Rethinking Collaborative Metric Learning: Toward an Efficient Alternative without Negative Sampling

Shilong Bao, Qianqian Xu*, *Senior Member, IEEE*, Zhiyong Yang,
Xiaochun Cao, *Senior Member, IEEE*, and Qingming Huang*, *Fellow, IEEE*

Abstract—The recently proposed Collaborative Metric Learning (CML) paradigm has aroused wide interest in the area of recommendation systems (RS) owing to its simplicity and effectiveness. Typically, the existing literature of CML depends largely on the *negative sampling* strategy to alleviate the time-consuming burden of pairwise computation. However, in this work, by taking a theoretical analysis, we find that negative sampling would lead to a biased estimation of the generalization error. Specifically, we show that the sampling-based CML would introduce a bias term in the generalization bound, which is quantified by the per-user *Total Variance* (TV) between the distribution induced by negative sampling and the ground truth distribution. This suggests that optimizing the sampling-based CML loss function does not ensure a small generalization error even with sufficiently large training data. Moreover, we show that the bias term will vanish without the negative sampling strategy. Motivated by this, we propose an efficient alternative without negative sampling for CML named *Sampling-Free Collaborative Metric Learning* (SFCML), to get rid of the sampling bias in a practical sense. Finally, comprehensive experiments over seven benchmark datasets speak to the supriority of the proposed algorithm.

Index Terms—Recommendation System, Collaborative Metric Learning, Negative Sampling, Machine Learning

1 Introduction

Nowadays, the explosion of Internet data poses an inevitable challenge of how to help users access their desirable information (say which book/news to read, which resturtant to eat and which anime to see, etc.). Consequently, *recommendation system* (short for RS) [1], [2], [3], [4], [5], [6], [7], [8] has recently emerged as a major solution and has broad applications in modern Internet enterprises, such as Amazon, Alibaba and Facebook.

The main purpose of RS is to leverage user preference prediction based on the historical data produced from user-item interactions. In practice, such interactions often exist as implicit feedback [9], [10] where no explicit ratings and only actions are provided (such as browses, clicks, purchases, etc.). It is well-known that implicit feedback only contains indirect records from user behavior, without explicit knowledge about their negative intentions. This poses a great challenge to RS-targeted machine learning algorithms and brings about a wave of relevant studies [11], [12], [13]. The vast majority of such work follows a standard paradigm known as One-Class Collaborative Filtering (OCCF) [14], [15], [16], [17], [18], [19], [20], where the items not being observed are usually assumed to be of less interest for a given user.

Over the past decade, Matrix Factorization (MF)-based algorithms are one of the most representative techniques among the OCCF community, where the user's preference toward an item is captured by an inner product between their latent factors, such as [21], [22], [23]. Unfortunately, some literature pointed out that the inner product violates the triangle inequality, which may lead to sub-optimal performance of recommendation [12], [24]. To tackle this problem, *Collaborative Metric Learning* (CML) [24] presents a novel OCCF framework via incorporating the strengths of *metric learning* [25], [26] into the *Collaborative Filtering* (CF) framework, and achieves a reasonable performance on a wide range of RS benchmark datasets. Nonetheless, the full-batch objective function of CML is featured with an $\mathcal{O}(\sum_{i=1}^{M} n_i^+ n_i^-)$ complexity, where M is the number of users and $n_i^+(n_i^-)$ is the number of positive (unobserved)

* *Corresponding authors*

- *Shilong Bao is with State Key Laboratory of Information Security (SKLOIS), Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China, and also with School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China (email: baoshilong@iie.ac.cn).*
- *Qianqian Xu is with the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China, (email: xuqianqian@ict.ac.cn).*
- *Zhiyong Yang is with the School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 101408, China (email: yangzhiyong21@ucas.ac.cn).*
- *Xiaochun Cao is with State Key Laboratory of Information Security (SKLOIS), Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China, also with School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China(email: caoxiaochun@iie.ac.cn).*
- *Qingming Huang is with the School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 101408, China, also with the Key Laboratory of Big Data Mining and Knowledge Management (BDKM), University of Chinese Academy of Sciences, Beijing 101408, China, also with the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China, and also with Peng Cheng Laboratory, Shenzhen 518055, China (e-mail: qmhuang@ucas.ac.cn).*

items for user u_i. Practically, CML adopts the standard negative sampling strategy [27], [28], [29], [30] to improve the efficiency, where merely a limited amount of unobserved items are selected to optimize the model for a given user. Hereafter, a series of the related studies has been carried out to improve the performance of CML, such as popularity-based [31], [32], [33], two-stage [33], and hard mining [34], [35], [36], [37] negative sampling strategies, translation-based CML [38], [39] and co-occurrence embedding regularized metric learning (CRML) [40]. *Without loss of generality, since the existing methods of CML cannot bypass the negative sampling in the training phase, we call this kind of algorithms sampling-based CML in this paper.*

Different from the sampling-based CML, we argue that the negative sampling strategy essentially alters the intrinsic distribution of the training data, leading to a biased estimation for the expected loss function over the unseen data. Therefore, in this paper, we are interested in the following question:

Could the sampling-based CML guarantee a good generalization performance?

In search of an answer to the question, we provide a systematic analysis of the generalization ability of the CML framework, based on the Rademacher Complexity-based arguments. The major challenge is that the CML framework adopts a pairwise loss function to capture the preference comparison between positive and the unobserved items, which could not be expressed as a sum of independently identically distributed (i.i.d) loss term. As a result, the standard theoretical arguments [41] are no longer available for our task. Therefore, we first extend the standard symmetrization regime and the definition of Rademacher complexity [42], [43] according to a specifically designed symmetrization strategy.

On top of the proposed complexity measure, we prove that the sampling-based CML would introduce an extra per-user *Total Variance* (TV) term in its generalization upper bound, which reflects the discrepancy between the sampling-strategy-induced distribution and the ground truth distribution. This implies that minimizing the sampling-based CML over the training data cannot ensure a small generalization error when the induced distribution behaves away from its ground truth. Meanwhile, we also demonstrate that the biased term will vanish in the sampling-free version. Therefore, in order to obtain a reasonable performance, we propose to learn CML in a sampling-free manner to get rid of the bias. However, as we mentioned above, we must face the heavy computational burden with a non-sampling favor.

Facing this challenge, we start the first exploration to develop an efficient alternative for CML without negative sampling. Specifically, by posing a ℓ_2 hyper-sphere constraint over the embedding space, we figure out the closed connection between CML and another technique called AUC optimization [44], [45]. Motivated by this fact, we construct an acceleration method to evaluate the full sample loss and gradient on top of a semi-regular comparison graph.

Finally, a systematic empirical study is conducted on seven real-world RS datasets, including MovieLens-100k, CiteULike, MovieLens-1m, Steam-200k, Anime, MovieLens-20m and Amazon-Book, the results of which consistently speak to the efficacy of our proposed method.

In a nutshell, the main contributions are summarized as follows:

- The pairwise formulation of the CML loss function makes it impossible to employ standard Rademacher complexity-based measures to analyze the generalization ability of CML framework. To address this issue, we propose an extended Rademacher complexity with the strength of a novel symmetrization scheme.
- According to the proposed complexity measure, we start an early trial to present theoretical analyses of the generalization ability of the CML framework with (without) the help of negative sampling strategy, which reflects the biased issue of the sampling-based process and suggests the benefit of the sampling-free manner.
- Motivated by the theoretical findings, we propose an efficient alternative called *Sampling-Free Collaborative Metric Learning* (SFCML) to deal with the full samples based CML, where the loss and gradient evaluations are accelerated with a graph-based reformulation.

The rest of the paper is organized as follows. Sec.2 presents a review of the most related studies. Sec.3 presents a brief introduction of the CML framework. In Sec.4, we propose the theoretical analysis for sampling-based and sampling-free CML. In Sec.5, the SFCML method is proposed. In Sec.6, we conduct empirical studies to show the efficacy of our proposed algorithm on seven real-world RS datasets. Finally, Sec.7 presents a concluding remark about the work.

2 Prior Art

In this section, we briefly review the closely related studies along with our main topic, including one-class collaborative filtering and the existing solutions without negative sampling.

2.1 One-Class Collaborative Filtering

In many real-world applications, the vast majority of interactions are implicitly expressed by users' behaviors, e.g, downloads of movies, clicks of products and browses of news. In order to develop RS from such implicit feedback, researchers usually formulate the recommendation task as the *One-Class Collaborative Filtering* (OCCF) problem [14], [15], [16], [19], [46], [47], [48], [49]. The existing OCCF algorithms can be mainly categorized into two fashions: a) Pointwise fashion: The goal of the pointwise-based algorithms is to recover the missing signals by minimizing the error between the estimated and observed implicit feedback [20], [50]. b) Pairwise fashion: Specifically, the pairwise-based solutions aim to construct a system that the observed items should be ranked higher than the unobserved items [51], [52].

Matrix Factorization (MF) based Algorithm. Over the past decades, the Matrix Factorization (MF)-based algorithms are one of the most classical OCCF solutions [48], [53]. The key idea of MF is to represent each user and item as a latent factor to recover the missing entries in a unified space. Typically, a user's preference toward an item is represented as

an inner product between their latent factors. For example, under the pointwise setting, [53] proposes an item-oriented MF method with implicit feedback. Neural Collaborative Filtering (NCF) [27] develops a general framework that unifies the MF and the neural networks together, and then regards the recommendation task as a regression problem. In addition, a pairwise-based algorithm is developed for MF-based recommendation [54]. However, some literature pointed out that the inner product violates the triangle inequality, which may lead to sub-optimal performance [12], [24].

Collaborative Metric Learning based Algorithm. Recently, there rises a new trend in the community to alleviate the intrinsic problem of MF-based algorithms [27], [55], [56]. Among them, a popular idea is to borrow the strengths of metric learning [57], [58], due to its simplicity and effectiveness [39], [59], [60], [61], [62]. Noteworthy is the work known as Collaborative Metric Learning (CML) [24], which is the first successful integration of metric learning and CF. Generally speaking, the idea of CML is to represent users and items in a unified Euclidean space, where the proximity between users and items naturally captures the preference. As a typical trait, CML employs a negative sampling strategy [4], [63], [64], [65], [66], [67] to generate the contrastive pairs, which could mitigate the high computational burden of pairwise learning. Thereafter, many efforts have been made to improve the recommendation performance of CML. The relevant studies fall into two camps. The first camp employs a model-oriented strategy, where the CML model is enriched with either more complicated structures or side information. Typically, inspired by the knowledge translation mechanism in knowledge graph embedding, [38], [39] propose to learn an exclusive latent relation vector for each user-item interaction to explore the preference of each user and item more precisely. Co-occurrence embedding Regularized Metric Learning (CRML) [40] presents an effective approach that optimizes the representations of both users and items by considering the global statistical information of user-user and item-item pairs. Another camp of the related work attempts to improve the effectiveness of the negative sampling process. For example, Popularity-based negative sampling (PopS) [31], [33] proposes to sample negative items based on their popularity to minimize the pairwise ranking loss. Tran et al. [33] propose a two-stage negative sampling (2stS), which first samples an items' candidate according to their popularity and then selects a negative item from the candidate based on their inner product with positive items. In addition, hard sample mining [24], [34], [36], [37] manners are also adopted to obtain negative items. However, in this paper, we prove that negative sampling would introduce a bias term in the generalization bound (shown in Sec.4), such that optimizing CML with the negative sampling strategies may fail to obtain a reasonable generalization performance. Consequently, different from both directions, we will study how to perform CML without the help of negative sampling.

2.2 Learning without negative sampling

Recently, some researchers have pointed out that negative sampling results in insufficient training of model and thus leads to performance degradation [53], [68], [69], [70]. This is because it inevitably discards some informative samples during the training process. On the contrary, non-sampling could avoid this problem since taking all training data into account directly induces a better solution. However, the main bottleneck is training efficiency for the sampling-free manner. Therefore, a growing number of researchers attempt to alleviate the efficiency issue of learning with a non-sampling paradigm in different tasks. He et al. [53] design an efficient element-wise alternating least squares (eALS) with non-uniform missing data. Yuan et al. [68] present a generic and fast batch gradient descent optimizer f_{BGD} that can learn embedding from the whole training dataset without a negative sampling strategy. Therefore, it can naturally be applied to the factorization-based CF recommendation. [71] proposes a general matrix factorization algorithm, i.e., Efficient Neural Matrix Factorization (ENMF) without non-sampling, which shows both effective and efficient performance. In order to sufficiently capture collaborative information among users, items and entities in knowledge graph (KG), a novel jointly non-sampling learning model for KG enhanced recommendation (JNSKR) [69] is proposed. In addition, recently, Chen et al. [72] study the flaws in heterogeneous collaborative filtering (HCF) and develop a novel and state-of-the-art non-sampling learning framework for the recommendation, i.e., Efficient Heterogeneous Collaborative Filtering (EHCF) .

The recent development of recommendations with implicit feedback has revolutionary motivated the advanced studies of learning without negative sampling. Nonetheless, optimizing CML in a sampling-free manner has long been left behind this wave of revolution. Specifically, **the existing non-sampling algorithms are designed for a factorization-based RS model or a pointwise loss function, which is not applicable for the CML framework based on the pairwise ranking loss**. On stark contrary, different from the above non-sampling recommendation models, in this work, we study how to learn CML in a sampling-free manner efficiently. Most importantly, we also present the issue of negative sampling from the theoretical perspective, which has not been explored in previous work.

3 Preliminary

In this section, we first introduce some notations and the task of recommendation learning from implicit feedback. Then, the concepts of CML framework are briefly presented. For clarity, a summary of key notations and their corresponding descriptions throughout this paper are listed in Tab.1.

3.1 Task Definition

In this work, our model is developed by the *implicit feedback* signals (e.g., clicks, thump ups, likes, etc.). Given the historical user-item interaction records, our primary aim is to infer the preference of users and then recommend unseen items that he/she is most likely to be interested in. Mathematically, assume that there are M users and N items in practical recommendation system, denoted as $\mathcal{U} = \{u_1, u_2, \ldots, u_M\}$ and $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$, respectively. Let $\mathcal{S}_i^+$ be the set of

TABLE 1: A summary of key notations and descriptions in ths work.

Notations	Descriptions
M	number of users
N	number of items
$\mathcal{U}$	the set of users
$\mathcal{V}$	the set of items
$\mathcal{S}_i^+$	the interacted items of u_i
v_*^+	the item contained in $\mathcal{S}_i^+$
v_*^-	the item not belonging to $\mathcal{S}_i^+$
y_{ij}	user u_i's preference toward item v_j
f	learned score function
d	the dimension of space
$\boldsymbol{W}_u$	learned transformation weight of users
$\boldsymbol{W}_v$	learned transformation weight of items
$\boldsymbol{e}_{u_i}$	embeddings of user u_i
$\boldsymbol{e}_{v_j}$	embeddings of item v_j
$\boldsymbol{d}(i,j)$	Euclidean distance between u_i and v_j
$\boldsymbol{u}_i$	one-hot encoding of user u_i
$\boldsymbol{v}_j$	one-hot encoding of item v_j
$\ell_{hinge}^{(i)}$	Hinge loss function
$\ell_{sq}^{(i)}$	Square loss function
n_i^+	number of observed items for user u_i
n_i^-	number of unobserved items for user u_i
$\tilde{\mathbb{P}}^{(i)}$	the sampling-strategy-induced distribution
$\hat{\mathbb{P}}^{(i)}$	the ground-truth distribution
$\boldsymbol{L}^{(i)}$	the Laplacian matrix of user u_i
$\mathcal{G}^{(i)}$	the graph of user u_i
$\mathcal{D}^{(i)}$	the adjacent matrix of user u_i
$\mathcal{S}^{(i)}$	the vertex set of $\mathcal{G}^{(i)}$
$\mathcal{E}^{(i)}$	the edge set of $\mathcal{G}^{(i)}$
$\nabla_{\boldsymbol{e}_{u_i}}$	gradient of variable $\boldsymbol{e}_{u_i}$
$\nabla_{\boldsymbol{W}_v}$	gradient of variable $\boldsymbol{W}_v$
$\boldsymbol{y}^{(i)}$	vector of u_i's preferences
$\boldsymbol{f}^{(i)}$	user u_i's score vectors toward all items
$\mathcal{H}_R$	the hypothesis space
$\|\boldsymbol{W}\|_*$	induced 2 norm of matrix $\boldsymbol{W}$
$\xi_t(\boldsymbol{W}_v)$	the t-th largest singular value of $\boldsymbol{W}_v$

interacted items of user u_i. Then, the interaction record of user u_i toward item v_j is defined as:

$$y_{ij} = \begin{cases} 1, & if \;\; v_j \in \mathcal{S}_i^+ \wedge v_j \in \mathcal{V}; \\ 0, & else. \end{cases} \tag{1}$$

where $y_{ij} = 1$ indicates the observed/positive actions with item v_j, and $y_{ij} = 0$ could imply user that dislikes or is unaware of the existence of item v_j [27], [39].

Motivated by the preference paradigm of OCCF, here for observed interactions, we assume that the users tend to have higher preferences for these interacted items than other unobserved items. Consequently, given a target user $u_i \in \mathcal{U}$ and his/her interaction records, the main task of recommendation is to seek a preference score function $f(v_j|u_i)$ out of a predefined hypothesis class $\mathcal{H}_R$ containing the candidate models. Finally, given a proper choice of f, the RS system will then recommend the items having the top-K ranking score.

3.2 Preference Consistency Model

Now we start to introduce how the preference score functions are formulated in CML. Borrowing the wisdom from metric learning, in order to capture the preferences, a distance metric should be learned from the implicit feedback. On top of the metric space, the users' preferences could be naturally specified by the value of distance through the learned metric.

To this end, we first project each user and item in a joint Euclidean space through the following lookup transformations [73], [74], [75]:

$$\begin{aligned} \boldsymbol{e}_{u_i} &= \boldsymbol{W}_u^\top \boldsymbol{u}_i, \\ \boldsymbol{e}_{v_j} &= \boldsymbol{W}_v^\top \boldsymbol{v}_j, \end{aligned} \tag{2}$$

where $\boldsymbol{e}_{u_i} \in \mathbb{R}^d$ and $\boldsymbol{e}_{v_j} \in \mathbb{R}^d$ are the embeddings of user u_i and item v_j, respectively; d is the dimension of space; $\boldsymbol{W}_u \in \mathbb{R}^{M \times d}$, $\boldsymbol{W}_v \in \mathbb{R}^{N \times d}$ are the learned transformation weight; $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$ are two different one-hot encodings in which the nonzero elements in $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$ correspond to the index of a particular user u_i and item v_j, respectively.

If the metric space is Euclidean, the value of metric between user u_i and item v_j can be intuitively measured by their distance:

$$\boldsymbol{d}(i,j) = ||\boldsymbol{e}_{u_i} - \boldsymbol{e}_{v_j}||^2.$$

Subsequently, in order to capture the preference of user u_i, one should push away the observed items and unobserved items through the lens of distance constraints. Therefore, if user u_i likes item v_j (i.e., $y_{ij} = 1$), a small value for $\boldsymbol{d}(i,j)$ should be assigned. If the opposite is the case (i.e., $y_{ij} = 0$), we then hope a large $\boldsymbol{d}(i,j)$. Mathematically, the following inequality should be held to reflect the relative preference of u_i toward different items v_j and v_k:

$$\begin{cases} \boldsymbol{d}(i,j) < \boldsymbol{d}(i,k), & v_j^+ \in \mathcal{S}_i^+, v_k^- \notin \mathcal{S}_i^+; \\ \boldsymbol{d}(i,j) > \boldsymbol{d}(i,k), & v_j^- \notin \mathcal{S}_i^+, v_k^+ \in \mathcal{S}_i^+ \end{cases} \tag{3}$$

where, with a slight abuse of notation, we let v_*^+ represent the item involved in $\mathcal{S}_i^+$ and that for $\mathcal{S}_i^-$ is denoted by v_*^- ($*$ represents any item here).

Motivated by Eq.(3), we only need to control the relative preference rather than their magnitude, since scaling both sides of inequalities does not change the partial order. Hence, the CML framework is to minimize the following pairwise empirical risk to reflect such preference consistency:

$$\begin{aligned} \hat{\mathcal{R}}_{\mathcal{S}}^{\text{cml}}(f) = \frac{1}{M} \sum_{u_i \in \mathcal{U}} \frac{1}{n_i^+ n_i^-} \sum_{j=1}^{n_i^+} \sum_{k=1}^{n_i^-} \ell^{(i)}(v_j^+, v_k^-), \\ s.t. \;\; ||\boldsymbol{e}_{u_i}||^2 \le R, \quad ||\boldsymbol{e}_{v_j}||^2 \le R, \;\; u_i \in \mathcal{U}, v_j \in \mathcal{V}. \end{aligned} \tag{4}$$

In respect of Eq.(4), we have the following explanations. Denote the number of observed (unobserved) items for a given user u_i as $n_i^+(n_i^-)$. Since the overall amount of items in the system is fixed, we naturally come to the fact that $n_i^+ + n_i^- = N$, where N is the total number of the items. Moreover, the user/item embeddings are constrained within a ℓ_2 ball with radius R to ensure a normalization. Finally, $\ell^{(i)}(v_j^+, v_k^-)$ is a differentiable ranking loss which is often set as the hinge loss

$$\ell_{hinge}^{(i)}(v_j^+, v_k^-) = \textbf{max}\left(0, \lambda + \boldsymbol{d}(i,j) - \boldsymbol{d}(i,k)\right),$$

where $\lambda > 0$ is the safe margin to ensure sufficient partition across different types of items.

Different from the traditional inner-product-based OCCF framework, CML induces the triangle inequality [76], [77] by means of the Euclidean metric. As a result, the learned embedding can automatically cluster 1) co-liked items from

the same user and 2) co-liked items of similar users, which suggests a better global consistency of the preference ranking [24], [59].

Finally, when the training is completed, one can easily leverage $f(v_j|u_i) = -\boldsymbol{d}(i,j)$ to calculate the rank of each item v_j and generate recommendations for user u_i.

3.3 Learning with Negative Sampling

Despite the strength of metric learning, CML leads to a heavy computational burden: *In Eq.(4), every item that the user interacted with needs to be paired with all remaining items, which induces a* $\mathcal{O}(\sum_{i=1}^{M} n_i^+ n_i^-)$ *time and space complexity.*

At present, to alleviate this situation, most of the existing literature usually resorts to the *negative sampling* strategy, i.e., sampling a few items from unobserved sets as negatives, such as uniform sampling [27], [28], [78], popular-based sampling [31], [32], [33], two-stage sampling technique [33] and hard sampling strategy [24], [34], [35], [36], [37].

Specifically, one usually samples U negative items for each positive user-item (u_i, v_j^+) interaction based on a predesigned sampling distribution. We could formulate such a process as modification toward the objective function in expectation. For user i, the negative samples are drawn from a sparse and discrete distribution $\tilde{\mathbb{P}}^{(i)}$, then the empirical expectation of the sampling-based CML could be rewritten as:

$$\tilde{\mathcal{R}}_{\mathcal{S}}^{\text{cml}}(f) = \frac{1}{M} \sum_{u_i \in \mathcal{U}} \sum_{j=1}^{n_i^+} \sum_{k=1}^{n_i^-} \tilde{\mathbb{P}}_{jk}^{(i)} \cdot \ell^{(i)}(v_j^+, v_k^-) \quad s.t. \left\|\boldsymbol{e}_{u_i}\right\|^2 \leq R, \quad \left\|\boldsymbol{e}_{v_j}\right\|^2 \leq R, \quad u_i \in \mathcal{U}, v_j \in \mathcal{V}, \tag{5}$$

where $\tilde{\mathbb{P}}_{jk}^{(i)} = \mathbb{P}(\boldsymbol{v}_j^+, \boldsymbol{v}_k^-)$ represents the probability that item v_k^- is sampled as a negative instance for v_j^+. The whole distribution for user i is then expressed in a compact form :

$$\tilde{\mathbb{P}}_j^{(i)} = [\tilde{\mathbb{P}}_{j1}^{(i)}, \tilde{\mathbb{P}}_{j2}^{(i)}, \ldots, \tilde{\mathbb{P}}_{jn_i^-}^{(i)}],$$

which is a sparse vector with U non-zero terms. Here, $\tilde{\mathbb{P}}_{jk}^{(i)} \neq 0$ only if item v_k^- is sampled as one of the negatives for v_j^+. Moreover, it is interesting to note that, the orginal objective function could be also regarded as a special case of the new formulation via setting $\mathbb{P}(\boldsymbol{v}_j^+, \boldsymbol{v}_k^-) \equiv \frac{1}{n_i^+ n_i^-} > 0$. In this sence, the negative sampling strategy introduces sparisity to the original distribution, which leads to a lighter $\mathcal{O}(\sum_{i=1}^{M} n_i^+ U)$ complexity than $\mathcal{O}(\sum_{i=1}^{M} n_i^+ n_i^-)$ (note that $U \ll n_i^-$).

Although the negative sampling schemes could reduce the computation burden of Eq.(4), **such strategies may cause some unexpected issues**, including

- The sampling distributions and the size of sampling (constant U) to a large extent determine the performance, which is generally difficult to choose. Moreover, this also makes the sampling-based CML unstable.
- Comparing Eq.(4) with Eq.(5), the sampling-based CML is a biased estimation of the original loss function. In other words, from the theoretical analysis in the next section (Sec.4), one can see that ***optimizing the sampling-based loss function will not necessarily lead to a small generalization error***.

4 Generalization Bounds for CML Framework

In this section, we provide a systematic theoretical discussion of the generalization ability of the CML framework. To do this, the basic notations and assumptions of theoretical analysis are first introduced. Subsequently, we extend the standard symmetrization regime and the definition of Rademacher Complexity. Finally, based on the proposed complexity arguments, we present the generalization bounds for the CML framework, including sampling-based and sampling-free manners.

Asymptotic Notations. In order to make our theoretical discussions more clear, we first provide some asymptotic notations that will be adopted throughout the generalization analysis.

- $x \lesssim y$ represents that there exists some universal constant $C > 0$ such that $x \leq Cy$.
- Similarly, $x \gtrsim y$ means that there exists some universal constant $C > 0$ such that $x \geq Cy$.
- $x \asymp y$ is equivalent to $y \lesssim x \lesssim y$.

Notably, other notations could be found in Tab.1.

4.1 Basic Assumptions on the Item Embeddings

Our analysis relies on two fundamental regularities of the item embedding matrix $\boldsymbol{W}_v \in \mathbb{R}^{N \times d}$, where we generally assume that $d \ll N$ in the practical RS.

Here, we need the basic notion of the induced matrix 2-Norm. Given a matrix $\boldsymbol{W}_v$, $\xi_1(\boldsymbol{W}_v) \geq \xi_2(\boldsymbol{W}_v) \geq, \cdots, \geq \xi_d(\boldsymbol{W}_v)$, are the singular values of $\boldsymbol{W}_v$. Then we have the following definition.

Definition 1. ***(Induced 2-Norm of a Matrix)***. *Let* $\|\boldsymbol{W}\|_*$ *be the induced 2-norm of a weight matrix* $\boldsymbol{W}$*. Then, the induced norm of* $\boldsymbol{W}$ *is defined as follows:*

$$\|\boldsymbol{W}\|_* = \max_{||\boldsymbol{x}||_2=1} \|\boldsymbol{W}\boldsymbol{x}\|_2 = \xi_1(\boldsymbol{W}). \tag{6}$$

Assumption 1. ***(Basic Assumptions)***. *We have the following assumptions:*

(a) ***Embedding Diversity Assumption:*** *Let* d *be the dimensionality of the embedding space, we assume that:*

$$\frac{\xi_1(\boldsymbol{W}_v)}{\xi_d(\boldsymbol{W}_v)} \asymp 1$$

(b) ***Embedding Capacity Assumption:*** *We assume that* d *is sufficiently large such that:*

$$(N)^{1/2} \lesssim d \ll N.$$

Remark 1. *We have the following remarks about the assumptions above.*

(a) Note that we call Assm.1-(a) the embedding diversity assumption, because when $\frac{\xi_1(\boldsymbol{W}_v)}{\xi_d(\boldsymbol{W}_v)} \asymp 1$*, the embedding matrix is of full column rank with each dimension exhibiting similar importance.*

(b) Note that, since $\left\|\boldsymbol{e}_{v_j}\right\|_2^2 \lesssim 1$, $\|\boldsymbol{W}_v\|_F^2 \lesssim N$*. Moreover,*

$$\|\boldsymbol{W}_v\|_F^2 = \xi_1^2 + \xi_2^2 + \cdots + \xi_d^2.$$

Then, we have

$$\|\boldsymbol{W}_v\|_* \lesssim \sqrt{\frac{n_i^+ + n_i^-}{d}} = \sqrt{\frac{N}{d}} \lesssim d^{1/2} \tag{7}$$

Overall, Assum.1 ensures that the learned embeddings are sufficiently informative to support a well-trained model.

4.2 The Rademacher Complexity Measure for CML Framework

According to the model constraints of CML, the user-item embeddings are chosen uniformly from the following embedding hypothesis space:

$$\mathcal{H}_R = \left\{ \boldsymbol{e} : \boldsymbol{e} \in \mathbb{R}^d, \ \|\boldsymbol{e}\|^2 \leq R \right\}, \tag{8}$$

where $\boldsymbol{e}_{u_i} \in \mathcal{H}_R, \ u_i \in \mathcal{U}$ and $\boldsymbol{e}_{v_j} \in \mathcal{H}_R, \ v_j \in \mathcal{V}$.

Based on the given hypothesis space, we will present a worse case generalization analysis to show that even the worst choice of the user-item embedding set has a reasonably small generalization error given a small training error. Following the standard learning theory arguments, such a bound relies on the Rademacher complexity measure of the given hypothesis $\mathcal{H}$. Traditionally, the Rademacher complexity is derived from the symmetrization technique as an upper bound for the largest deviation over a given hypothesis $\mathcal{H}$:

$$\mathop{\mathbb{E}}_{\mathcal{S}} \left[\sup_{f \in \mathcal{H}} \mathop{\mathbb{E}}_{\mathcal{S}} (\hat{\mathcal{R}}_{\mathcal{S}}) - \hat{\mathcal{R}}_{\mathcal{S}} \right].$$

Unfortunately, the standard argument of the symmetrization technique requires the empirical risk $\hat{\mathcal{R}}_{\mathcal{S}}$ to be a sum of independent terms. This is not available for the CML framework-based loss, which is essentially a sum of pairwise terms. For example, the terms $\ell^{(i)}(v_j^+, v_k^-)$ and $\ell^{(i)}(\tilde{v}_j^+, \tilde{v}_k^-)$ are interdependent as long as one of them is the same (i.e., $v_j^+ = \tilde{v}_j^+$ or $v_k^- = \tilde{v}_k^-$).

To solve this problem, we present a novel symmetrization technique to construct an extended Rademacher complexity defined as follows:

Definition 2. *(CML **Rademacher Complexity**). Given the sample set $\mathcal{S} = \bigcup_{u_i \in \mathcal{U}} \mathcal{S}_i$ where $\mathcal{S}_i = \{v_j^+\}_{j=1}^{n_i^+} \cup \{v_k^-\}_{k=1}^{n_i^-}, n_i^+ + n_i^- = N$ and the hypothesis space $\mathcal{H}_R$, then the empirical CML Rademacher Complexity with respect to the sample $\mathcal{S}$ is defined as:*

$$\hat{\mathfrak{R}}_{\ell,\mathcal{S}}^{cml}(\mathcal{H}_R) = \frac{1}{M} \sum_{u_i \in \mathcal{U}} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i} \left[\sup_{\mathcal{H}_R} \frac{1}{n_i^+ n_i^-} \sum_{j=1}^{n_i^+} \sum_{k=1}^{n_i^-} \mathcal{Q}_{(i)}^{jk} \right], \tag{9}$$

where

$$\mathcal{Q}_{(i)}^{jk} = \frac{\sigma_{ij}^+ + \sigma_{ik}^-}{2} \cdot \ell^{(i)}(v_j^+, v_k^-);$$

$\boldsymbol{\sigma}_i = [\sigma_{i1}^+, \sigma_{i2}^+, \ldots, \sigma_{in_i^+}^+, \sigma_{i1}^-, \sigma_{i2}^-, \ldots, \sigma_{in_i^-}^-]$ is i.i.d Rademacher random variables uniformly chosen from $\{-1, +1\}$, i.e., $\mathbb{P}(\sigma = 1) = \mathbb{P}(\sigma = -1) = 0.5$. Next, the population version of the Rademacher Complexity of CML is expressed as $\mathfrak{R}_{\ell,\mathcal{S}}^{cml}(\mathcal{H}_R) = \mathop{\mathbb{E}}_{\mathcal{S}} \left[\hat{\mathfrak{R}}_{\ell,\mathcal{S}}^{cml}(\mathcal{H}_R) \right]$.

According to this extended form of Rademacher complexity, we establish a symmetrization result expressed in the following theorem. The proof of Thm.1 is involved in Appendix B.1 in the supplementary materials.

Theorem 1 (CML Symmetrization). *Let $\mathcal{S}$ and $\mathcal{S}'$ be the two independent datasets of interactions that only one sample is different. In terms of any the hypothesis set $\mathcal{H}_R$ and loss function ℓ, the following holds:*

$$\begin{aligned} &\mathop{\mathbb{E}}_{\mathcal{S}} \left[\sup_{\mathcal{H}_R} \left[\mathop{\mathbb{E}}_{\mathcal{S}} (\hat{\mathcal{R}}_{\mathcal{S}}^{cml}) - \hat{\mathcal{R}}_{\mathcal{S}}^{cml} \right] \right] \\ &\leq \mathop{\mathbb{E}}_{\mathcal{S},\mathcal{S}'} \left[\sup_{\mathcal{H}_R} \left[\left(\hat{\mathcal{R}}_{\mathcal{S}'}^{cml}(f) - \hat{\mathcal{R}}_{\mathcal{S}}^{cml}(f) \right) \right] \right] \\ &\leq 2\mathfrak{R}_{\ell,\mathcal{S}}^{cml}(\mathcal{H}_R). \end{aligned} \tag{10}$$

Moreover, the generalization analysis also relies on an upper bound of the empirical Rademacher complexity $\hat{\mathfrak{R}}_{\ell,\mathcal{S}}^{\text{cml}}(\mathcal{H}_R)$. Here, we need the following notion of the effective sample size.

Definition 3. *(**Essential Sample Size**) Given the dataset $\mathcal{S}$, and n_i^+, n_i^- for each specific user, the effective sample size is defined as:*

$$\tilde{N} = \left(\sum_{u_i \in \mathcal{U}} \sqrt{\frac{1}{n_i^+} + \frac{1}{n_i^-}} \right)^{-2}.$$

Note that we call $\tilde{N}$ Essential Sample Size since it behaves like an ordinary sample size in the traditional generalization bound. Specifically, the traditional generalization bounds enjoy an order of $O((1/N)^{1/2})$, while our results scale as $O((1/\tilde{N})^{1/2})$. Moreover, the following remark presents an interesting property of the Essential Sample Size.

Remark 2. *Compared with the true sample size N, $\tilde{N}$ could better reflect the long-tail nature of the implicit feedback since increasing n_i^+ brings way shaper influence to $\tilde{N}$ than n_i^- (Note that the number of unobserved items (n_i^-) often dominates the observed ones (n_i^+) in practical RS).*

Based on the effective sample size, we reach the following upper bound for the Rademacher complexity. Refer to Appendix B.2 for the details of its proof.

Theorem 2. *(**Upper Bound of empirical Rademacher Complexity**). Given the sample dataset $\mathcal{S} = \bigcup_{u_i \in \mathcal{U}} \mathcal{S}_i$ where $\mathcal{S}_i = \{v_j^+\}_{j=1}^{n_i^+} \cup \{v_k^-\}_{k=1}^{n_i^-}, n_i^+ + n_i^- = N$. If ℓ is ϕ-Lipschitz continuous, then the following inequality holds:*

$$\hat{\mathfrak{R}}_{\ell,\mathcal{S}}^{cml}(\mathcal{H}_R) \lesssim \phi \cdot \frac{\mathbf{max}(\lambda, \sqrt{R \cdot d})}{M} \cdot \tilde{N}^{-1/2}. \tag{11}$$

Based on the two theorems above, we provide generalization bounds for the sampling-based and sampling-free CML framework respectively in the following two subsections.

4.3 Generalization Bound of Sampling-Free CML

We start our discussion with the sampling-free CML. Specifically, the main result is summarized in the following theorem.

Theorem 3. *(**Generalization Upper Bound of CML with Eq.(4)**). Let $\mathcal{H}_R$ be the hypothesis space and ℓ be ϕ-Lipschitz continuous. Given the sample set $\mathcal{S} = \bigcup_{u_i \in \mathcal{U}} \mathcal{S}_i$ where $\mathcal{S}^{(i)} =$*

$\{v_j^+\}_{j=1}^{n_i^+} \cup \{v_k^-\}_{k=1}^{n_i^-}, n_i^+ + n_i^- = N$, *for any* $\delta \in (0,1)$, *with probability at least* $1-\delta$, *the following inequation holds:*

$$\begin{aligned}\mathcal{R}_\ell^{cml}(f) \lesssim \hat{\mathcal{R}}_{\mathcal{S}}^{cml}(f) + \phi \cdot \frac{\max(\lambda, \sqrt{R \cdot d})}{M} \cdot \sqrt{\frac{1}{\tilde{N}}} \\ + \phi \cdot \frac{R}{M} \cdot \sqrt{\frac{\log 2/\delta}{2}} \cdot \sqrt{\frac{1}{\tilde{N}}},\end{aligned} \tag{12}$$

where $\mathcal{R}_\ell^{cml}(f)$ *is the expectation risk.*

Proof. *Equipped with Assum.1 and Def.2, by applying Talagrand contraction (Lem.3), we could complete the proof. More details are presented in the Appendix B.3.*

Based on a proper choice of λ, the theorem above shows that the generalization gap

$$\Delta_{\mathcal{S}} = \mathcal{R}_\ell^{\text{cml}}(f) - \hat{\mathcal{R}}_{\mathcal{S}}^{\text{cml}}(f)$$

satisfies $\Delta_{\mathcal{S}} \lesssim \frac{1}{M}\sqrt{\frac{d}{\tilde{N}}}$. This shows that $\Delta_{\mathcal{S}}$ vanishes with a sufficiently large data size and a moderate magnitude of d.

4.4 Generalization Bound of Sampling-based CML

For the sampling-based CML framework we have the following generalization upper bound.

Theorem 4. ***(Generalization Upper Bound of sampling-based CML Eq.(5)).*** *Let* $\mathcal{H}_R$ *be the hypothesis set and* ℓ *be* ϕ*-Lipschitz. Given the sample set* $\mathcal{S} = \bigcup_{u_i \in \mathcal{U}} \mathcal{S}_i$ *where* $\mathcal{S}^{(i)} = \{v_j^+\}_{j=1}^{n_i^+} \cup \{v_k^-\}_{k=1}^{n_i^-}, n_i^+ + n_i^- = N$, *for any* $\delta \in (0,1)$, *with probability at least* $1-\delta$, *the following holds for all possible embedding* $\mathcal{H}_R$*:*

$$\begin{aligned}\mathcal{R}_\ell^{cml}(f) \lesssim \hat{\mathcal{R}}_{\mathcal{S}}^{cml}(f) + \phi \cdot \frac{\max(\lambda, \sqrt{R \cdot d})}{M} \cdot \sqrt{\frac{1}{\tilde{N}}} \\ + \frac{(\lambda + 4R)}{M} \cdot \sum_{u_i \in \mathcal{U}} D_{TV}(\hat{\mathbb{P}}^{(i)}, \tilde{\mathbb{P}}^{(i)}) \\ + \phi \cdot \frac{R}{M} \cdot \sqrt{\frac{\log 2/\delta}{2}} \cdot \sqrt{\frac{1}{\tilde{N}}}\end{aligned} \tag{13}$$

where $\hat{\mathbb{P}}^{(i)}$ *is the original distribution with* $\hat{\mathbb{P}}_{ik}^{(i)} \equiv \frac{1}{n_i^+ n_i^-}$; $D_{TV}(\hat{\mathbb{P}}^{(i)}, \tilde{\mathbb{P}}^{(i)}) = \frac{1}{2} \cdot \left\| \hat{\mathbb{P}}^{(i)} - \tilde{\mathbb{P}}^{(i)} \right\|_1$ *is the per-user Total Variance (TV) between two probability distributions* $\hat{\mathbb{P}}^{(i)}$ *and* $\tilde{\mathbb{P}}^{(i)}$ *on* $\mathcal{S}$ *for a specific user* u_i*, which characterizes the difference between two probability distributions.*

Proof. *The proof of Thm.4 follows those of Thm.3 and Thm.2, and we refer the readers to see more details in the Appendix B.4.*

It is easy to see that the generalization upper bound for sampling-based CML has an extra term

$$\frac{(\lambda + 4R)}{M} \cdot \sum_{u_i \in \mathcal{U}} D_{TV}\left(\hat{\mathbb{P}}^{(i)}, \tilde{\mathbb{P}}^{(i)}\right),$$

which captures the distance between the sampling-strategy-induced distribution and the ground-truth distribution for per user. This brings about a biased estimation.

4.5 Summary and Discussion

Thm.4 intuitively reveals the shortcomings of sampling-based CML, due to the extra bias term $D_{TV}(\hat{\mathbb{P}}^{(i)}, \tilde{\mathbb{P}}^{(i)})$. This term describes the distribution deviation between leveraging whole samples for learning and adopting the negative sampling strategy. In order to eliminate this deviation and improve the performance of recommendation, we must learn from all samples instead of adopting negative sampling. *This motivates us to develop an efficient alternative without negative sampling.*

5 Sampling-Free Acceleration

Following our theoretical results, we propose an efficient Sample-Free CML (SFCML) acceleration algorithm in this section.

5.1 Modifying the Pairwise Loss

Since we focus on developing an efficient algorithm for CML without negative sampling, our acceleration method is based on a modification of the full samples-based CML loss function Eq.(4). First of all, we replace the hinge loss with widely adopted square loss $\ell_{sq}(x) = (\lambda - x)^2$. In addition, without loss of generality, we restrict the bounded norm of all users and items on a R-radius hyper-sphere rather than a bounded ball. Putting them into Eq.(4), we arrive at our new sampling-free CML loss function:

$$\begin{aligned}\hat{\mathcal{R}}_{\mathcal{S}}^{\text{sfcml}}(f) = \frac{1}{M} \sum_{u_i \in \mathcal{U}} \frac{1}{n_i^+ n_i^-} \sum_{j=1}^{n_i^+} \sum_{k=1}^{n_i^-} \ell_{sq}^{(i)}(v_j^+, v_k^-), \\ s.t. \left\|\boldsymbol{e}_{u_i}\right\|^2 = R, \quad \left\|\boldsymbol{e}_{v_j}\right\|^2 = R, \quad u_i \in \mathcal{U}, v_j \in \mathcal{V},\end{aligned} \tag{14}$$

where

$$\ell_{sq}^{(i)}(v_j^+, v_k^-) = \left(\lambda + \boldsymbol{d}(i,j) - \boldsymbol{d}(i,k)\right)^2. \tag{15}$$

and $\lambda > 0$ could also be regarded as the safe margin. Since we let $\boldsymbol{e}_{u_i}$ and $\boldsymbol{e}_{v_j}$ distribute on a hyper-sphere with R radius ($R = 1.0$ in the experiment), i.e., $\left\|\boldsymbol{e}_{u_i}\right\|^2 = R$ and $\left\|\boldsymbol{e}_{v_j}\right\|^2 = R$, $\ell_{sq}^{(i)}(v_j^+, v_k^-)$ could be further simplified as

$$\ell_{sq}^{(i)}(v_j^+, v_k^-) = \left(\lambda - 2\boldsymbol{e}_{u_i}^\top (\boldsymbol{e}_{v_j^+} - \boldsymbol{e}_{v_k^-})\right)^2.$$

Therefore, the final CML loss function is modified as follows:

$$\begin{aligned}\hat{\mathcal{R}}_{\mathcal{S}}^{\text{sfcml}}(f) = \frac{1}{M} \sum_{u_i \in \mathcal{U}} \frac{1}{n_i^+ n_i^-} \sum_{j=1}^{n_i^+} \sum_{k=1}^{n_i^-} \left(\lambda - 2\boldsymbol{e}_{u_i}^\top (\boldsymbol{e}_{v_j^+} - \boldsymbol{e}_{v_k^-})\right)^2, \\ s.t. \left\|\boldsymbol{e}_{u_i}\right\|^2 = R, \quad \left\|\boldsymbol{e}_{v_j}\right\|^2 = R, \quad u_i \in \mathcal{U}, v_j \in \mathcal{V}.\end{aligned} \tag{16}$$

Motivated by the modified pairwise loss, the following corollary demonstrates that the bias term caused by the per-user TV term will vanish in the generalization upper bound of SFCML. This proves the effectiveness of our proposed SFCML method.

Corollary 1. ***(Generalization Upper Bound of SFCML with Eq.(16)).*** *Given the sample set* $\mathcal{S} = \bigcup_{u_i \in \mathcal{U}} \mathcal{S}_i$ *where* $\mathcal{S}_i =$

$\{v_j^+\}_{j=1}^{n_i^+} \cup \{v_k^-\}_{k=1}^{n_i^-}, n_i^+ + n_i^- = N$, *for any* $\delta \in (0,1)$, *with probability at least* $1-\delta$, *the following inequation holds:*

$$\begin{aligned}\mathcal{R}_\ell^{cml}(f) \lesssim \hat{\mathcal{R}}_{\mathcal{S}}^{sfcml}(f) \\ + (\lambda + 4R) \cdot \frac{\max(\lambda, \sqrt{R \cdot d})}{M} \cdot \sqrt{\frac{1}{\tilde{N}}} \\ + \frac{(\lambda + 4R) \cdot R}{M} \cdot \sqrt{\frac{\log 2/\delta}{2}} \cdot \sqrt{\frac{1}{\tilde{N}}}.\end{aligned} \tag{17}$$

Proof. *Note that, changing the constraint of embeddings from* $\|\boldsymbol{e}_{u_i}\|^2 \le R, \|\boldsymbol{e}_{v_j}\|^2 \le R$ *to* $\|\boldsymbol{e}_{u_i}\|^2 = R, \|\boldsymbol{e}_{v_j}\|^2 = R$ *will not change the result of Thm.3, since the constraints are only employed to bound the supremum. It is easy to show that* $\ell_{sq}^{(i)}$ *is* $(\lambda + 4R)$*-Lipschitz continuous. Then the proof is completed via setting* $\phi = \lambda + 4R$ *in Thm.3.* □

5.2 Efficient Alternative without Negative Sampling

With the help of the squared loss, it is interesting to note that Eq.(16) could be regarded as an AUC optimization problem [44], [79]. Recall that area under the ROC curve (AUC) measures the probability of the score of a positive sample higher than a negative sample [80], [81], [82]. Then, assuming there are no ties in the scores of samples, AUC could be optimized by the following empirical minimization problem [44], [83]:

$$\min_f \frac{1}{n_+ n_-} \sum_{x_+} \sum_{x_-} \ell_{0-1}\left(f(x_+) - f(x_-)\right).$$

Generally speaking, x_+ (x_-) denotes the positive (negative) samples, and n_+ (n_-) represents the number of positive (negative) instances. f is the score/decision function representing the probability of an instance to be predicted as a positive sample. $\ell_{0-1}(z)$ is the 0-1 loss, which returns 1 if $z < 0$, otherwise 0 is returned. Since ℓ_{0-1} is not continuous, it is often replaced by a continuous surrogate loss ℓ_{sur} [45], [84], [85], which induces a surrogate AUC optimization problem:

$$\min_f \frac{1}{n_+ n_-} \sum_{x_+} \sum_{x_-} \ell_{sur}\left(f(x_+) - f(x_-)\right). \tag{18}$$

Therefore, if we regard f and ℓ_{sur} in Eq.(18) as $f(v_j|u_i) = 2\boldsymbol{e}_{u_i}^\top \boldsymbol{e}_{v_j}$ and ℓ_{sq} respectively, it then recovers our objective function.

Next, we will elaborate on how to develop an efficient algorithm without negative sampling strategies. Note that, since the loss function is calculated separately for different users, we only consider one specific user (taking $u_i \in \mathcal{U}$ as an example), while the overall objective function could simply be obtained by taking an average. For the convenience of the subsequent description, we let $\hat{\mathcal{R}}_{\mathcal{S}_i}^{\text{sfcml}}(f)$ be the empirical risk of the specific user u_i.

At first, we can find that, for every single item $v_j^+ \in \mathcal{S}_i^+$, it is only paired with the remaining negative items $v_k^- \notin \mathcal{S}_i^+$. This observation helps us to decouple the time-consuming pairwise computation and develop our efficient algorithm SFCML. Let us construct a graph defined as $\mathcal{G}^{(i)} = (\mathcal{S}^{(i)}, \mathcal{E}^{(i)}, \mathcal{D}^{(i)})$, where the vertex set $\mathcal{S}^{(i)} = \{(v_j, y_{ij}) | v_j \in \mathcal{V}\}$ is the set of preferences for user u_i, $\mathcal{E}^{(i)} = \{(j,k) | y_{ij} \neq y_{ik}\}$ is the edge of the graph and $\mathcal{D}^{(i)}$ is the adjacent matrix defined as follows:

$$\mathcal{D}_{jk}^{(i)} = \begin{cases} \frac{1}{n_i^+ n_i^-}, & (j,k) \in \mathcal{E}^{(i)}, \\ 0, & (j,k) \notin \mathcal{E}^{(i)}. \end{cases} \tag{19}$$

Subsequently, the graph Laplacian matrix could be defined as:

$$\boldsymbol{L}^{(i)} = diag(\mathcal{D}^{(i)}\mathbf{1}) - \mathcal{D}^{(i)}, \tag{20}$$

where $\mathbf{1} \in \mathbb{R}^N$ is an all-one vector; $diag(\mathcal{D}^{(i)}\mathbf{1}) \in \mathbb{R}^{N \times N}$ is a diagonal matrix, with the $diag(\mathcal{D}^{(i)}\mathbf{1})_{i,i}$ representing the degree of vertex i in the graph.

Based the graph-theoretic machinery, Eq.(16) can be reformulated as follows:

$$\begin{aligned}&\hat{\mathcal{R}}_{\mathcal{S}_i}^{\text{sfcml}}(f) = (\boldsymbol{f}^{(i)} - \lambda \cdot \boldsymbol{y}^{(i)})^\top \boldsymbol{L}^{(i)} (\boldsymbol{f}^{(i)} - \lambda \cdot \boldsymbol{y}^{(i)}), \\ &s.t. \quad \|\boldsymbol{e}_{u_i}\|^2 = R, \quad \|\boldsymbol{e}_{v_j}\|^2 = R, \quad v_j \in \mathcal{V},\end{aligned} \tag{21}$$

where $\boldsymbol{y}^{(i)} = [y_{i1}, y_{i2}, \ldots, y_{iN}]^\top$ is a vector representing the preferences of user u_i toward all items, $\boldsymbol{f}^{(i)} = [2\boldsymbol{e}_{u_i}^\top \boldsymbol{e}_{v_1}, 2\boldsymbol{e}_{u_i}^\top \boldsymbol{e}_{v_2}, \ldots, 2\boldsymbol{e}_{u_i}^\top \boldsymbol{e}_{v_N}]^\top$ could be seen as a score vector in terms of all items.

However, Eq.(21) still brings a rather heavy computation burden due to the inefficiency of naive matrix multiplication of Eq.(21), almost $\mathcal{O}(N^2 + N)$ *for each user.*

This drives us to adopt the following proposition to further accelerate the calculation.

Proposition 1. *Let both* p *and* q *be the positive integers. Then, for any matrix* $\boldsymbol{P} \in \mathbb{R}^{N \times p}$ *and* $\boldsymbol{Q} \in \mathbb{R}^{N \times q}$, *the calculation of* $\boldsymbol{P}^\top \boldsymbol{L}^{(i)} \in \mathbb{R}^{p \times N}$ *could be nearly finished within* $\mathcal{O}(pN)$ *and* $\boldsymbol{P}^\top \boldsymbol{L}^{(i)} \boldsymbol{Q} \in \mathbb{R}$ *could be almost completed within* $\mathcal{O}(pqN)$.

Proof. *Firstly, according to the above definition, we can show that,* $\mathcal{D}^{(i)}$ *can be reformulated as*

$$\mathcal{D}^{(i)} = \frac{1}{n_i^+ n_i^-}[\boldsymbol{y}^{(i)}(\mathbf{1} - \boldsymbol{y}^{(i)})^\top + (\mathbf{1} - \boldsymbol{y}^{(i)})(\boldsymbol{y}^{(i)})^\top], \tag{22}$$

where $\mathbf{1} \in \mathbb{R}^N$ *is a vector where all values are* 1.

Meanwhile, given $\mathcal{D}^{(i)}$, $\boldsymbol{L}^{(i)} \in \mathbb{R}^{N \times N}$ *could be rewritten as*

$$\begin{aligned}\boldsymbol{L}^{(i)} &= diag(\mathcal{D}^{(i)}\mathbf{1}) - \mathcal{D}^{(i)} \\ &= diag\left(\frac{\boldsymbol{y}^{(i)}}{n_i^+} + \frac{(\mathbf{1} - \boldsymbol{y}^{(i)})}{n_i^-}\right) - \mathcal{D}^{(i)}\end{aligned} \tag{23}$$

Correspondingly, according to Eq.(23), the following equation holds for $\boldsymbol{P}^\top \boldsymbol{L}^{(i)} \in \mathbb{R}^{p \times N}$

$$\begin{aligned}\boldsymbol{P}^\top \boldsymbol{L}^{(i)} =& \boldsymbol{P}^\top \left(diag\left(\frac{\boldsymbol{y}^{(i)}}{n_i^+} + \frac{(\mathbf{1} - \boldsymbol{y}^{(i)})}{n_i^-}\right)\right) \\ &- \frac{\boldsymbol{P}^\top \boldsymbol{y}^{(i)}(\mathbf{1} - \boldsymbol{y}^{(i)})^\top}{n_i^+ n_i^-} \\ &- \frac{\boldsymbol{P}^\top (\mathbf{1} - \boldsymbol{y}^{(i)})(\boldsymbol{y}^{(i)})^\top}{n_i^+ n_i^-}\end{aligned} \tag{24}$$

According to Eq.(24), it is easy to conclude that $\boldsymbol{P}^\top \boldsymbol{L}^{(i)}$ *could be completed within almost* $\mathcal{O}(pN)$ *while it should be computed within almost* $\mathcal{O}(pN^2)$ *with the naive matrix multiplication.*

In addition, with the acceleration of $\boldsymbol{P}^\top \boldsymbol{L}^{(i)} \in \mathbb{R}^{p \times N}$, *it is obvious to show that the calculation of* $\boldsymbol{P}^\top \boldsymbol{L}^{(i)} \boldsymbol{Q} \in \mathbb{R}$ *is reduced from* $\mathcal{O}(pqN^2)$ *to almost* $\mathcal{O}(pqN)$.

This proved the proposition. □

Remark 3. *Equipped with Prop.1, Eq.(21) could be almost finished within* $\mathcal{O}(N) = \mathcal{O}(n_i^+ + n_i^-)$ *by replacing either of the two parts in Eq.(21), i.e., let* $\boldsymbol{P} = (\boldsymbol{f}^{(i)} - \lambda \cdot \boldsymbol{y}^{(i)}) \in \mathbb{R}^N$ *or* $\boldsymbol{Q} = (\boldsymbol{f}^{(i)} - \lambda \cdot \boldsymbol{y}^{(i)}) \in \mathbb{R}^N$. *This is a significant efficiency improvement against naive CML with* $\mathcal{O}(n_i^+ n_i^-)$ *time complexity for each user, due to* $n_i^+ + n_i^- \ll n_i^+ n_i^-$ *in the practical RS. At the same time, Prop.1 also guarantees that our algorithm SFCML enjoys practically* $\mathcal{O}(N)$ *space complexity per user, while the naive CML (refer to Eq.(4)) has almost* $\mathcal{O}(n_i^+ n_i^-) = \mathcal{O}(N^2)$ *space complexity per user.*

5.3 Optimization and Algorithm

5.3.1 The Overall Objective Function

Now we come to the objective function of our efficient alternative SFCML with all users by taking an average

$$\hat{\mathcal{R}}_{\mathcal{S}}^{\text{sfcml}}(f) = \frac{1}{M} \sum_{u_i \in \mathcal{U}} \hat{\mathcal{R}}_{\mathcal{S}_i}^{\text{sfcml}}(f), \quad s.t. \;\; ||\boldsymbol{e}_{u_i}||^2 = R, \quad ||\boldsymbol{e}_{v_j}||^2 = R, \;\; u_i \in \mathcal{U}, v_j \in \mathcal{V}. \tag{25}$$

5.3.2 Optimization

We employ the gradient descent method as the optimizer to learn our proposed algorithm SFCML. The optimization of user u_i and items' weight $\boldsymbol{W}_v$ could be summarized as follows.

Optimization of user u_i. In order to minimize Eq.(25), we first rewrite the score function of u_i as $\boldsymbol{f}^{(i)} = 2\boldsymbol{W}_v \boldsymbol{e}_{u_i}$ where $\boldsymbol{W}_v \in \mathbb{R}^{N \times d}$ is the learned transformation weight (review Eq.(2)). Then, the gradient descent method updates the variable $\boldsymbol{e}_{u_i}$ according to:

$$\boldsymbol{e}_{u_i} = \boldsymbol{e}_{u_i} - \eta \cdot \nabla_{\boldsymbol{e}_{u_i}} \hat{\mathcal{R}}_{\mathcal{S}_i}^{\text{sfcml}}(f) \tag{26}$$

where η is the learning rate, and

$$\nabla_{\boldsymbol{e}_{u_i}} \hat{\mathcal{R}}_{\mathcal{S}_i}^{\text{sfcml}}(f) = 2\boldsymbol{W}_v^\top \boldsymbol{L}^{(i)} \left(\boldsymbol{f}^{(i)} - \lambda \cdot \boldsymbol{y}^{(i)} \right).$$

Note that, here we just present the derivation of $\boldsymbol{e}_{u_i}$. One can easily match the $\boldsymbol{e}_{u_i}$ and $\boldsymbol{W}_u$ based on Eq.(2), i.e., $\boldsymbol{e}_{u_i}$ corresponding the i-th row in the matrix $\boldsymbol{W}_u$.

Remark 4. *Owing to the strengths of Prop.1, the computation complexity of Eq.(26) is still reasonable, which could be almost completed within* $\mathcal{O}(dN) = \mathcal{O}(d(n_i^+ + n_i^-))$ *for a specific user* u_i.

Optimization of items' weight $\boldsymbol{W}_v$. In the same way, the gradient descent method updates the variable $\boldsymbol{W}_v$ according to:

$$\boldsymbol{W}_v = \boldsymbol{W}_v - \eta \cdot \nabla_{\boldsymbol{W}_v} \hat{\mathcal{R}}_{\mathcal{S}_i}^{\text{sfcml}}(f) \tag{27}$$

where

$$\nabla_{\boldsymbol{W}_v} \hat{\mathcal{R}}_{\mathcal{S}_i}^{\text{sfcml}}(f) = 2\boldsymbol{L}^{(i)} \left(\boldsymbol{f}^{(i)} - \lambda \cdot \boldsymbol{y}^{(i)} \right) \boldsymbol{e}_{u_i}^\top$$

Remark 5. *Similarly, following the Prop.1, we could demonstrate that, for any* $\boldsymbol{Q} \in \mathbb{R}^{N \times q}$ *where* q *is a positive integer,* $\boldsymbol{L}^{(i)}\boldsymbol{Q} \in \mathbb{R}^{N \times q}$ *could be finished within* $\mathcal{O}(qN)$. *According to this, by setting* $\boldsymbol{Q} = (\boldsymbol{f}^{(i)} - \lambda \cdot \boldsymbol{y}^{(i)}) \in \mathbb{R}^N$, *Eq.(27) could also be updated within* $\mathcal{O}(dN + N) = \mathcal{O}\left((d+1)(n_i^+ + n_i^-)\right)$ *for all items' embeddings.*

Finally, we summarize all the details of SFCML in Alg.1.

Algorithm 1: Sampling-Free Collaborative Metric Learning

Input: User set $\mathcal{U} = \{u_1, u_2, \ldots, u_M\}$
Input: Item set $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$
Input: Preference set: $\{\boldsymbol{y}^{(i)} | u_i \in \mathcal{U}\}$
Input: Hypersphere radius: R
Input: Safe margin λ
Input: Learning rate η
Output: User transformation matrix: $\boldsymbol{W}_u$
Output: Item transformation matrix: $\boldsymbol{W}_v$

```
Initialize W_u;
Initialize W_v;
while Not Converged do
    Restrict the norm of all users' and items' embeddings on a R-radius hypersphere;
    for u_i in user set U do
        Project user u_i and items into the metric space by Eq.(2);
        Index the corresponding preference set y^(i);
        Optimize the weights via Eq.(26) and Eq.(27);
        Restrict the norm of all items' embeddings on a R-radius hypersphere;
    end
end
Restrict the norm of all users' embeddings on a R-radius hypersphere;
return W_u and W_v
```

6 Experiments

In this section, we conduct comprehensive experiments on a wide range of benchmark datasets to show the superiority of our proposed method.

6.1 Dataset Descriptions

We perform the empirical studies over seven widely adopted benchmark datasets to evaluate the performance, including:

- **MovieLens** [1] - A series of benchmark datasets are popularly and widely used in RS. There are many versions of MovieLens, and we adopt **MovieLens-100k**[2], **MovieLens-1m**[3] and **MovieLens-20m**[4] here to test the performance. Specifically, it includes ratings ranging from 1 to 5 on various movies. Following the previous work [24], [86], if the score of item v_j rated by user u_i is no less than 4, we regard item v_j as a positive item for user u_i.

- **CiteULike**[5] [87] - An implicit feedback dataset that allows users to create their own collections of articles. There are two configurations of CiteULike collected from CiteULike and Google Scholar. Following [24], we adopt **CiteULike-T** here to evaluate the performance.

1. https://grouplens.org/datasets/movielens/
2. https://grouplens.org/datasets/movielens/100k/
3. https://grouplens.org/datasets/movielens/1m/
4. https://grouplens.org/datasets/movielens/20m/
5. http://www.citeulike.org/faq/data.adp

TABLE 2: Basic Information of the Datasets. %Density is defined as $\frac{\#Ratings}{\#Users \times \#Items} \times 100\%$.

Datasets	MovieLens-100K	CiteULike-T	MovieLens-1M	Steam-200k	Anime	MovieLens-20M	Amazon-Book
Domain	Movie	Paper	Movie	Game	Anime	Movie	Book
#Users	938	7,947	11,209	3,757	54,190	136,677	64,937
#Items	1,447	25,975	7,491	5,113	6,967	17,679	181,152
#Ratings	55,361	125,580	85,341	115,139	7,634,542	9,986,829	2,880,930
%Density	4.0788	0.0608	0.1016	0.5994	2.0221	0.4133	0.0245

- **Steam-200k**[6] - This dataset is collected from the Steam which is the world's most popular PC gaming hub. The observed behaviors of users include 'purchase' and 'play' signals. In order to obtain the implicit feedback, if user has purchased a game as well as the playing hours $play > 0$, we treat this game as a positive item.
- **Anime**[7] - This dataset is collected by the myanimelist.net API which records the preferences of users toward several animes. Its ratings range from -1 to 10, where -1 represents the user watched an anime but didn't make an assessment for it. Moreover, the higher the ratings for an item is, and the more the user likes it. Similarly, if the item v_j's ratings produced by user u_i is no less than 5, we regard item v_j as a positive item in terms of user u_i.
- **Amazon-Book** [8] [88] - This dataset includes ratings and various metadata collected from Amazon. The ratings therein range from 1 to 5. We conduct the same preprocess as MoiveLens to obtain the implicit signals.

More detailed statistics with respect to these datasets are summarized in Tab.2.

6.2 Competitors

Note that, the starting point of this work is to develop an efficient CML-based algorithm without negative sampling to get rid of the bias caused by the sampling-based CML. Therefore, to show the superiority of the proposed algorithm, we evaluate SFCML against the following 12 competitors:

- **itemKNN** [89], [90] is a simple but classical item-based collaborative filtering method. It recommends new items to the target user based on the similarities with his/her interacted items. Generally speaking, one usually adopts the cosine function to measure the similarities between different items.
- **Generalized Matrix Factorization** (GMF) can be regarded as a generalized and extended MF method, which is more expressive than the traditional MF algorithm. It is one of the instantiates in [27], which applies a linear kernel to model the latent user-item interactions.
- **Multi-Layer Perceptron** (MLP) is a deep learning-based framework [27], which adopts a non-linearity multi-layer perceptron to learn the interaction between users and items. In this way, the model could be endowed with reasonable flexibility and non-linearity to capture the preference of users.
- **Neural network-based Collaborative Filtering** (NCF)[9] [27] is a popular and competitive deep learning-based framework bridging the gap of GMF and MLP. NCF concatenates the output of GMF and MLP, and regards the recommendation task as a regression problem. Notably, it computes the ranking scores with a neural network instead of the inner product.
- **Uniform Negative Sampling** (UniS) [14], [24] leverages a uniform negative Sampling strategy to alleviate the heavy burden of computations for CML. Specifically, for every user, uniformly sample U items from unobserved interactions as negatives to minimize Eq.(5).
- **Popularity-based Negative Sampling** (PopS) [31], [32], [33] optimizes the pairwise ranking loss Eq.(5) with a popularity-based negative sampling strategy, i.e., sampling U negative candidates from unobserved interactions based on their frequencies.
- **Two-Stage Negative Sampling** (2stS) [10] [33] is an effective and competitive method. To increase the number of informative items to optimize Eq.(5), 2stS adopts a two-stage sampling strategy. Firstly, a candidate set of items are sampled based on their popularity. Secondly, according to their inner product values with anchors (positive items), the most informative samples are selected from this candidate.
- **Hard Negative Sampling** (HarS) [11] [24] is similar to the negative sample mining process widely adopted in the object detection [34], [35], [36], [37]. Specifically, it includes two stages: 1) uniformly sample U candidates from unobserved items; 2) select the hard item from the candidates as negative item to train based on the distance between targeted user and items. Note that, when the number of U is set as 1, the HarS is the same as the UniS strategy.
- **Collaborative Translational Metric Learning (TransCF)** [38] is a translation-based method. Specifically, such translation-based algorithms employ $\boldsymbol{d}(i,j) = ||\boldsymbol{e}_{u_i} + \boldsymbol{e}_{r_{ij}} - \boldsymbol{e}_{v_j}||^2$ as the distance/score between user u_i and item v_j instead of $||\boldsymbol{e}_{u_i} - \boldsymbol{e}_{v_j}||^2$, where $\boldsymbol{e}_{r_{ij}}$ is a specific translation vector for u_i and v_j. In light of this, TransCF discovers such user–item translation vectors via the users' relationships with their neighbor items.
- **Latent Relational Metric Learning (LRML)** [39] is also a translation-based CML method. As a whole, the key idea of LRML is similar to TransCF. The main difference is how to access the translation vectors effectively. Concretely, TransCF leverages the neighborhood information of users and items to acquire the translation vectors while LRML introduces an attention-based memory-

6. https://www.kaggle.com/tamber/steam-video-games
7. https://www.kaggle.com/CooperUnion/anime-recommendations-database
8. https://jmcauley.ucsd.edu/data/amazon/

augmented neural architecture to learn the exclusive and optimal translation vectors.

- **Co-occurrence embedding Regularized Metric Learning (CRML)** [40] considers the global statistical information of user-user and item-item pairs by involving a co-occurrence embedding to regularize the metric learning model. Then, CRML regards the optimization problem as a multi-task learning problem to boost the performance of CML, including the primary CML recommendation task and two auxiliary representation learning tasks.
- **Efficient Heterogeneous Collaborative Filtering (EHCF)** [72] is a state-of-the-art non-sampling-based neural framework, which presents a sampling-free strategy to optimize an NCF-like model using the whole heterogeneous data without negative sampling.

Discussions of the competitors. The most related competitors to our proposed SFCML roughly fall into two groups: **a) Sampling-based CML methods**, including UniS, PopS, 2stS, HarS, TransCF, LRML and CRML. **b) The state-of-the-art sampling-free algorithms**, i.e., EHCF. Our work differs from both of them. In terms of a), some of them (i.e., UniS, PopS, 2stS and HarS) try to improve the performance of CML by directly developing more effective negative sampling strategies. The others (including TransCF, LRML and CRML) introduce more complicated structures or auxiliary statistical information to improve the CML. *Yet they still need to adopt the negative sampling (usually employing one of the above-mentioned sampling strategies) in the training phase to alleviate the heavy burden of pairwise computations. Such circumstance implies they would still encounter the generalization problem more or less as discussed in Sec.4.5.* Different from the existing sampling-based CML algorithms, SFCML attempts to boost the recommendation performance from a sampling-free aspect, i.e., directly optimize CML leveraging the whole data under a relatively acceptable efficiency. In terms of b), however, it still differs significantly from our work. Practically, EHCF presents an effective heterogeneous CF-based framework learned from the whole data instead of negative sampling. *However, such a method is specifically tailored for the pointwise-based recommendation (such as NCF-like algorithms and some other factorization-based models), which is not suitable for the CML framework based on the pairwise ranking loss.*

6.3 Evaluation Metrics

In some typical recommendation systems, users often care about the top-K items in recommendation lists, so the most relevant items should be ranked first as much as possible. In light of this, we evaluate the performance of competitors and our algorithm with the following extensively adopted six metrics: **Precision** (P@K), **Recall** (R@K), **Normalized Discounted Cumulative Gain** (NDCG@K), **Mean Average Precision** (MAP), **Mean Reciprocal Rank** (MRR) and **Area Under ROC Curve** (AUC). Note that, for all the above metrics, the higher the metric is, the better the performance the algorithm achieves. See Appendix.C.1 for more details.

9. https://github.com/guoyang9/NCF
10. https://github.com/deezer/sigir2019-2stagesampling
11. https://github.com/changun/CollMetric

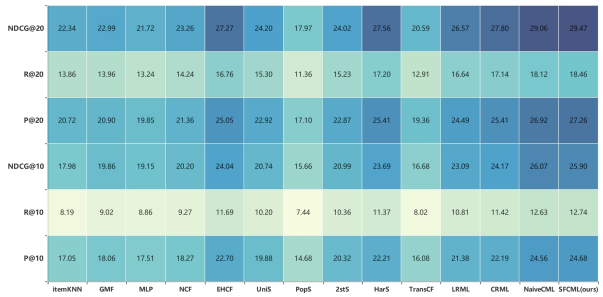


Fig. 1: Heat map of performance results on MovieLens-100k in terms of $K = \{10, 20\}$. Please see Appendix.C.3 for more results.

6.4 Implementation Details

All the experiments are conducted on a Ubuntu 16.04.6 server equipped with 256GB RAM, Intel(R) Xeon(R) Gold-5218 CPU, and an RTX 3090 GPU. We implement our model with PyTorch [12] [91] and adopt *Adagrad* [92] as the optimizer to minimize the objective loss function. For all datasets, each user's interactions are divided into training/validation/test sets with a $60\%/20\%/20\%$ split. Based on this split ratio, to ensure that each user has at least one positive interaction in training/validation/test, we thus filter out users that have less than five interactions. We conduct the grid search to find the best parameters based on the validation set and report the corresponding performance on the test set. Specifically, for all methods, the batch size is set to 256 and the learning rate is tuned amongst $\{0.001, 0.003, 0.005, 0.01, 0.03, 0.05\}$. The number of epochs is set as 200 on all datasets. In addition, to further avoid the over-fitting problem, if the performance according to the AUC metric (with error range $\epsilon = 10^{-5}$) on the validation set does not improve after 15 epochs, the early-stopping is executed. With respect to the CML-based algorithms, the dimension of embedding d is fixed as 256, and the margin λ is tuned amongst $\{1.0, 1.5, 2.0\}$. Moreover, we test the sampling-based CML with different sampling constant $U = \{1, 3, 5, 8, 10\}$ and then report the best performance according to AUC metric. For the other parameters of baseline models, we follow their tuning strategies in the original papers. Finally, in terms of the top-K recommendation, we evaluate the performance at $K \in \{3, 5, 10, 20\}$, respectively.

6.5 Experiments Results

6.5.1 Overall Performance

Some experimental results are presented in Tab.3 and Tab.4. The others are shown in Appendix.C.3 due to the limitation of space. From these results, we can draw the following interesting observations:

- Our proposed SFCML shows competitive performance on all benchmark datasets, and, in most cases, its performance surpasses all the involved competitors. For example, the significant improvement of performance between SFCML and the best competitor (achieved

12. https://pytorch.org/

TABLE 3: Performance comparisons on MovieLens-100k, CiteULike and MovieLens-1m datasets, where '-' means that we cannot complete the experiments due to the out-of-memory issue. The best and second-best are highlighted in bold and underlined, respectively.

	Method	P@3	R@3	NDCG@3	P@5	R@5	NDCG@5	MAP	MRR	AUC
MovieLens-100k	itemKNN	11.35	2.41	11.57	12.96	4.11	13.45	8.49	24.63	85.68
	GMF	14.35	3.37	15.20	16.43	5.79	17.21	9.82	31.00	86.12
	MLP	14.98	3.93	15.57	15.51	5.70	16.54	10.09	31.99	87.09
	NCF	15.94	4.11	16.75	17.26	6.45	18.25	11.35	34.34	88.03
	EHCF	21.13	6.99	21.80	20.89	8.82	22.08	16.51	41.77	92.18
	UniS	15.94	4.43	16.06	17.04	6.23	17.40	13.21	33.07	92.27
	PopS	13.05	3.99	13.36	13.38	5.10	13.93	9.49	29.13	80.51
	2stS	15.50	4.42	15.77	16.76	6.21	17.18	13.35	32.95	92.01
	HarS	20.76	6.51	21.05	21.36	8.86	22.10	15.94	40.02	91.66
	TransCF	12.90	3.72	13.32	14.35	5.70	14.76	11.19	29.88	87.53
	LRML	20.65	6.65	21.44	20.36	8.24	21.75	13.48	37.93	90.38
	CRML	20.94	6.43	21.80	21.14	8.53	22.44	16.33	41.14	92.07
	NaiveCML	22.51	7.26	22.79	**23.85**	9.81	24.42	17.62	42.35	**93.24**
	SFCML(ours)	**23.40**	**7.62**	**23.63**	23.74	**9.95**	**24.65**	**18.00**	**43.13**	93.11
CiteULike	itemKNN	1.20	0.83	1.23	1.15	0.77	1.16	1.44	3.78	69.94
	GMF	1.86	0.96	2.05	2.15	0.97	2.40	1.34	5.53	65.38
	MLP	1.76	0.77	1.94	2.42	0.98	2.67	1.52	5.70	78.14
	NCF	2.06	1.04	2.21	2.36	1.16	2.64	1.66	6.20	77.88
	EHCF	4.91	2.69	5.21	5.88	3.07	6.29	3.78	12.60	76.40
	UniS	5.84	3.04	6.07	7.58	3.96	7.91	4.67	14.51	**86.62**
	PopS	7.25	3.82	7.55	9.14	4.96	9.72	5.47	17.06	84.79
	2stS	7.16	3.69	7.44	8.95	4.71	9.66	5.34	16.96	84.79
	HarS	6.05	3.11	6.35	7.84	4.16	8.29	5.02	15.46	83.80
	TransCF	5.84	3.12	6.21	7.29	3.92	7.76	4.40	14.43	83.62
	LRML	2.93	1.38	3.05	3.84	1.88	4.13	2.05	7.96	74.89
	CRML	6.71	3.47	7.08	8.61	4.65	9.21	5.40	16.85	84.09
	NaiveCML	-	-	-	-	-	-	-	-	-
	SFCML(ours)	**8.28**	**4.65**	**8.57**	**9.69**	**5.38**	**10.29**	**6.70**	**19.43**	83.44
MovieLens-1m	itemKNN	12.24	2.90	12.41	12.43	4.29	12.79	8.34	26.16	88.70
	GMF	14.03	2.79	14.35	14.28	4.08	14.80	8.24	29.51	88.56
	MLP	13.95	2.78	14.22	14.06	3.98	14.56	8.30	29.33	88.88
	NCF	16.43	3.20	16.87	16.73	4.68	17.40	9.69	33.23	90.07
	EHCF	17.82	4.21	18.18	18.08	6.06	18.67	12.12	36.23	90.42
	UniS	12.46	2.40	12.60	12.98	3.72	13.31	8.47	27.10	91.84
	PopS	9.07	1.98	9.32	9.07	2.94	9.52	5.39	21.74	81.02
	2stS	12.42	2.27	12.80	12.64	3.46	13.27	8.43	27.48	89.96
	HarS	18.75	4.04	19.23	19.09	5.93	19.88	12.84	37.48	92.78
	TransCF	10.55	2.25	10.77	10.16	3.17	10.75	6.44	23.75	86.94
	LRML	15.37	2.91	15.78	15.84	4.37	16.43	9.25	31.67	90.21
	CRML	19.13	4.10	19.64	19.28	5.99	20.13	13.13	37.80	93.90
	NaiveCML	-	-	-	-	-	-	-	-	-
	SFCML(ours)	**22.71**	**5.39**	**23.18**	**22.66**	**7.56**	**23.49**	**15.30**	**42.88**	**94.02**

by EHCF) on the MovieLens-100k dataset are 2.27%, 1.49% and 1.36% with respect to P@3, MAP and MRR, respectively. This validates the superiority of our proposed SFCML algorithm.

- The deep learning-based algorithms (such NCF) show inferior performance than the sampling-based CML competitors on CiteULike and Amazon-Book datasets. A possible reason is that deep-learning-based models can only have access to sparse feedback information on CiteULike and Amazon-Book datasets, which makes them generalize poorly to the test set. The same reason also causes the degraded performance of LRML, where a memory-augmented neural architecture is applied to CML.
- With respect to the CML-based frameworks, the performance improvement between SFCML and other sampling-based CML is significant. The reason might be that the sampling-based CML essentially alters the intrinsic distribution of the training data. This results in a biased estimation for the expected risk over the unseen data and thus degrades the recommendation performance. On the contrary, SFCML, in a sampling-free manner, could boost the generalization performance. This confirms the arguments of this work and the effectiveness of SFCML.
- For the non-sampling algorithms, we see that, SFCML consistently outperforms the state-of-the-art sampling-free algorithm EHCF in most cases, which shows the superiority of our proposed SFCML method.

In summary, the above empirical discussions could support the aforementioned theoretical arguments, i.e., the sampling-based CML methods could not guarantee a small generalization error and sometimes result in sub-optimal generalization performance. By contrast, SFCML could

TABLE 4: Performance comparisons on Steam-200k and Anime datasets, where '-' means that we cannot complete the experiments due to the out-of-memory issue. The best and second-best are highlighted in bold and underlined, respectively.

	Method	P@3	R@3	NDCG@3	P@5	R@5	NDCG@5	MAP	MRR	AUC
Steam-200k	itemKNN	12.58	9.47	13.23	6.47	3.9	7.23	11.74	23.33	86.81
	GMF	9.28	3.85	9.52	12.94	5.73	13.41	7.32	22.35	87.25
	MLP	13.09	6.64	13.66	14.34	6.92	15.48	10.02	28.64	91.45
	NCF	12.97	6.58	13.58	14.26	6.90	15.42	10.10	28.71	91.51
	EHCF	22.76	13.33	24.07	21.13	10.87	22.57	19.98	43.82	93.17
	UniS	13.33	8.14	13.73	12.09	5.73	12.65	13.00	27.25	93.59
	PopS	18.84	11.54	19.51	16.23	8.26	17.09	15.33	35.38	84.46
	2stS	13.65	8.33	14.22	12.44	5.84	13.02	13.44	27.91	92.22
	HarS	20.27	11.50	20.87	20.80	10.04	21.39	18.16	37.97	94.17
	TransCF	15.23	9.48	15.93	12.88	6.75	13.74	13.64	31.04	91.91
	LRML	14.54	7.14	14.86	16.15	7.72	17.28	11.45	30.70	92.06
	CRML	20.51	11.40	21.47	21.10	10.17	22.27	18.42	39.46	94.19
	NaiveCML	25.78	15.22	27.01	**24.17**	**12.29**	**25.63**	21.67	46.74	**94.94**
	SFCML(ours)	**26.34**	**16.00**	**27.23**	23.76	12.25	24.84	**23.17**	**47.03**	94.58
Anime	itemKNN	16.93	3.51	17.15	16.21	4.97	16.67	9.79	33.76	93.52
	GMF	18.69	3.31	19.40	17.26	4.76	18.40	9.57	37.26	92.32
	MLP	20.45	3.71	21.24	19.13	5.35	20.32	10.95	39.96	93.94
	NCF	24.09	4.29	24.84	22.90	6.28	24.06	13.41	44.47	94.93
	EHCF	28.72	5.88	29.45	27.55	8.56	28.65	18.06	50.55	96.03
	UniS	14.81	2.33	15.16	14.24	3.48	14.83	9.30	30.76	94.99
	PopS	15.20	2.92	15.66	14.40	4.22	15.10	7.80	31.93	87.66
	2stS	16.86	2.84	17.28	16.31	4.27	16.96	10.34	34.39	93.72
	HarS	20.57	3.54	21.07	19.87	5.30	20.65	12.77	39.57	94.97
	TransCF	13.78	2.73	14.50	12.33	3.69	13.42	7.44	29.82	91.21
	LRML	17.69	3.06	18.42	16.49	4.41	17.56	9.17	36.03	92.25
	CRML	27.05	5.05	27.73	25.93	7.41	27.00	17.23	48.16	96.74
	NaiveCML	-	-	-	-	-	-	-	-	-
	SFCML(ours)	**30.31**	**6.36**	**30.82**	**29.50**	**9.37**	**30.30**	**20.48**	**52.12**	**97.04**

boost the performance since it can get rid of this bias via the sampling-free algorithm.

6.5.2 Comparison against NaiveCML

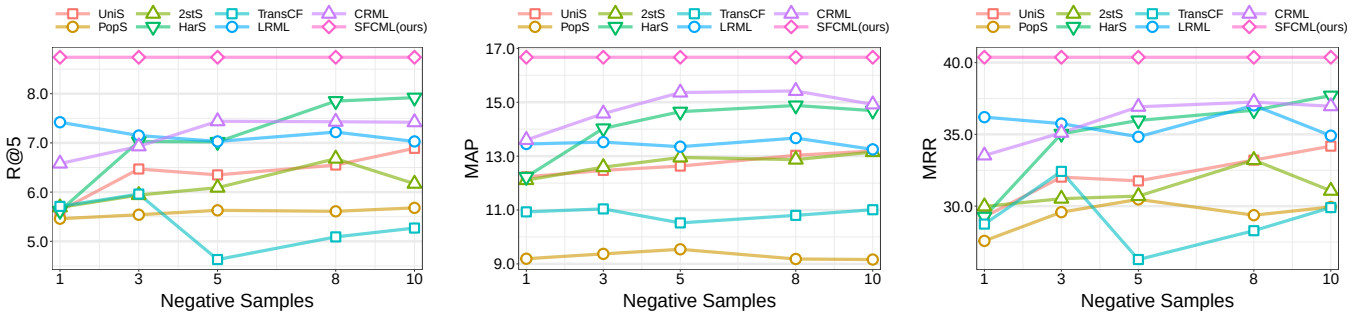
In order to further demonstrate the effectiveness of the acceleration of SFCML, we also report the performance comparison between SFCML and NaiveCML. Unfortunately, as we discussed in Rem.3, NaiveCML leads to almost $\mathcal{O}(\sum_{i=1}^{M} n_i^+ n_i^-) = \mathcal{O}(MN^2)$ space complexity to store the contrastive triplets, while SFCML achieves almost $\mathcal{O}(MN)$ time and space complexity. Therefore, due to the out-of-memory exception, we only conduct the experiments of NaiveCML on MovieLens-100k and Steam-200k datasets. The performance results are reported in Tab.3, Tab.4, Fig.1 and Fig.8(c) (in the Appendix.C.3). From the empirical results, we can observe that, SFCML can achieve comparable performance against NaiveCML. Since different loss functions usually lead to different optimization goals, it is reasonable that there exists a slight performance gap between SFCML and NaiveCML. This warrants the rationality and effectiveness of our proposed algorithms.

6.5.3 Adverse evidence of sampling-based CML

As we argued in Sec.3.3, the recommendation performance of sampling-based CML is largely determined by the negative sampling strategy and the number of sampled items. Since it is difficult to find the best sampling strategy and the number of sampled items, this makes the sampling-based CML methods perform unstably. To validate this claim, we report the performance of different negative sampling strategies and the sampling number U on validation set of the MovieLens-100k dataset. Fig.2 shows the empirical results in terms of CML framework competitors. The sampling number U is chosen from $\{1, 3, 5, 8, 10\}$. As depicted in Fig.2, we can conclude that, given a fixed sampling number, choosing different negative sampling strategies end up with quite different performances. This is due to the fact that different sampling strategies utilize different items to optimize the model. Moreover, given a fixed sampling strategy, the sampling-based CML algorithms also exhibit quite different performances across different U. Theoretically, altering different negative sampling strategies and the number of sampled items both induce a different sampling distribution, resulting in a nonzero D_{TV} term in its generalization upper bound (see Thm.4). This may lead to different generalization performance for sampling-based CML. By contrast, SFCML consistently outperforms all its sampling-based counterparts through the lens of a sampling-free fashion.

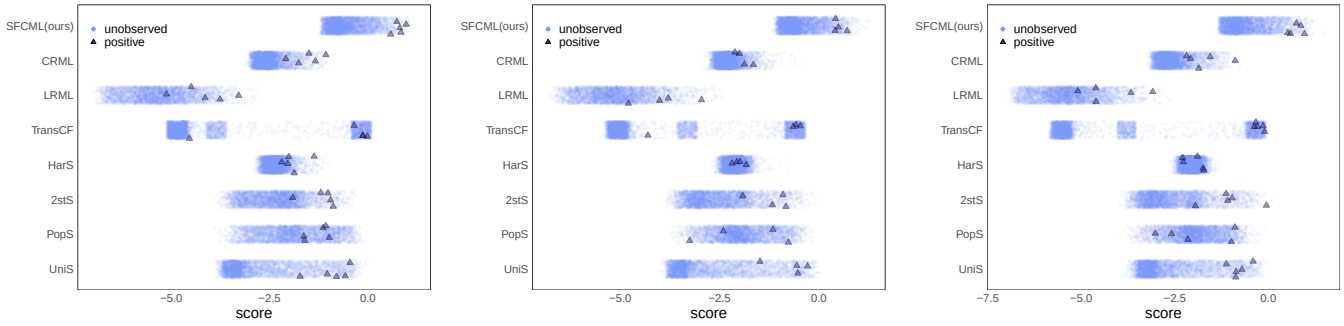
6.5.4 Fine-grained Performance Visualization

Recall that, in Sec.3.2, with respect to user u_i, we say that a contrastive pair (v_j^+, v_k^-) meets the preference consistency if the score $f(v_j^+|u_i) > f(v_k^-|u_i)$. This could be exactly measured by AUC, as discussed in Sec.5.2. According to this, at first, we separately report the fine-grained AUC comparison with respect to six users on the MovieLens-1m and Anime datasets, respectively. Specifically, in terms of each positive item, we evaluate its AUC score against all remaining unobserved items. The results are shown in Fig.4. We can notice that the AUC values of SFCML are consistently higher than the sampling-based competitors.



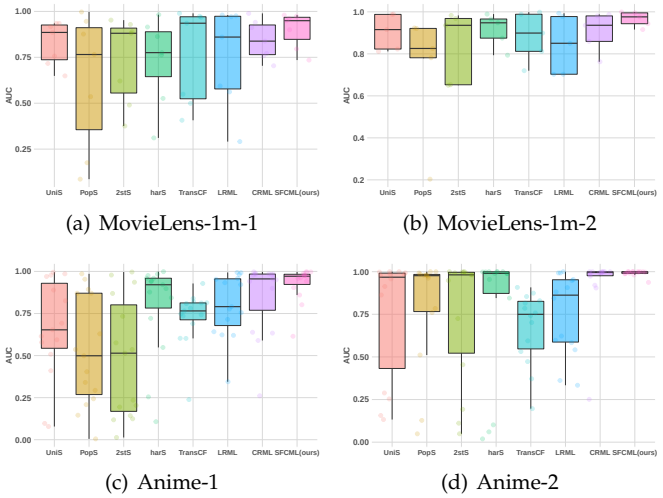
(a) R@5 (b) MAP (c) MRR

Fig. 2: Performance comparisons on validation set of MovieLens-100k with respect to different negative sampling strategies and different sampling numbers $U \in \{1, 3, 5, 8, 10\}$. Please refer to Appendix.C.2 for more results.



(a) MovieLens-1m-1 (b) MovieLens-1m-2 (c) MovieLens-1m-3

Fig. 3: The graphical visualization of score distribution of positive and unobserved items on MovieLens-1m.



(a) MovieLens-1m-1 (b) MovieLens-1m-2

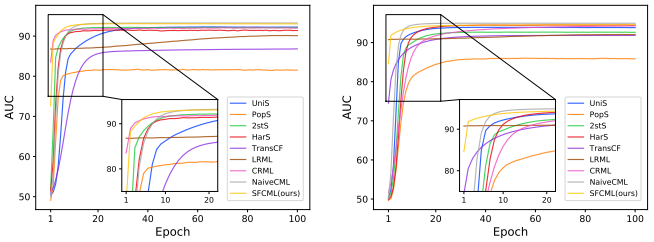
(c) Anime-1 (d) Anime-2

Fig. 4: Fine-grained AUC performance with respect to two users on MovieLens-1m and Anime, respectively. Please refer to Appendix.C.4.1 for more visualizations.

This indicates that SFCML tends to provide a better preference consistency. By contrast, in most cases, the sampling-based CML results in a heavy tail positive score distribution (i.e., lower AUC score of positive items). This is because they merely leverage a small part of unobserved items to train CML while ignoring other informative samples. In this way, the relative preference consistency could not be well preserved, leading to the degradation of performance. Secondly, we separately select three users on these two datasets. Then, for each user, we plot all of their scores in a scatted plot. The visualization results are presented in Fig.3 and Fig.11 in Appendix.C.4.1. The dots here are unobserved instances, and the triangles are positive ones. In the plots, the unobserved examples form a long band, and the dark part of the band corresponds to a higher density of the unobserved distribution. For our algorithm, we see that the triangles tend to form clear cliques and they only overlap with the top and light part of the unobserved band. While for other algorithms, the triangles either span the band or overlap with a dark part of the band. This shows that our proposed method could better separate the positive and unobserved examples apart.

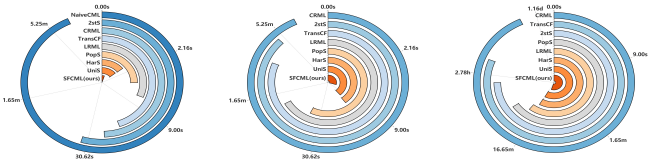
6.5.5 *Empirical analysis*

Fig.5 reports the convergence of AUC on MovieLens-100k and Steam-200k datasets. Grounded on the results, we can observe that, with regard to most algorithms, the AUC metric increases to a high value rapidly over only several epochs, and then tends to be stable. Especially, LRML can achieve a competitive performance over fewer iterations owing to the strong expression of its attention-based neural network. Moreover, different sampling-based CML algorithms require different epochs to converge, since they



(a) MovieLens-100k (b) Steam-200k

Fig. 5: Empirical converge analysis of testing AUC of all CML-based algorithms, reporting 100 epochs here.



(a) MovieLens-100k (b) MovieLens-1m (c) MovieLens-20m

Fig. 6: Comparisons against average running time with respect to CML framework algorithms and SFCML(ours). The method closer to the center of the circle enjoys better efficiency. Note that, here the 's', 'm', 'h' and 'd' represent the second, minute, hour and day respectively. Please see Appendix.C.6 for more results.

TABLE 5: Statistics information of ml-100k dataset with different preference thresholds t.

Thresholds	#Users	#Items	#Ratings	%Density
t=1	943	1,682	100,000	6.3047
t=2	943	1,612	93,890	5.9264
t=3	943	1,574	82,520	5.5596
t=4	938	1,447	55,361	4.0788
t=5	779	1,172	20,805	2.2788

usually leverage different negative items to train the model. Finally, SFCML performs consistently better than other competitors and shows a close performance to the NaiveCML algorithm.

6.5.6 Comparison against the efficiency

In order to show the efficiency improvements of SFCML against other competitors, we also report the running time of CML competitors, including: a) **UniS** b) **PopS** c) **2stS** d) **HarS** e) **TransCF** f) **LRML** g) **CRML** h) **NaiveCML** and i) ours **SFCML**. Notably, for all sampling-based CML methods, we set the sampling number $U = 10$.

Fig.6 shows the average running time over 10 epochs on all benchmark datasets, and the concrete training time per epoch could be found in the Appendix C.6. Let us first define the following acceleration ratio (acc.r):

$$\text{acc.r} = \frac{\text{Running time of the slower algorithm}}{\text{Running time of the faster algorithm}}$$

According to Fig.6, we can draw the following conclusions. At first, SFCML demonstrates even higher efficiency

TABLE 6: The empirical performance of AUC with respect to different preference thresholds $t \in \{1, 2, 3, 4, 5\}$ on MovieLens-100k. The best and second-best performances are highlighted in bold and underlined, respectively. Please refer to Appendix.C.5 for more results.

Method	Trend	t=1	t=2	t=3	t=4	t=5
itemKNN		82.60	84.09	83.93	85.68	76.23
GMF		84.08	83.82	83.74	86.12	83.34
MLP		85.11	85.21	85.05	87.09	84.93
NCF		87.52	88.03	87.52	88.03	84.68
EHCF		91.59	91.42	91.27	92.18	85.55
UniS		89.59	91.42	89.85	92.27	86.08
PopS		80.53	80.13	79.97	80.51	74.84
2stS		90.21	91.19	89.87	92.01	85.84
HarS		90.91	91.02	90.54	91.66	88.62
TransCF		85.75	85.53	85.12	87.53	84.72
LRML		90.37	89.97	89.97	90.38	85.43
CRML		90.76	90.48	89.58	92.07	87.72
SFCML(ours)		**92.85**	**92.97**	**92.66**	**93.11**	**89.11**

than the sampling-based CML on MovieLens-100k (4.1x speed-up against the second-best algorithm), MovieLens-1m (1.9x speed-up), Anime (4.7x speed-up) and MovieLens-20m (1.4x speed-up) datasets. The possible reason lies in that, since the sampling-based algorithms usually need to traverse all observed user-item interactions and then sample unobserved items to generate contrastive pairs for each interaction, this hurts their efficiency for datasets with dense interactions. Accordingly, for the medium/large datasets, such as Anime and MovieLens-20m, TransCF and CRML are much more inefficient than SFCML. Last but not least, compared with the NaiveCML, SFCML significantly reduces the running time without any negative sampling strategies. Although the experiments of NaiveCML on CiteULike, MovieLens-1m, Anime, MovieLens-20m and Amazon-Book could not be finished due to the out-of-memory issue, the efficiency gaps between SFCML and NaiveCML are already sharp on MovieLens-100k and Steam-200k datasets, where the improvements are up to 1114.1x speed-up and 743.0x speed-up on MovieLens-100k and Steam-200k, respectively. This validates the effectiveness of proposed accelerations in Sec.5.2, making it possible to learn from the whole data under a relatively acceptable efficiency.

6.5.7 Sensitivity Analysis of Preference thresholds

As we mentioned in Sec.6.1, to exert the explicit feedback to develop the implicit-feedback-driven recommendation, we follow a widely adopted setting in RS [24], [33], [59], [86]. Concretely, if the score of item v_j rated by user u_i is no less than a preference thresholds t, then v_j is treated as a positive item for u_i. To figure out the influence of t to the performance, we conduct the sensitivity analysis on MovieLens-100k data with different preference thresholds $t \in \{1, 2, 3, 4, 5\}$. The corresponding statistics of data used in this experiment are listed in Tab.5 and the empirical results of all algorithms are shown in Tab.6. Grounded on these results, we have the following conclusions. Firstly, we see that, almost all the algorithms achieve the best AUC performance at $t = 4$, which means that the models enjoy the best preference consistency at this threshold. Secondly, for the rest of the metrics, most algorithms tend to

demonstrate a similar trend except for the R@K metric with respect to all thresholds. According to the calculations of metrics in Appendix.C.1, we know that both numerator and denominator of R@K are related to the number of positive interactions $|I_{u_i}|$ in the dataset, while the other metrics are only influenced by the numerator. Since the numerator for all the involved metrics is inversely proportional to t, we see that, in most cases, all the metrics except R@K are inversely proportional to t correspondingly. Most importantly, even under different preference thresholds, SFCML still achieves the best performance consistently on all metrics, and the performance improvements are significant compared with other competitors. This further supports the superiority of SFCML.

7 Conclusion

In this paper, we study the issue of sampling-based CML framework and then start an early trial to develop an efficient alternative for CML without negative sampling. Specifically, based on the extended Rademacher Complexity and the specifically designed symmetrization regime, we provide a systematic analysis of the generalization ability of the CML framework. The theoretical analysis shows that the sampling-based CML may fail to obtain a reasonable generalization performance, due to the per-user TV bias term in its generalization upper bound. Meanwhile, we also prove that the biased term would be eliminated in a sampling-free manner. Motivated by this, we propose to learn CML without negative sampling to get rid of the bias and construct an acceleration method to overcome the heavy computational burden. Finally, empirical studies conducted on seven benchmark datasets demonstrate the superiority of our proposed SFCML algorithm.

8 Acknowledgment

This work was supported in part by the National Key R&D Program of China under Grant 2018AAA0102003, in part by National Natural Science Foundation of China: U21B2038, U2001202, U1936208, 61620106009, 62025604, 61931008, 6212200758 and 61976202, in part by the Fundamental Research Funds for the Central Universities, in part by the National Postdoctoral Program for Innovative Talents under Grant BX2021298, in part by the Youth Innovation Promotion Association CAS, in part by the Strategic Priority Research Program of Chinese Academy of Sciences, Grant No. XDB28000000.

References

[1] Y. Cao, X. Wang, X. He, Z. Hu, and T. Chua, "Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences," in *WWW*, 2019, pp. 151–161.
[2] X. Wang, X. He, L. Nie, and T.-S. Chua, "Item silk road: Recommending items from information domains to social users," in *SIGIR*, 2017, pp. 185–194.
[3] C. Wang, T. Zhou, C. Chen, T. Hu, and G. Chen, "Off-policy recommendation system without exploration," in *PAKDD*, vol. 12084. Springer, 2020, pp. 16–27.
[4] J. Ma, C. Zhou, P. Cui, H. Yang, and W. Zhu, "Learning disentangled representations for recommendation," in *NeurIPS*, 2019, pp. 5712–5723.
[5] P. Cui, F. Wang, S. Liu, M. Ou, S. Yang, and L. Sun, "Who should share what?: item-level social influence prediction for users and posts ranking," in *SIGIR*, 2011, pp. 185–194.
[6] J. Ma, C. Zhou, H. Yang, P. Cui, X. Wang, and W. Zhu, "Disentangled self-supervision in sequential recommenders," in *KDD*, 2020, pp. 483–491.
[7] X. He and T. Chua, "Neural factorization machines for sparse predictive analytics," in *SIGIR*, 2017, pp. 355–364.
[8] Y. Lv, Y. Zheng, F. Wei, C. Wang, and C. Wang, "AICF: attention-based item collaborative filtering," *Adv. Eng. Informatics*, vol. 44, pp. 101 090:1–11, 2020.
[9] D. W. Oard and J. Kim, "Implicit feedback for recommender systems," in *AAAI*, 1998, pp. 81–83.
[10] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *ICDM*, 2008, pp. 263–272.
[11] L. Lerche, "Using implicit feedback for recommender systems: characteristics, applications, and challenges," in *Doctoral dissertation*, 2016.
[12] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *CSUR*, vol. 52, no. 1, 2019.
[13] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. Artif. Intell.*, vol. 2009, pp. 421 425:1–421 425:19, 2009.
[14] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. M. Lukose, M. Scholz, and Q. Yang, "One-class collaborative filtering," in *ICDM*, 2008, pp. 502–511.
[15] Y. Yao, H. Tong, G. Yan, F. Xu, X. Zhang, B. K. Szymanski, and J. Lu, "Dual-regularized one-class collaborative filtering with implicit feedback," *WWW*, vol. 22, no. 3, pp. 1099–1129, 2019.
[16] R. Heckel and K. Ramchandran, "The sample complexity of online one-class collaborative filtering," in *ICML*, 2017, pp. 1452–1460.
[17] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, and T. Chua, "Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention," in *SIGIR*, 2017, pp. 335–344.
[18] X. Xin, X. He, Y. Zhang, Y. Zhang, and J. M. Jose, "Relational collaborative filtering: Modeling multiple item relations for recommendation," in *SIGIR*, 2019, pp. 125–134.
[19] W. Wang, F. Feng, X. He, L. Nie, and T. Chua, "Denoising implicit feedback for recommendation," in *WSDM*, 2021, pp. 373–381.
[20] X. He, X. Du, X. Wang, F. Tian, J. Tang, and T. Chua, "Outer product-based neural collaborative filtering," in *IJCAI*, 2018, pp. 2227–2233.
[21] Q. Zhang and F. Ren, "Prior-based bayesian pairwise ranking for one-class collaborative filtering," *Neurocomputing*, vol. 440, pp. 365–374, 2021.
[22] J. Chen, D. Lian, and K. Zheng, "Improving one-class collaborative filtering via ranking-based implicit regularizer," in *AAAI*, 2019, pp. 37–44.
[23] G. Li, Z. Zhang, L. Wang, Q. Chen, and J. Pan, "One-class collaborative filtering based on rating prediction and ranking prediction," *Knowl. Based Syst.*, vol. 124, pp. 46–54, 2017.
[24] C.-K. Hsieh, L. Yang, Y. Cui, T.-Y. Lin, S. Belongie, and D. Estrin, "Collaborative metric learning," in *WWW*, 2017, pp. 193–201.
[25] F. Liu, Z. Cheng, C. Sun, Y. Wang, L. Nie, and M. S. Kankanhalli, "User diverse preference modeling by multimodal attentive metric learning," in *MM*, 2019, pp. 1526–1534.
[26] S. Janarthan, S. Thuseethan, S. Rajasegarar, Q. Lyu, Y. Zheng, and J. Yearwood, "Deep metric learning based citrus disease classification with sparse data," *IEEE Access*, vol. 8, pp. 162 588–162 600, 2020.
[27] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua, "Neural collaborative filtering," in *WWW*, 2017, pp. 173–182.
[28] X. Wang, X. He, M. Wang, F. Feng, and T. Chua, "Neural graph collaborative filtering," in *SIGIR*, 2019, pp. 165–174.
[29] X. Wang, X. He, Y. Cao, M. Liu, and T. Chua, "KGAT: knowledge graph attention network for recommendation," in *KDD*, 2019, pp. 950–958.
[30] C. Chen, M. Zhang, Y. Liu, and S. Ma, "Social attentional memory network: Modeling aspect- and friend-level differences in recommendation," in *WSDM*, 2019, pp. 177–185.
[31] T. Chen, Y. Sun, Y. Shi, and L. Hong, "On sampling strategies for neural network-based collaborative filtering," in *KDD*, 2017, pp. 767–776.
[32] G. Wu, M. Volkovs, C. L. Soon, S. Sanner, and H. Rai, "Noise contrastive estimation for one-class collaborative filtering," in *SIGIR*, 2019, pp. 135–144.

[33] V.-A. Tran, R. Hennequin, J. Royo-Letelier, and M. Moussallam, "Improving collaborative metric learning with efficient negative sampling," in *SIGIR*, 2019, pp. 1201–1204.

[34] O. Canévet and F. Fleuret, "Efficient sample mining for object detection," in *ACML*, 2014, pp. 48–63.

[35] J. Wang, L. Yu, W. Zhang, Y. Gong, Y. Xu, B. Wang, P. Zhang, and D. Zhang, "IRGAN: A minimax game for unifying generative and discriminative information retrieval models," in *SIGIR*, 2017, pp. 515–524.

[36] D. H. Park and Y. Chang, "Adversarial sampling and training for semi-supervised information retrieval," in *WWW*, 2019, pp. 1443–1453.

[37] J. Ding, Y. Quan, X. He, Y. Li, and D. Jin, "Reinforced negative sampling for recommendation with exposure data," in *IJCAI*, 2019, pp. 2230–2236.

[38] C. Park, D. Kim, X. Xie, and H. Yu, "Collaborative translational metric learning," in *ICDM*, 2018, pp. 367–376.

[39] Y. Tay, L. A. Tuan, and S. C. Hui, "Latent relational metric learning via memory-based attention for collaborative ranking," in *WWW*, 2018, pp. 729–739.

[40] H. Wu, Q. Zhou, R. Nie, and J. Cao, "Effective metric learning with co-occurrence embedding for collaborative recommendations," *Neural Networks*, vol. 124, pp. 308–318, 2020.

[41] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*, 2014.

[42] P. L. Bartlett and S. Mendelson, "Rademacher and gaussian complexities: Risk bounds and structural results," in *COLT*, vol. 2111, 2001, pp. 224–240.

[43] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. MIT Press, 2012.

[44] W. Gao, R. Jin, S. Zhu, and Z. Zhou, "One-pass AUC optimization," in *ICML*, vol. 28, 2013, pp. 906–914.

[45] W. Gao and Z. Zhou, "On the consistency of AUC pairwise optimization," in *IJCAI*, 2015, pp. 939–945.

[46] R. Pan and M. Scholz, "Mind the gaps: weighting the unknown in large-scale one-class collaborative filtering," in *KDD*, 2009, pp. 667–676.

[47] U. Paquet and N. Koenigstein, "One-class collaborative filtering with random graphs," in *WWW*, 2013, pp. 999–1008.

[48] V. Sindhwani, S. S. Bucak, J. Hu, and A. Mojsilovic, "One-class matrix completion with low-density factorizations," in *ICDM*, 2010, pp. 1055–1060.

[49] N. Pappas and A. Popescu-Belis, "Sentiment analysis of user comments for one-class collaborative filtering over ted talks," in *SIGIR*, 2013, pp. 773–776.

[50] Y. Fang and L. Si, "Matrix co-factorization for recommendation with rich side information and implicit feedback," in *Hetrec*, 2011, pp. 65–69.

[51] W. Pan and L. Chen, "Gbpr: Group preference based bayesian personalized ranking for one-class collaborative filtering," in *IJCAI*, 2013, pp. 2691–2697.

[52] R. He and J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in *WWW*, 2016, pp. 507–517.

[53] X. He, H. Zhang, M. Kan, and T. Chua, "Fast matrix factorization for online recommendation with implicit feedback," in *SIGIR*, 2016, pp. 549–558.

[54] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: bayesian personalized ranking from implicit feedback," in *UAI*, 2009, pp. 452–461.

[55] B. C. Turnbull, "Learning intent to book metrics for airbnb search," in *WWW*, 2019, pp. 3265–3271.

[56] F. Xu, W. Zhang, Y. Cheng, and W. Chu, "Metric learning with equidistant and equidistributed triplet-based loss for product image search," in *WWW*, 2020, pp. 57–65.

[57] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *CVPR*, 2017, pp. 6738–6746.

[58] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *JMLR*, vol. 10, no. 2, pp. 207–244, 2009.

[59] S. Bao, Q. Xu, K. Ma, Z. Yang, X. Cao, and Q. Huang, "Collaborative preference embedding against sparse labels," in *ACM MM*, 2019, pp. 2079–2087.

[60] W. Li, M. Gao, W. Rong, J. Wen, Q. Xiong, R. Jia, and T. Dou, "Social recommendation using euclidean embedding," in *IJCNN*, 2017, pp. 589–595.

[61] C. Park, D. Kim, X. Xie, and H. Yu, "Collaborative translational metric learning," in *ICDM*, 2018, pp. 367–376.

[62] S. Feng, X. Li, Y. Zeng, G. Cong, Y. M. Chee, and Q. Yuan, "Personalized ranking metric embedding for next new poi recommendation," in *IJCAI*, 2015.

[63] Z. Yang, M. Ding, C. Zhou, H. Yang, J. Zhou, and J. Tang, "Understanding negative sampling in graph representation learning," in *KDD*, 2020, pp. 1666–1676.

[64] J. Ding, F. Feng, X. He, G. Yu, Y. Li, and D. Jin, "An improved sampler for bayesian personalized ranking by leveraging view data," in *WWW*, 2018, pp. 13–14.

[65] X. Wang, Y. Xu, X. He, Y. Cao, M. Wang, and T. Chua, "Reinforced negative sampling over knowledge graph for recommendation," in *WWW*, 2020, pp. 99–109.

[66] X. He, J. Tang, X. Du, R. Hong, T. Ren, and T.-S. Chua, "Fast matrix factorization with nonuniform weights on missing data," *TNNLS*, vol. 31, no. 8, pp. 2791–2804, 2019.

[67] S. Rendle and C. Freudenthaler, "Improving pairwise learning for item recommendation from implicit feedback," in *WSDM*, 2014, pp. 273–282.

[68] F. Yuan, X. Xin, X. He, G. Guo, W. Zhang, T. Chua, and J. M. Joemon, "f_{bgd}: Learning embeddings from positive unlabeled data with BGD," in *UAI*, 2018, pp. 198–207.

[69] C. Chen, M. Zhang, W. Ma, Y. Liu, and S. Ma, "Jointly non-sampling learning for knowledge graph enhanced recommendation," in *SIGIR*, 2020, pp. 189–198.

[70] X. Xin, F. Yuan, X. He, and J. M. Jose, "Batch IS NOT heavy: Learning word representations from all samples," in *ACL*, 2018, pp. 1853–1862.

[71] C. Chen, M. Zhang, Y. Zhang, Y. Liu, and S. Ma, "Efficient neural matrix factorization without sampling for recommendation," *T-IS*, vol. 38, no. 2, pp. 14:1–14:28, 2020.

[72] C. Chen, M. Zhang, Y. Zhang, W. Ma, Y. Liu, and S. Ma, "Efficient heterogeneous collaborative filtering without negative sampling for recommendation," in *AAAI*, 2020, pp. 19–26.

[73] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi, "Personalizing session-based recommendations with hierarchical recurrent neural networks," in *RecSys*, 2017, pp. 130–137.

[74] X. Chen, H. Xu, Y. Zhang, J. Tang, Y. Cao, Z. Qin, and H. Zha, "Sequential recommendation with user memory networks," in *WSDM*, 2018, pp. 108–116.

[75] P. Cui, X. Wang, J. Pei, and W. Zhu, "A survey on network embedding," *TKDE*, vol. 31, no. 5, pp. 833–852, 2019.

[76] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. J. Russell, "Distance metric learning with application to clustering with side-information," in *NeurIPS*, 2002, pp. 505–512.

[77] B. Kulis, "Metric learning: A survey," *Found. Trends Mach. Learn.*, vol. 5, no. 4, pp. 287–364, 2013.

[78] S. Rendle, W. Krichene, L. Zhang, and J. R. Anderson, "Neural collaborative filtering vs. matrix factorization revisited," in *RecSys*, 2020, pp. 240–248.

[79] B. Zhou, Y. Ying, and S. Skiena, "Online AUC optimization for sparse high-dimensional datasets," in *ICDM*, 2020, pp. 881–890.

[80] Z. Yang, W. Shen, Y. Ying, and X. Yuan, "Stochastic auc optimization with general loss," *Communications on Pure & Applied Analysis*, vol. 19, no. 8, 2020.

[81] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (roc) curve." *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[82] C. Cortes and M. Mohri, "Auc optimization vs. error rate minimization," in *NeurIPS*, 2003, pp. 313–320.

[83] P. Zhao, S. C. H. Hoi, R. Jin, and T. Yang, "Online AUC maximization," in *ICML*, 2011, pp. 233–240.

[84] Y. Ying, L. Wen, and S. Lyu, "Stochastic online auc maximization," in *NeurIPS*, 2016, pp. 451–459.

[85] S. Lyu and Y. Ying, "A univariate bound of area under ROC," in *UAI*, 2018, pp. 43–52.

[86] H. Wang, N. Wang, and D. Yeung, "Collaborative deep learning for recommender systems," in *KDD*, 2015, pp. 1235–1244.

[87] H. Wang, B. Chen, and W. Li, "Collaborative topic regression with social regularization for tag recommendation," in *IJCAI*, 2013, pp. 2719–2725.

[88] R. He and J. J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in *WWW*, 2016, pp. 507–517.

[89] G. Linden, B. Smith, and J. York, "Amazon. com recommendations: Item-to-item collaborative filtering," *IEEE Internet computing*, no. 1, pp. 76–80, 2003.
[90] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *WWW*, 2001, pp. 285–295.
[91] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NeurIPS Workshop*, 2017.
[92] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *JMLR*, vol. 12, no. 7, pp. 2121–2159, 2011.
[93] C. McDiarmid, "Concentration," in *Probabilistic methods for algorithmic discrete mathematics*, 1998, pp. 195–248.

Shilong Bao received the B.S. degree in College of Computer Science and Technology from Qingdao University in 2019. He is currently pursuing the Ph.D. degree with University of Chinese Academy of Sciences. His research interest is machine learning and data mining. He has authored or coauthored several academic papers in top-tier international conferences and journals including T-PAMI, ICML, and ACM Multimedia.

Qianqian Xu received the B.S. degree in computer science from China University of Mining and Technology in 2007 and the Ph.D. degree in computer science from University of Chinese Academy of Sciences in 2013. She is currently an Associate Professor with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. Her research interests include statistical machine learning, with applications in multimedia and computer vision. She has authored or coauthored 50+ academic papers in prestigious international journals and conferences (including T-PAMI, IJCV, T-IP, NeurIPS, ICML, CVPR, AAAI, etc). Moreover, she has served as the Senior Program Committee (SPC) of AAAI and IJCAI, Area Chair of ACM MM and ICME, and Reviewer of many leading journals and conferences (including TPAMI, TNNLS, TMM, TCSVT, ICML, NeurIPS, CVPR, ICCV, AAAI, IJCAI, ACM Multimedia, and ICLR).

Zhiyong Yang received the M.Sc. degree in computer science and technology from University of Science and Technology Beijing (USTB) in 2017, and the Ph.D. degree from University of Chinese Academy of Sciences (UCAS) in 2021. He is currently a postdoctoral research fellow with the University of Chinese Academy of Sciences. His research interests lie in machine learning and learning theory, with special focus on AUC optimization, meta-learning/multi-task learning, and learning theory for recommender systems. He has authored or coauthored several academic papers in top-tier international conferences and journals including T-PAMI/ICML/NeurIPS/CVPR. He served as a reviewer for several top-tier journals and conferences such as T-PAMI, ICML, NeurIPS and ICLR.

Xiaochun Cao , Professor of the Institute of Information Engineering, Chinese Academy of Sciences. He received the B.E. and M.E. degrees both in computer science from Beihang University (BUAA), China, and the Ph.D. degree in computer science from the University of Central Florida, USA, with his dissertation nominated for the university level Outstanding Dissertation Award. After graduation, he spent about three years at ObjectVideo Inc. as a Research Scientist. From 2008 to 2012, he was a professor at Tianjin University. He has authored and coauthored over 100 journal and conference papers. In 2004 and 2010, he was the recipients of the Piero Zamperoni best student paper award at the International Conference on Pattern Recognition. He is a fellow of IET and a Senior Member of IEEE. He is an associate editor of IEEE Transactions on Image Processing, IEEE Transactions on Circuits and Systems for Video Technology and IEEE Transactions on Multimedia.

Qingming Huang is a chair professor in the University of Chinese Academy of Sciences and an adjunct research professor in the Institute of Computing Technology, Chinese Academy of Sciences. He graduated with a Bachelor degree in Computer Science in 1988 and Ph.D. degree in Computer Engineering in 1994, both from Harbin Institute of Technology, China. His research areas include multimedia computing, image processing, computer vision and pattern recognition. He has authored or coauthored more than 400 academic papers in prestigious international journals and top-level international conferences. He was the associate editor of IEEE Trans. on CSVT and Acta Automatica Sinica, and the reviewer of various international journals including IEEE Trans. on PAMI, IEEE Trans. on Image Processing, IEEE Trans. on Multimedia, etc. He is a Fellow of IEEE and has served as general chair, program chair, area chair and TPC member for various conferences, including ACM Multimedia, CVPR, ICCV, ICME, ICMR, PCM, BigMM, PSIVT, etc.

CONTENTS

APPENDIX A
PRELIMINARY FOR GENERALIZATION ANALYSIS

A.1 Preliminary Lemmas

In this section, we first briefly review some preparatory knowledge for the proof.

Definition 4 (Bounded Difference Property). *Given a group of independent random variables $X_1, X_2, \cdots, X_n$ where $X_t \in \mathbb{X}, \forall t$, $f(X_1, X_2, \cdots, X_n)$ is satisfied with the bounded difference property, if there exists some non-negative constants $c_1, c_2, \cdots, c_n$, such that:*

$$\sup_{x_1, x_2, \cdots, x_n, x_t'} |f(x_1, \cdots, x_n) - f(x_1, \cdots, x_{t-1}, x_t', \cdots, x_n)| \leq c_t, \ \forall t, 1 \leq t \leq n. \tag{28}$$

Hereafter, if any functions f holds the Bounded Difference Property, the following Mcdiarmid's inequality is always satisfied.

Lemma 1 (Mcdiarmid's Inequality [93]). *Assume we have n independent random variables $X_1, X_2, \ldots, X_n$ that all of them are chosen from the set $\mathcal{X}$. For a function $f : \mathcal{X} \to \mathbb{R}$, $\forall t, 1 \leq t \leq n$, if the following inequality holds:*

$$\sup_{x_1, x_2, \cdots, x_n, x_t'} |f(x_1, \cdots, x_n) - f(x_1, \cdots, x_{t-1}, x_t', \cdots, x_n)| \leq c_t, \ \forall t, 1 \leq t \leq n.$$

with $\boldsymbol{x} \neq \boldsymbol{x}'$, then for all $\epsilon > 0$, we have

$$\mathbb{P}[\mathbb{E}(f) - f \geq \epsilon] \leq \exp\left(\frac{-2\epsilon^2}{\sum_{t=1}^n c_t^2}\right),$$

$$\mathbb{P}[f - \mathbb{E}(f) \geq \epsilon] \leq \exp\left(\frac{-2\epsilon^2}{\sum_{t=1}^n c_t^2}\right).$$

Lemma 2 (ϕ-Lipschitz Continuous). *Given a set $\mathcal{X}$ and a function $f : \mathcal{X} \to \mathbb{R}$, if f is continuously differentiable on $\mathcal{X}$ and the derivative of f is Lipschitz continuous on $\mathcal{X}$ with constant μ:*

$$\|f(x) - f(y)\| \leq \phi \|x - y\|.$$

Thereafter, f is said to a ϕ-Lipschitz continuous function.

Lemma 3 (Talagrand Contraction Lemma). *Let $h_1, h_2, \ldots, h_n$ be a series of ϕ-Lipschitz continuous function and $\sigma_1, \ldots, \sigma_n$ be the independent Rademacher random variables, the following holds*

$$\frac{1}{n}\mathop{\mathbb{E}}_{\sigma}\left[\sup_{f\in\mathcal{H}}\sum_{i=1}^{n}\sigma_i \cdot (h_i \circ f)(z)\right] \leq \frac{\phi}{n}\mathop{\mathbb{E}}_{\sigma}\left[\sup_{f\in\mathcal{H}}\sum_{i=1}^{n}\sigma_i \cdot f(z)\right]$$

APPENDIX B
GENERALIZATION BOUNDS OF CML FRAMEWORK AND ITS PROOFS

Restate of Definition 2. *(CML **Rademacher Complexity**). Given the sample set $\mathcal{S} = \bigcup_{u_i \in \mathcal{U}} \mathcal{S}_i$ and $\mathcal{S}_i = \{v_j^+\}_{j=1}^{n_i^+} \cup \{v_k^-\}_{k=1}^{n_i^-}$ with $n_i^+ + n_i^- = N$ and the hypothesis space $\mathcal{H}_R$, then the empirical CML Rademacher Complexity with respect to the sample $\mathcal{S}$ is defined as:*

$$\hat{\mathfrak{R}}_{\ell,\mathcal{S}}^{cml}(\mathcal{H}_R) = \frac{1}{M}\sum_{u_i \in \mathcal{U}} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sup_{\mathcal{H}_R} \frac{1}{n_i^+ n_i^-}\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-} \mathcal{Q}_{(i)}^{jk}\right], \tag{29}$$

where

$$\mathcal{Q}_{(i)}^{jk} = \frac{\sigma_{ij}^+ + \sigma_{ik}^-}{2} \cdot \ell^{(i)}(v_j^+, v_k^-);$$

$\boldsymbol{\sigma}_i = [\sigma_{i1}^+, \sigma_{i2}^+, \ldots, \sigma_{in_i^+}^+, \sigma_{i1}^-, \sigma_{i2}^-, \ldots, \sigma_{in_i^-}^-]^\top$ is i.i.d Rademacher random variables uniformly chosen from $\{-1, +1\}$ and we have $\mathbb{P}(\sigma = 1) = \mathbb{P}(\sigma = -1) = 0.5$. Next, the population version of the Rademacher Complexity of CML is represented as $\mathfrak{R}_{\ell,\mathcal{S}}^{cml}(\mathcal{H}_R) = \mathop{\mathbb{E}}_{\mathcal{S}}\left[\hat{\mathfrak{R}}_{\ell,\mathcal{S}}^{cml}(\mathcal{H}_R)\right]$.

B.1 CML Symmetrization scheme

In this section, we provide a brief proof of our extended symmetrization scheme for CML framework. The basic idea is that exchanging item instead of term does not change the value of

$$\mathop{\mathbb{E}}_{\mathcal{S},\mathcal{S}'}\left[\sup_{\mathcal{H}_R}\left(\hat{\mathcal{R}}^{\text{cml}}_{\mathcal{S}'}(f)-\hat{\mathcal{R}}^{\text{cml}}_{\mathcal{S}}(f)\right)\right]$$

Restate of Theorem 1 (CML Symmetrization). *Let $\mathcal{S}$ and $\mathcal{S}'$ be the two independent datasets of interactions that only one sample is different. In terms of any the hypothesis set $\mathcal{H}_R$ and loss function ℓ, the following holds:*

$$\mathop{\mathbb{E}}_{\mathcal{S},\mathcal{S}'}\left[\sup_{\mathcal{H}_R}\left(\hat{\mathcal{R}}^{cml}_{\mathcal{S}'}(f)-\hat{\mathcal{R}}^{cml}_{\mathcal{S}}(f)\right)\right]\leq 2\mathfrak{R}^{cml}_{\ell,\mathcal{S}}(\mathcal{H}_R) \tag{30}$$

Proof. *Define*

$$\mathcal{Q}^{jk}_{i,\sigma}=\frac{\sigma^+_{ij}+\sigma^-_{ik}}{2}\ell^{(i)}(\tilde{v}^+_j,\tilde{v}^-_k)+\frac{\sigma^+_{ij}-\sigma^-_{ik}}{2}\ell^{(i)}(\tilde{v}^+_j,v^-_k)-\frac{\sigma^+_{ij}-\sigma^-_{ik}}{2}\ell^{(i)}(v^+_j,\tilde{v}^-_k)-\frac{\sigma^+_{ij}+\sigma^-_{ik}}{2}\ell^{(i)}(v^+_j,v^-_k) \tag{31}$$

where $\sigma^+_{ij},\forall j=1,2,\ldots,n^+_i$ and $\sigma^-_{ik},\forall k=1,2,\ldots,n^-_i$ are i.i.d Rademacher random variables uniformly chosen from $\{-1,+1\}$ with $\mathbb{P}(\sigma=1)=\mathbb{P}(\sigma=-1)=0.5$.

To complete the proof, first of all, we need to prove that

$$\mathop{\mathbb{E}}_{\mathcal{S},\mathcal{S}'}\left[\sup_{\mathcal{H}_R}\left(\hat{\mathcal{R}}^{cml}_{\mathcal{S}'}(f)-\hat{\mathcal{R}}^{cml}_{\mathcal{S}}(f)\right)\right]\leq\frac{1}{M}\sum_{u_i\in\mathcal{U}}\mathop{\mathbb{E}}_{\mathcal{S}_i,\mathcal{S}'_i}\mathop{\mathbb{E}}_{\sigma_i}\left[\sup_{\mathcal{H}_R}\frac{1}{n^+_in^-_i}\sum_{j=1}^{n^+_i}\sum_{k=1}^{n^-_i}\mathcal{Q}^{jk}_{i,\sigma}\right], \tag{32}$$

To prove this, we have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathcal{S},\mathcal{S}'}\left[\sup_{\mathcal{H}_R}\left(\hat{\mathcal{R}}^{cml}_{\mathcal{S}'}(f)-\hat{\mathcal{R}}^{cml}_{\mathcal{S}}(f)\right)\right]\\
=&\mathop{\mathbb{E}}_{\mathcal{S},\mathcal{S}'}\left[\sup_{\mathcal{H}_R}\left(\left(\hat{\mathcal{R}}^{cml}_{\mathcal{S}'_1}(f)-\hat{\mathcal{R}}^{cml}_{\mathcal{S}_1}(f)\right)\ldots\left(\hat{\mathcal{R}}^{cml}_{\mathcal{S}'_i}(f)-\hat{\mathcal{R}}^{cml}_{\mathcal{S}_i}(f)\right)+\cdots+\left(\hat{\mathcal{R}}^{cml}_{\mathcal{S}'_M}(f)-\hat{\mathcal{R}}^{cml}_{\mathcal{S}_M}(f)\right)\right)\right]\\
\overset{(*)}{\leq}&\sum_{u_i\in\mathcal{U}}\mathop{\mathbb{E}}_{\mathcal{S}_i,\mathcal{S}'_i}\left[\sup_{\mathcal{H}_R}\left(\hat{\mathcal{R}}^{cml}_{\mathcal{S}'_i}(f)-\hat{\mathcal{R}}^{cml}_{\mathcal{S}_i}(f)\right)\right]
\end{aligned} \tag{33}$$

where

$$\hat{\mathcal{R}}^{cml}_{\mathcal{S}_i}(f)=\frac{1}{M}\cdot\frac{1}{n^+_in^-_i}\sum_{j=1}^{n^+_i}\sum_{k=1}^{n^-_i}\ell^{(i)}(v^+_j,v^-_k).$$

and $(*)$ achieves by the inequality: $\sup(x+y)\leq\sup(x)+\sup(y)$.

Equipped with Eq.(33), we could hold Eq.(32) by proving the following equation satisfied:

$$\sum_{u_i\in\mathcal{U}}\mathop{\mathbb{E}}_{\mathcal{S}_i,\mathcal{S}'_i}\left[\sup_{\mathcal{H}_R}\left(\hat{\mathcal{R}}^{cml}_{\mathcal{S}'_i}(f)-\hat{\mathcal{R}}^{cml}_{\mathcal{S}_i}(f)\right)\right]=\frac{1}{M}\sum_{u_i\in\mathcal{U}}\mathop{\mathbb{E}}_{\mathcal{S}_i,\mathcal{S}'_i}\mathop{\mathbb{E}}_{\sigma_i}\left[\sup_{\mathcal{H}_R}\sum_{j=1}^{n^+_i}\sum_{k=1}^{n^-_i}\frac{1}{n^+_in^-_i}\mathcal{Q}^{jk}_{i,\sigma}\right], \tag{34}$$

It is interesting to note that, Eq.(34) is calculated separately for each user. Therefore, we only need to consider each user to prove the CML symmetrization such that the overall symmetrization of CML could be simply proved by taking a sum. Taking u_i as an example, we need to prove the following equation

$$\mathop{\mathbb{E}}_{\mathcal{S}_i,\mathcal{S}'_i}\left[\sup_{\mathcal{H}_R}\left(\hat{\mathcal{R}}^{cml}_{\mathcal{S}'_i}(f)-\hat{\mathcal{R}}^{cml}_{\mathcal{S}_i}(f)\right)\right]=\frac{1}{M}\mathop{\mathbb{E}}_{\mathcal{S}_i,\mathcal{S}'_i}\mathop{\mathbb{E}}_{\sigma_i}\left[\sup_{\mathcal{H}_R}\sum_{j=1}^{n^+_i}\sum_{k=1}^{n^-_i}\frac{1}{n^+_in^-_i}\mathcal{Q}^{jk}_{i,\sigma}\right]. \tag{35}$$

Let $\mathcal{S}_i=\{v^+_j\}_{j=1}^{n^+_i}\cup\{v^-_k\}_{k=1}^{n^-_i}$ and $\mathcal{S}'_i=\{\tilde{v}^+_j\}_{j=1}^{n^+_i}\cup\{\tilde{v}^-_k\}_{k=1}^{n^-_i}$ be the two independent interaction datasets that only one item is different. To this end, considering the process of exchanging instances, since the items are drawn independently, the following holds

$$\mathop{\mathbb{E}}_{\mathcal{S}_i,\mathcal{S}'_i}\left[\sup_{\mathcal{H}_R}\left(\hat{\mathcal{R}}^{cml}_{\mathcal{S}'_i}(f)-\hat{\mathcal{R}}^{cml}_{\mathcal{S}_i}(f)\right)\right]=\mathop{\mathbb{E}}_{\mathcal{S}_i,\mathcal{S}'_i}\left[\sup_{\mathcal{H}_R}\left(\hat{\mathcal{R}}^{cml}_{\tilde{\mathcal{S}}_i}(f)-\hat{\mathcal{R}}^{cml}_{\tilde{\mathcal{S}}_i}(f)\right)\right] \tag{36}$$

where $\tilde{\mathcal{S}}_i{}'$ and $\tilde{\mathcal{S}}_i$ are two different datasets determined by exchanging the corresponding indexed sample between $\mathcal{S}'_i$ and $\mathcal{S}_i$.

Next, we will employ the induction method to prove this conclusion Eq.(36).

Trivial Case. *Let us first consider that there are only one positive and negative item for the specific user u_i, i.e., $n^+_i=1$, $n^-_i=1$, $\mathcal{S}_i=\{v^+_1,v^-_1\}$, $\mathcal{S}'_i=\{\tilde{v}^+_1,\tilde{v}^-_1\}$ and $\sigma_i=\{\sigma^+_{i1},\sigma^-_{i1}\}$. In this configuration, the value of σ_i can be permuted as follows:*

1) $\sigma_{i1}^{+}=1$ *and* $\sigma_{i1}^{-}=1$. *We have*

$$\mathcal{Q}_{i,\sigma}^{11}=\ell^{(i)}(\tilde{v}_1^{+},\tilde{v}_1^{-})-\ell_i(v_1^{+},v_1^{-}) \tag{37}$$

This motivates us to set $\tilde{\mathcal{S}}_i^{'}=\mathcal{S}_i',\tilde{\mathcal{S}}_i=\mathcal{S}_i$ *to satisfy Eq.(36).*

2) $\sigma_{i1}^{+}=-1$ *and* $\sigma_{i1}^{-}=-1$. *Here, we have*

$$\mathcal{Q}_{i,\sigma}^{11}=\ell^{(i)}(v_1^{+},v_1^{-})-\ell^{(i)}(\tilde{v}_1^{+},\tilde{v}_1^{-}) \tag{38}$$

This motivates us to set $\tilde{\mathcal{S}}_i^{'}=\mathcal{S}_i,\tilde{\mathcal{S}}_i=\mathcal{S}_i'$ *to satisfy Eq.(36).*

3) $\sigma_{i1}^{+}=1$ *and* $\sigma_{i1}^{-}=-1$. *We have*

$$\mathcal{Q}_{i,\sigma}^{11}=\ell^{(i)}(\tilde{v}_1^{+},v_1^{-})-\ell^{(i)}(v_1^{+},\tilde{v}_1^{-}) \tag{39}$$

To hold Eq.(36), let $\mathcal{S}_i$ *and* $\mathcal{S}_i'$ *exchange sample* v_1^{-} *and* $\tilde{v}_1^{-}$, *i.e.,* $\mathcal{S}_i'=\{\tilde{v}_1^{+},v_1^{-}\}$ *and* $\mathcal{S}_i=\{v_1^{+},\tilde{v}_1^{-}\}$, *and let* $\tilde{\mathcal{S}}_i^{'}=\mathcal{S}_i',\tilde{\mathcal{S}}_i=\mathcal{S}_i$ *complete the proof.*

4) $\sigma_{i1}^{+}=-1$ *and* $\sigma_{i1}^{-}=1$.

$$\mathcal{Q}_{i,\sigma}^{11}=\ell^{(i)}(v_1^{+},\tilde{v}_1^{-})-\ell^{(i)}(\tilde{v}_1^{+},v_1^{-}). \tag{40}$$

Needless to say, one can let $\tilde{\mathcal{S}}_i=\mathcal{S}_i,\tilde{\mathcal{S}}_i^{'}=\mathcal{S}_i'$ *by exchanging* v_1^{+} *and* $\tilde{v}_1^{+}$ *between* $\mathcal{S}_i$ *and* $\mathcal{S}_i'$ *to hold the conclusion.*

In summary, the above discussions support the proof of Eq.(36) in terms of trivial case.

Recursion*. Given* $1<n_{+}<n_i^{+},1<n_{-}<n_i^{-}$ *samples, and*

$$\mathcal{S}_{i,0}=\{v_j^{+}\}_{j=1}^{n_{+}}\cup\{v_k^{-}\}_{k=1}^{n_{-}},\ \ \mathcal{S}_{i,0}'=\{\tilde{v}_j^{+}\}_{j=1}^{n_{+}}\cup\{\tilde{v}_k^{-}\}_{k=1}^{n_{-}},\sigma_{i,0}=\{\sigma_{ij}^{+}\}_{j=1}^{n_{+}}\cup\{\sigma_{ik}^{-}\}_{k=1}^{n_{-}},$$

assume $\mathcal{S}_{i,0}$, $\mathcal{S}_{i,0}'$ *and* $\sigma_{i,0}$ *could hold Eq.(36) by* $\tilde{\mathcal{S}}_{i,0}$ *and* $\tilde{\mathcal{S}}_{i,0}'$. *Next, we will show that for any fresh samples* $(v_t^{+},\tilde{v}_t^{+},\sigma_{it}^{+})$ *or* $(v_t^{-},\tilde{v}_t^{-},\sigma_{it}^{-})$, *our conclusion Eq.(36) is still satisfied.*

Let us first consider $\sigma_{it}^{+}=1$. *Obviously, merely* $\mathcal{Q}_{i,\sigma}^{tk},k=1,2,\ldots,n_{-}$ *has the contribution to this new sample.*

1) *if* $\sigma_{ik}^{-}=1$ *is the case, we have*

$$\mathcal{Q}_{i,\sigma}^{tk}=\ell^{(i)}(\tilde{v}_t^{+},\tilde{v}_k^{-})-\ell^{(i)}(v_t^{+},v_k^{-}).$$

One can let

$$\tilde{\mathcal{S}}_i^{'}=\mathcal{S}_{i,0}'\cup\{\tilde{v}_t^{+}\},\tilde{\mathcal{S}}_i=\mathcal{S}_{i,0}\cup\{v_t^{+}\}$$

to complete the proof.

2) *if* $\sigma_{ik}^{-}=-1$ *is the case, we have*

$$\mathcal{Q}_{i,\sigma}^{tk}=\ell^{(i)}(\tilde{v}_t^{+},v_k^{-})-\ell^{(i)}(v_t^{+},\tilde{v}_k^{-}),$$

and one can let

$$\tilde{\mathcal{S}}_i^{'}=\mathcal{S}_{i,0}\cup\{\tilde{v}_t^{+}\},\tilde{\mathcal{S}}_i=\mathcal{S}_{i,0}'\cup\{v_t^{+}\}$$

to complete the proof.

In addition, with respect to $\sigma_{it}^{+}=-1$, *we have*

1) $\sigma_{ik}^{-}=1$.

$$\mathcal{Q}_{i,\sigma}^{tk}=\ell^{(i)}(v_t^{+},\tilde{v}_k^{-})-\ell^{(i)}(\tilde{v}_t^{+},v_k^{-}),$$

and one can let

$$\tilde{\mathcal{S}}_i^{'}=\mathcal{S}_{i,0}'\cup\{v_t^{+}\},\tilde{\mathcal{S}}_i=\mathcal{S}_{i,0}\cup\{\tilde{v}_t^{+}\}$$

to complete the proof.

2) $\sigma_{ik}^{-}=-1$.

$$\mathcal{Q}_{i,\sigma}^{tk}=\ell^{(i)}(v_t^{+},v_k^{-})-\ell^{(i)}(\tilde{v}_t^{+},\tilde{v}_k^{-}),$$

we have

$$\tilde{\mathcal{S}}_i^{'}=\mathcal{S}_{i,0}\cup\{v_t^{+}\},\tilde{\mathcal{S}}_i=\mathcal{S}_{i,0}'\cup\{\tilde{v}_t^{+}\}$$

to statisfy the conclusion.

In the same way, in terms of σ_{it}^{-}, *we have*

1) $\sigma_{it}^{-}=1,\sigma_{ij}^{+}=1$. *This suggests that*

$$\tilde{\mathcal{S}}_i^{'}=\mathcal{S}_{i,0}'\cup\{\tilde{v}_t^{-}\},\tilde{\mathcal{S}}_i=\mathcal{S}_{i,0}\cup\{v_t^{-}\}$$

2) $\sigma_{it}^{-}=1,\sigma_{ij}^{+}=-1$.

$$\tilde{\mathcal{S}}_i^{'}=\mathcal{S}_{i,0}\cup\{\tilde{v}_t^{-}\},\tilde{\mathcal{S}}_i=\mathcal{S}_{i,0}'\cup\{v_t^{-}\}$$

3) $\sigma_{it}^{-}=-1,\sigma_{ij}^{+}=1$.

$$\tilde{\mathcal{S}}_i^{'}=\mathcal{S}_{i,0}'\cup\{v_t^{-}\},\tilde{\mathcal{S}}_i=\mathcal{S}_{i,0}\cup\{\tilde{v}_t^{-}\}$$

4) $\sigma_{it}^- = -1, \sigma_{ij}^+ = -1$.

$$\tilde{\mathcal{S}}_i{}' = \mathcal{S}_{i,0} \cup \{v_t^-\}, \tilde{\mathcal{S}}_i = \mathcal{S}'_{i,0} \cup \{\tilde{v}_t^-\}$$

It is interesting to note that, in either case, $\tilde{\mathcal{S}}_i$ and $\tilde{\mathcal{S}}_i{}'$ are both obtained from $\mathcal{S}_i$ and $\mathcal{S}'_i$, which only swaps the single corresponding instance rather than a pair. *Taking the trivial case and recursion into account together, we have proved that*

$$\mathop{\mathbb{E}}_{\mathcal{S}_i,\mathcal{S}'_i}\left[\sup_{\mathcal{H}_R}\left(\hat{\mathcal{R}}^{cml}_{\mathcal{S}'_i}(f) - \hat{\mathcal{R}}^{cml}_{\mathcal{S}_i}(f)\right)\right] = \frac{1}{M}\mathop{\mathbb{E}}_{\mathcal{S}_i,\mathcal{S}'_i}\mathop{\mathbb{E}}_{\sigma_i}\left[\sup_{\mathcal{H}_R}\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\frac{1}{n_i^+ n_i^-}\mathcal{Q}^{jk}_{i,\sigma}\right]. \tag{41}$$

since we have

$$\begin{aligned}
&\frac{1}{M}\mathop{\mathbb{E}}_{\mathcal{S}_i,\mathcal{S}'_i}\mathop{\mathbb{E}}_{\sigma_i}\left[\sup_{\mathcal{H}_R}\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\frac{1}{n_i^+ n_i^-}\mathcal{Q}^{jk}_{i,\sigma}\right]\\
=&\frac{1}{M}\cdot\frac{1}{2^N}\sum_{\sigma_i}\mathop{\mathbb{E}}_{\mathcal{S}_i,\mathcal{S}'_i}\left[\sup_{\mathcal{H}_R}\frac{1}{n_i^+ n_i^-}\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\mathcal{Q}^{jk}_{i,\sigma}\right]\\
=&\frac{1}{2^N}\sum_{\sigma_i}\mathop{\mathbb{E}}_{\mathcal{S}_i,\mathcal{S}'_i}\left[\sup_{\mathcal{H}_R}\left(\hat{\mathcal{R}}^{cml}_{\tilde{\mathcal{S}}_i'}(f) - \hat{\mathcal{R}}^{cml}_{\tilde{\mathcal{S}}_i}(f)\right)\right]\\
=&\frac{1}{2^N}\sum_{\sigma_i}\mathop{\mathbb{E}}_{\mathcal{S}_i,\mathcal{S}'_i}\left[\sup_{\mathcal{H}_R}\left(\hat{\mathcal{R}}^{cml}_{\mathcal{S}'_i}(f) - \hat{\mathcal{R}}^{cml}_{\mathcal{S}_i}(f)\right)\right]\\
=&\frac{2^N}{2^N}\mathop{\mathbb{E}}_{\mathcal{S}_i,\mathcal{S}'_i}\left[\sup_{\mathcal{H}_R}\left(\hat{\mathcal{R}}^{cml}_{\mathcal{S}'_i}(f) - \hat{\mathcal{R}}^{cml}_{\mathcal{S}_i}(f)\right)\right]\\
=&\mathop{\mathbb{E}}_{\mathcal{S}_i,\mathcal{S}'_i}\left[\sup_{\mathcal{H}_R}\left(\hat{\mathcal{R}}^{cml}_{\mathcal{S}'_i}(f) - \hat{\mathcal{R}}^{cml}_{\mathcal{S}_i}(f)\right)\right],
\end{aligned} \tag{42}$$

where again $\mathcal{S}_i$ and $\mathcal{S}'_i$ are the two independent datasets of interactions that only one sample is different.

Equipped with Eq.(42), by taking a sum over all users, it is easy to obtain Eq.(34) and thus Eq.(32) holds. Finally, according to the property that the sign of the Rademacher random variables do not affect its expectation, we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{\mathcal{S},\mathcal{S}'}\left[\sup_{\mathcal{H}_R}\left(\hat{\mathcal{R}}^{cml}_{\mathcal{S}'}(f) - \hat{\mathcal{R}}^{cml}_{\mathcal{S}}(f)\right)\right] &\le \frac{1}{M}\sum_{u_i\in\mathcal{U}}\mathop{\mathbb{E}}_{\mathcal{S}_i,\mathcal{S}'_i}\mathop{\mathbb{E}}_{\sigma_i}\left[\sup_{\mathcal{H}_R}\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\frac{1}{n_i^+ n_i^-}\mathcal{Q}^{jk}_{i,\sigma}\right]\\
&\le \frac{2}{M}\sum_{u_i\in\mathcal{U}}\mathop{\mathbb{E}}_{\mathcal{S}_i,\sigma_i}\left[\sup_{\mathcal{H}_R}\frac{1}{n_i^+ n_i^-}\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\mathcal{Q}^{jk}_{(i)}\right]\\
&= 2\mathfrak{R}^{cml}_{\ell,\mathcal{S}}(\mathcal{H}_R)
\end{aligned} \tag{43}$$

where again

$$\mathcal{Q}^{jk}_{(i)} = \frac{\sigma_{ij}^+ + \sigma_{ik}^-}{2}\cdot\ell^{(i)}(v_j^+, v_k^-)$$

This completed the proof.

B.2 Upper Bound of empirical Rademacher Complexity

Restate of Theorem 2. *(**Upper Bound of empirical Rademacher Complexity**). Given the user set $\mathcal{U}$ and corresponding sample set $\mathcal{S} = \bigcup_{u_i\in\mathcal{U}}\mathcal{S}_i$ where $\mathcal{S}_i = \{v_j^+\}_{j=1}^{n_i^+}\cup\{v_k^-\}_{k=1}^{n_i^-}, n_i^+ + n_i^- = N$. If ℓ is a ϕ Lipschitz continuous, then the following inequality holds:*

$$\hat{\mathfrak{R}}^{cml}_{\ell,\mathcal{S}}(\mathcal{H}_R) \lesssim \frac{\phi}{M}\cdot\mathbf{max}(\lambda, \sqrt{R\cdot d})\cdot\tilde{N}^{-1/2}. \tag{44}$$

Proof. *At first, according to Definition.2, we have*

$$\begin{aligned}
\hat{\mathfrak{R}}^{cml}_{\ell,\mathcal{S}}(\mathcal{H}_R) &= \frac{1}{M}\sum_{u_i\in\mathcal{U}}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sup_{\mathcal{H}_R}\frac{1}{n_i^+ n_i^-}\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\mathcal{Q}^{jk}_{(i)}\right]\\
=&\frac{1}{M}\left(\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_1}\left[\sup_{\mathcal{H}_R}\frac{1}{n_1^+ n_1^-}\sum_{j=1}^{n_1^+}\sum_{k=1}^{n_1^-}\mathcal{Q}^{jk}_{(1)}\right] + \cdots + \mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sup_{\mathcal{H}_R}\frac{1}{n_i^+ n_i^-}\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\mathcal{Q}^{jk}_{(i)}\right] + \cdots + \mathop{\mathbb{E}}_{\boldsymbol{\sigma}_M}\left[\sup_{\mathcal{H}_R}\frac{1}{n_M^+ n_M^-}\sum_{j=1}^{n_M^+}\sum_{k=1}^{n_M^-}\mathcal{Q}^{jk}_{(M)}\right]\right),
\end{aligned} \tag{45}$$

where

$$\mathcal{Q}_{(i)}^{jk} = \frac{\sigma_{ij}^{+} + \sigma_{ik}^{-}}{2} \cdot \ell^{(i)}(v_j^{+}, v_k^{-}),$$

and $\ell^{(i)}(v_j^{+}, v_k^{-})$ *is a differentiable ranking loss, such as hinge-loss and square loss:*

$$\ell^{(i)}(v_j^{+}, v_k^{-}) = \ell \circ (\lambda + \boldsymbol{d}(i,j) - \boldsymbol{d}(i,k)).$$

Next, according to Eq.(45), we first derive the following bound of a specific user u_i*:*

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sup_{\mathcal{H}_R}\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\frac{1}{n_i^+ n_i^-}\mathcal{Q}_{(i)}^{jk}\right] = \mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sup_{\mathcal{H}_R}\frac{1}{n_i^+ n_i^-}\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\frac{\sigma_{ij}^{+}+\sigma_{ij}^{k}}{2}\cdot\ell^{(i)}(v_j^{+},v_k^{-})\right]\\
=&\frac{1}{n_i^+ n_i^-}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sup_{\mathcal{H}_R}\left(\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\frac{\sigma_{ij}^{+}}{2}\cdot\ell^{(i)}(v_j^{+},v_k^{-})+\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\frac{\sigma_{ik}^{-}}{2}\cdot\ell^{(i)}(v_j^{+},v_k^{-})\right)\right]\\
\overset{(*)}{\leq}&\frac{1}{n_i^+ n_i^-}\left(\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sup_{\mathcal{H}_R}\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\frac{\sigma_{ij}^{+}}{2}\cdot\ell^{(i)}(v_j^{+},v_k^{-})\right]+\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sup_{\mathcal{H}_R}\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\frac{\sigma_{ik}^{-}}{2}\cdot\ell^{(i)}(v_j^{+},v_k^{-})\right]\right)\\
\overset{(a)}{\leq}&\frac{1}{n_i^+ n_i^-}\left(\phi\cdot\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sup_{\mathcal{H}_R}\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\frac{\sigma_{ij}^{+}}{2}\cdot(\lambda+\boldsymbol{d}(i,j)-\boldsymbol{d}(i,k))\right]+\phi\cdot\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sup_{\mathcal{H}_R}\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\frac{\sigma_{ik}^{-}}{2}\cdot(\lambda+\boldsymbol{d}(i,j)-\boldsymbol{d}(i,k))\right]\right)\\
\overset{(*)}{\leq}&\underbrace{\frac{\phi}{n_i^+ n_i^-}\left(\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sup_{\mathcal{H}_R}\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\frac{\sigma_{ij}^{+}}{2}\cdot\lambda\right]+\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sup_{\mathcal{H}_R}\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\frac{\sigma_{ik}^{-}}{2}\cdot\lambda\right]\right)}_{①}\\
&+\underbrace{\frac{\phi}{n_i^+ n_i^-}\left(\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sup_{\mathcal{H}_R}\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\frac{\sigma_{ij}^{+}}{2}\cdot(\boldsymbol{d}(i,j)-\boldsymbol{d}(i,k))\right]+\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sup_{\mathcal{H}_R}\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\frac{\sigma_{ik}^{-}}{2}\cdot(\boldsymbol{d}(i,j)-\boldsymbol{d}(i,k))\right]\right)}_{②}
\end{aligned} \tag{46}$$

where (a) follows the Lem.3 and $(*)$ *achieves by the inequality:* $\sup(x+y) \leq \sup(x) + \sup(y)$.

Now, it is easy to show that, for ①*, the following holds:*

$$\begin{aligned}
&\frac{\phi}{n_i^+ n_i^-}\left(\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sup_{\mathcal{H}_R}\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\frac{\sigma_{ij}^{+}}{2}\cdot\lambda\right]+\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sup_{\mathcal{H}_R}\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\frac{\sigma_{ik}^{-}}{2}\cdot\lambda\right]\right)\\
\leq&\frac{\lambda\phi}{n_i^+ n_i^-}\left(\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\left|\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\frac{\sigma_{ij}^{+}}{2}\right|\right]+\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\left|\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\frac{\sigma_{ik}^{-}}{2}\right|\right]\right)\\
\leq&\frac{\lambda\phi}{n_i^+ n_i^-}\left(\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sum_{k=1}^{n_i^-}\left|\sum_{j=1}^{n_i^+}\frac{\sigma_{ij}^{+}}{2}\right|\right]+\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sum_{j=1}^{n_i^+}\left|\sum_{k=1}^{n_i^-}\frac{\sigma_{ik}^{-}}{2}\right|\right]\right)\\
=&\frac{\lambda\phi}{n_i^+ n_i^-}\left(\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sum_{k=1}^{n_i^-}\left|\sum_{j=1}^{n_i^+}\frac{\sigma_{ij}^{+}}{2}\right|+\sum_{j=1}^{n_i^+}\left|\sum_{k=1}^{n_i^-}\frac{\sigma_{ik}^{-}}{2}\right|\right]\right)\\
\overset{(**)}{\leq}&\frac{\lambda\phi\cdot\sqrt{n_i^+ + n_i^-}}{n_i^+ n_i^-}\left(\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sqrt{n_i^-\left(\sum_{j=1}^{n_i^+}\frac{\sigma_{ij}^{+}}{2}\right)^2+n_i^+\left(\sum_{k=1}^{n_i^-}\frac{\sigma_{ik}^{-}}{2}\right)^2}\right]\right)\\
\overset{(Jen.)}{\leq}&\frac{\lambda\phi\cdot\sqrt{n_i^+ + n_i^-}}{n_i^+ n_i^-}\sqrt{n_i^-\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\left(\sum_{j=1}^{n_i^+}\frac{\sigma_{ij}^{+}}{2}\right)^2\right]+n_i^+\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\left(\sum_{k=1}^{n_i^-}\frac{\sigma_{ik}^{-}}{2}\right)^2\right]}\\
\leq&\lambda\phi\cdot\sqrt{\frac{n_i^+ + n_i^-}{2n_i^+ n_i^-}}\\
\lesssim&\lambda\phi\cdot\sqrt{\frac{1}{n_i^+}+\frac{1}{n_i^-}}
\end{aligned} \tag{47}$$

where $(**)$ is because of the fact $\|\boldsymbol{x}\|_1 \leq \sqrt{b}\,\|\boldsymbol{x}\|_2, \boldsymbol{x} \in \mathbb{R}^b$, and $(Jen.)$ follows the Jensen's Inequality.

Subsequently, for the last term ②, we have

$$
\begin{aligned}
&\frac{\phi}{n_i^+ n_i^-}\left(\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sup_{\mathcal{H}_R}\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\frac{\sigma_{ij}^+}{2}\cdot(\boldsymbol{d}(i,j)-\boldsymbol{d}(i,k))\right]+\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sup_{\mathcal{H}_R}\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\frac{\sigma_{ik}^-}{2}\cdot(\boldsymbol{d}(i,j)-\boldsymbol{d}(i,k))\right]\right)\\
&=\frac{\phi}{n_i^+ n_i^-}\left(\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sup_{\mathcal{H}_R}\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\sigma_{ij}^+\cdot\boldsymbol{e}_{u_i}^\top(\boldsymbol{e}_{v_k}-\boldsymbol{e}_{v_j})\right]+\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sup_{\mathcal{H}_R}\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\sigma_{ik}^-\cdot\boldsymbol{e}_{u_i}^\top(\boldsymbol{e}_{v_k}-\boldsymbol{e}_{v_j})\right]\right)\\
&\leq\frac{\phi}{n_i^+ n_i^-}\left(\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sup_{\mathcal{H}_R}\left|\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\sigma_{ij}^+\cdot\boldsymbol{e}_{u_i}^\top(\boldsymbol{e}_{v_k}-\boldsymbol{e}_{v_j})\right|\right]+\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sup_{\mathcal{H}_R}\left|\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\sigma_{ik}^-\cdot\boldsymbol{e}_{u_i}^\top(\boldsymbol{e}_{v_k}-\boldsymbol{e}_{v_j})\right|\right]\right)\\
&\overset{(***)}{\leq}\frac{\phi}{n_i^+ n_i^-}\left(\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sup_{\mathcal{H}_R}\|\boldsymbol{e}_{u_i}\|\cdot\left\|\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\sigma_{ij}^+\cdot(\boldsymbol{e}_{v_k}-\boldsymbol{e}_{v_j})\right\|\right]+\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sup_{\mathcal{H}_R}\|\boldsymbol{e}_{u_i}\|\cdot\left\|\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\sigma_{ik}^-\cdot(\boldsymbol{e}_{v_k}-\boldsymbol{e}_{v_j})\right\|\right]\right)\\
&=\frac{\sqrt{R}\cdot\phi}{n_i^+ n_i^-}\left(\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sup_{\mathcal{H}_R}\left\|\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\sigma_{ij}^+\cdot\boldsymbol{W}_v^\top(\boldsymbol{v}_k-\boldsymbol{v}_j)\right\|\right]+\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sup_{\mathcal{H}_R}\left\|\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\sigma_{ik}^-\cdot\boldsymbol{W}_v^\top(\boldsymbol{v}_k-\boldsymbol{v}_j)\right\|\right]\right)\\
&\overset{(***)}{\leq}\frac{\sqrt{R}\cdot\phi}{n_i^+ n_i^-}\cdot\|\boldsymbol{W}_v\|_*\cdot\left(\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\left\|\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\sigma_{ij}^+\cdot(\boldsymbol{v}_k-\boldsymbol{v}_j)\right\|\right]+\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\left\|\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\sigma_{ik}^-\cdot(\boldsymbol{v}_k-\boldsymbol{v}_j)\right\|\right]\right)
\end{aligned}
\tag{48}
$$

where $(***)$ is achieved by the Cauchy-Buniakowsky-Schwarz Inequality $\|\boldsymbol{x}^\top\boldsymbol{y}\| \leq \|\boldsymbol{x}\|\cdot\|\boldsymbol{y}\|$ here.

Next, according to Assumption.1, we have $\|\boldsymbol{W}_v\|_2 \lesssim \sqrt{\frac{n_i^+ + n_i^-}{d}}$ and $\sqrt{n_i^+ + n_i^-} \lesssim d$, holding that

$$
\begin{aligned}
&\frac{\phi}{n_i^+ n_i^-}\left(\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sup_{\mathcal{H}_R}\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\frac{\sigma_{ij}^+}{2}\cdot(\boldsymbol{d}(i,j)-\boldsymbol{d}(i,k))\right]+\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sup_{\mathcal{H}_R}\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\frac{\sigma_{ik}^-}{2}\cdot(\boldsymbol{d}(i,j)-\boldsymbol{d}(i,k))\right]\right)\\
&\lesssim\frac{\sqrt{Rd}\cdot\phi}{n_i^+ n_i^-}\left(\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sum_{k=1}^{n_i^-}\left\|\sum_{j=1}^{n_i^+}\sigma_{ij}^+\cdot(\boldsymbol{v}_k-\boldsymbol{v}_j)\right\|\right]+\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\sum_{j=1}^{n_i^+}\left\|\sum_{k=1}^{n_i^-}\sigma_{ik}^-\cdot(\boldsymbol{v}_k-\boldsymbol{v}_j)\right\|\right]\right)\\
&\overset{(**)}{\lesssim}\phi\cdot\frac{\sqrt{Rd\cdot(n_i^+ + n_i^-)}}{n_i^+ n_i^-}\left(\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\sqrt{n_i^-\cdot\left\|\sum_{j=1}^{n_i^+}\sigma_{ij}^+\cdot(\boldsymbol{v}_k-\boldsymbol{v}_j)\right\|_2^2+n_i^+\cdot\left\|\sum_{j=1}^{n_i^+}\sigma_{ik}^-\cdot(\boldsymbol{v}_k-\boldsymbol{v}_j)\right\|_2^2}\right)\\
&\overset{(Jen.)}{\lesssim}\phi\cdot\frac{\sqrt{Rd\cdot(n_i^+ + n_i^-)}}{n_i^+ n_i^-}\left(\sqrt{n_i^-\cdot\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left(\left\|\sum_{j=1}^{n_i^+}\sigma_{ij}^+\cdot(\boldsymbol{v}_k-\boldsymbol{v}_j)\right\|_2^2\right)+n_i^+\cdot\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left(\left\|\sum_{j=1}^{n_i^+}\sigma_{ik}^-\cdot(\boldsymbol{v}_k-\boldsymbol{v}_j)\right\|_2^2\right)}\right)\\
&\overset{(b)}{\lesssim}\phi\cdot\sqrt{\frac{Rd\cdot(n_i^+ + n_i^-)}{n_i^+ n_i^-}}\\
&\lesssim\phi\cdot\sqrt{R\cdot d}\cdot\sqrt{\frac{1}{n_i^+}+\frac{1}{n_i^-}}
\end{aligned}
\tag{49}
$$

where (b) according to the fact that $\boldsymbol{v}_k$ and $\boldsymbol{v}_j$ are two one-hot vectors that only one nonzero term there.

Therefore, taking Eq.(45), Eq.(47), Eq.(49) and Definition.3 into account, we hold the following bounds for empirical Rademacher complexity by taking a sum of all users:

$$
\begin{aligned}
\hat{\mathfrak{R}}_{\ell,\mathcal{S}}^{cml}(\mathcal{H}_R)\ &\lesssim\phi\cdot\frac{\mathbf{max}(\lambda,\sqrt{R\cdot d})}{M}\cdot\left(\sum_{u_i\in\mathcal{U}}\sqrt{\frac{1}{n_i^+}+\frac{1}{n_i^-}}\right)\\
&\lesssim\frac{\phi}{M}\cdot\mathbf{max}(\lambda,\sqrt{R\cdot d})\cdot\tilde{N}^{-1/2}
\end{aligned}
\tag{50}
$$

This completed the proof.

B.3 Generalization Bound of Sampling-Free CML

Restate of Theorem 3. *(Generalization Upper Bound of CML with Eq.(4)). Let $\mathcal{H}_R$ be the hypothesis space and ℓ be ϕ-Lipschitz continuous. Given the sample set $\mathcal{S} = \underset{u_i \in \mathcal{U}}{\cup} \mathcal{S}_i$ where $\mathcal{S}^{(i)} = \{v_j^+\}_{j=1}^{n_i^+} \cup \{v_k^-\}_{k=1}^{n_i^-}$, $n_i^+ + n_i^- = N$, for any $\delta \in (0,1)$, with probability at least $1-\delta$, the following inequation holds:*

$$\begin{aligned} \mathcal{R}_\ell^{cml}(f) \lesssim \hat{\mathcal{R}}_{\mathcal{S}}^{cml}(f) + \phi \cdot \frac{\mathbf{max}(\lambda, \sqrt{R \cdot d})}{M} \cdot \sqrt{\frac{1}{\tilde{N}}} \\ + \phi \cdot \frac{R}{M} \cdot \sqrt{\frac{\log 2/\delta}{2}} \cdot \sqrt{\frac{1}{\tilde{N}}}, \end{aligned} \tag{51}$$

where $\mathcal{R}_\ell^{cml}(f)$ is the expectation risk.

Proof. *Step 1. Let*

$$\Phi(\mathcal{S}) = \sup_{\mathcal{H}_R} \left(\mathcal{R}_\ell^{cml}(f) - \hat{\mathcal{R}}_{\mathcal{S}}^{cml}(f) \right)$$

The first aim is to prove that $\Phi(\mathcal{S})$ satisfies the condition of Lem.1. To this end, let $\mathcal{S}$ and $\mathcal{S}'$ be two independent datasets where exactly one item is different with respect to the specific user u_i. Subsequently, we have the following two possible cases for u_i:

- ***Case 1:*** *Only one positive item is different, i.e.,*

$$\mathcal{S}_i = \{v_j^+\}_{j=1}^{n_i^+} \cup \{v_k^-\}_{k=1}^{n_i^-}, \quad \mathcal{S}_i' = (\mathcal{S}_i \backslash \{v_{t_1}^+\}) \cup \{\tilde{v}_{t_1}^+\}, \tag{52}$$

where $\forall t_1, t_1 = 1, 2, \ldots, n_i^+$ and $n_i^+ + n_i^- = N$.
According to this, the upper bound on $\Phi(\mathcal{S}') - \Phi(\mathcal{S})$ could be bounded as follows:

$$\begin{aligned} |\Phi(\mathcal{S}') - \Phi(\mathcal{S})| &\leq \left| \sup_{\mathcal{H}_R} \left(\hat{\mathcal{R}}_{\mathcal{S}}^{cml}(f) - \hat{\mathcal{R}}_{\mathcal{S}'}^{cml}(f) \right) \right| \\ &\leq \sup_{\mathcal{H}_R} \left| \hat{\mathcal{R}}_{\mathcal{S}}^{cml}(f) - \hat{\mathcal{R}}_{\mathcal{S}'}^{cml}(f) \right| \\ &= \sup_{\mathcal{H}_R} \left| \hat{\mathcal{R}}_{\mathcal{S}_i}^{cml}(f) - \hat{\mathcal{R}}_{\mathcal{S}_i'}^{cml}(f) \right| \end{aligned} \tag{53}$$

where again

$$\hat{\mathcal{R}}_{\mathcal{S}_i}^{cml}(f) = \frac{1}{M} \cdot \sum_{j=1}^{n_i^+} \sum_{k=1}^{n_i^-} \ell^{(i)}(v_j^+, v_k^-)$$

and $\ell^{(i)}(v_j^+, v_k^-)$ is a differentiable ranking loss, such as hinge-loss and square loss:

$$\ell^{(i)}(v_j^+, v_k^-) = \ell \circ (\lambda + \boldsymbol{d}(i,j) - \boldsymbol{d}(i,k)).$$

Since v_j^+ and $\tilde{v}_j^+$ are different in this case, we have

$$\begin{aligned} |\Phi(\mathcal{S}') - \Phi(\mathcal{S})| &\leq \frac{1}{M} \sup_{\mathcal{H}_R} \left| \frac{1}{n_i^+ n_i^-} \sum_{v_j^+} \sum_{k=1}^{n_i^-} \ell^{(i)}(v_j^+, v_k^-) - \frac{1}{n_i^+ n_i^-} \sum_{\tilde{v}_j^+} \sum_{k=1}^{n_i^-} \ell^{(i)}(\tilde{v}_j^+, v_k^-) \right| \\ &\overset{(a)}{\leq} \frac{1}{M} \cdot \frac{\phi}{n_i^+ n_i^-} \sum_{k=1}^{n_i^-} \left(\boldsymbol{d}(i,j) - \boldsymbol{d}(i,\tilde{j}) \right) \\ &\leq \phi \cdot \frac{4R}{M n_i^+} \end{aligned} \tag{54}$$

- ***Case 2:*** *Only one negative item is different, i.e.,*

$$\mathcal{S}_i = \{v_j^+\}_{j=1}^{n_i^+} \cup \{v_k^-\}_{k=1}^{n_i^-}, \quad \mathcal{S}_i' = (\mathcal{S}_i \backslash \{v_{t_2}^-\}) \cup \{\tilde{v}_{t_2}^-\}. \tag{55}$$

where $\forall t_2, t_2 = 1, 2, \ldots, n_i^-$. Similarly, if v_k^- and $\tilde{v}_k^-$ are different, we can also hold

$$|\Phi(\mathcal{S}') - \Phi(\mathcal{S})| \leq \phi \cdot \frac{4R}{M n_i^-} \tag{56}$$

Therefore, according to Eq.(54) and Eq.(56) show that $\Phi(\mathcal{S})$ satisfies the Bounded Difference Property (Lem.4). Subsequently, for any $\delta \in (0,1)$, with probability at least $1-\delta$, the following holds:

$$\Phi(\mathcal{S}) \lesssim \mathop{\mathbb{E}}_{\mathcal{S}}[\Phi(\mathcal{S})] + \phi \cdot \frac{R}{M} \cdot \sqrt{\frac{\log 1/\delta}{2} \cdot \left(\frac{1}{n_i^+} + \frac{1}{n_i^-}\right)} \tag{57}$$

Step 2. *Next, we need to bound the expectation of the right-hand side $\mathop{\mathbb{E}}_{\mathcal{S}}[\Phi(\mathcal{S})]$ in Eq.(57). We have*

$$\begin{aligned}
\mathop{\mathbb{E}}_{\mathcal{S}}[\Phi(\mathcal{S})] &= \mathop{\mathbb{E}}_{\mathcal{S}}\left[\sup_{\mathcal{H}_R}\left(\mathcal{R}_\ell^{cml}(f) - \hat{\mathcal{R}}_{\mathcal{S}}^{cml}(f)\right)\right] \\
&= \mathop{\mathbb{E}}_{\mathcal{S}}\left[\sup_{\mathcal{H}_R}\mathop{\mathbb{E}}_{\mathcal{S}'}\left[\hat{\mathcal{R}}_{\mathcal{S}'}^{cml}(f) - \hat{\mathcal{R}}_{\mathcal{S}}^{cml}(f)\right]\right] \\
&\leq \mathop{\mathbb{E}}_{\mathcal{S},\mathcal{S}'}\left[\sup_{\mathcal{H}_R}\left(\hat{\mathcal{R}}_{\mathcal{S}'}^{cml}(f) - \hat{\mathcal{R}}_{\mathcal{S}}^{cml}(f)\right)\right]
\end{aligned} \tag{58}$$

Now, the most crucial step of the proof is to conduct CML symmetrization, which represents that introducing Rademacher variable $\boldsymbol{\sigma}$ (i.e., exchange single instance instead of pairs) does not change the expectation. By applying Thm.B.1 to Eq.(58), we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{\mathcal{S}}[\Phi(\mathcal{S})] &\leq \mathop{\mathbb{E}}_{\mathcal{S},\mathcal{S}'}\left[\sup_{\mathcal{H}_R}\left(\hat{\mathcal{R}}_{\mathcal{S}'}^{cml}(f) - \hat{\mathcal{R}}_{\mathcal{S}}^{cml}(f)\right)\right] \\
&= \frac{1}{M}\sum_{u_i \in \mathcal{U}} \mathop{\mathbb{E}}_{\mathcal{S}_i,\mathcal{S}_i'}\mathop{\mathbb{E}}_{\sigma_i}\left[\sup_{\mathcal{H}_R}\frac{1}{n_i^+ n_i^-}\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\mathcal{Q}_{i,\sigma}^{jk}\right] \\
&\leq \frac{2}{M}\sum_{u_i \in \mathcal{U}} \mathop{\mathbb{E}}_{\mathcal{S}_i,\sigma_i}\left[\sup_{\mathcal{H}_R}\frac{1}{n_i^+ n_i^-}\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\mathcal{Q}_{(i)}^{jk}\right] \\
&= 2\mathfrak{R}_{\ell,\mathcal{S}}^{cml}(\mathcal{H}_R)
\end{aligned} \tag{59}$$

where

$$\mathcal{Q}_{(i)}^{jk} = \frac{\sigma_{ij}^+ + \sigma_{ik}^-}{2} \cdot \ell^{(i)}(v_j^+, v_k^-).$$

Step 3. *Similarly, we can follow the proof of $\Phi(\mathcal{S})$ to derive a bound with respect of $\mathfrak{R}_{\ell,\mathcal{S}}^{cml}(\mathcal{H}_R)$ by applying Mcdiarmid's Inequality again, and hence, for any $\delta \in (0,1)$, with probability at least $1-\delta/2$, the following holds*

$$\mathfrak{R}_{\ell,\mathcal{S}}^{cml}(\mathcal{H}_R) \lesssim \hat{\mathfrak{R}}_{\ell,\mathcal{S}}^{cml}(\mathcal{H}_R) + \phi \cdot \frac{R}{M} \cdot \sqrt{\frac{\log 2/\delta}{2} \cdot \left(\frac{1}{n_i^+} + \frac{1}{n_i^-}\right)}. \tag{60}$$

This immediatly suggests that, for any $\delta \in (0,1)$, with probability at least $1-\delta$, we have

$$\Phi(\mathcal{S}) \lesssim 2\hat{\mathfrak{R}}_{\ell,\mathcal{S}}^{cml}(\mathcal{H}_R) + \phi \cdot \frac{R}{M} \cdot \sqrt{\frac{\log 2/\delta}{2} \cdot \left(\frac{1}{n_i^+} + \frac{1}{n_i^-}\right)}. \tag{61}$$

Finally, based on Thm.2 and the union bound, for any $\delta \in (0,1)$, with probability at least $1-\delta$, we have proved

$$\begin{aligned}
\mathcal{R}_\ell^{cml}(f) \lesssim \hat{\mathcal{R}}_{\mathcal{S}}^{cml}(f) &+ \phi \cdot \frac{\mathbf{max}(\lambda, \sqrt{R \cdot d})}{M} \cdot \sqrt{\frac{1}{\tilde{N}}} \\
&+ \phi \cdot \frac{R}{M} \cdot \sqrt{\frac{\log 2/\delta}{2}} \cdot \sqrt{\frac{1}{\tilde{N}}}.
\end{aligned} \tag{62}$$

B.4 Generalization Bound of sampling-based CML

Restate of Theorem 4. ***(Generalization Upper Bound of sampling-based CML Eq.(5)).*** *Let $\mathcal{H}_R$ be the hypothesis set and ℓ be ϕ-Lipschitz. Given the sample set $\mathcal{S} = \bigcup_{u_i \in \mathcal{U}} \mathcal{S}_i$ where $\mathcal{S}^{(i)} = \{v_j^+\}_{j=1}^{n_i^+} \cup \{v_k^-\}_{k=1}^{n_i^-}, n_i^+ + n_i^- = N$, for any $\delta \in (0,1)$, with*

probability at least $1-\delta$, the following equation holds for all possible embedding $\mathcal{H}_R$:

$$\begin{aligned}\mathcal{R}_{\ell}^{cml}(f) \lesssim \hat{\mathcal{R}}_{\mathcal{S}}^{cml}(f) &+ \phi \cdot \frac{\textbf{max}(\lambda, \sqrt{R \cdot d})}{M} \cdot \sqrt{\frac{1}{\tilde{N}}} \\ &+ \frac{(\lambda + 4R)}{M} \cdot \sum_{u_i \in \mathcal{U}} D_{TV}(\hat{\mathbb{P}}^{(i)}, \tilde{\mathbb{P}}^{(i)}) \\ &+ \phi \cdot \frac{R}{M} \cdot \sqrt{\frac{\log 2/\delta}{2}} \cdot \sqrt{\frac{1}{\tilde{N}}}\end{aligned} \tag{63}$$

where $\hat{\mathbb{P}}^{(i)}$ is the original distribution with $\hat{\mathbb{P}}_{ik}^{(i)} \equiv \frac{1}{n_i^+ n_i^-}$; $D_{TV}(\hat{\mathbb{P}}^{(i)}, \tilde{\mathbb{P}}^{(i)}) = \frac{1}{2} \cdot \left\| \hat{\mathbb{P}}^{(i)} - \tilde{\mathbb{P}}^{(i)} \right\|_1$ is the Total Variance (TV) between two probability distributions $\hat{\mathbb{P}}^{(i)}$ and $\tilde{\mathbb{P}}^{(i)}$ on $\mathcal{S}$, which characterizes the difference between two probability distributions.

Proof. *Similarly, define*

$$\begin{aligned}\tilde{\Phi}(\mathcal{S}) &= \sup_{\mathcal{H}_R} \left(\mathcal{R}_{\ell}^{cml}(f) - \tilde{\mathcal{R}}_{\mathcal{S}}^{cml}(f) \right) \\ &= \sup_{\mathcal{H}_R} \left(\mathcal{R}_{\ell}^{cml}(f) - \hat{\mathcal{R}}_{\mathcal{S}}^{cml}(f) + \hat{\mathcal{R}}_{\mathcal{S}}^{cml}(f) - \tilde{\mathcal{R}}_{\mathcal{S}}^{cml}(f) \right) \\ &\leq \underbrace{\sup_{\mathcal{H}_R} \left(\mathcal{R}_{\ell}^{cml}(f) - \hat{\mathcal{R}}_{\mathcal{S}}^{cml}(f) \right)}_{\Phi(\mathcal{S})} + \underbrace{\sup_{\mathcal{H}_R} \left(\hat{\mathcal{R}}_{\mathcal{S}}^{cml}(f) - \tilde{\mathcal{R}}_{\mathcal{S}}^{cml}(f) \right)}_{③}\end{aligned} \tag{64}$$

***Step 1**. First of all, following the proof of Thm.2 and Thm.3, for any $\delta \in (0,1)$, with probability at least $1-\delta$, we have*

$$\begin{aligned}\Phi(\mathcal{S}) &= \sup_{\mathcal{H}_R} \left(\mathcal{R}_{\ell}^{cml}(f) - \hat{\mathcal{R}}_{\mathcal{S}}^{cml}(f) \right) \\ &\lesssim \phi \cdot \frac{\textbf{max}(\lambda, \sqrt{R \cdot d})}{M} \cdot \sqrt{\frac{1}{\tilde{N}}} + \phi \cdot \frac{R}{M} \cdot \sqrt{\frac{\log 2/\delta}{2}} \cdot \sqrt{\frac{1}{\tilde{N}}}.\end{aligned} \tag{65}$$

***Step 2**. In order to clarify ③ thoroughly, recall that we have*

$$\begin{aligned}\hat{\mathcal{R}}_{\mathcal{S}}^{cml}(f) &= \frac{1}{M} \sum_{u_i \in \mathcal{U}} \sum_{j=1}^{n_i^+} \sum_{k=1}^{n_i^-} \hat{\mathbb{P}}_{jk}^{(i)} \cdot \ell^{(i)}(v_j^+, v_k^-) \\ \tilde{\mathcal{R}}_{\mathcal{S}}^{cml}(f) &= \frac{1}{M} \sum_{u_i \in \mathcal{U}} \sum_{j=1}^{n_i^+} \sum_{k=1}^{n_i^-} \tilde{\mathbb{P}}_{jk}^{(i)} \cdot \ell^{(i)}(v_j^+, v_k^-)\end{aligned} \tag{66}$$

where $\tilde{\mathbb{P}}_{jk}^{(i)} = \mathbb{P}(\boldsymbol{v}_j^+, \boldsymbol{v}_k^-)$ represents the probability that item v_k^- is sampled as a negative instance with respect to v_j^+ and $\hat{\mathbb{P}}_{jk}^{(i)} = \mathbb{P}(\boldsymbol{v}_j^+, \boldsymbol{v}_k^-) \equiv \frac{1}{n_i^+ n_i^-} > 0$ could be regarded as the ground truth probability.

According to this, we have

$$\sup_{\mathcal{H}_R} \left(\hat{\mathcal{R}}_{\mathcal{S}}^{cml}(f) - \tilde{\mathcal{R}}_{\mathcal{S}}^{cml}(f) \right) = \frac{1}{M} \sup_{\mathcal{H}_R} \left(\sum_{u_i \in \mathcal{U}} \sum_{j=1}^{n_i^+} \sum_{k=1}^{n_i^-} \ell^{(i)}(v_j^+, v_k^-) \cdot \left(\hat{\mathbb{P}}_{jk}^{(i)} - \tilde{\mathbb{P}}_{jk}^{(i)} \right) \right). \tag{67}$$

Let

$$\boldsymbol{\ell}^{(i)} = \left[\ell^{(i)}(v_1^+, v_1^-), \ell^{(i)}(v_1^+, v_2^-), \ldots, \ell^{(i)}(v_{n_i^+}^+, v_{n_i^-}^-) \right]^\top,$$

$$\hat{\mathbb{P}}^{(i)} = \left[\hat{\mathbb{P}}_{11}^{(i)}, \ldots, \hat{\mathbb{P}}_{21}^{(i)}, \ldots, \hat{\mathbb{P}}_{n_i^+ 1}^{(i)}, \ldots, \hat{\mathbb{P}}_{n_i^+ n_i^-}^{(i)} \right]^\top,$$

$$\tilde{\mathbb{P}}^{(i)} = \left[\tilde{\mathbb{P}}_{11}^{(i)}, \ldots, \tilde{\mathbb{P}}_{21}^{(i)}, \ldots, \tilde{\mathbb{P}}_{n_i^+ 1}^{(i)}, \ldots, \tilde{\mathbb{P}}_{n_i^+ n_i^-}^{(i)} \right]^\top,$$

and then ③ could be reformulated as

$$\begin{aligned}\sup_{\mathcal{H}_R}\left(\hat{\mathcal{R}}_{\mathcal{S}}^{cml}(f)-\tilde{\mathcal{R}}_{\mathcal{S}}^{cml}(f)\right) &= \frac{1}{M}\cdot\sup_{\mathcal{H}_R}\left(\sum_{u_i\in\mathcal{U}}\sum_{j=1}^{n_i^+}\sum_{k=1}^{n_i^-}\ell^{(i)}(v_j^+,v_k^-)\cdot\left(\hat{\mathbb{P}}_{jk}^{(i)}-\tilde{\mathbb{P}}_{jk}^{(i)}\right)\right)\\ &\overset{(***)}{\leq} \frac{1}{M}\sup_{\mathcal{H}_R}\sum_{u_i\in\mathcal{U}}\left(\left\|\boldsymbol{\ell}^{(i)}\right\|_\infty\cdot\left\|\hat{\mathbb{P}}^{(i)}-\tilde{\mathbb{P}}^{(i)}\right\|_1\right)\\ &= \frac{2\left(\lambda+4R\right)}{M}\sum_{u_i\in\mathcal{U}}\frac{1}{2}\cdot\left\|\hat{\mathbb{P}}^{(i)}-\tilde{\mathbb{P}}^{(i)}\right\|_1\\ &\lesssim \frac{(\lambda+4R)}{M}\cdot\sum_{u_i\in\mathcal{U}}D_{TV}(\hat{\mathbb{P}}^{(i)},\tilde{\mathbb{P}}^{(i)})\end{aligned}\tag{68}$$

where again $(***)$ is achieved by the Cauchy-Buniakowsky-Schwarz Inequality $\left\|\boldsymbol{x}^\top\boldsymbol{y}\right\|\leq\|\boldsymbol{x}\|\cdot\|\boldsymbol{y}\|$ here; $D_{TV}(\hat{\mathbb{P}}^{(i)},\tilde{\mathbb{P}}^{(i)})=\frac{1}{2}\cdot\left\|\hat{\mathbb{P}}^{(i)}-\tilde{\mathbb{P}}^{(i)}\right\|_1$ is the Total Variance(TV), which reflects the discrepancy between sampling-strategy-induced distribution and the ground truth distribution.

***Step 3**. Taking **Step 1** and **Step 2** into account, for any $\delta\in(0,1)$, with probability at least $1-\delta$, the following holds:*

$$\begin{aligned}\tilde{\Phi}(\mathcal{S}) &\lesssim \phi\cdot\frac{\mathbf{max}(\lambda,\sqrt{R\cdot d})}{M}\cdot\sqrt{\frac{1}{\tilde{N}}}\\ &+\phi\cdot\frac{R}{M}\cdot\sqrt{\frac{\log 2/\delta}{2}}\cdot\sqrt{\frac{1}{\tilde{N}}}\\ &+\frac{(\lambda+4R)}{M}\cdot\sum_{u_i\in\mathcal{U}}D_{TV}(\hat{\mathbb{P}}^{(i)},\tilde{\mathbb{P}}^{(i)})\end{aligned}\tag{69}$$

Therefore, for any $\delta\in(0,1)$, with probability at least $1-\delta$, we have

$$\begin{aligned}\mathcal{R}_\ell^{cml}(f) &\lesssim \tilde{\mathcal{R}}_{\mathcal{S}}^{cml}(f)\\ &+\phi\cdot\frac{\mathbf{max}(\lambda,\sqrt{R\cdot d})}{M}\cdot\sqrt{\frac{1}{\tilde{N}}}\\ &+\frac{(\lambda+4R)}{M}\cdot\sum_{u_i\in\mathcal{U}}D_{TV}(\hat{\mathbb{P}}^{(i)},\tilde{\mathbb{P}}^{(i)})\\ &+\phi\cdot\frac{R}{M}\cdot\sqrt{\frac{\log 2/\delta}{2}}\cdot\sqrt{\frac{1}{\tilde{N}}}\end{aligned}\tag{70}$$

This completed the proof.

APPENDIX C
ADDITIONAL EXPERIMENT RESULTS

C.1 Details of evaluation metrics

In some typical recommendation systems, users often care about the top-K items in recommendation lists, so the most relevant items should be ranked first as much as possible. Motivated by this, we evaluate the performance of competitors and our algorithm with the following extensively adopted six metrics, including:

- **Precision** (P@K) counts the proportion that the ground-truth items are among the Top-K recommended list.

$$\text{P@}K = \frac{1}{M}\sum_{u_i \in \mathcal{U}} \frac{|I_{u_i} \cap R_{u_i}|}{K}$$

  where I_{u_i} is the set of ground-truth items of user u_i; R_{u_i} is the top-K recommendation list for user u_i; and $|\cdot|$ means the size of set.
- **Recall** (R@K) is defined as the number of the ground-truth items in top-K recommendation list divided by the amount of totally ground-truth items. This reflects the ability of model to find the relevant items.

$$\text{R@}K = \frac{1}{M}\sum_{u_i \in \mathcal{U}} \frac{|I_{u_i} \cap R_{u_i}|}{|I_{u_i}|}$$

- **Normalized Discounted Cumulative Gain** (NDCG@K) counts the ground-truth items in the top-K recommendation list with a position weighting strategy, i.e., assigning larger value on top items than bottom ones.

$$\text{NDCG@}K = \frac{1}{M}\sum_{u_i \in \mathcal{U}} \frac{\text{DCG}_{u_i}@K}{\text{IDCG}_{u_i}@K}$$

  Specifically, the $\text{DCG}_{u_i}@K$ and $\text{IDCG}_{u_i}@K$ are defined as:

$$\text{DCG}_{u_i}@K = \sum_{j=1}^{|R_{u_i}|} \frac{1 \cdot \mathbb{I}(R_{u_i,j} \in I_{u_i})}{\log_2(j+1)},$$

$$\text{IDCG}_{u_i}@K = \sum_{k=1}^{\min(K,|I_{u_i}|)} \frac{1}{\log_2(k+1)},$$

  where $R_{u_i,j}$ respresents the j-th item in the top-K recommendation list; $\mathbb{I}(\cdot)$ is an indicator function that returns one if the statement is true and returns zero, otherwise.
- **Mean Average Precision** (MAP) is an extension of Average Precision(AP). AP is the average of precision values at all positions where ground-truth items are found.

$$\text{AP}_{u_i} = \frac{1}{|I_{u_i}|}\sum_{j=1}^{|\hat{R}_{u_i}|} \frac{|I_{u_i} \cap \hat{R}_{u_i,1:j}| \cdot \mathbb{I}(j \in I_{u_i})}{rank_j^{u_i}}$$

$$\text{MAP} = \frac{1}{M}\sum_{u_i \in \mathcal{U}} \text{AP}_{u_i}$$

  where different from R_{u_i}, $\hat{R}_{u_i}$ is the recommendation rankings in terms of all items for user u_i; $\hat{R}_{u_i,1:j}$ represents the top-j recommendation list for user u_i; and $rank_j^{u_i}$ means the ranking of item j in $\hat{R}_{u_i}$.
- **Mean Reciprocal Rank** (MRR) takes the rank of each recommended item into account. It is the average of reciprocal ranks of the desired item:

$$\text{MRR} = \frac{1}{M}\sum_{u_i \in \mathcal{U}}\sum_{j=1}^{|\hat{R}_{u_i}|} \frac{1}{rank_j^{u_i}} \cdot \mathbb{I}(\hat{R}_{u_i,j} \in I_{u_i})$$

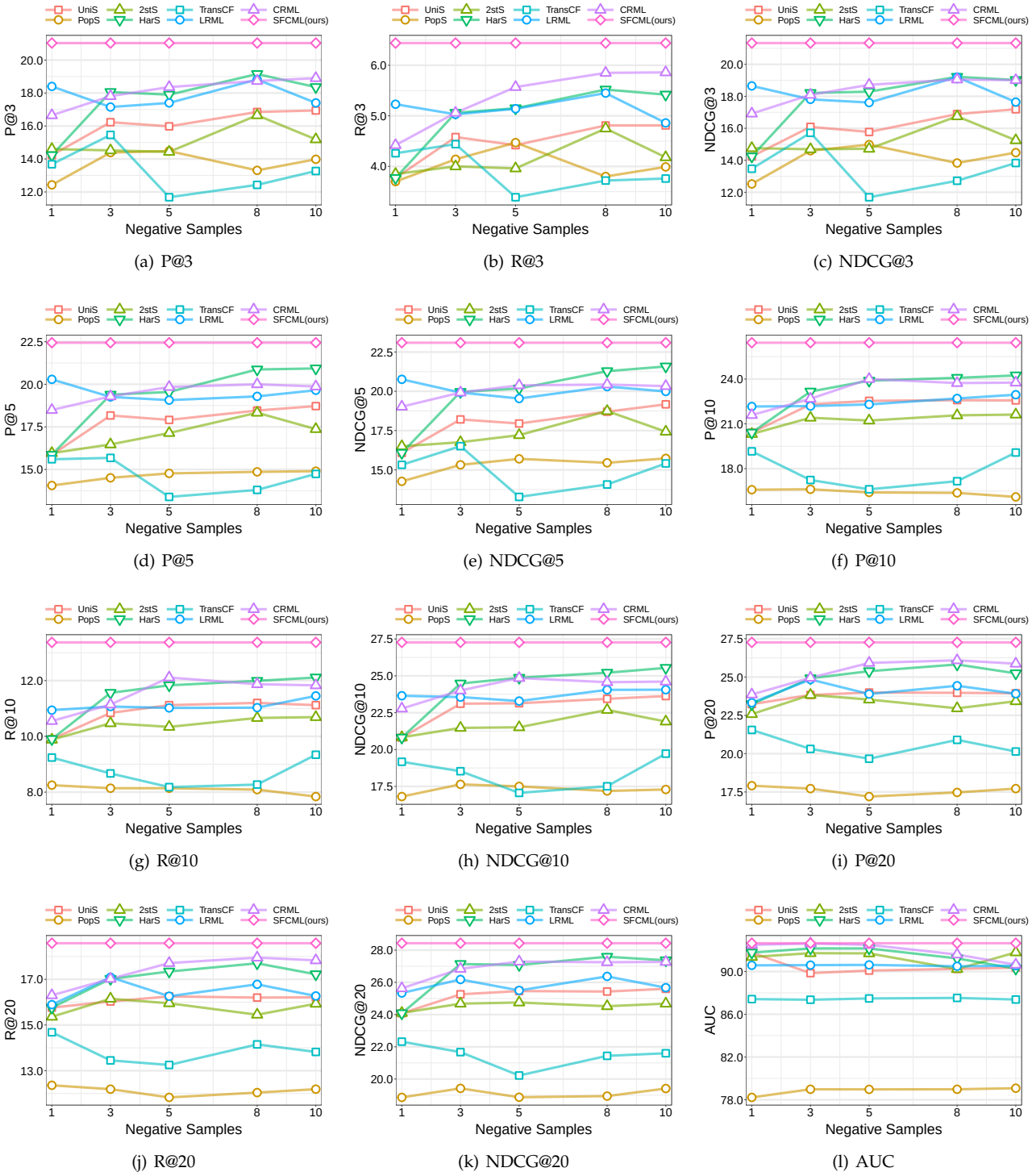
- **Area Under ROC Curve** (AUC) is the probability that a ground-truth item has a higher rank than a negative item.

$$\text{AUC} = \frac{1}{M}\sum_{u_i \in \mathcal{U}} \text{AUC}_{u_i}$$

$$\text{AUC}_{u_i} = \frac{\sum_{j \in \mathcal{V}} rank_j^{u_i} - \frac{|I_{u_i}|(1+|I_{u_i}|)}{2}}{N\,(N - |I_{u_i}|)}$$

Note that, for all the above metrics, the higher the metric is, the better the performance the algorithm achieves.

C.2 Adverse evidence of sampling-based CML



(a) P@3 (b) R@3 (c) NDCG@3

(d) P@5 (e) NDCG@5 (f) P@10

(g) R@10 (h) NDCG@10 (i) P@20

(j) R@20 (k) NDCG@20 (l) AUC

Fig. 7: Performance comparisons on validation set of MovieLens-100k with respect to different negative sampling strategies and different sampling numbers $U \in \{1, 3, 5, 8, 10\}$.

C.3 More evaluation results

C.3.1 More datasets

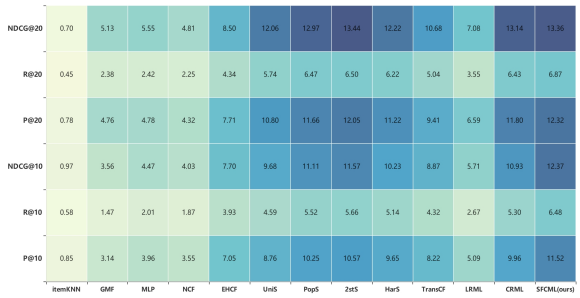
Tab.7 shows the performance comparisons with respect to two larger datasets, i.e., MovieLens-20m and Amazon-Book.

TABLE 7: Performance comparisons on MovieLens-20m and Amazon-Book datasets, where '-' means that we cannot complete the experiments due to the out-of-memory issue. The best and second-best performance are highlighted in bold and underlined, respectively.

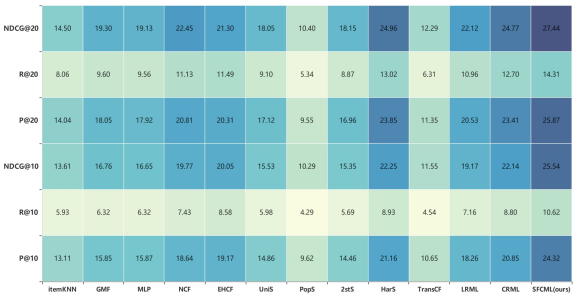
	Method	P@3	R@3	NDCG@3	P@5	R@5	NDCG@5	MAP	MRR	AUC
MovieLens-20m	itemKNN	12.01	3.77	12.33	12.44	4.96	12.94	8.04	26.05	95.82
	GMF	12.45	3.52	12.89	13.40	4.88	13.94	7.94	27.31	96.15
	MLP	14.55	4.02	14.90	15.74	5.72	16.26	10.64	30.52	97.59
	NCF	15.79	4.39	16.07	16.80	6.21	17.42	11.03	31.82	97.49
	EHCF	17.04	5.39	**17.43**	17.66	7.19	**18.34**	12.66	34.61	96.20
	UniS	10.11	2.57	10.38	10.86	3.72	11.35	7.38	23.17	97.71
	PopS	10.31	3.32	10.68	10.38	4.29	10.96	6.40	23.68	89.89
	2stS	12.70	3.50	13.01	13.47	4.90	14.04	9.00	27.76	94.90
	HarS	12.60	3.31	12.95	13.49	4.75	14.09	9.51	27.68	97.60
	TransCF	7.60	2.19	7.77	8.13	3.09	8.41	5.75	18.78	96.06
	LRML	12.57	3.48	12.96	13.22	4.65	13.84	7.71	27.08	96.05
	CRML	14.94	4.01	15.33	16.01	5.73	16.68	11.00	31.23	97.98
	NaiveCML	-	-	-	-	-	-	-	-	-
	SFCML(ours)	**17.16**	**5.78**	**17.43**	**17.90**	**7.52**	18.26	**14.69**	**35.49**	**98.07**
Amazon-Book	itemKNN	1.87	0.83	1.92	1.73	1.10	1.87	1.29	5.25	71.64
	GMF	0.90	0.33	0.90	0.97	0.48	0.97	0.66	3.08	83.11
	MLP	0.91	0.32	0.91	1.04	0.51	1.03	0.85	3.47	89.15
	NCF	1.37	0.44	1.40	1.52	0.72	1.56	1.04	4.56	89.30
	EHCF	3.11	1.09	3.24	3.31	1.69	3.50	2.01	8.66	83.26
	UniS	1.82	0.67	1.93	1.81	0.98	1.97	1.26	5.69	91.56
	PopS	3.20	1.27	3.32	3.33	1.87	3.49	2.06	8.83	88.73
	2stS	3.26	1.18	3.43	3.08	1.62	3.41	1.80	8.28	82.32
	HarS	3.43	1.26	3.53	3.52	1.87	3.72	2.25	9.46	91.31
	TransCF	2.25	0.84	2.33	2.31	1.22	2.43	1.38	6.50	87.16
	LRML	0.47	0.17	0.46	0.40	0.17	0.42	0.29	1.56	77.23
	CRML	3.77	1.34	3.96	3.88	2.10	4.17	2.34	9.98	89.82
	NaiveCML	-	-	-	-	-	-	-	-	-
	SFCML(ours)	**4.13**	**1.61**	**4.24**	**4.05**	**2.26**	**4.30**	**2.79**	**11.14**	**92.66**

C.3.2 More K values

In order to prove the effectiveness of our proposed SFCML algorithm, we further report the performance of SFCML and its competitors with larger $K \in \{10, 20\}$. See Fig.8.

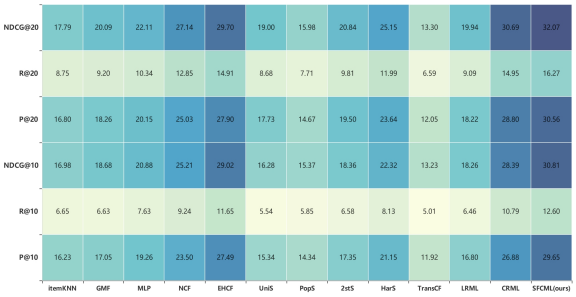


(a) CiteULike

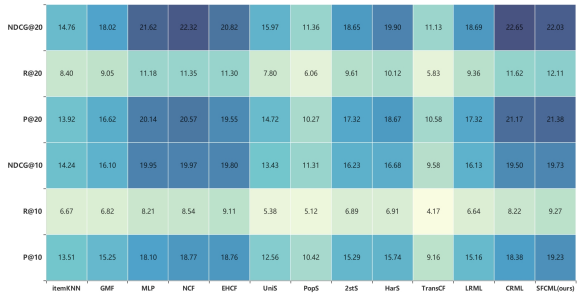


(b) MovieLens-1m

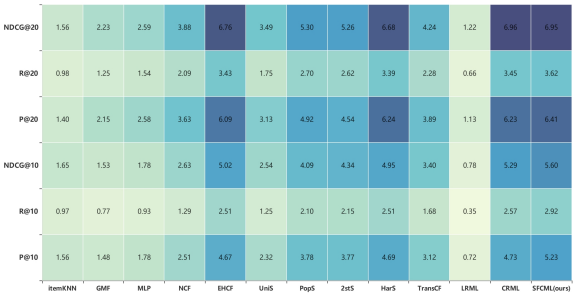
(c) Steam-200k



(d) Anime



(e) MovieLens-20m



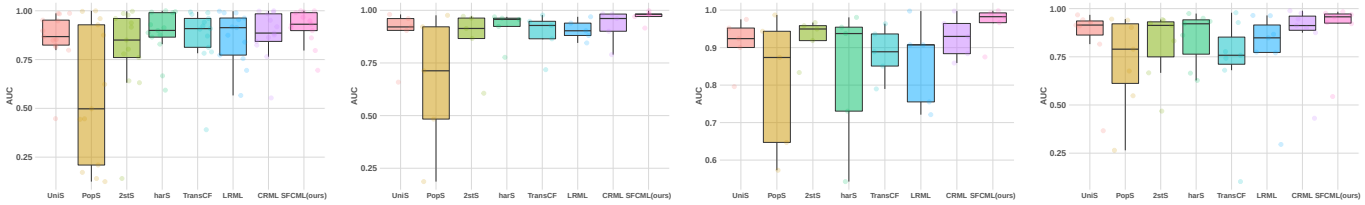
(f) Amazon-Book

Fig. 8: Performance comparisons on CiteULike, Steam-200k, MovieLens-1m, Anime, MovieLens-20m and Amazon-Book datasets with respect to $K \in \{10, 20\}$.

C.4 Fine-grained performance visualization

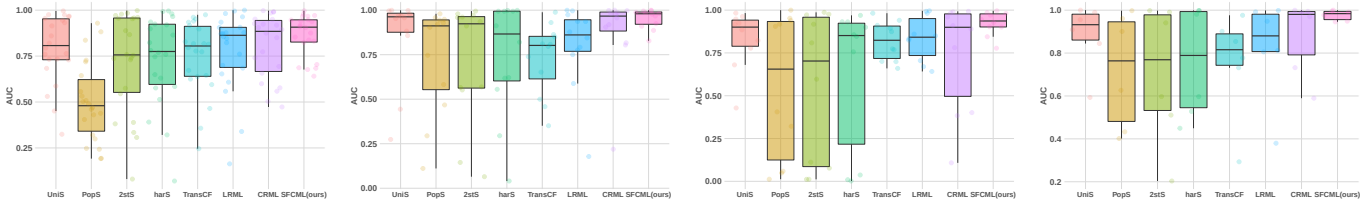
C.4.1 Fine-grained AUC comparison

Please see Fig.9 and Fig.10.



(a) MovieLens-1m-3 (b) MovieLens-1m-4 (c) MovieLens-1m-5 (d) MovieLens-1m-6

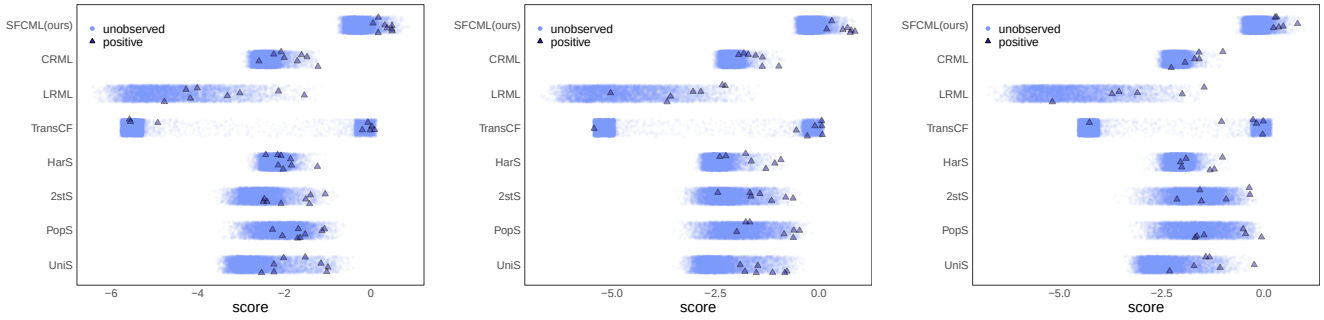
Fig. 9: Fine-grained AUC performance in terms of four users on MovieLens-1m dataset.



(a) Anime-3 (b) Anime-4 (c) Anime-5 (d) Anime-6

Fig. 10: Fine-grained AUC performance with respect to four users on Anime dataset.

C.4.2 Visualization of score density



(a) Anime-1 (b) Anime-2 (c) Anime-3

Fig. 11: The graphical visualization of score distribution of positive and unobserved items on Anime.

C.5 Sensitivity analysis of preference thresholds

TABLE 8: The empirical results of P@3 with respect to different preference thresholds t on MovieLens-100k. The best and second-best are highlighted in bold and underlined, respectively.

Method	Trend	t=1	t=2	t=3	t=4	t=5
itemKNN		11.63	15.34	14.59	11.35	1.32
GMF		18.77	19.09	17.25	14.35	10.23
MLP		18.52	19.26	16.36	14.98	10.02
NCF		25.27	27.00	20.67	15.94	10.07
EHCF		27.43	25.59	22.94	21.13	12.82
UniS		21.85	20.50	21.24	15.94	10.65
PopS		17.99	17.39	16.19	13.05	8.64
2stS		22.16	21.42	20.67	15.50	10.97
HarS		27.29	26.05	25.29	20.76	10.92
TransCF		18.06	18.59	16.86	12.90	9.43
LRML		25.63	26.16	22.87	20.65	11.08
CRML		29.20	29.55	25.29	20.94	10.12
SFCML(ours)		**32.56**	**32.24**	**27.64**	**23.40**	**14.89**

TABLE 9: The empirical results of R@3 with respect to different preference thresholds t on MovieLens-100k. The best and second-best are highlighted in bold and underlined, respectively.

Method	Trend	t=1	t=2	t=3	t=4	t=5
itemKNN		1.77	2.19	2.57	2.41	0.86
GMF		2.85	3.08	3.13	3.37	4.81
MLP		3.24	3.81	3.46	3.93	4.82
NCF		5.26	5.86	4.32	4.11	4.75
EHCF		5.93	5.73	5.63	6.99	6.40
UniS		4.23	3.84	4.79	4.43	4.86
PopS		3.56	3.61	3.74	3.99	4.15
2stS		4.39	4.22	4.67	4.42	4.93
HarS		5.36	5.34	5.79	6.51	5.20
TransCF		3.45	3.85	3.25	3.72	4.46
LRML		5.22	5.74	4.98	6.65	5.07
CRML		5.98	6.65	5.79	6.43	4.94
SFCML(ours)		**6.83**	**7.24**	**6.89**	**7.62**	**7.20**

TABLE 10: The empirical results of NDCG@3 with respect to different preference thresholds t on MovieLens-100k. The best and second-best are highlighted in bold and underlined, respectively.

Method	Trend	t=1	t=2	t=3	t=4	t=5
itemKNN		11.67	15.11	14.51	11.57	1.33
GMF		19.79	20.13	18.05	15.20	10.79
MLP		19.23	20.00	17.30	15.57	10.90
NCF		26.66	27.66	21.65	16.75	11.00
EHCF		28.07	25.86	23.50	21.80	13.21
UniS		22.52	20.81	21.64	16.06	11.18
PopS		18.47	17.75	16.72	13.36	8.60
2stS		22.87	21.76	20.97	15.77	11.26
HarS		27.64	26.52	25.61	21.05	10.90
TransCF		18.45	19.45	16.73	13.32	9.85
LRML		26.43	26.78	23.13	21.44	11.75
CRML		29.87	30.20	25.99	21.80	9.90
SFCML(ours)		**33.50**	**32.94**	**28.00**	**23.63**	**15.19**

TABLE 11: The empirical results of P@5 with respect to different preference thresholds t on MovieLens-100k. The best and second-best are highlighted in bold and underlined, respectively.

Method	Trend	t=1	t=2	t=3	t=4	t=5
itemKNN		12.25	15.08	14.70	12.96	1.38
GMF		17.80	17.35	17.48	16.43	12.06
MLP		18.16	17.71	16.66	15.51	11.53
NCF		24.39	24.65	20.93	17.26	12.06
EHCF		26.02	25.31	23.06	20.89	14.92
UniS		21.19	19.69	21.19	17.04	13.12
PopS		18.11	16.82	16.45	13.38	10.21
2stS		21.91	20.90	20.54	16.76	12.86
HarS		26.45	25.27	23.92	21.36	13.39
TransCF		17.43	18.04	17.27	14.35	11.06
LRML		24.26	24.63	22.36	20.36	14.18
CRML		28.14	27.98	25.30	21.14	11.90
SFCML(ours)		**30.85**	**29.97**	**26.91**	**23.74**	**15.66**

TABLE 12: The empirical results of R@5 with respect to different preference thresholds t on MovieLens-100k. The best and second-best are highlighted in bold and underlined, respectively.

Method	Trend	t=1	t=2	t=3	t=4	t=5
itemKNN		3.14	3.60	3.81	4.11	0.85
GMF		4.40	4.44	4.97	5.79	6.72
MLP		5.18	5.33	5.06	5.70	6.59
NCF		7.65	8.33	6.77	6.45	6.80
EHCF		8.67	9.05	8.35	8.82	8.82
UniS		6.29	5.96	7.21	6.23	7.49
PopS		5.53	5.68	5.57	5.10	5.70
2stS		6.76	6.37	6.89	6.21	7.25
HarS		8.24	8.39	8.26	8.86	7.56
TransCF		5.11	5.88	5.10	5.70	6.14
LRML		7.60	8.27	7.33	8.24	8.02
CRML		9.07	9.93	8.82	8.53	6.65
SFCML(ours)		**10.27**	**10.59**	**9.58**	**9.95**	**9.12**

TABLE 13: The empirical results of NDCG@5 with respect to different preference thresholds t on MovieLens-100k. The best and second-best are highlighted in bold and underlined, respectively.

Method	Trend	t=1	t=2	t=3	t=4	t=5
itemKNN		12.23	15.17	14.92	13.45	1.18
GMF		19.05	18.88	18.44	17.21	13.08
MLP		18.98	18.91	17.63	16.54	12.67
NCF		25.81	26.07	22.01	18.25	13.18
EHCF		27.00	25.80	23.84	22.08	15.41
UniS		22.06	20.39	21.81	17.40	13.66
PopS		18.52	17.39	17.00	13.93	10.30
2stS		22.60	21.49	21.15	17.18	13.60
HarS		27.11	26.09	25.04	22.10	13.49
TransCF		18.01	19.03	17.40	14.76	11.58
LRML		25.46	25.70	23.15	21.75	14.91
CRML		29.19	29.14	26.29	22.44	11.74
SFCML(ours)		**32.22**	**31.49**	**27.80**	**24.65**	**16.46**

TABLE 14: The empirical results of P@10 with respect to different preference thresholds t on MovieLens-100k. The best and second-best are highlighted in bold and underlined, respectively.

Method	Trend	t=1	t=2	t=3	t=4	t=5
itemKNN		15.09	18.97	18.33	17.05	1.09
GMF		20.70	21.00	20.02	18.06	15.43
MLP		20.47	19.97	19.34	17.51	14.42
NCF		25.75	26.19	24.04	18.27	15.58
EHCF		27.02	25.82	24.64	<u>22.70</u>	16.16
UniS		23.85	22.63	23.19	19.88	16.16
PopS		19.82	18.68	17.90	14.68	11.09
2stS		24.39	23.08	22.94	20.32	15.29
HarS		28.07	27.45	26.12	22.21	<u>16.23</u>
TransCF		20.73	20.00	20.73	16.08	14.71
LRML		26.84	25.94	25.36	21.38	16.16
CRML		<u>29.69</u>	<u>29.07</u>	<u>27.14</u>	22.19	14.78
SFCML(ours)		**31.82**	**31.40**	**29.16**	**24.68**	**18.26**

TABLE 15: The empirical results of R@10 with respect to different preference thresholds t on MovieLens-100k. The best and second-best are highlighted in bold and underlined, respectively.

Method	Trend	t=1	t=2	t=3	t=4	t=5
itemKNN		5.40	6.66	7.09	8.19	0.85
GMF		7.83	7.86	7.97	9.02	10.82
MLP		7.98	8.09	8.11	8.86	10.34
NCF		10.09	10.60	10.30	9.27	11.11
EHCF		11.30	10.91	11.16	<u>11.69</u>	11.40
UniS		9.46	8.93	9.85	10.20	11.51
PopS		7.71	7.55	7.60	7.44	7.96
2stS		9.53	9.11	9.77	10.36	10.64
HarS		11.39	11.29	11.58	11.37	<u>11.58</u>
TransCF		8.04	8.40	8.47	8.02	10.29
LRML		10.74	10.64	11.17	10.81	11.50
CRML		<u>12.01</u>	<u>12.06</u>	<u>12.12</u>	11.42	10.36
SFCML(ours)		**13.15**	**13.30**	**13.17**	**12.74**	**13.08**

TABLE 16: The empirical results of NDCG@10 with respect to different preference thresholds t on MovieLens-100k. The best and second-best are highlighted in bold and underlined, respectively.

Method	Trend	t=1	t=2	t=3	t=4	t=5
itemKNN		15.29	19.57	18.95	17.98	1.16
GMF		22.60	23.15	21.76	19.86	16.45
MLP		21.89	21.62	20.77	19.15	15.54
NCF		27.58	28.20	25.56	20.20	16.78
EHCF		28.52	26.89	25.84	24.04	<u>17.47</u>
UniS		24.98	23.73	24.37	20.74	17.09
PopS		20.55	19.51	18.84	15.66	12.02
2stS		25.14	24.27	24.00	20.99	16.79
HarS		29.59	28.72	27.65	23.69	16.38
TransCF		21.36	21.46	21.39	16.68	14.80
LRML		28.20	27.62	26.51	23.09	17.45
CRML		<u>31.39</u>	<u>30.56</u>	<u>28.95</u>	<u>24.17</u>	15.11
SFCML(ours)		**33.82**	**33.29**	**30.57**	**25.90**	**18.91**

TABLE 17: The empirical results of P@20 with respect to different preference thresholds t on MovieLens-100k. The best and second-best are highlighted in bold and underlined, respectively.

Method	Trend	t=1	t=2	t=3	t=4	t=5
itemKNN		17.90	22.94	23.04	20.72	0.87
GMF		23.69	23.65	22.84	20.90	17.17
MLP		23.31	22.24	21.55	19.85	16.30
NCF		28.13	27.38	26.18	21.36	19.34
EHCF		28.43	28.03	25.74	25.05	18.70
UniS		27.18	26.30	27.23	22.92	19.13
PopS		21.65	20.37	20.09	17.10	13.70
2stS		27.27	26.22	26.65	22.87	18.70
HarS		30.20	30.26	28.71	<u>25.41</u>	18.48
TransCF		23.54	19.81	24.15	19.36	17.39
LRML		28.15	26.95	27.16	24.49	<u>19.35</u>
CRML		<u>31.42</u>	<u>30.88</u>	<u>29.76</u>	<u>25.41</u>	17.39
SFCML(ours)		**32.88**	**32.42**	**31.05**	**27.26**	**21.09**

TABLE 18: The empirical results of R@20 with respect to different preference thresholds t on MovieLens-100k. The best and second-best are highlighted in bold and underlined, respectively.

Method	Trend	t=1	t=2	t=3	t=4	t=5
itemKNN		9.16	12.15	13.13	13.86	0.83
GMF		12.24	12.70	13.07	13.96	13.66
MLP		12.13	12.16	12.46	13.24	13.07
NCF		14.80	14.88	15.05	14.24	15.31
EHCF		15.54	15.76	15.07	16.76	14.55
UniS		14.35	14.24	15.63	15.30	15.34
PopS		11.41	11.00	11.46	11.36	10.71
2stS		14.38	14.23	15.20	15.23	14.98
HarS		16.21	16.83	16.92	<u>17.20</u>	14.57
TransCF		12.43	10.78	13.41	12.91	13.84
LRML		15.01	14.71	15.83	16.64	<u>15.52</u>
CRML		<u>16.82</u>	<u>17.16</u>	<u>17.29</u>	17.14	13.46
SFCML(ours)		**17.89**	**18.11**	**18.10**	**18.46**	**16.46**

TABLE 19: The empirical results of NDCG@20 with respect to different preference thresholds t on MovieLens-100k. The best and second-best are highlighted in bold and underlined, respectively.

Method	Trend	t=1	t=2	t=3	t=4	t=5
itemKNN		18.54	24.27	24.14	22.34	0.72
GMF		25.92	26.15	25.06	22.99	18.24
MLP		24.83	23.85	23.34	21.72	17.14
NCF		30.21	29.85	28.07	23.26	19.77
EHCF		30.08	29.31	27.16	27.27	19.65
UniS		28.55	27.51	28.36	24.20	20.61
PopS		23.14	21.55	21.54	17.97	13.65
2stS		28.56	27.41	27.77	24.02	20.60
HarS		32.08	31.79	30.28	27.56	19.56
TransCF		24.49	21.71	25.35	20.59	17.74
LRML		30.30	28.98	29.06	26.57	<u>21.08</u>
CRML		<u>33.63</u>	<u>32.62</u>	<u>31.41</u>	<u>27.80</u>	18.06
SFCML(ours)		**35.38**	**34.93**	**32.61**	**29.47**	**21.34**

TABLE 20: The empirical results of MAP with respect to different preference thresholds t on MovieLens-100k. The best and second-best are highlighted in bold and underlined, respectively.

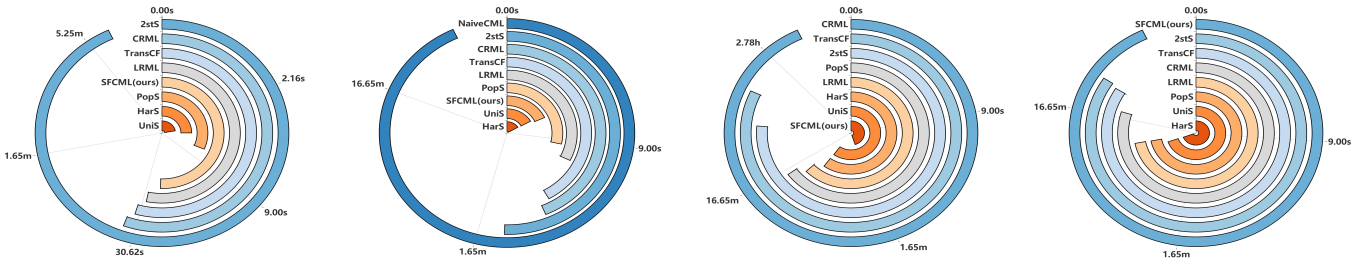
Method	Trend	t=1	t=2	t=3	t=4	t=5
itemKNN		8.81	10.09	9.96	8.49	3.43
GMF		11.24	11.10	10.64	9.82	9.19
MLP		11.52	11.93	10.92	10.09	9.74
NCF		16.33	16.32	14.26	11.35	9.87
EHCF		19.51	19.15	17.61	<u>16.51</u>	<u>11.45</u>
UniS		16.05	15.27	15.93	13.21	10.10
PopS		12.06	11.79	11.11	9.49	7.24
2stS		16.63	15.76	15.78	13.35	9.89
HarS		18.80	18.85	17.68	15.94	11.01
TransCF		12.90	12.97	11.94	11.19	9.10
LRML		16.82	16.92	15.65	13.48	10.40
CRML		<u>19.64</u>	<u>20.17</u>	<u>17.86</u>	16.33	10.31
SFCML(ours)		**22.51**	**22.49**	**20.22**	**18.00**	**13.34**

TABLE 21: The empirical results of MRR with respect to different preference thresholds t on MovieLens-100k. The best and second-best are highlighted in bold and underlined, respectively.

Method	Trend	t=1	t=2	t=3	t=4	t=5
itemKNN		24.27	28.75	28.05	24.63	6.10
GMF		37.44	37.39	34.76	31.00	24.76
MLP		37.99	38.95	35.36	31.99	25.99
NCF		47.99	48.20	41.29	34.34	26.11
EHCF		49.02	46.46	43.86	<u>41.77</u>	<u>29.02</u>
UniS		42.26	38.90	40.48	33.07	26.11
PopS		36.90	35.59	33.97	29.13	20.56
2stS		42.99	40.54	39.68	32.95	25.44
HarS		47.55	47.05	45.04	40.02	25.81
TransCF		36.48	38.52	32.83	29.88	23.64
LRML		46.98	47.24	42.15	37.93	27.03
CRML		<u>50.73</u>	<u>51.54</u>	<u>46.21</u>	41.14	23.45
SFCML(ours)		**55.03**	**54.58**	**48.81**	**43.13**	**31.87**

C.6 Additional results of efficiency

C.6.1 Additional average efficiency



(a) CiteULike (b) Steam-200k (c) Anime (d) Amazon-Book

Fig. 12: Comparisons against average running time with respect to CML framework algorithms and SFCML(ours) on CiteULike, Steam-200k, Anime and Amazon-Book datasets. The method closer to the center of the circle enjoys better efficiency. Note that, here the 's', 'm', and 'h' represent the second, minute and hour, respectively.

C.6.2 Running time per epoch

TABLE 22: Comparisons against running time with respect to CML framework algorithms and SFCML(ours) on MovieLens-100k and Steam-200k, where '-' means that we cannot complete the experiments due to the out-of-memory issue. Note that, here the 's' and 'm' represent the second and minute, respectively. The best and second-best are highlighted in bold and underlined, respectively.

	Method	Epoch 1	Epoch 2	Epoch 3	Epoch 4	Epoch 5	Epoch 6	Epoch 7	Epoch 8	Epoch 9	Epoch 10	Average ↓
MovieLens-100k	UniS	1.52s	1.52s	1.51s	1.51s	1.50s	1.50s	1.49s	1.51s	1.49s	1.50s	<u>1.51s</u>
	PopS	4.10s	4.07s	4.11s	4.09s	4.09s	4.11s	4.09s	4.07s	4.10s	4.12s	4.10s
	2stS	31.70s	32.97s	32.58s	33.68s	33.96s	33.79s	32.43s	35.43s	34.59s	35.55s	33.67s
	HarS	1.54s	1.55s	1.55s	1.55s	1.56s	1.57s	1.54s	1.54s	1.54s	1.54s	1.55s
	TransCF	17.94s	16.17s	17.20s	17.60s	16.66s	16.94s	16.80s	17.41s	16.93s	16.88s	17.05s
	LRML	6.67s	6.60s	6.46s	6.66s	6.49s	6.54s	6.61s	6.57s	6.66s	6.59s	6.59s
	CRML	23.41s	23.52s	23.61s	23.42s	23.44s	23.60s	23.33s	23.51s	23.53s	23.63s	23.50s
	NaiveCML	6.87m	6.86m	6.88m	6.87m	6.88m	6.86m	6.87m	6.88m	6.88m	6.88m	6.87m
	SFCML(ours)	0.37s	0.37s	0.37s	0.38s	0.38s	0.37s	0.37s	0.37s	0.37s	0.37s	**0.37s**
Steam-200k	UniS	3.22s	3.25s	3.21s	3.25s	3.20s	3.18s	3.18s	3.19s	3.21s	3.20s	<u>3.21s</u>
	PopS	10.38s	10.33s	10.38s	10.39s	10.34s	10.36s	10.35s	10.32s	10.35s	10.32s	10.35s
	2stS	1.22m	1.23m	1.17m	1.19m	1.17m	1.17m	1.16m	1.24m	1.16m	1.19m	1.19m
	HarS	3.03s	3.11s	3.10s	3.06s	2.99s	3.07s	3.09s	3.11s	3.07s	3.05s	**3.07s**
	TransCF	34.10s	33.66s	33.74s	33.68s	33.80s	33.70s	34.11s	33.75s	33.81s	33.75s	33.81s
	LRML	14.37s	14.55s	14.55s	14.43s	14.55s	14.54s	14.37s	14.47s	14.31s	14.49s	14.46s
	CRML	39.78s	39.76s	39.55s	40.18s	38.99s	39.38s	40.44s	39.64s	39.7s	39.96s	39.74s
	NaiveCML	45.59m	45.57m	45.7m	45.76m	45.69m	45.52m	45.46m	45.56m	45.47m	45.35m	45.57m
	SFCML(ours)	3.81s	3.68s	3.64s	3.72s	3.68s	3.59s	3.64s	3.62s	3.72s	3.73s	3.68s

TABLE 23: Comparisons against running time with respect to CML framework algorithms and SFCML(ours) on CiteULike, MovieLens-1m and Anime, where '-' means that we cannot complete the experiments due to the out-of-memory issue. Note that, here the 's', 'm' and 'h' represent the second, minute and hour, respectively. The best and second-best are highlighted in bold and underlined, respectively.

	Method	Epoch 1	Epoch 2	Epoch 3	Epoch 4	Epoch 5	Epoch 6	Epoch 7	Epoch 8	Epoch 9	Epoch 10	Average ↓
CiteULike	UniS	3.16s	3.13s	3.14s	3.15s	3.08s	3.15s	3.10s	3.04s	3.13s	3.12s	**3.12s**
	PopS	6.49s	6.52s	6.50s	6.45s	6.49s	6.44s	6.46s	6.48s	6.53s	6.48s	6.48s
	2stS	6.35m	6.60m	6.56m	6.56m	6.39m	6.34m	6.45m	6.33m	6.50m	6.27m	6.43m
	HarS	3.92s	3.96s	3.98s	3.93s	3.93s	3.92s	3.97s	3.94s	3.91s	3.91s	3.94s
	TransCF	32.30s	32.35s	31.97s	32.24s	31.30s	31.60s	32.30s	31.68s	32.33s	32.25s	32.03s
	LRML	29.29s	28.93s	29.06s	29.36s	29.35s	29.08s	29.24s	29.42s	29.34s	29.05s	29.21s
	CRML	34.29s	34.42s	34.41s	34.25s	34.30s	34.40s	34.23s	34.39s	34.27s	34.31s	34.33s
	NaiveCML	-	-	-	-	-	-	-	-	-	-	-
	SFCML(ours)	24.21s	24.15s	24.22s	24.26s	24.20s	24.21s	24.17s	24.18s	24.20s	24.21s	24.20s
MovieLens-1m	UniS	11.13s	11.52s	11.41s	11.59s	11.37s	11.11s	11.17s	11.61s	11.24s	11.39s	11.35s
	PopS	40.82s	40.63s	40.41s	40.12s	40.51s	40.12s	40.38s	40.67s	40.54s	40.37s	40.46s
	2stS	4.80m	4.85m	4.87m	4.78m	4.87m	4.83m	4.90m	4.89m	4.89m	4.80m	4.85m
	HarS	12.18s	12.15s	12.10s	12.01s	12.13s	12.03s	12.19s	12.04s	12.14s	12.14s	12.11s
	TransCF	3.23m	3.24m	3.2m	3.2m	3.18m	3.14m	3.2m	3.17m	3.17m	3.16m	3.19m
	LRML	1.25m	1.24m	1.26m	1.26m	1.26m	1.23m	1.25m	1.25m	1.26m	1.24m	1.25m
	CRML	6.93m	6.92m	6.94m	6.94m	6.96m	6.9m	6.93m	6.93m	6.91m	6.90m	6.93m
	NaiveCML	-	-	-	-	-	-	-	-	-	-	-
	SFCML(ours)	5.88s	5.89s	5.88s	5.87s	5.86s	5.90s	5.89s	5.87s	5.90s	5.85s	**5.88s**
Anime	UniS	8.77m	8.77m	8.74m	8.73m	8.75m	8.79m	8.77m	8.78m	8.73m	8.79m	8.76m
	PopS	14.96m	14.70m	14.69m	14.59m	15.02m	14.81m	14.80m	15.04m	14.98m	14.93m	14.85m
	2stS	49.55m	48.37m	48.98m	48.95m	47.92m	49.60m	49.80m	49.27m	49.84m	48.94m	49.12m
	HarS	8.88m	8.93m	9.00m	8.93m	8.85m	8.91m	8.83m	8.86m	8.98m	8.80m	8.90m
	TransCF	1.49h	1.49h	1.44h	1.49h	1.49h	1.48h	1.50h	1.46h	1.47h	1.48h	1.48h
	LRML	12.03m	12.04m	12.04m	11.97m	11.97m	11.97m	11.93m	11.96m	11.99m	12.00m	11.99m
	CRML	4.71h	4.73h	4.72h	4.72h	4.69h	4.74h	4.73h	4.7h	4.72h	4.71h	4.72h
	NaiveCML	-	-	-	-	-	-	-	-	-	-	-
	SFCML(ours)	1.84m	1.85m	1.84m	1.84m	1.85m	1.85m	1.85m	1.84m	1.85m	1.84m	**1.85m**

TABLE 24: Comparisons against running time with respect to CML framework algorithms and SFCML(ours) on MovieLens-20m and Amazon-Book, where '-' means that we cannot complete the experiments due to the out-of-memory issue. Note that, here the 's', 'm' and 'h' represent the second, minute and hour, respectively. The best and second-best are highlighted in bold and underlined, respectively.

	Method	Epoch 1	Epoch 2	Epoch 3	Epoch 4	Epoch 5	Epoch 6	Epoch 7	Epoch 8	Epoch 9	Epoch 10	Average ↓
MovieLens-20m	UniS	17.34m	16.98m	17.21m	17.10m	16.86m	17.00m	16.93m	17.05m	17.00m	17.18m	17.07m
	PopS	41.52m	41.15m	41.44m	41.60m	41.18m	41.26m	41.40m	41.45m	41.43m	41.50m	41.39m
	2stS	2.22h	2.20h	2.17h	2.15h	2.16h	2.17h	2.06h	2.05h	2.19h	2.07h	2.14h
	HarS	17.66m	17.66m	17.80m	17.70m	17.73m	17.33m	17.43m	17.50m	17.69m	17.45m	17.60m
	TransCF	4.49h	4.43h	4.47h	4.51h	4.50h	4.48h	4.44h	4.47h	4.46h	4.47h	4.47h
	LRML	23.35m	23.46m	23.56m	23.35m	23.37m	23.72m	23.60m	23.56m	23.35m	23.46m	23.48m
	CRML	18.52h	18.49h	18.47h	18.51h	18.42h	18.51h	18.53h	18.54h	18.52h	18.48h	18.50h
	NaiveCML	-	-	-	-	-	-	-	-	-	-	-
	SFCML(ours)	11.99m	12.03m	11.99m	11.97m	12.04m	11.96m	11.99m	12.02m	12.01m	11.97m	**12.00m**
Book	UniS	7.53m	7.60m	7.48m	7.59m	7.65m	7.55m	7.66m	7.68m	7.53m	7.56m	7.58m
	PopS	9.40m	9.20m	9.27m	9.39m	9.35m	9.23m	9.33m	9.33m	9.20m	9.19m	9.29m
	2stS	28.46m	27.69m	27.13m	28.02m	27.03m	27.19m	28.03m	27.87m	28.03m	28.37m	27.78m
	HarS	7.52m	7.44m	7.41m	7.45m	7.51m	7.37m	7.31m	7.37m	7.27m	7.46m	**7.41m**
	TransCF	26.57m	26.57m	26.59m	26.57m	26.59m	26.59m	26.59m	26.58m	26.58m	26.57m	26.58m
	LRML	9.37m	9.42m	9.38m	9.38m	9.35m	9.36m	9.41m	9.39m	9.38m	9.38m	9.38m
	CRML	18.93m	18.80m	18.71m	18.94m	18.81m	18.62m	18.84m	18.77m	18.82m	18.80m	18.80m
	NaiveCML	-	-	-	-	-	-	-	-	-	-	-
	SFCML(ours)	59.77m	59.53m	59.90m	59.76m	58.98m	59.90m	59.73m	60.00m	60.00m	60.00m	59.79m