., Menick, J., Munos, R., and Kavukcuoglu, K. Automated curriculum learning for neural networks. In *international conference on machine learning*, pp. 1311–1320. PMLR, 2017.

Hacohen, G. and Weinshall, D. On the power of curriculum learning in training deep networks. In *International Conference on Machine Learning*, pp. 2535–2544. PMLR, 2019.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, R., Lee, W. S., Ng, H. T., and Dahlmeier, D. An unsupervised neural attention model for aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 388–397, 2017.

He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., and Wang, M. 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*, pp. 639–648, 2020.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2017.

Jain, S., Banner, E., van de Meent, J.-W., Marshall, I. J., and Wallace, B. C. Learning disentangled representations of texts with application to biomedical abstracts. 2018:4683, 2018.

Jamali, M. and Ester, M. A matrix factorization technique with trust propagation for recommendation in social networks. In *Recsys*, pp. 135–142. ACM, 2010.

Jang, E., Gu, S., and Poole, B. Categorical reparametrization with gumble-softmax. In *International Conference on Learning Representations (ICLR 2017)*. OpenReview. net, 2017.

Kernighan, B. W. and Lin, S. An efficient heuristic procedure for partitioning graphs. *The Bell system technical journal*, 49(2):291–307, 1970.

Kim, H. and Mnih, A. Disentangling by factorising. pp. 2649–2658, 2018.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kingma, D. P., Mohamed, S., Jimenez Rezende, D., and Welling, M. Semi-supervised learning with deep generative models. volume 27, 2014.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Kumar, M., Packer, B., and Koller, D. Self-paced learning for latent variable models. *Advances in neural information processing systems*, 23, 2010.

Lan, X., Yuan, Y., Chen, H., Wang, X., Jie, Z., Ma, L., Wang, Z., and Zhu, W. Curriculum multi-negative augmentation for debiased video grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

Li, H., Wang, X., Zhang, Z., Yuan, Z., Li, H., and Zhu, W. Disentangled contrastive learning on graphs. *Advances in Neural Information Processing Systems*, 34:21872–21884, 2021.

Li, H., Zhang, Z., Wang, X., and Zhu, W. Disentangled graph contrastive learning with independence promotion. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

Li, L., Jamieson, K., Rostamizadeh, A., Gonina, E., Hardt, M., Recht, B., and Talwalkar, A. Massively parallel hyperparameter tuning. *arXiv preprint arXiv:1810.05934*, 5, 2018.

Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J. E., and Stoica, I. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*, 2018.

Locatello, F., Tschannen, M., Bauer, S., Rätsch, G., Schölkopf, B., and Bachem, O. Disentangling factors of variations using few labels. In *International Conference on Learning Representations*, 2019.

Ma, H., King, I., and Lyu, M. R. Learning to recommend with social trust ensemble. In *SIGIR*, pp. 203–210. ACM, 2009.

Ma, H., Zhou, D., Liu, C., Lyu, M. R., and King, I. Recommender systems with social regularization. In *WSDM*, pp. 287–296. ACM, 2011.

Ma, J., Cui, P., Kuang, K., Wang, X., and Zhu, W. Disentangled graph convolutional networks. In *International conference on machine learning*, pp. 4212–4221. PMLR, 2019a.

Ma, J., Zhou, C., Cui, P., Yang, H., and Zhu, W. Learning disentangled representations for recommendation. *Advances in neural information processing systems*, 32, 2019b.

Ma, J., Zhou, C., Yang, H., Cui, P., Wang, X., and Zhu, W. Disentangled self-supervision in sequential recommenders. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 483–491, 2020.

Maddison, C., Mnih, A., and Teh, Y. The concrete distribution: A continuous relaxation of discrete random variables. In *Proceedings of the international conference on learning Representations*. International Conference on Learning Representations, 2017.

Massa, P. and Avesani, P. Trust-aware recommender systems. RecSys '07, pp. 17–24, 2007.

McAuley, J., Targett, C., Shi, Q., and van den Hengel, A. Image-based recommendations on styles and substitutes. SIGIR '15, pp. 43–52, 2015.

Purushotham, S., Liu, Y., and Kuo, C.-C. J. Collaborative topic regression with social matrix factorization for recommendation systems. In *International Conference on Machine Learning*, pp. 691–698, 2012.

Qian, X., Feng, H., Zhao, G., and Mei, T. Personalized recommendation combining user interest and social circle. *TKDE*, 26(7):1763–1777, 2014.

Rabinovich, R. and Forschungsgebiet, L.-u. Complexity measures of directed graphs. *Diss., Rheinisch-Westfälische Technische Hochschule Aachen*, pp. 123, 2008.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on International Conference on Machine Learning-Volume 32*, pp. II–1278. JMLR. org, 2014.

Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., and Vandergheynst, P. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine*, 30(3):83–98, 2013.

Sinha, S., Garg, A., and Larochelle, H. Curriculum by smoothing. *Advances in Neural Information Processing Systems*, 33, 2020.

Spitkovsky, V. I., Alshawi, H., and Jurafsky, D. From baby steps to leapfrog: How "less is more" in unsupervised dependency parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 751–759, 2010.

Tishby, N., Pereira, F. C., and Bialek, W. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Wang, X., Lu, W., Ester, M., Wang, C., and Chen, C. Social recommendation with strong and weak ties. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pp. 5–14, 2016.

Wang, X., Hoi, S. C., Ester, M., Bu, J., and Chen, C. Learning personalized preference of strong and weak ties for social recommendation. In *Proceedings of the 26th International Conference on World Wide Web*, pp. 1601–1610, 2017a.

Wang, X., Hoi, S. C., Liu, C., and Ester, M. Interactive social recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 357–366, 2017b.

Wang, X., Zhu, W., and Liu, C. Social recommendation with optimal limited attention. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1518–1527, 2019.

Wang, X., Chen, H., and Zhu, W. Multimodal disentangled representation for recommendation. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE, 2021a.

Wang, X., Chen, Y., and Zhu, W. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021b.

Wang, X., Chen, H., Tang, S., Wu, Z., and Zhu, W. Disentangled representation learning. *arXiv preprint arXiv:2211.11695*, 2022a.

Wang, X., Chen, H., Zhou, Y., Ma, J., and Zhu, W. Disentangled representation learning for recommendation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022b.

Weinshall, D., Cohen, G., and Amir, D. Curriculum learning by transfer learning: Theory and experiments with deep networks. In *International Conference on Machine Learning*, pp. 5238–5246. PMLR, 2018.

Wu, L., Sun, P., Hong, R., Ge, Y., and Wang, M. Collaborative neural social recommendation. *IEEE transactions on systems, man, and cybernetics: systems*, 51(1):464–476, 2018.

Wu, L., Sun, P., Fu, Y., Hong, R., Wang, X., and Wang, M. A neural influence diffusion model for social recommendation. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pp. 235–244, 2019.

Yang, B., Lei, Y., Liu, J., and Li, W. Social collaborative filtering by trust. *IEEE transactions on pattern analysis and machine intelligence*, 39(8):1633–1647, 2016.

Yang, S.-H., Long, B., Smola, A., Sadagopan, N., Zheng, Z., and Zha, H. Like like alike: joint friendship and interest propagation in social networks. In *ACM international conference on World wide web*, pp. 537–546, 2011.

Ye, M., Liu, X., and Lee, W.-C. Exploring social influence for recommendation: a generative model approach. In *SIGIR*, pp. 671–680. ACM, 2012.

Yin, H., Wang, Q., Zheng, K., Li, Z., Yang, J., and Zhou, X. Social influence-based group representation learning for group recommendation. In *IEEE International Conference on Data Engineering*, pp. 566–577, 2019.

Yu, J., Yin, H., Gao, M., Xia, X., Zhang, X., and Viet Hung, N. Q. Socially-aware self-supervised tri-training for recommendation. KDD '21, pp. 2084–2092, 2021a.

Yu, J., Yin, H., Li, J., Wang, Q., Hung, N. Q. V., and Zhang, X. Self-supervised multi-channel hypergraph convolutional network for social recommendation. In *ACM the Web Conference 2021*, WWW '21, pp. 413–424, New York, NY, USA, 2021b.

Zhang, Y., Zuo, W., Shi, Z., Yue, L., and Liang, S. Social bayesian personal ranking for missing data in implicit feedback recommendation. In *KSEM*, pp. 299–310. Springer, 2018.

Zhang, Y., Wang, X., Chen, H., and Zhu, W. Adaptive disentangled transformer for sequential recommendation. In *Proceedings of the 29th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2023.

Zhang, Z., Zhang, Z., Wang, X., and Zhu, W. Learning to solve travelling salesman problem with hardness-adaptive curriculum. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 9136–9144, 2022.

Zhao, M., Wu, H., Niu, D., and Wang, X. Reinforced curriculum learning on pre-trained neural machine translation models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 9652–9659, 2020.

Zhao, T., McAuley, J., and King, I. Leveraging social connections to improve personalized ranking for collaborative filtering. In *CIKM*, pp. 261–270. ACM, 2014.

Zhao, Z., Lu, H., Cai, D., He, X., and Zhuang, Y. User preference learning for online social recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 28(9):2522–2534, 2016.

Zhou, Y., Wang, X., Chen, H., Duan, X., Guan, C., and Zhu, W. Curriculum-nas: Curriculum weight-sharing neural architecture search. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 6792–6801, 2022.

A. Proofs

Property 1. $\ln p_\Theta(\mathbf{x}_u)$ *is bounded as follows:*

$$\begin{aligned}\ln p_\Theta(\mathbf{x}_u) \geq & \mathbb{E}_{p_\Theta(\mathbf{C})}\Big[\mathbb{E}_{q_\Theta(\mathbf{z}_u|\mathbf{x}_u,\mathbf{C})}\big[\ln p_\Theta(\mathbf{x}_u \mid \mathbf{z}_u, \mathbf{C})\big] \\ & - D_{\mathrm{KL}}\big(q_\Theta(\mathbf{z}_u \mid \mathbf{x}_u, \mathbf{C}) \| p_\Theta(\mathbf{z}_u)\big)\Big].\end{aligned} \tag{4}$$

The proof is as follows.

Proof. Given the following equation,

$$q_\Theta(\mathbf{z}_u, \mathbf{C} \mid \mathbf{x}_u) = q_\Theta(\mathbf{z}_u \mid \mathbf{x}_u, \mathbf{C}) p_\Theta(\mathbf{C}),$$

then we have the following inequality,

$$\begin{aligned}
& \ln p_\Theta(\mathbf{x}_u) \\
=& \mathbb{E}_{q_\Theta(\mathbf{z}_u,\mathbf{C}|\mathbf{x}_u)}\big[\ln p_\Theta(\mathbf{x}_u)\big] \\
=& \mathbb{E}_{q_\Theta(\mathbf{z}_u,\mathbf{C}|\mathbf{x}_u)}\left[\ln \frac{p_\Theta(\mathbf{x}_u, \mathbf{z}_u, \mathbf{C})}{p_\Theta(\mathbf{z}_u, \mathbf{C} \mid \mathbf{x}_u)}\right] \\
=& \mathbb{E}_{q_\Theta(\mathbf{z}_u,\mathbf{C}|\mathbf{x}_u)}\left[\ln \frac{q_\Theta(\mathbf{z}_u, \mathbf{C} \mid \mathbf{x}_u)}{p_\Theta(\mathbf{z}_u, \mathbf{C} \mid \mathbf{x}_u)}\right] + \mathbb{E}_{q_\Theta(\mathbf{z}_u,\mathbf{C}|\mathbf{x}_u)}\left[\ln \frac{p_\Theta(\mathbf{x}_u, \mathbf{z}_u, \mathbf{C})}{q_\Theta(\mathbf{z}_u, \mathbf{C} \mid \mathbf{x}_u)}\right] \\
=& \mathbb{E}_{q_\Theta(\mathbf{z}_u,\mathbf{C}|\mathbf{x}_u)}\left[\ln \frac{q_\Theta(\mathbf{z}_u, \mathbf{C} \mid \mathbf{x}_u)}{p_\Theta(\mathbf{z}_u, \mathbf{C} \mid \mathbf{x}_u)}\right] + \mathbb{E}_{q_\Theta(\mathbf{z}_u,\mathbf{C}|\mathbf{x}_u)}\big[\ln p_\Theta(\mathbf{x}_u \mid \mathbf{z}_u, \mathbf{C})\big] + \mathbb{E}_{q_\Theta(\mathbf{z}_u,\mathbf{C}|\mathbf{x}_u)}\left[\ln \frac{p_\Theta(\mathbf{z}_u, \mathbf{C})}{q_\Theta(\mathbf{z}_u, \mathbf{C} \mid \mathbf{x}_u)}\right] \\
=& D_{\mathrm{KL}}\big(q_\Theta(\mathbf{z}_u, \mathbf{C} \mid \mathbf{x}_u) \| p_\Theta(\mathbf{z}_u, \mathbf{C} \mid \mathbf{x}_u)\big) + \mathbb{E}_{q_\Theta(\mathbf{z}_u,\mathbf{C}|\mathbf{x}_u)}\big[\ln p_\Theta(\mathbf{x}_u \mid \mathbf{z}_u, \mathbf{C})\big] - D_{\mathrm{KL}}\big(q_\Theta(\mathbf{z}_u, \mathbf{C} \mid \mathbf{x}_u) \| p_\Theta(\mathbf{z}_u, \mathbf{C})\big) \\
\geq & \mathbb{E}_{q_\Theta(\mathbf{z}_u,\mathbf{C}|\mathbf{x}_u)}\big[\ln p_\Theta(\mathbf{x}_u \mid \mathbf{z}_u, \mathbf{C})\big] - D_{\mathrm{KL}}\big(q_\Theta(\mathbf{z}_u, \mathbf{C} \mid \mathbf{x}_u) \| p_\Theta(\mathbf{z}_u, \mathbf{C})\big) \\
=& \mathbb{E}_{p_\Theta(\mathbf{C})}\big[\mathbb{E}_{q_\Theta(\mathbf{z}_u|\mathbf{x}_u,\mathbf{C})}\big[\ln p_\Theta(\mathbf{x}_u \mid \mathbf{z}_u, \mathbf{C})\big] - D_{\mathrm{KL}}\big(q_\Theta(\mathbf{z}_u \mid \mathbf{x}_u, \mathbf{C}) \| p_\Theta(\mathbf{z}_u)\big)\big].
\end{aligned}$$

Note that in the last line above, we have used

$$\begin{aligned}
& D_{\mathrm{KL}}\big(q_\Theta(\mathbf{z}_u, \mathbf{C} \mid \mathbf{x}_u) \| p_\Theta(\mathbf{z}_u, \mathbf{C})\big) \\
&= D_{\mathrm{KL}}\big(q_\Theta(\mathbf{z}_u \mid \mathbf{x}_u, \mathbf{C}) p_\theta(\mathbf{C}) \| p_\Theta(\mathbf{z}_u) p_\theta(\mathbf{C})\big) \\
&= \mathbb{E}_{p_\Theta(\mathbf{C})}\left[D_{\mathrm{KL}}\big(q_\Theta(\mathbf{z}_u \mid \mathbf{x}_u, \mathbf{C}) \| p_\Theta(\mathbf{z}_u)\big)\right],
\end{aligned}$$

which completes the proof. □

Property 2. *A reformulation of KL term in Eq.* (4)*:*

$$\begin{aligned}
& \mathbb{E}_{p_{\mathrm{data}}(\mathbf{x}_u)}\left[D_{\mathrm{KL}}(q_\Theta(\mathbf{z}_u \mid \mathbf{x}_u, \mathbf{C}) \| p_\Theta(\mathbf{z}_u))\right] \\
=& I_q(\mathbf{x}_u; \mathbf{z}_u) + D_{\mathrm{KL}}(q_\Theta(\mathbf{z}_u \mid \mathbf{C}) \| p_\Theta(\mathbf{z}_u)).
\end{aligned} \tag{6}$$

The proof is as follows.

Proof.

$$
\begin{aligned}
&\mathbb{E}_{p_{\text{data}}(\mathbf{x}_u)}\left[D_{\text{KL}}\big(q_\Theta(\mathbf{z}_u \mid \mathbf{x}_u, \mathbf{C}) \| p_\Theta(\mathbf{z}_u)\big)\right] \\
=&\mathbb{E}_{p_{\text{data}}(\mathbf{x}_u)}\left[\mathbb{E}_{q_\Theta(\mathbf{z}_u|\mathbf{x}_u,\mathbf{C})}\left[\ln \frac{q_\Theta(\mathbf{z}_u \mid \mathbf{x}_u, \mathbf{C})}{p_\Theta(\mathbf{z}_u)}\right]\right] \\
=&\mathbb{E}_{p_{\text{data}}(\mathbf{x}_u)}\left[\mathbb{E}_{q_\Theta(\mathbf{z}_u|\mathbf{x}_u,\mathbf{C})}\left[\ln \frac{q_\Theta(\mathbf{z}_u \mid \mathbf{x}_u, \mathbf{C})}{q_\Theta(\mathbf{z}_u \mid \mathbf{C})}\frac{q_\Theta(\mathbf{z}_u \mid \mathbf{C})}{p_\Theta(\mathbf{z}_u)}\right]\right] \\
=&\mathbb{E}_{p_{\text{data}}(\mathbf{x}_u)}\left[\mathbb{E}_{q_\Theta(\mathbf{z}_u|\mathbf{x}_u,\mathbf{C})}\left[\ln \frac{q_\Theta(\mathbf{z}_u \mid \mathbf{x}_u, \mathbf{C})}{q_\Theta(\mathbf{z}_u \mid \mathbf{C})} + \ln\frac{q_\Theta(\mathbf{z}_u \mid \mathbf{C})}{p_\Theta(\mathbf{z}_u)}\right]\right] \\
=&\mathbb{E}_{p_{\text{data}}(\mathbf{x}_u)}\left[D_{\text{KL}}\big(q_\Theta(\mathbf{z}_u \mid \mathbf{x}_u, \mathbf{C}) \| q_\Theta(\mathbf{z}_u \mid \mathbf{C})\big)\right] + \mathbb{E}_{q_\Theta(\mathbf{z}_u|\mathbf{x}_u,\mathbf{C})p_{\text{data}}(\mathbf{x}_u)}\left[\ln \frac{q_\Theta(\mathbf{z}_u \mid \mathbf{C})}{p_\Theta(\mathbf{z}_u)}\right] \\
=&I_q(\mathbf{x}_u; \mathbf{z}_u) + \mathbb{E}_{q_\Theta(\mathbf{z}_u|\mathbf{C})}\left[\ln \frac{q_\Theta(\mathbf{z}_u \mid \mathbf{C})}{p_\Theta(\mathbf{z}_u)}\right] \\
=&I_q(\mathbf{x}_u; \mathbf{z}_u) + D_{\text{KL}}\big(q_\Theta(\mathbf{z}_u \mid \mathbf{C}) \| p_\Theta(\mathbf{z}_u)\big).
\end{aligned}
$$

Note that $p_{\text{data}}(\mathbf{x}_u \mid \mathbf{C}) = p_{\text{data}}(\mathbf{x}_u)$, and the mutual information $I_q(\mathbf{x}_u; \mathbf{z}_u)$ is under the joint distribution

$$
\begin{aligned}
&q_\Theta(\mathbf{z}_u, \mathbf{x}_u \mid \mathbf{C}) \\
=&q_\Theta(\mathbf{z}_u \mid \mathbf{x}_u, \mathbf{C}) p_{\text{data}}(\mathbf{x}_u \mid \mathbf{C}) \\
=&q_\Theta(\mathbf{z}_u \mid \mathbf{x}_u, \mathbf{C}) p_{\text{data}}(\mathbf{x}_u),
\end{aligned}
$$

which completes the proof. □

With the Gaussian Mixture initialization from PROTOTYPE LEARNING, we derive the following theorem on convergence properties:

Theorem 1. *The* SOCIALDYNAMICROUTING *procedure is equivalent to an expectation-maximization (EM) algorithm for the mixture model. In particular, it converges to a point estimate of* $\{\mathbf{r}\}_{k=1}^K$ *that maximizes the marginal likelihood* $l\left(\{\mathbf{s}_{v,k} : (u,v) \in E, 1 \le k \le K\}; \{\mathbf{r}\}_{k=1}^K\right)$.

The proof is as follows.

Proof. Let

$$\theta = \{\mathbf{r}_k\}_{k=1}^K, A = \{a_v : (u,v) \in E\},$$

and

$$S = \{\mathbf{s}_{v,k} : (u,v) \in E, 1 \le k \le K)\}.$$

Factor a_v is the unknown factor of why neighbor v and user u are connected. To derive an EM algorithm that maximizes $l(A;\theta) = \sum_A l(A, S; \theta)$, we introduce here an additional auxiliary distribution $q(A)$ over A.

Let

$$L(\theta, q) = \sum_A q(A) \ln \frac{l(A, S; \theta)}{q(A)},$$

and

$$D_{\text{KL}}\left(q \| l_\theta\right) = \sum_A q(A) \ln \frac{q(A)}{l(A \mid S; \theta)}.$$

We can then verify the following:

$$\ln l(S;\theta) = L(\theta, q) + D_{\text{KL}}\left(q \| l_\theta\right),$$

where the second term here is the Kullback-Leibler (KL) divergence from $l(A \mid S; \theta)$ towards the auxiliary distribution $q(A)$. The KL divergence is non-negative. As a result, $L(\theta, q)$ is a lower bound of $\ln l(S;\theta)$.

The E-step of the EM algorithm is to find $q(A)$ that tightens the lower bound. This can be achieved by setting $q(A)$ to $l(A \mid S;\theta)$, since the KL divergence will become zero. Given that

$$l(A \mid S;\theta) = \prod_v l(a_v \mid S;\theta),$$

and

$$\begin{aligned} & l(a_v = k \mid S;\theta) \\ \propto & l(a_v = k, S;\theta) \\ \propto & \exp(\eta \cdot {\mathbf{s}_{v,k}}^\top \mathbf{r}_k), \end{aligned}$$

the optimal $q(A)$ that tightens the lower bound can be written in the following:

$$q(a_v = k) \propto \exp(\eta \cdot {\mathbf{s}_{v,k}}^\top \mathbf{r}_k).$$

This proves that

$$l_{v,k}(t) = \frac{\exp\left(\eta \cdot {\mathbf{s}_{v,k}}^\intercal \mathbf{r}_{u,k}(t-1)\right)}{\sum_{k'=1}^{K} \exp\left(\eta \cdot {\mathbf{s}_{v,k'}}^\intercal \mathbf{r}_{u,k'}(t-1)\right)}$$

will be performing the E-step, and that

$$l_{v,k} = q(a_v = k) = l(a_v = k \mid S;\theta).$$

After every E-step, the EM algorithm performs an M step to maximize the lower bound $L(\theta, q)$ w.r.t. θ, with $q(A)$ fixed to the value found in the E-step. Note that we have

$$\frac{\partial L(\theta, q)}{\mathbf{r}_k} = {\mathbf{r}_k}^\top \left(\mathbf{s}_{u,k} + \sum_v l_{v,k} \mathbf{s}_{v,k} \right).$$

We optimize $\mathbf{r}_k$ via setting $\frac{\partial L(\theta,q)}{\partial \mathbf{r}_k}$ to zero, and it turns out that the optimal $\mathbf{r}_k$ can be expressed exactly as follows,

$$\mathbf{r}_{u,k}(t) = \frac{\mathbf{s}_{u,k} + \sum_{v:(u,v)\in E} \; l_{v,k}(t) \; \mathbf{s}_{v,k}}{1 + \sum_{v:(u,v)\in E} \; l_{v,k}(t),}$$

which is in fact performing the M-step.

Let $q^{(t)}(A)$ and $\theta^{(t)}$ be the result of the t^{th} E-step and the t^{th} M-step, respectively, then

$$\begin{aligned} & \ln l(S;\theta^{(t-1)}) \\ =& L(\theta^{(t-1)}, q) + D_{\text{KL}}\left(q \| l_{\theta^{(t-1)}}\right) \\ =& L(\theta^{(t-1)}, q^{(t)}) \\ \leq& L(\theta^{(t)}, q^{(t)}) \\ \leq& L(\theta^{(t)}, q^{(t)}) + D_{\text{KL}}\left(q^{(t)} \| l_{\theta^{(t)}}\right) \\ =& \ln l(S;\theta^{(t)}). \end{aligned}$$

Thus the likelihood will increase monotonically, at the same time being upper-bounded by zero. Therefore, the algorithm converges, which completes the proof. □

B. Additional Experimental Settings

B.1. Dataset Preprocessing

We split the whole dataset into training set, validation set and test set according to the ratio of 8:1:1. Particularly for Amazon dataset which is based on explicit ratings, we binarize it by labeling ratings higher than or equal to 4 as 1, and only keep those users who write at least 5 reviews. Since the connections are not originally provided in Amazon dataset, we utilize the categories of the items bought by each user to simulate a social network

$$G = (V, E).$$

Concretely, we calculate the preference vector

$$p_u = [p_u^1; p_u^2; \cdots; p_u^k]$$

for user u, where

$$p_u^i = \sum_{c_j = i} x_{u,j}$$

denotes the summation of user u's preference towards item i under category i, i.e., the preference of user u over all the items under category i.

Then we use the cosine similarity between p_u and p_v to approximate the affinity between user u and v. We add an edge

$$(u, v) \in E,$$

if and only if

$$consine(u, v) > \gamma,$$

where γ is a parameter controlling the edge density of graph G.

B.2. Explainability for Recommendation

We in detail illustrate our strategy to retrieve items in the representation space. Let us assume that $\mathbf{y}_*$ is the original representation, which can be either the item representation $\mathbf{m}_i$ or a component of the user representation $\mathbf{z}_u^{(i)}$, and that prototype k_* is the prototype closest to $\mathbf{y}_*$.

We then determine A consecutive intervals

$$(a_i, a_{i+1}], i = 1, 2, \cdots, A$$

for the j^{th} dimension of $\mathbf{y}_*$ such that when $y_{*,j}$ is altered within the range

$$(a_1, a_{A+1}],$$

the prototype assigned, i.e., k_*, remains unchanged. In addition, we ensure that approximately the same number of items within prototype k_* will fall into each interval. Finally, we derive A items $\{i_t\}_{t=1}^A$ by maximizing the following objective:

$$\sum_{1 \le t \le A} e^{\frac{\text{COSINE}(\mathbf{y}_{i_t,-j}, \mathbf{y}_{*,-j})}{\tau}} + \gamma \cdot \sum_{1 \le t_1 < t_2 \le A} e^{\frac{\text{COSINE}(\mathbf{y}_{i_{t_1},-j}, \mathbf{y}_{i_{t_2},-j})}{\tau}},$$

where

$$\mathbf{y}_{i,-j} = [y_{i,1}; y_{i,2}; \cdots; y_{i,j-1}; y_{i,j+1}; \cdots; y_{i,d}] \in \mathcal{R}^d.$$

Each item i_t is chosen from the t^{th} interval, i.e.,

$$y_{i^t,j} \in (a_t, a_{t+1}]$$

and is within prototype k_*. The maximization is solved using beam search.

C. Full Tables for Experimental Results

We show the full experimental results of recommendation accuracy in Table 5 and the full ablation studies on the effectiveness of the Kernighan-Lin (KL) algorithm as well as curriculum subgraph weighing strategy in Table 8.

Dataset	Method	Metric		
		NDCG@100	Recall@20	Recall@50
Lastfm	Diffnet	0.26318(±0.00552)	0.22919(±0.00559)	0.34557(±0.00652)
	LightGCN	0.28691(±0.00573)	0.24333(±0.00559)	0.36899(±0.00650)
	MHCN	**0.32702(±0.00597)**	0.29121(±0.00595)	0.41715(±0.00674)
	SEPT	0.32216(±0.00588)	0.28305(±0.00593)	0.41141(±0.00673)
	DISGCN	0.28555(±0.00478)	0.28092(±0.00548)	0.41243(±0.00655)
	CurCoDis	0.30714(±0.00519)	**0.30172(±0.00607)**	**0.43236(±0.00676)**
	Improvement	-	3.61%	3.65%
Yelp	Diffnet	0.08594(±0.00122)	0.08638(±0.00187)	0.15670(±0.00245)
	LightGCN	0.09857(±0.00130)	0.09656(±0.00196)	0.17686(±0.00255)
	MHCN	0.11114(±0.00142)	0.11384(±0.00212)	0.19489(±0.00265)
	SEPT	0.10695(±0.00136)	0.10995(±0.00207)	0.19243(±0.00264)
	DISGCN	0.10329(±0.00232)	0.11803(±0.00409)	0.20128(±0.00537)
	CurCoDis	**0.11191(±0.00134)**	**0.12846(±0.00223)**	**0.21820(±0.00279)**
	Improvement	0.69%	8.84%	8.41%
Amazon	Diffnet	0.04745(±0.00420)	0.06325(±0.00714)	0.10538(±0.00906)
	LightGCN	0.07470(±0.00557)	0.09335(±0.00850)	0.14926(±0.01053)
	MHCN	0.07237(±0.00533)	0.08289(±0.00818)	0.14603(±0.01049)
	SEPT	0.04336(±0.00369)	0.06047(±0.00711)	0.10792(±0.00923)
	DISGCN	0.07046(±0.00322)	0.09964(±0.00809)	0.16245(±0.01140)
	CurCoDis	**0.08047(±0.00513)**	**0.11665(±0.00953)**	**0.19126(±0.01173)**
	Improvement	7.72%	17.1%	17.7%
Epinion	Diffnet	0.04334(±0.00069)	0.04709(±0.00118)	0.08448(±0.00155)
	LightGCN	0.05532(±0.00080)	0.06199(±0.00134)	0.10698(±0.00173)
	MHCN	0.06070(±0.00086)	0.06612(±0.00137)	0.11309(±0.00175)
	SEPT	0.06557(±0.00089)	0.07502(±0.00147)	0.12615(±0.00186)
	DISGCN	0.05680(±0.00370)	0.06760(±0.00804)	0.11839(±0.00804)
	CurCoDis	**0.07431(±0.00095)**	**0.08908(±0.00160)**	**0.14578(±0.00199)**
	Improvement	13.3%	18.7%	15.6%

Table 4. Full table of Comparisons between our proposed CurCoDis model and baselines on all four datasets, with bold font denoting the best approach. The relative improvement of our model over the best performing baseline is recorded in row *Improvement* for each dataset.

Amazon				
Metric	**Method**	**Metric@5**	**Metric@10**	**Metric@15**
Recall	Diffnet	0.02563(±0.00361)	0.02228(±0.00352)	0.01664(±0.00332)
	LightGCN	0.04699(±0.00504)	0.04160(±0.00496)	0.03689(±0.00485)
	MHCN	0.04140(±0.00461)	0.03641(±0.00451)	0.02855(±0.00429)
	SEPT	0.02610(±0.00377)	0.02191(±0.00366)	0.01834(±0.00355)
	DISGCN	0.05290(±0.00139)	0.04577(±0.00136)	0.03591(±0.00129)
	CurCoDis	**0.06246(±0.00582)**	**0.05721(±0.0057)**	**0.04779(±0.00561)**
	Improvement	18.1%	25.0%	29.5%
NDCG	Diffnet	0.05268(±0.00656)	0.04031(±0.00577)	0.02352(±0.00447)
	LightGCN	0.08481(±0.00817)	0.06507(±0.00722)	0.05119(±0.00650)
	MHCN	0.08005(±0.00823)	0.06547(±0.00741)	0.04120(±0.00591)
	SEPT	0.05236(±0.00659)	0.03666(±0.00553)	0.02646(±0.00480)
	DISGCN	0.10407(±0.00239)	0.07723(±0.00208)	0.04775(±0.00165)
	CurCoDis	**0.11384(±0.00944)**	**0.09481(±0.00875)**	**0.06537(±0.00734)**
	Improvement	9.39%	22.8%	27.7%

Table 5. Extra experiments on Amazon under the same experimental settings in terms of several commonly used evaluation metrics, i.e., Recall@K and NDCG@K, where K is set to 5, 10, 15. We can observe that our proposed CurCoDis still outperforms other baselines.

Value of ρ	NDCG@100		Recall@20		Recall@50	
	vanilla	CurCoDis	vanilla	CurCoDis	vanilla	CurCoDis
$\rho = 0.5$	0.29920(±0.00518)	**0.30230(±0.00520)**	**0.30029(±0.00604)**	0.29889(±0.00608)	**0.42818(±0.00682)**	0.42435(±0.00673)
$\rho = 1$	**0.30341(±0.00524)**	0.30294(±0.00521)	0.29750(±0.00600)	**0.30026(±0.00609)**	0.42085(±0.00681)	**0.42734(±0.00683)**
$\rho = 2$	**0.30271(±0.00515)**	0.30055(±0.00517)	**0.30151(±0.00608)**	0.29865(±0.00608)	0.42108(±0.00669)	**0.42443(±0.00672)**
$\rho = 4$	0.29826(±0.00516)	**0.29920(±0.00517)**	0.29048(±0.00596)	**0.29324(±0.00599)**	**0.42359(±0.00676)**	0.42224(±0.00671)

Table 6. Full table of comparisons between CurCoDis and vanilla, which is trained using the whole social graph.

Value of ρ	NDCG@100		Recall@20		Recall@50	
	vanilla	CurCoDis	vanilla	CurCoDis	vanilla	CurCoDis
$\rho = 0.5$	0.25863(±0.00506)	**0.30230(±0.00520)**	0.25074(±0.00590)	**0.29889(±0.00608)**	0.36875(±0.00679)	**0.42435(±0.00673)**
$\rho = 1$	0.25418(±0.00503)	**0.30294(±0.00521)**	0.24912(±0.00588)	**0.30026(±0.00609)**	0.36448(±0.00675)	**0.42734(±0.00683)**
$\rho = 2$	0.25534(±0.00502)	**0.30055(±0.00517)**	0.24840(±0.00584)	**0.29865(±0.00608)**	0.37040(±0.00675)	**0.42443(±0.00672)**
$\rho = 4$	0.25533(±0.00501)	**0.29920(±0.00517)**	0.24809(±0.00590)	**0.29324(±0.00599)**	0.36605(±0.00674)	**0.42224(±0.00671)**

Table 7. Full table of comparisons between CurCoDis and vanilla, which substitutes the social dynamic routing with classic GCN.

Dataset	Method	Metric		
		NDCG@100	Recall@20	Recall@50
Lastfm	vanilla	0.28634(±0.00501)	0.28315(±0.00589)	0.41080(±0.00668)
	Kernighan-Lin (KL)	0.30343(±0.00521)	0.29830(±0.00607)	0.43100(±0.00676)
	KL+Cur (CurCoDis)	**0.30714(±0.00519)**	**0.30172(±0.00607)**	**0.43236(±0.00676)**
	KL Improvement	5.97%	5.35%	4.92%
	Curriculum Improvement	1.22%	1.15%	0.32%
	Overall Improvement	7.26%	6.56%	5.25%
Yelp	vanilla	0.10833(±0.00132)	0.12382(±0.00220)	0.20740(±0.00272)
	Kernighan-Lin (KL)	0.11146(±0.00133)	0.12817(±0.00223)	0.21575(±0.00277)
	KL+Cur (CurCoDis)	**0.11191(±0.00134)**	**0.12846(±0.00223)**	**0.21820(±0.00279)**
	KL Improvement	2.89%	3.51%	4.03%
	Curriculum Improvement	0.40%	0.23%	1.14%
	Overall Improvement	3.30%	3.75%	5.21%
Amazon	vanilla	0.05959(±0.00444)	0.08947(±0.00846)	0.14334(±0.01036)
	Kernighan-Lin (KL)	0.07801(±0.00366)	0.10501(±0.00915)	0.18114(±0.01145)
	KL+Cur (CurCoDis)	**0.08047(±0.00513)**	**0.11665(±0.00953)**	**0.19126(±0.01173)**
	KL Improvement	30.3%	17.4%	26.4%
	Curriculum Improvement	3.15%	11.1%	5.59%
	Overall Improvement	35.0%	30.4%	33.4%
Epinion	vanilla	0.07343(±0.00095)	0.08632(±0.00158)	0.14108(±0.00196)
	Kernighan-Lin (KL)	0.07402(±0.00095)	0.08781(±0.00159)	0.14294(±0.00197)
	KL+Cur (CurCoDis)	**0.07431(±0.00095)**	**0.08908(±0.00160)**	**0.14578(±0.00199)**
	KL Improvement	0.80%	1.73%	1.32%
	Curriculum Improvement	0.39%	1.45%	1.99%
	Overall Improvement	1.20%	3.20%	3.33%

Table 8. Full results of ablation study. The relative improvement of Kernighan-Lin (KL) over vanilla, KL+Cur (CurCoDis) over Kernighan-Lin (KL) and KL+Cur (CurCoDis) over vanilla are presented in row *KL Improvement*, *Curriculum Improvement* and *Overall Improvement* respectively for each dataset in terms of different evaluation metrics.