# O$^3$ERS: An Explainable Recommendation System with Online Learning, Online Recommendation, and Online Explanation

Qianqiao Liang[a], Xiaolin Zheng[a,*], Yan Wang[b] and Mengying Zhu[a]

[a]College of Computer Science, Zhejiang University, Hangzhou 310027, China
[b]Department of Computing, Macquarie University, Sydney, NSW 2109, Australia

ARTICLE INFO

*Keywords*:
Explainable recommendation systems
Online learning
Factorization bandit

ABSTRACT

Explainable recommendation systems (ERSs) have attracted increasing attention from researchers, which generate high-quality recommendations with intuitive explanations to help users make appropriate decisions. However, most of the existing ERSs are designed with an offline setting, which can hardly adjust their models using the online feedback instantly for improved performance. To overcome the limitations of ERSs with the offline setting, we propose a novel online setting for ERSs and devise an effective model called O$^3$ERS in this online setting, which can perform online learning with good scalability and rigorous theoretical guides for better online recommendations and online explanations. O$^3$ERS also addresses two challenging problems in real scenarios, namely, the sparsity and delay of online explanations' feedback as well as the partialness and insufficiency of online recommendations' feedback. Specifically, O$^3$ERS not only instantly leverages the knowledge learned from the recommendations' feedback to adjust the sparse and delayed explanations' feedback for better explanations but also utilizes a novel exploitation–exploration strategy that incorporates the explanations' feedback to adjust the partial and insufficient recommendations' feedback for better recommendations. Our theoretical analysis and empirical studies on one simulated and two real-world datasets show that our model outperforms the state-of-the-art models in online scenarios remarkably.

## 1. Introduction

Explainable recommendation systems (ERSs) have attracted increasing attention from researchers. ERSs provide recommended items to users along with explanations to clarify why users might like the recommended items [47]. For example, Figure 1 presents a common type of ERSs called tag-based ERS, whose explainable recommendation includes a recommended movie attached with tags for explanations [30, 37]. ERSs not only aim to improve the accuracy of the recommendation task but also target three goals of the explanation task, namely, (i) *effectiveness*: identifying the features of the recommended items correctly, (ii) *persuasiveness*: convincing users to accept the recommended items, and (iii) *satisfaction*: increasing users' satisfaction degree on the recommendation system [12].

Most of the existing ERSs are designed with an offline setting, that is, their models are trained using users' historical behavior and remain static after deployment [18, 42, 50]. This setting for ERS often violates the reality that users' historical behavior cannot always reveal users' true preference. Therefore, ERSs with the offline setting can hardly be practical because they fail to analyze users' behavior from instant feedback dynamically. Take the ERS in Figure 1(b) as an example. At round $t$, the ERS recommended a movie attached with tags for explanations to a user who arrived at this round. At round $t+1$, this user interacted with the ERS and provided feedback (i.e., recommendation's feedback and explanation's feedback) to evaluate the recommended movies, which may provide hints about the user's true preference. An ideal ERS is supposed to optimize its explainable recommendation model based on the instant feedback. However, the ERS with the offline setting is unable to utilize the feedback instantly at round $t+1$ to adjust its model for improved performance. A simple way for existing ERSs to utilize the feedback is to store the feedback and refresh their models periodically [26]. However, even in a batch-update fashion, ERSs still suffer from (i) poor scalability because they often have to re-train their models from scratch with new feedback, and (ii) lack of rigorous theoretical guarantee on the online performance [17].

To overcome the aforementioned limitations, we propose a new setting for ERSs, namely, an online setting, with which ERSs adopt some online learning techniques to learn from the instant feedback at each round continuously and

*Corresponding author
liangqq@zju.edu.cn (Q. Liang); xlzheng@zju.edu.cn (X. Zheng); yan.wang@mq.edu.au (Y. Wang); mengyingzhu@zju.edu.cn (M. Zhu)

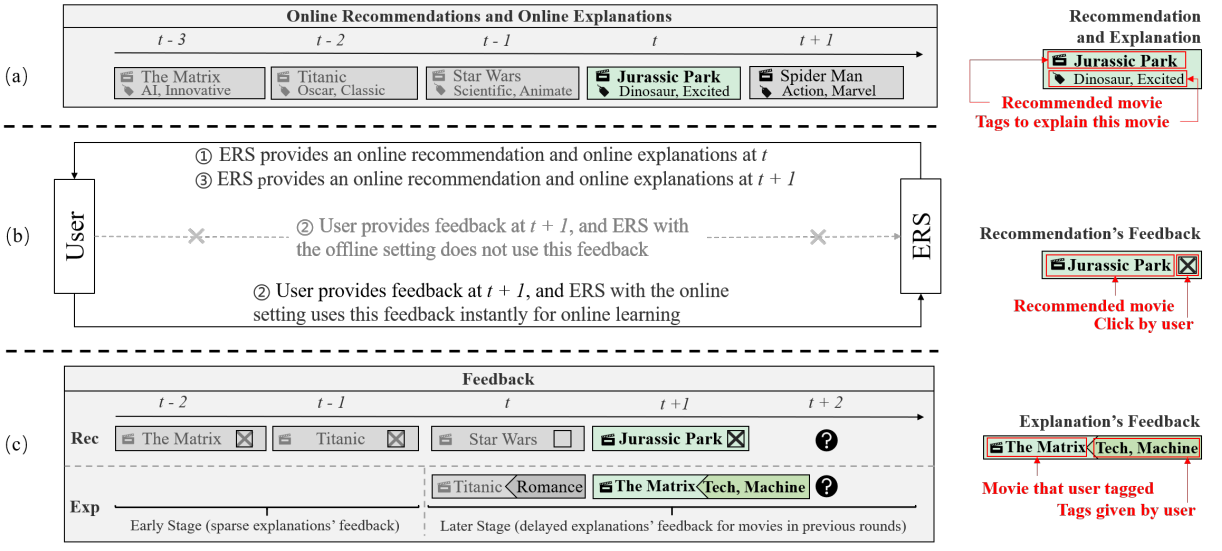



**Figure 1**: An example of a tag-based ERS. (a) Online recommendation attached with online explanation at each round. (b) Process of an ERS with the offline/online setting. (c) Recommendation's feedback (Rec) and explanation's feedback (Exp) at each round.

provide online recommendations attached with online explanations (also depicted in Figure 1(b)). Notably, online learning techniques are often based on solid theories with rigorous regret bounds on their online performance. Therefore, ERSs with the online setting may provide better scalability and stronger guarantee on the performance compared with existing ERSs with the offline setting.

However, ERSs with the online setting face two challenging problems in real scenarios that are needed to be addressed for greater feasibility.

**Problem 1:** How to ensure the effectiveness, persuasiveness, and users' satisfaction of the explanation task with sparse and delayed online explanations' feedback?

Sparse and delayed online explanations' feedback affects the performance of the explanation task negatively. The effectiveness, persuasiveness, and users' satisfaction of an explanation task can be achieved if the explanation can describe the item's features accurately and match the user's preference exactly [37], which can be learned from adequate feedback. However, in real scenarios, online explanations' feedback is sparse and delayed. Take the tag-based ERS in Figure 1(c) as an example, wherein explanations' feedback refers to the tagging behavior from users. Little is known about the movie 'Titanic' after its first release because tags about it were *sparse* at an early stage. Likewise, we could not recognize 'Titanic' as a romantic movie until the user took some time to identify the features of this movie and labeled it at a later stage, thereby indicating that the feedback is *delayed*. The sparsity and delay of online explanations' feedback increase the difficulty in learning items' features and users' preference, thereby degrading the overall performance of the explanation task.

**Problem 2:** How to ensure the performance of the recommendation task with partial and insufficient online recommendations' feedback?

Partial and insufficient online recommendations' feedback affects the performance of the recommendation task negatively. Most existing models for recommendation systems perform well if they have access to full and sufficient recommendations' feedback for learning [10, 28]. However, in real scenarios, online recommendations' feedback is *partial* and *insufficient* [17]. Take the tag-based ERS in Figure 1(c) as an example, wherein recommendations' feedback refers to whether the user has clicked the recommended items. This recommendations' feedback only indicates whether the user is satisfied with the recommended items instead of the user's particular favorite item in hindsight. In other words, ERSs are unaware of the recommendations' feedback for other items outside the recommendation list. If a model trusts the recommendations' feedback fully and always selects the same item for recommendations, then how would it know if other items are better? Therefore, exploiting the partial and insufficient recommendations' feedback fully to update the model degrades the performance of the recommendation task. The most common solution for this problem is to explore some currently less-promising items properly for model correction. However, little is known

about deriving a new exploitation–exploration strategy with a theoretical guarantee on the performance if we need to perform the explanation task simultaneously.

To adapt to the ERS with the online setting fully and achieve greater feasibility in real scenarios, we propose an effective model called O$^3$ERS, which can perform online learning with good scalability and rigorous theoretical guides for better online recommendations and online explanations. At each round, O$^3$ERS first leverages both recommendations' feedback (i.e., click or not) and explanations' feedback (i.e., tagging behavior) to learn the latent vector of each user, item, and tag adaptively, which indicates their latent features accordingly. Then, O$^3$ERS selects an item for recommendation based on an effective exploitation–exploration strategy. Finally, O$^3$ERS constructs an intuitive tag-cloud for explanation based on the latent vectors of the user, the recommended item, and all tags in this round. Specifically, our proposed model can address the two aforementioned challenging problems of ERSs with the online setting. Targeting **Problem 1**, we relate the recommendation and the explanation tasks in O$^3$ERS, which can leverage the knowledge learned from the recommendations' feedback instantly at every round to reconcile the sparse and delayed explanations' feedback for better explanations. Targeting **Problem 2**, we derive a novel confidence set for each latent vector to construct an effective exploitation–exploration strategy in O$^3$ERS, which incorporates the explanations' feedback to adjust the partial and insufficient recommendations' feedback for better recommendations.

The main contributions of our work are detailed as follows:

1. We propose a novel online setting for ERSs and point out two challenging problems of ERSs with the online setting in real scenarios, namely, the sparsity and delay of online explanations' feedback and the partialness and insufficiency of online recommendations' feedback, both of which affect ERSs' performance negatively.
2. We propose an effective model called O$^3$ERS in the online setting, which not only provides good scalability and rigorous theoretical guarantee on the performance but also takes advantage of the interrelation between the recommendation and explanation tasks to address the two challenging problems in real scenarios.
3. We have performed rigorous analysis to prove that O$^3$ERS achieves a sub-linear regret upper bound, which guarantees that the number of sub-optimal recommendations from O$^3$ERS reduces rapidly over time. This regret upper bound has the same scaling as the state-of-the-art online learning recommendation models which perform the recommendation task only, rather than both the recommendation and the explanation tasks.
4. We have conducted extensive experiments on one simulated and two real-world datasets, which show that O$^3$ERS not only outperforms the state-of-the-art recommendation models remarkably in terms of cumulative regret and cumulative reward but also provides effective, persuasive, and satisfactory explanations. Moreover, O$^3$ERS has better efficiency in terms of execution time than the state-of-the-art models for ERSs.

The remainder of this paper is organized as follows. Section 2 provides a summary of prior works relevant to this study. Section 3 formulates the underlying explainable recommendation problem and provides our solution with rigorous theoretical analysis. Section 4 reports extensive evaluation results. Section 5 concludes this paper.

## 2. Related Work

In this section, we review the studies of two different groups of recommendation systems related to this study, namely, explainable recommendation systems and online learning recommendation systems.

### 2.1. Explainable Recommendation Systems

Explainable recommendation systems provide personalized recommended items and intuitive explanations to users. Studies show that explanations play an important role in helping users evaluate a recommendation system [34, 35], and various models have been proposed to generate explanations [47], whose solutions can be classified into two categories, namely, content-based collaborative filtering models and latent factor models.

#### *2.1.1. Content-based Collaborative Filtering Models*

This model category is a common type in early studies. Models under this category build up users' and items' profiles with various available content information, such as the demographic information of the users, or the brands of the items in recommendation systems [4, 29]. Some studies focused on the profiles' similarities showed that providing content-based explanations can help improve the recommendations performance empirically. Pazzani et al. [30] and Cramer et al. [11] matched pre-processed users' profiles with the items' profiles to generate recommendations and explanations. Vig et al. [37] displayed the items' profiles as tags and informed each user about the relevance of each

tag to him/her. Considering that items' contents are easily understandable to users, they are usually intuitive to explain to users why the items are recommended. Some researchers further exploit these contents to link similar users and items. They studied user-based and item-based explanations, which find a set of similar users or similar items, for the target user or recommended item and explain that the recommendation is based on such similarities [10, 16, 31]. The problems of user-based collaborative filtering explanations include trustworthiness and privacy concerns, because the target user may have no idea about other users or items who have 'similar contents'. To let the explanations become more persuasive, Sharma and Cosley [32] adopted the content of social relationships for explanations, which informed users that their friends had similar interests in the recommended item. Content-based collaborative filtering models are the most common way in early studies because can be interpreted easily. However, their quality is known to be inferior to modern latent factor models.

#### *2.1.2. Latent Factor Models*

This model category is the most common type in later studies because of its promising quality. Models under this category map users, recommendation candidates, and different explanation interfaces (e.g., tags [50], reviews texts [13], photos [14]) to a lower-dimensional latent space and obtains their latent vectors, which encode affinities among different entities and are proven to be more expressive. Among all the explanation interfaces in the latent factor models for ERSs, tag-based explanation has become a popular one because of the increasing abundance of textual information on websites, such as Delicious[1] and Movielens[2]. For this reason, the tag-based explanation is also the explanation interface that our study focuses on. Several studies learned the latent vectors of tags to provide tag-based explanations. Zhang et al. [48] proposed the first explicit factor model for explainable recommendations, which presented tag clouds as explanations to highlight the performance of the recommended items on certain aspects. Chang et al. [9] and Balog et al. [5] embedded personalized tags into a natural language sentence to generate more expressive explanations. Hou et al. [18] grouped similar tags to generate topics of each item for explanations. To investigate the semantic information of tags further, Zheng et al. [50] integrated the idea of topic modeling into tag-based explanations for improved explanations. However, these existing models are designed with the offline setting, which are hardly practical because they fail to adjust their models dynamically using instant feedback. McInerney et al. [26] attempted to update their explainable recommendation model using online feedback, which could also provide tag-based explanations after updating the latent vectors of users, items, and tags. However, as emphasized in their work, updating their model at every round required the increase in training time because their model update was based on recalculation from scratch using all the data seen up to the latest observation. For this reason, in practice, their model is retrained periodically in batch mode (e.g., once per day or once per training dataset), which still suffers from poor scalability and lack of theoretical guarantee on their performance of the recommendation task when deployed in real scenarios.

In conclusion, both categories of the existing ERSs models can hardly learn from the online feedback effectively and fail to address the sparse and delayed explanations' feedback as discussed in **Problem 1**, thereby degrading the performance of ERSs in real scenarios.

### 2.2. Online Learning Recommendation Systems

Online learning recommendation systems provide users with personalized recommended items and analyze their real-time feedback for improved recommendations. The solutions for online learning recommendation systems can be classified into two categories, namely, online supervised learning models and factorization bandit learning models.

#### *2.2.1. Online Supervised Learning Models*

Online supervised learning models perform supervised learning tasks, which can learn users' preference and items' features correctly if explicit recommendation feedback is provided. Liu et al. [25] provided an efficient similarity score updating model to perform online memory-based collaborative filtering. Abernethy et al. [2] applied an online supervised learning algorithm, namely, online gradient descent, to the regularized loss function of the latent matrix factorization model for recommendations. Later, several improved algorithms were proposed, which adopted more advanced update strategies beyond online gradient descent and thus could achieve faster adaptation for rapid users' preference changes in the real world. Such advanced update strategies include the multi-task collaborative filtering

[1] http://del.icio.us
[2] https://movielens.org/

algorithm [41], dual-averaging online probabilistic matrix factorization algorithm [23], adaptive gradient online probabilistic matrix factorization algorithm [23], and alternative least square based fast matrix factorization algorithm [15]. However, in real scenarios of the ERSs, users' feedback for recommendations is partial and insufficient, as pointed out in **Problem 2**. Considering that online supervised learning models fully trust the recommendation feedback without any exploration, they may provide sub-optimal recommended items and are not the best choice for online learning recommendation systems.

#### *2.2.2. Factorization Bandit Learning Models*

Factorization bandit learning models introduce an exploration–exploitation mechanism to address the problems that occurred with partial and insufficient recommendations' feedback [33]. Specifically, because the latent vectors of users and items updated at each round are uncertain, a factorization bandit-based model will derive a confidence set for each of them with a rigorous theoretical guarantee on its performance to guide the item selection in the recommendation task [3]. Factorization bandits can be divided into two types. The first type refers to context-free factorization bandits, which can update the latent vectors of users and items in real time using recommendations' feedback. Kawale et al. [19] used Thompson sampling for online matrix completion. Based on this sampling, Wang et al. [43] further learned the dependencies among items. However, context-free bandits are unable to use side information, thereby having difficulties in linking similar items or similar users when the feedback is sparse. The second type is contextual bandit, which introduces side information into bandit algorithms to pre-train items' latent vectors [21, 24, 39, 40, 49], to link the relations among items. Some contextual bandit models utilize the context of social relations to pre-define the relations among users [8, 40, 45]. However, in practice, obtaining all side information ahead of time is challenging. Moreover, the obtained side information may be inaccurate, which may result in the bias of the pre-trained items' latent vectors of the contextual bandit. Both types of factorization bandit learning models cannot perform the explanation task because little is known about how to derive a new confidence set for each of the latent vectors if we want to learn the latent vectors of the user, items, and the explanation interface jointly. Notably, McInerney et al. [26] attempted to devise an exploration–exploitation strategy in the ERSs. However, the proposed strategy was based on $\epsilon$-greedy exploration, which pre-defined a threshold for exploration without any theoretical guide to guarantee the performance in real scenarios.

In conclusion, no existing study can perform the online recommendation and online explanation tasks with good scalability and rigorous theoretical guarantee on the performance while addressing **Problem 1** and **Problem 2** simultaneously in real scenarios for ERSs with the online setting.

## 3. Methodology

In this section, we present the details of our proposed O$^3$ERS. First, we formulate the overall process of our tag-based ERS with the online setting and define some notations of O$^3$ERS. Second, we elaborate the process of O$^3$ERS, which includes online learning, online recommendation, and online explanation. Finally, we provide rigorous theoretical analysis and discuss the properties of our proposed model.

### 3.1. Notations and Problem Settings

We first describe the overall process of our tag-based ERS in real scenarios. At each round $t$, user $u_t$ first provides recommendation's feedback $r^{rec}_{u_t,a_t}$ for the item $a_t$ (i.e., whether the user has clicked the item) after receiving the explainable recommendations at the previous round. The user may also provide some explanations' feedback $\{r^{exp}_{u_t,a_1,v_1}, ..., r^{exp}_{u_t,a_n,v_n}\}$ as additional assessment of the previously recommended items (i.e., label some items with tags $\{v_{t_1}, ..., v_{t_n}\}$). According to the feedback, the explainable recommendation strategy of the ERS is updated to predict which item the user will be interested in and which tags explain the recommended item best in the future rounds when this user comes again.

To investigate the recommendations' feedback and explanations' feedback fully, our proposed O$^3$ERS considers two advanced reward assumptions, namely, recommendation reward and explanation reward. *Recommendation reward* predicts whether a user will click an item. It assumes that the expected reward of an item with respect to a given user is an inner product of their latent vectors. *Explanation reward* predicts whether a user will label an item with a tag. It assumes that the expected reward of a tag with respect to a given user–item pair is an inner product of the tag's latent vector and the concatenated latent vector of the user–item pair.

O$^3$ERS relates the two reward assumptions through the user and item latent vectors. Specifically, some parts of

the users' and items' latent vectors can be shared between two reward functions while other parts remain personalized and unshared in the recommendation task. Formally, both reward can be determined by Equation (1) and (2):

$$r_{u,a}^{rec} = \boldsymbol{m}_u^{*\top} \boldsymbol{n}_a^* + \boldsymbol{g}_u^{*\top} \boldsymbol{h}_a^* + \eta^{rec}, \tag{1}$$

$$r_{u,a,v}^{exp} = \left(\boldsymbol{m}_u^*, \boldsymbol{n}_a^*\right)^\top \left(\boldsymbol{s}_v^{u*}, \boldsymbol{s}_v^{a*}\right) + \eta^{exp}, \tag{2}$$

where $r_{u,a}^{rec}$ denotes whether user $u$ will click item $a$ (i.e., recommendation's feedback), $r_{u,a,v}^{exp}$ denotes whether user $u$ will label item $a$ with tag $v$ (i.e., explanation's feedback), $\eta^{rec}$ and $\eta^{exp}$ denote the Gaussian noise drawn from a Gaussian distribution $\mathcal{N}(0, \sigma)$ in observations at each round , and the letters with $^*$ denote the ground truth latent vectors. The main notations mentioned in this paper are summarized in Table 1.

**Table 1**
Descriptions of the main notations

| Notation | Description |
|---|---|
| $u, a, v$ | ID of each user, item, and tag |
| $u_t, a_t$ | Arrived user and the selected item at time $t$ |
| $\boldsymbol{m}_{u,t}$ | Estimated shared latent vector of user $u$ at time $t$ |
| $\boldsymbol{n}_{a,t}$ | Estimated shared item vector of item $a$ at time $t$ |
| $\boldsymbol{g}_{u,t}$ | Estimated personal latent vector of user $u$ at time $t$ |
| $\boldsymbol{h}_{a,t}$ | Estimated personal latent vector of item $a$ at time $t$ |
| $\boldsymbol{s}_{v,t} = \left(\boldsymbol{s}_{v,t}^u, \boldsymbol{s}_{v,t}^a\right)$ | Estimated latent vector of tag $v$ at time $t$, in which $\boldsymbol{s}_{v,t}^u$ and $\boldsymbol{s}_{v,t}^a$ are the first half and second half of $\boldsymbol{s}_{v,t}$ |
| $\boldsymbol{m}_u^*, \boldsymbol{n}_a^*, \boldsymbol{g}_u^*, \boldsymbol{h}_a^*, \boldsymbol{s}_v^*$ | Ground-truth of $\boldsymbol{m}_u, \boldsymbol{n}_a, \boldsymbol{g}_u, \boldsymbol{h}_a, \boldsymbol{s}_v$ |
| $\hat{r}_{u,a}^{rec}, r_{u,a}^{rec}$ | Estimated and observed recommendation feedback |
| $\hat{r}_{u,a,v}^{exp}, r_{u,a,v}^{exp}$ | Estimated and observed explanation feedback |
| $\mathcal{U}, \mathcal{I}, \mathcal{T}$ | All user set, item set, and tag set |
| $\mathcal{A}_t$ | Candidate item set at round $t$ |
| $\mathcal{T}_t$ | Tag set that user label at round $t$ |
| $\lambda_s,\ \lambda_m,\ \lambda_n,\ \lambda_g,\ \lambda_h$ | Regularization coefficients of $\boldsymbol{s}, \boldsymbol{m}, \boldsymbol{n}, \boldsymbol{g}, \boldsymbol{h}$ |

## 3.2. O$^3$ERS

The process of our proposed O$^3$ERS at each round can be divided into three phases, namely, **Phase 1:** Online Learning, **Phase 2:** Online Recommendation, and **Phase 3:** Online Explanation. A graphical representation of the O$^3$ERS model within one round is shown in Figure 2.

In Phase 1, in the current round, O$^3$ERS leverages the recommendation's feedback and explanation's feedback from the previous user to update the relevant latent vectors of the user, item, and tags in an online learning manner. Then, in Phase 2, O$^3$ERS receives a user who arrives at this round and utilizes a novel upper confidence bound algorithm to recommend an item to the user. Finally, in Phase 3, O$^3$ERS selects tags that match the user's preference and the item's features for an online explanation based on the latent vectors of the user, the recommended item, and all tags in this round. The three phases of O$^3$ERS are presented in Algorithm 1 with details explained in the following sections.

### *3.2.1. Phase 1: Online learning*

At each round $t$, O$^3$ERS first collects the recommendation's feedback and explanation's feedback from the previous user (i.e., user's evaluation on the previously recommended items after the user has interacted with the ERS) and leverages the feedback to update the relevant latent vectors of the user, item, and tags in an online learning manner based on the two reward assumptions in Equations (1) and (2). We appeal to the alternating least square algorithm built on ridge regression for parameter estimation because of the coupling among the latent vectors in the two reward assumptions [6]. Formally, let $\lambda_s$, $\lambda_m$, $\lambda_n$, $\lambda_g$, and $\lambda_h$ denote the regularization coefficients of vectors $\boldsymbol{s}$, $\boldsymbol{m}$, $\boldsymbol{n}$, $\boldsymbol{g}$, and

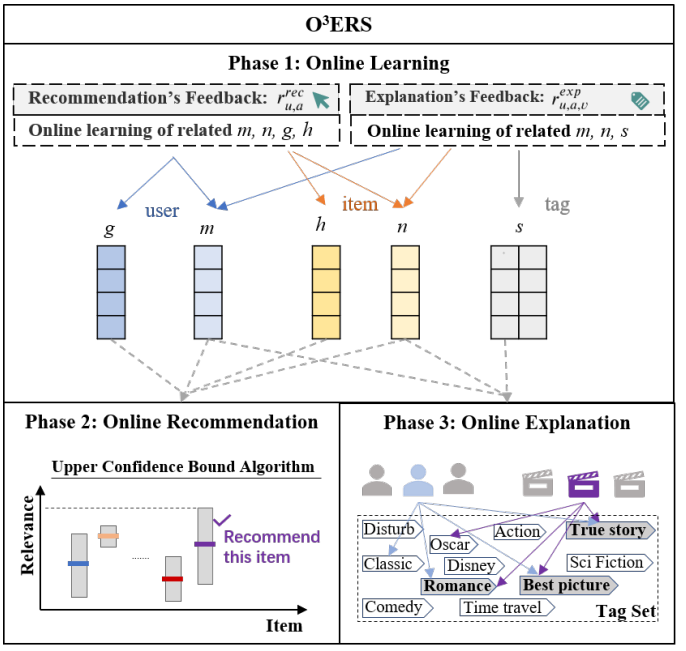



**Figure 2**: Graphic representation of the O³ERS model within one round. First (Phase 1), in the current round, O³ERS leverages the recommendation's feedback and explanation's feedback from the previous user to update the relevant latent vectors of the user, item, and tags in an online learning manner. Then (Phase 2), O³ERS receives a user who arrives at this round and utilizes a novel upper confidence bound algorithm to recommend an item to the user at this round. Finally (Phase 3), O³ERS selects tags that match both the user's preference and the item's features for an online explanation based on the latent vectors of the user, the recommended item, and all tags in this round.

$\boldsymbol{h}$ respectively. Then, the objective function of O³ERS is defined in Equation (3).

$$\begin{aligned}\min \frac{1}{2}\sum_{t=1}^{T}\left(\boldsymbol{m}_{u_t}^{\top}\boldsymbol{n}_{a_t}+\boldsymbol{g}_{u_t}^{\top}\boldsymbol{h}_{a_t}-r^{rec}_{u_t,a_t}\right)^2+\frac{1}{2}\sum_{t=1}^{T}\sum_{v\in\mathcal{T}_t}\left(\left(\boldsymbol{m}_{u_t}^{\top},\boldsymbol{n}_{a_t}^{\top}\right)\boldsymbol{s}_v-r^{exp}_{u_t,a_t,v}\right)^2+\frac{\lambda_s}{2}\|\boldsymbol{s}\|^2 \\ +\frac{\lambda_m}{2}\|\boldsymbol{m}\|^2+\frac{\lambda_n}{2}\|\boldsymbol{n}\|^2+\frac{\lambda_g}{2}\|\boldsymbol{g}\|^2+\frac{\lambda_h}{2}\|\boldsymbol{h}\|^2.\end{aligned} \tag{3}$$

We should note that the inclusion of the regularization terms is critical to our solution because it not only makes the subproblems in the coordinate descent based optimization well-posed with closed-form solution but also makes the $q$-linear convergence rate of parameter estimation achievable [36, 44].

By plugging Equation (1) and Equation (2) into Equation (3), the closed-form estimation of $\boldsymbol{m}$, $\boldsymbol{n}$, $\boldsymbol{g}$, $\boldsymbol{h}$ and $\boldsymbol{s}$ at round $t$ can be achieved by $\boldsymbol{m}_{u,t}=\mathbf{A}_{u,t}^{-1}\mathbf{a}_{u,t}$, $\boldsymbol{n}_{a,t}=\mathbf{B}_{a,t}^{-1}\mathbf{b}_{a,t}$, $\boldsymbol{g}_{u,t}=\mathbf{C}_{u,t}^{-1}\mathbf{c}_{u,t}$, $\boldsymbol{h}_{a,t}=\mathbf{D}_{a,t}^{-1}\mathbf{d}_{a,t}$, and $\boldsymbol{s}_{v,t}=\mathbf{E}_{v,t}^{-1}\mathbf{e}_{v,t}$, where $\mathbf{A}_{u,t},\mathbf{B}_{a,t},\mathbf{C}_{u,t},\mathbf{D}_{a,t},\mathbf{E}_{v,t}$ are auxiliary matrices and $\mathbf{a}_{u,t},\mathbf{b}_{a,t},\mathbf{c}_{u,t},\mathbf{d}_{a,t},\mathbf{e}_{v,t}$ are auxiliary vectors for parameters update. The detailed updating formula of the corresponding parameters can be found lines 6-22 in Algorithm 1.

### *3.2.2. Phase 2: Online Recommendation*

After learning the relevant latent vectors at this round, O³ERS performs online recommendation for the user who arrived at this round. The estimated latent vectors $\boldsymbol{m}_t$, $\boldsymbol{n}_t$, $\boldsymbol{g}_t$, and $\boldsymbol{h}_t$ represent the model's current best knowledge about the users' preference and the items' features at time $t$ and therefore are used for exploitation purposes. In other words, at time $t$, a pure-exploitation strategy without exploration is to select an item $a_t=\arg\max_{a\in\mathcal{A}_t}(\boldsymbol{m}_{u_t,t}^{\top}\boldsymbol{n}_{a,t}+\boldsymbol{g}_{u_t,t}^{\top}\boldsymbol{h}_{a,t})$ to the arrived user $u_t$ for recommendation. However, exploiting the currently trained model fully without any exploration may unfortunately reinforce bias in a currently inaccurate model and easily get stuck in the sub-optimal items because the observed online recommendations' feedback is partial and insufficient. As an alternative, properly exploring some currently less promising items for model correction becomes necessary for long-term optimality.

In O$^3$ERS, to balance the exploitation and exploration, we derive a novel algorithm based on the upper confidence bound framework, which estimates the confidence set of each latent vector and utilize these confidence sets to guide the recommendation task. Specifically, our algorithm follows three steps in each round:

1. The high probability confidence sets for the estimated users' and items' latent vectors are constructed.
2. The upper confidence bound of the estimated reward of each user–item pair is calculated.
3. The item with the largest upper confidence bound on the estimated reward is selected.

We now elaborate the details in each of the three steps.

In the first step, the high probability confidence sets for the estimated users' latent vectors and items' latent vectors, namely, $\boldsymbol{m}_{u,t}, \boldsymbol{n}_{a,t}, \boldsymbol{g}_{u,t}$ and $\boldsymbol{h}_{a,t}$, are $\left\|\boldsymbol{m}_u^* - \boldsymbol{m}_{u,t}\right\|_{\mathbf{A}_{u,t}} \leq \alpha_t^{m_u}$, $\left\|\boldsymbol{n}_a^* - \boldsymbol{n}_{a,t}\right\|_{\mathbf{B}_{a,t}} \leq \alpha_t^{n_a}$, $\left\|\boldsymbol{g}_u^* - \boldsymbol{g}_{u,t}\right\|_{\mathbf{C}_{u,t}} \leq \alpha_t^{g_u}$, and $\left\|\boldsymbol{h}_a^* - \boldsymbol{h}_{a,t}\right\|_{\mathbf{D}_{a,t}} \leq \alpha_t^{h_a}$ accordingly, where $\alpha_t^{m_u}, \alpha_t^{n_a}, \alpha_t^{g_u}$, and $\alpha_t^{h_a}$ can be computed by Lemma 1. In this Lemma, the $Q-linear\ convergence$ holds for every $\epsilon > 0$ and every $q$ can be explicitly estimated with the corresponding $\epsilon$ as described in [36]. The proof sketch of Lemma 1 can be found in the appendix.

**Lemma 1.** ***(Upper Confidence Bound of Latent Vectors)*** *Let $\boldsymbol{n}_1, \boldsymbol{n}_2, ..., \boldsymbol{n}_t \in \mathbb{R}^d$ be the shared latent vectors of the items and $\boldsymbol{s}_1^u, \boldsymbol{s}_2^u, ..., \boldsymbol{s}_t^u \in \mathbb{R}^d$ be the first half of the latent vectors of the tags that user $u$ has interacted with until time $t$. To avoid clutter, let $\boldsymbol{N}_{u,t} = \boldsymbol{n}_{1:t}$ and $\boldsymbol{S}_{u,t}^u = \boldsymbol{s}_{1:t}^u$. Similarly, let $\boldsymbol{m}_1, \boldsymbol{m}_2, ..., \boldsymbol{m}_t \in \mathbb{R}^d$ be the shared latent vectors of the users and $\boldsymbol{s}_1^a, \boldsymbol{s}_2^a, ..., \boldsymbol{s}_t^a \in \mathbb{R}^d$ be the second half of the latent vectors of the tags that item $a$ has interacted with until time $t$, then $\boldsymbol{M}_{a,t} = \boldsymbol{m}_{1:t}$ and $\boldsymbol{S}_{a,t}^a = \boldsymbol{s}_{1:t}^a$.*

*Suppose that $\left\|\boldsymbol{m}_u^*\right\|_2 \leq L_m$, $\left\|\boldsymbol{n}_a^*\right\|_2 \leq L_n$, $\left\|\boldsymbol{g}_u^*\right\|_2 \leq L_g$, $\left\|\boldsymbol{h}_a^*\right\|_2 \leq L_h$, $\left\|\boldsymbol{s}_v^*\right\|_2 \leq L_s$. Then similar to the proof in [39], at time $t$, $\left\|\boldsymbol{m}_u^* - \boldsymbol{m}_{u,t}\right\|_2$, $\left\|\boldsymbol{n}_a^* - \boldsymbol{n}_{a,t}\right\|_2$, $\left\|\boldsymbol{g}_u^* - \boldsymbol{g}_{u,t}\right\|_2$, $\left\|\boldsymbol{h}_a^* - \boldsymbol{h}_{a,t}\right\|_2$, and $\left\|\boldsymbol{s}_v^* - \boldsymbol{s}_{v,t}\right\|_2$, is upper bounded by $Q_m, Q_n, Q_g, Q_h, Q_s$, respectively, where $Q_m = \frac{(q_m+\epsilon_m)\left(1-(q_m+\epsilon_m)^t\right)}{1-(q_m+\epsilon_m)}$ and $Q_n, Q_g, Q_h, Q_s$ is similarly defined with different $q \in (0,1)$ and $\epsilon > 0$ (Q-linear convergence).*

*Similar to most studies [1, 24, 40], we assume that $\eta_{rec}$ and $\eta_{exp}$ are conditionally $1/\sqrt{2}-$sub-Gaussian. Then at time $t$, for $\delta \in (0,1)$, with probability $1-\delta$, $\left\|\boldsymbol{m}_u^* - \boldsymbol{m}_{u,t}\right\|_{\mathbf{A}_{u,t}} \leq \alpha_t^{m_u}$, $\left\|\boldsymbol{n}_a^* - \boldsymbol{n}_{a,t}\right\|_{\mathbf{B}_{a,t}} \leq \alpha_t^{n_a}$, $\left\|\boldsymbol{g}_u^* - \boldsymbol{g}_{u,t}\right\|_{\mathbf{C}_{u,t}} \leq \alpha_t^{g_u}$, and $\left\|\boldsymbol{h}_a^* - \boldsymbol{h}_{a,t}\right\|_{\mathbf{D}_{a,t}} \leq \alpha_t^{h_a}$, where*

$$\alpha_t^{m_u} = \sqrt{\ln \frac{\det\left(\mathbf{A}_{u,t}\right)^{1/2}}{\det\left(\mathbf{S}_{u,t}^u \mathbf{S}_{u,t}^{u\top} + \lambda_m \mathbf{I}\right)\delta}} + \frac{2L_m L_n^2}{\sqrt{\lambda_m}} Q_n + \sqrt{\ln \frac{\det\left(\mathbf{A}_{u,t}\right)^{1/2}}{\det\left(\mathbf{N}_{u,t}\mathbf{N}_{u,t}^{\top} + \lambda_m \mathbf{I}\right)^{1/2}\delta}} + \sqrt{\lambda_m} L_m + \frac{2L_m L_s^2}{\sqrt{\lambda_m}} Q_s, \tag{4}$$

$$\alpha_t^{n_a} = \sqrt{\ln \frac{\det\left(\mathbf{B}_{a,t}\right)^{1/2}}{\det\left(\mathbf{S}_{v,t}^a \mathbf{S}_{v,t}^{a\top} + \lambda_n \mathbf{I}\right)^{1/2}\delta}} + \frac{2L_n L_m^2}{\sqrt{\lambda_n}} Q_m + \sqrt{\ln \frac{\det\left(\mathbf{B}_{a,t}\right)^{1/2}}{\det\left(\mathbf{M}_{a,t}\mathbf{M}_{a,t}^{\top} + \lambda_n \mathbf{I}\right)^{1/2}\delta}} + \sqrt{\lambda_n} L_n + \frac{2L_m L_s^2}{\sqrt{\lambda_n}} Q_s, \tag{5}$$

$$\alpha_t^{g_u} = \frac{2L_g L_h^2}{\sqrt{\lambda_g}} Q_h + \sqrt{d \ln \frac{1 + tL_h/\lambda_g}{\delta}} + \sqrt{\lambda_g} L_g, \tag{6}$$

$$\alpha_t^{h_a} = \frac{2L_h L_g^2}{\sqrt{\lambda_h}} Q_g + \sqrt{d \ln \frac{1 + tL_g/\lambda_h}{\delta}} + \sqrt{\lambda_h} L_h. \tag{7}$$

In the second step, let $\hat{r}_{u,a}^{rec} = \boldsymbol{m}_{u,t}^{\top} \boldsymbol{n}_{a,t} + \boldsymbol{g}_{u,t}^{\top} \boldsymbol{h}_{a,t}$ be the estimated reward that user $u$ may give item $a$ at time $t$. Then, plugging in the recommendation reward assumption defined in Equation (1), the confidence set of the estimated reward

can be derived as

$$\begin{aligned}
\|r^{*rec}_{u,a} - \hat{r}^{rec}_{u,a}\|_2 =& \|\boldsymbol{m}^{*\top}_u \boldsymbol{n}^*_a - \boldsymbol{m}^\top_{u,t} \boldsymbol{n}_{a,t} + \boldsymbol{p}^{*\top}_u \boldsymbol{q}^*_a - \boldsymbol{p}^\top_{u,t} \boldsymbol{q}_{a,t}\|_2 \\
\leq& \left\|\boldsymbol{m}^*_u - \boldsymbol{m}_{u,0}\right\|_2 \left\|\boldsymbol{n}^*_a - \boldsymbol{n}_{a,0}\right\|_2 \left(q_m + \epsilon_m\right)^t \left(q_n + \epsilon_n\right)^t \\
&+ \left\|\boldsymbol{p}^{*\top}_u - \boldsymbol{p}^\top_{u,t}\right\|_2 \left\|\boldsymbol{q}^*_a - \boldsymbol{q}_{a,t}\right\|_2 \left(q_p + \epsilon_p\right)^t \left(q_q + \epsilon_q\right)^t \\
&+ \left\|\boldsymbol{m}_{u,t}\right\|_{\mathbf{B}^{-1}_{a,t}} \left\|\boldsymbol{n}^*_a - \boldsymbol{n}_{a,t}\right\|_{\mathbf{B}_{a,t}} + \left\|\boldsymbol{n}_{a_t}\right\|_{\mathbf{A}^{-1}_{u,t}} \left\|\boldsymbol{m}^*_u - \boldsymbol{m}_{u,t}\right\|_{\mathbf{A}_{u,t}} \\
&+ \left\|\boldsymbol{p}^\top_{u,t}\right\|_{\mathbf{D}^{-1}_{a,t}} \left\|\boldsymbol{q}^*_a - \boldsymbol{q}_{a,t}\right\|_{\mathbf{D}_{a,t}} + \left\|\boldsymbol{q}_{a,t}\right\|_{\mathbf{C}^{-1}_{u,t}} \left\|\boldsymbol{p}^{*\top}_u - \boldsymbol{p}^\top_{u,t}\right\|_{\mathbf{C}_{u,t}} \\
\leq& 2L_m L_n \left(q_m + \epsilon_m\right)^t \left(q_n + \epsilon_n\right)^t + 2L_p L_q \left(q_p + \epsilon_p\right)^t \left(q_q + \epsilon_q\right)^t \\
&+ \alpha_t^{n_a} \left\|\boldsymbol{m}_{u,t}\right\|_{\mathbf{B}^{-1}_{a,t}} + \alpha_t^{m_u} \left\|\boldsymbol{n}_{a,t}\right\|_{\mathbf{A}^{-1}_{u,t}} + \alpha_t^{q_a} \left\|\boldsymbol{p}_{u,t}\right\|_{\mathbf{D}^{-1}_{a,t}} + \alpha_t^{p_u} \left\|\boldsymbol{q}_{a,t}\right\|_{\mathbf{C}^{-1}_{u,t}} \\
=& CB(u,a),
\end{aligned} \tag{8}$$

where the first inequality can be derived using the Cauchy-Schwartz inequality [1] and the last inequality can be derived using Lemma 1. At round $t$, given a user $u$ and an item $a$ in the candidate set, O$^3$ERS calculate the upper confidence bound for the estimated reward of this user-item pair as:

$$UCB(u,a) = \boldsymbol{m}^\top_{u,t} \boldsymbol{n}_{a,t} + \boldsymbol{g}^\top_{u,t} \boldsymbol{h}_{a,t} + CB(u,a), \tag{9}$$

where the first two terms represent the estimation of the recommendation reward and the last term represents the uncertainty of this estimation for exploration. In other words, the first two terms reflect the tendency of exploiting the current estimations, and the last term reflects the tendency of exploring currently less promising but uncertain items.

In the third step, O$^3$ERS selects an item with the largest upper confidence bound for recommendation.

### *3.2.3. Phase 3: Online Explanation*

O$^3$ERS provides explanations via a tag cloud interface based on the estimated latent vectors of users, items, and tags. A user cares about different aspects of various items, and the explanations should consider the user's preference and item's features at the same time to guarantee effectiveness, persuasiveness, and satisfaction. Therefore, a tag cloud is attached to the selected item $a_t$, which represents the most related features of the item that the user $u_t$ cares about. Specifically, at round $t$, we first predict the relevance degree of each tag by

$$\hat{r}^{exp}_{u,a,v} = \left(\boldsymbol{m}_{u,t}, \boldsymbol{n}_{a,t}\right)^\top \boldsymbol{s}_{v,t}. \tag{10}$$

Subsequently, a tag cloud is constructed, in which the size of each tag is weighted by its predicted relevance degree. Such tag clouds are constructed automatically, which do not depend on other pre-defined rules. In the explanation task, the users' true assessment of the recommended item is not limited to the tags we present for explanations and users can make any labels or comments to reflect the user's true preference and the item's true features. Therefore, in the explanation task, a pure-exploitation strategy is sufficient for tag selection.

### *3.2.4. Overall Process of O$^3$ERS*

Putting the aforementioned process together, the pseudocode for O$^3$ERS is provided in Algorithm 1. O$^3$ERS starts with the MAIN procedure. It first initializes the auxiliary matrices, auxiliary vectors, and latent vectors of all users, items, and tags with dimension $d$, identity matrix $\mathbf{I}$, zeros vector $\mathbf{0}$, and uniform distribution $U$ (Line 2). At each round $t$, O$^3$ERS first collects the feedback from the previous user $u_{t-1}$, which is the user's evaluation on the recommended items after the user has interacted with the ERS (Lines 4-5). Based on both the recommendation feedback and explanation feedback, O$^3$ERS updates all relevant latent vectors (Lines 6-22). Then, O$^3$ERS receives the user $u_t$ who arrives at this round and recommends an item to this user based on the upper confidence bound algorithm (Lines 23-24). Finally, To explain the recommended item, O$^3$ERS constructs the tag clouds as described in Section 3.2.3 (Line 25).

**Algorithm 1** O$^3$ERS

**procedure** MAIN($d$) ▷ Main entry
  INITIALIZATION($d$)
  **for** $t = 1, 2, ..., T$ **do**
    Observe the recommendation's feedback $r^{rec}_{u_{t-1},a_{t-1}}$. ▷ Collect feedback from the previous user
    Observe the explanation's feedback $r^{exp}_{u_{t-1},a,v}$ for some item $a$ with tag $v$ **for** $v \in \mathcal{T}_{t-1}$
    **if** $u_{t-1}$ labeled some item $a$ with tag $v$ **for** $v \in \mathcal{T}_{t-1}$ **then** ▷ Phase 1: Online learning
      $\mathbf{E}_{v,t} = \mathbf{E}_{v,t-1} + \left(\boldsymbol{m}_{u_{t-1},t-1}, \boldsymbol{n}_{a,t-1}\right)\left(\boldsymbol{m}_{u_{t-1},t-1}, \boldsymbol{n}_{a,t-1}\right)^\top$.
      $\mathbf{e}_{v,t} = \mathbf{e}_{v,t-1} + r^{exp}_{u_{t-1},a,v}\left(\boldsymbol{m}_{u_{t-1},t-1}, \boldsymbol{n}_{a,t-1}\right)$.
      $\boldsymbol{s}_{v,t} = \mathbf{E}^{-1}_{v,t}\mathbf{e}_{v,t}$.
    **end if**
    $\mathbf{A}_{u_t,t} = \mathbf{A}_{u_{t-1},t-1} + \boldsymbol{n}_{a_{t-1},t-1}\boldsymbol{n}^\top_{a_{t-1},t-1} + \sum_{v\in\mathcal{T}_{t-1}} \boldsymbol{s}^u_{v,t-1}\boldsymbol{s}^{u\top}_{v,t-1}$.
    $\mathbf{a}_{u_{t-1},t} = \mathbf{a}_{u_{t-1},t-1} + \left(r^{rec}_{u_{t-1},a_{t-1}} - \boldsymbol{g}^\top_{u_{t-1},t-1}\boldsymbol{h}_{a_{t-1},t-1}\right)\boldsymbol{n}_{a_{t-1},t-1} + \sum_{v\in\mathcal{T}_{t-1}}\left(r^{exp}_{u_{t-1},a_{t-1},v} - \boldsymbol{n}^\top_{a_{t-1},t-1}\boldsymbol{s}^a_{v,t-1}\right)\boldsymbol{s}^{u\top}_{v,t-1}$.
    $\boldsymbol{m}_{u_{t-1},t} = \mathbf{A}^{-1}_{u_{t-1},t}\mathbf{a}_{u_{t-1},t}$.
    $\mathbf{B}_{a_{t-1},t} = \mathbf{B}_{a_{t-1},t-1} + \boldsymbol{m}_{u_{t-1},t-1}\boldsymbol{m}^\top_{u_{t-1},t-1} + \sum_{v\in\mathcal{T}_{t-1}} \boldsymbol{s}^a_{v,t-1}\boldsymbol{s}^{a\top}_{v,t-1}$.
    $\mathbf{b}_{a_{t-1},t} = \mathbf{b}_{a_{t-1},t-1} + \left(r^{rec}_{u_{t-1},a_{t-1}} - \boldsymbol{g}^\top_{u_{t-1},t-1}\boldsymbol{h}_{a_{t-1},t-1}\right)\boldsymbol{m}^\top_{u_{t-1},t-1} + \sum_{v\in\mathcal{T}_{t-1}}\left(r^{tag}_{u_{t-1},a_{t-1},v} - \boldsymbol{m}^\top_{u_{t-1},t-1}\boldsymbol{s}^u_{v,t-1}\right)\boldsymbol{s}^{a\top}_{v,t-1}$.
    $\boldsymbol{n}_{a_{t-1},t} = \mathbf{B}^{-1}_{a_{t-1},t}\mathbf{b}_{a_{t-1},t}$.
    $\mathbf{C}_{u_{t-1},t} = \mathbf{C}_{u_{t-1},t-1} + \boldsymbol{h}_{a_{t-1},t-1}\boldsymbol{h}^\top_{a_{t-1},t-1}$.
    $\mathbf{c}_{u_{t-1},t} = \mathbf{c}_{u_{t-1},t-1} + \left(r^{rec}_{u_{t-1},a_{t-1}} - \boldsymbol{m}^\top_{u_{t-1},t-1}\boldsymbol{n}_{a_{t-1},t-1}\right)\boldsymbol{h}^\top_{a_{t-1},t-1}$.
    $\boldsymbol{g}_{u_{t-1},t} = \mathbf{C}^{-1}_{u_{t-1},t}\mathbf{c}_{u_{t-1},t}$.
    $\mathbf{D}_{a_{t-1},t} = \mathbf{D}_{a_{t-1},t-1} + \boldsymbol{g}_{u_{t-1},t-1}\boldsymbol{g}^\top_{u_{t-1},t-1}$.
    $\mathbf{d}_{a_{t-1},t} = \mathbf{d}_{a_{t-1},t-1} + \left(r^{rec}_{u_{t-1},a_{t-1}} - \boldsymbol{m}^\top_{u_{t-1},t-1}\boldsymbol{n}_{a_{t-1},t-1}\right)\boldsymbol{g}^\top_{u_{t-1},t-1}$.
    $\boldsymbol{h}_{a_{t-1},t} = \mathbf{D}^{-1}_{a_{t-1},t}\mathbf{d}_{a_{t-1},t}$.
    Receive the arrived user $u_t$ and the current candidate item set $\mathcal{A}_t$ ▷ Phase 2: Online recommendation
    Select item by $a_t = \arg\max_{a\in\mathcal{A}_t} UCB(u_t, a)$ and recommend it to user $u_t$
    Explain by a tag cloud constructed in Section 3.2.3 ▷ Phase 3: Online explanation
  **end for**
**end procedure**
**procedure** INITIALIZATION($d$) ▷ Initialize all latent vectors
  **for** $u \in \mathcal{U}$ **do** $\mathbf{A}_u = \lambda_m\mathbf{I}, \mathbf{a}_u = \mathbf{0}_d, \boldsymbol{m}_{u,0} = U(0,1,d)$.
  **for** $a \in \mathcal{I}$ **do** $\mathbf{B}_a = \lambda_n\mathbf{I}, \mathbf{b}_a = \mathbf{0}_d, \boldsymbol{n}_{a,0} = U(0,1,d)$.
  **for** $u \in \mathcal{U}$ **do** $\mathbf{C}_u = \lambda_g\mathbf{I}, \mathbf{c}_u = \mathbf{0}_d, \boldsymbol{g}_{u,0} = U(0,1,d)$.
  **for** $a \in \mathcal{I}$ **do** $\mathbf{D}_a = \lambda_h\mathbf{I}, \mathbf{d}_a = \mathbf{0}_d, \boldsymbol{h}_{a,0} = U(0,1,d)$.
  **for** $v \in \mathcal{T}$ **do** $\mathbf{E}_v = \lambda_s\mathbf{I}, \mathbf{e}_v = \mathbf{0}_{2d}, \boldsymbol{s}_{v,0} = U(0,1,2d)$.
**end procedure**

## 3.3. Regret Analysis of O$^3$ERS

We now analyze the long-term performance of the O$^3$ERS, which aims to reduce its cumulative regret after $T$ rounds as:

$$R(T) = \sum_{t=1}^{T} r^{rec}_{u_t,a^*_t} - r^{rec}_{u_t,a_t}. \tag{11}$$

$R(T)$ quantifies a policy's effectiveness by the difference between the expected reward of the optimal item $a^*_t$ and that of the selected item $a_t$ [7]. Based on Lemma 1 and the item recommendation strategy in Section 3.2.2, we can derive Lemmas 2 and 3, which can be further used to derive the upper bound of O$^3$ERS's cumulative regret in Theorem 1.

**Lemma 2.** *(**Upper Bound of the True Reward**) Let $a_t$ be the selected item at time t for user u and $a_t^*$ be the best item in hindsight at time t. According to the item recommendation strategy, if item $a_t$ is chosen at time t, $UCB(u, a_t)$ is the highest among all items for a given user u at time t, which means: $\boldsymbol{m}_{u,t}^\top \boldsymbol{n}_{a_t,t} + \boldsymbol{p}_{u,t}^\top \boldsymbol{q}_{a_t,t} + CB(u, a_t) \geq \boldsymbol{m}_{u,t}^\top \boldsymbol{n}_{a_t^*,t} + \boldsymbol{p}_{u,t}^\top \boldsymbol{q}_{a_t^*,t} + CB(u, a_t^*)$. With simple derivations, we have:*

$$\boldsymbol{m}_u^{*\top} \boldsymbol{n}_{a_t^*}^* + \boldsymbol{p}_u^{*\top} \boldsymbol{q}_{a_t^*}^* \leq \boldsymbol{m}_{u,t}^\top \boldsymbol{n}_{a_t,t} + \boldsymbol{p}_{u,t}^\top \boldsymbol{q}_{a_t,t} + \mathcal{B}_{a_t,t} + \mathcal{F}_t, \tag{12}$$

*where* $\mathcal{B}_{a,t} = \alpha_t^{p_u} \left\|\boldsymbol{q}_{a,t}\right\|_{\mathbf{C}_{u,t}^{-1}} + \alpha_t^{q_a} \left\|\boldsymbol{p}_{u,t}\right\|_{\mathbf{D}_{a,t}^{-1}} + \alpha_t^{n_a} \left\|\boldsymbol{m}_{u,t}\right\|_{\mathbf{B}_{a,t}^{-1}} + \alpha_t^{m_u} \left\|\boldsymbol{n}_{a,t}\right\|_{\mathbf{A}_{u,t}^{-1}}$,

$\mathcal{F}_t = \alpha_t^{n_{a_t^*}} \left\|\boldsymbol{m}_u^* - \boldsymbol{m}_{u,t}\right\|_{\mathbf{B}_{a_t^*,t}^{-1}} + \alpha_t^{q_{a_t^*}} \left\|\boldsymbol{p}_u^* - \boldsymbol{p}_{u,t}\right\|_{\mathbf{C}_{a_t^*,t}^{-1}}$.

**Lemma 3.** *(**Regret at Time** t) According to the inequality in Lemma 2, the expected regret $\mathcal{R}_t$ at time t is*

$$\begin{aligned}
\mathcal{R}_t =& r_{u_t,a_t^*} - r_{u_t,a_t} = \boldsymbol{m}_{u_t}^{*\top} \boldsymbol{n}_{a_t^*}^* + \boldsymbol{p}_{u_t}^{*\top} \boldsymbol{q}_{a_t^*}^* - \boldsymbol{m}_{u_t}^{*\top} \boldsymbol{n}_{a_t}^* - \boldsymbol{p}_{u_t}^{*\top} \boldsymbol{q}_{a_t}^* \\
\leq& \boldsymbol{m}_{u_t,t}^\top \boldsymbol{n}_{a_t,t} + \boldsymbol{p}_{u_t,t}^\top \boldsymbol{q}_{a_t,t} + \mathcal{B}_{a_t,t} + \mathcal{F}_t - \boldsymbol{m}_{u_t}^{*\top} \boldsymbol{n}_{a_t}^* - \boldsymbol{p}_{u_t}^{*\top} \boldsymbol{q}_{a_t}^* \\
\leq& 2\alpha_t^{n_{a_t}} \left\|\boldsymbol{m}_{u_t,t}\right\|_{\mathbf{B}_{a_t,t}^{-1}} + 2\alpha_t^{q_{a_t}} \left\|\boldsymbol{p}_{u_t,t}\right\|_{\mathbf{D}_{a_t,t}^{-1}} + 2\alpha_t^{m_{u_t}} \left\|\boldsymbol{n}_{a_t,t}\right\|_{\mathbf{A}_{u_t,t}^{-1}} + 2\alpha_t^{p_{u_t}} \left\|\boldsymbol{q}_{a_t,t}\right\|_{\mathbf{C}_{u_t,t}^{-1}} \\
&+ \alpha_t^{n_{a_t}} \left\|\boldsymbol{m}_{u_t}^* - \boldsymbol{m}_{u_t,t}\right\|_{\mathbf{B}_{a_t,t}^{-1}} + \alpha_t^{q_{a_t}} \left\|\boldsymbol{p}_{u_t}^* - \boldsymbol{p}_{u_t,t}\right\|_{\mathbf{D}_{a_t,t}^{-1}} + \alpha_t^{n_{a_t^*}} \left\|\boldsymbol{m}_{u_t}^* - \boldsymbol{m}_{u_t,t}\right\|_{\mathbf{B}_{a_t^*,t}^{-1}} + \alpha_t^{q_{a_t^*}} \left\|\boldsymbol{p}_{u_t}^* - \boldsymbol{p}_{u_t,t}\right\|_{\mathbf{D}_{a_t^*,t}^{-1}}
\end{aligned} \tag{13}$$

**Theorem 1.** *(**Cumulative Regret until Time** T) Under the same notations and assumptions as in Lemma 1, with probability $1-\delta$, O$^3$ERS's cumulative regret satisfies*

$$\begin{aligned}
R(T) = \sum_{t=1}^{T} \mathcal{R}_t \leq& \alpha_T^n \frac{2L_m}{\sqrt{\lambda_n}} Q_m + \alpha_T^h \frac{2L_g}{\sqrt{\lambda_g}} Q_g + 2\alpha_T^g \sqrt{2dT \ln\left(1 + \frac{TL_h}{\lambda_g d}\right)} + 2\alpha_T^h \sqrt{2dT \ln\left(1 + \frac{TL_g}{\lambda_h d}\right)} \\
&+ 2\alpha_T^m \sqrt{2T \ln \sum_{u \in \mathcal{U}} \frac{\det\left(\mathbf{A}_{u,T}\right)^{1/2}}{\det\left(\mathbf{S}_{u,T}^u \mathbf{S}_{u,T}^{u\top} + \lambda_m \mathbf{I}\right) \delta}} + 2\alpha_T^n \sqrt{2T \ln \sum_{a \in \mathcal{I}} \frac{\det\left(\mathbf{B}_{a,T}\right)^{1/2}}{\det\left(\mathbf{S}_{a,T}^a \mathbf{S}_{a,T}^{a\top} + \lambda_n \mathbf{I}\right)^{1/2} \delta}}
\end{aligned} \tag{14}$$

*where $\alpha_T^m$, $\alpha_T^n$, $\alpha_T^g$, and $\alpha_T^h$ are the upper bound of all $\alpha^{m_u}$, $\alpha^{n_a}$, $\alpha^{g_u}$, and $\alpha^{h_a}$ respectively over all $t \in 1, ..., T$ and $\delta$ is also encoded in $\alpha_T^m$, $\alpha_T^n$, $\alpha_T^g$, and $\alpha_T^h$ as shown in Lemma 1.*

The detailed proofs of Lemma 2, Lemma 3, and Theorem 1 are included in the appendix. From those lemmas and the theorem, we can make two conclusions which point out the importance of exploration of the recommendation task and explanation feedback in the explanation task.

First, thanks to our online recommendation strategy, O$^3$ERS achieves a sub-linear regret upper bound, which demonstrates that the average number of sub-optimal recommendations made in O$^3$ERS over time vanishes with high probability. Specifically, the exploration can bound the reward of the best item by the estimated reward of the selected item (Lemma 2), thereby bounding the difference between the true reward and the estimated reward at every round $t$ (Lemma 3). Then, because the confidence interval is shrinking via exploration (Lemma 1), a sub-linear regret of order $\mathcal{O}(\sqrt{T}\log T)$ is achieved after $T$ rounds of interactions, which is a sub-linear regret upper bound with the same scaling as the state-of-the-art online learning recommendation models [1, 40, 45]. In other words, a sub-linear regret is achieved with the proper balance between exploitation and exploration in the online recommendation task; otherwise without proper exploration, such as in the conventional offline training models for ERSs, a linear regret is inevitable.

Second, thanks to our online explanation task, O$^3$ERS achieves a reduced regret upper bound with an improved convergence rate, which demonstrates that it requires less explorations to make the optimal recommendations. Specifically, when the explanations' feedback is provided, the largest eigenvalue of $\mathbf{S}_{u,T}^u \mathbf{S}_{u,T}^{u\top}$ or $\mathbf{S}_{a,T}^a \mathbf{S}_{a,T}^{a\top}$ is nonzero, which accelerates the shrinkage of the confidence set in Lemma 3, thereby reducing the cumulative regret in Theorem 1. In other words, a regret reduction is achieved because of the O$^3$ERS's superiority to utilize the explanations' feedback.

In conclusion, both the recommendation task and the explanation task contribute to a sub-linear and reduced regret upper bound on the long-term performance of O$^3$ERS.

### 3.4. Discussions

In this section, we discuss two advantages of our proposed O$^3$ERS.

One advantage of our proposed O$^3$ERS is that it provides better scalability and theoretical guarantee on its online performance compared with existing models for ERSs. On the one hand, O$^3$ERS has better scalability compared with existing models for ERSs. O$^3$ERS does not need to recalculate from scratch in every update even though the feedback in the ERSs is continuously streaming in because the updated model is stored in the auxiliary matrices and vectors at every round, whereas existing models for ERSs need to recalculate from scratch when dealing with new training data. On the other hand, O$^3$ERS has a smaller regret upper bound to guarantee the performance compared with existing models for ERSs because of its novel strategy based on both the recommendations' feedback and explanations' feedback, as proved rigorously in Section 3.3.

Another advantage of our proposed O$^3$ERS is that it can address two challenging problems, as mentioned in Section 1, because of its superiority to unify the recommendation task and the explanation task into its reward assumptions (i.e., Equations (1) and (2)). On the one hand, because of its two related reward assumptions, O$^3$ERS transfers users' latent vectors and items' latent vectors learned from the recommendation task to the explanation task instantly at each round, thereby making up for the sparsity and delay of online explanations' feedback in **Problem 1** for better explanations. On the other hand, with the online learning algorithm based on this two related reward assumptions, O$^3$ERS is able to include online explanations' feedback into a novel exploitation–exploration strategy, which can address the partial and insufficient online recommendations' feedback in **Problem 2** for better recommendations.

## 4. Experiments

In this section, we conduct comprehensive experiments on three datasets and compared O$^3$ERS with several models to evaluate their performance.

### 4.1. Experiment Setup

#### *4.1.1. Dataset Descriptions*

Given that our proposed framework exploits users' preference and items' features based on the users' recommendations' feedback (i.e., click or not) and explanations' feedback (i.e., tagging behaviors) in an online setting, we need some datasets that contain both kinds of feedback with corresponding time stamps. For this purpose, we devise one simulated dataset to model the users' behaviors in a completely online environment and select two real-world offline datasets that are often used to validate recommendations models in the online setting [8, 40]. Notably, a simulated dataset is often required to validate the theoretical regret analysis of a recommendation model in existing studies because ground truth cannot be obtained from real-world offline datasets [19, 39, 40]. The two real-world datasets are extracted from two websites (i.e., Movielens and Delicious). We provide a brief description of these three datasets in Table 2 and present the details in subsequent paragraphs.

**Table 2**
Datasets descriptions

| Dataset | # user | # item | # tags | # recommendation feedback | # explanation feedback |
|---|---|---|---|---|---|
| Simulation | 100 | 1,000 | 3,000 | 500,000 | 500,000 |
| Movielens | 2,113 | 10,197 | 9,079 | 376,105 | 47,957 |
| Delicious | 1,861 | 69,226 | 40,897 | 437,593 | 437,593 |

**Simulated dataset:** Because the simulated dataset is used to validate the theoretical regret analysis, its generation needs to be consistent with the reward assumptions in Equations.(1) and (2). Specially, we generate two $d$-dimensional shared vectors, i.e., $\boldsymbol{m}$ or $\boldsymbol{n}$, and two $d$-dimensional personal vectors, i.e., $\boldsymbol{g}$ or $\boldsymbol{h}$, for each of the $M$ users and $N$ items. Then we generates a $2d$-dimensional latent vector (i.e., $\boldsymbol{s}$) for each of the $S$ tags. Each of the dimensions of all vectors is drawn from a Gaussian distribution $\mathcal{N}(0, \sigma)$, where $\sigma \sim U(0, 1)$. We normalize all simulated vectors such that each of their norm is upper bounded by one. These vectors are treated as the ground truth for reward generation and are unknown to all algorithms. We simulate an online environment as follows. In each round $t \in T$, the same candidate item set containing randomly selected $K$ items is presented to all the models; and the Gaussian noises in Equation (1) and Equation (2) are sampled once for all those items. After each model selects an item for the recommendation,

a reward for that item is generated by Equation (1) and the best explanation tag for that item with the corresponding reward is generated by Equation (2). Both rewards are used to update the model. We fix $d$ to 50, $M$ to 100, $N$ to 1000, $S$ to 3000, $K$ to 25, and $T$ to 500,000.

**Movielens**[3]**:** This dataset is extracted from a movie rating website (i.e., Movielens) with users' rating scores ranging from 1 to 5 and users' labeling behavior toward movies. We follow the same settings in [43] to pre-process this dataset, which provides an unbiased offline evaluation for the online recommendation task via this dataset. Specifically, we gather all the rated movies of a user to construct the candidate item set for this user. Then, we generate the recommendation's feedback of each item in the candidate item set as follows: if a user's rating toward an item is higher than 3, then the feedback is 1; otherwise the feedback is 0. At each round $t$, a user arrives, and the same candidate item set of this user is presented to all the models to replay each user's visit. After each model selects an item for the recommendation, the recommendation's feedback and the explanation's feedback (i.e., tagging behavior) of the user are observed to update the model.

**Delicious** [3]**:** This dataset is extracted from a bookmarking web service (i.e., Delicious) with users' labeling behavior toward URLs. We follow the same settings in [8, 40] to pre-process this dataset, which is based on the same offline evaluation theory for the online recommendation task that is proven in [20]. Specifically, we generate the recommendations' feedback for each user toward each URL as follows: the feedback is 1 if the user bookmarked a particular URL, otherwise, it is 0. Then, at each round $t$, we fix the size of the candidate item set to be 25 and generate the candidate item set for a particular user as follows: we select one URL among the nonzero-reward URLs, and randomly choose the 24 other URLs among the zero-reward URLs. As a result, only one item for the arrived user is relevant in each candidate item set. After each model selects an item, the recommendation's feedback and the explanation's feedback of the user are observed to update the model.

Additionally, some of our comparison models need the context of each item, which we will introduce in Section 4.1.2. Therefore, we pre-process the Movielens and Delicious datasets to obtain the context following the settings in [8, 39, 40]. Specifically, we use all tags associated with a single item to create its TF-IDF feature vector. Subsequently, we used PCA to reduce the feature dimension. In both datasets, we only consider the first 25 principal components to construct the context vectors, that is, the dimension of the observed context is 25.

### *4.1.2. Comparison models*

Given that our model is the first online learning model for ERSs with the online setting, we can only compare it with two categories of existing models (i.e., online learning recommendation models without explanations and offline learning recommendation models with explanations). The details of each category are described as follows.

**Online Learning Recommendation Models without Explanations**: As discussed in Section 2.2, factorization bandit-based algorithms are more suitable for the real scenarios of the ERSs. Thus, we select online learning recommendation models that are based on factorization bandit as comparison models.

- **LinUCB** [21] is a conventional contextual bandit baseline for online learning recommendation. It selects an item based on an upper confidence bound of the estimated reward given the pre-processed items' context vectors.
- **hLinUCB** [39] is a state-of-the-art contextual bandit model that assumes that the hidden context contributes to the reward apart from the observed context. It models the hidden context by matrix factorization.
- **FactorUCB** [40] is a is a state-of-the-art contextual bandit model that uses social relations to further learn the users' latent vector and items' latent vectors as well as their upper confidence bound.
- **PTS** [19] is a conventional context-free bandit baseline that performs matrix factorization for recommendation. It approximates the posterior of latent vectors by updating a set of particles.
- **ICTR** [43] is a state-of-the-art context-free bandit model based on PTS, which further utilizes the topic modeling algorithms to learn the dependencies among items in the bandit setting.

Notably, among the above factorization bandit models, the contextual bandit models have a different reward assumption from our proposed models, whereas the context-free bandit models have the same reward function as ours.

**Offline Learning Recommendation Models with Explanations**: As discussed in Section 2.1, latent factor models are more effective for ERSs. Thus we select explainable recommendation models that are based on matrix factorization and those that can perform tag-based explanations as our comparison models. These models are not online learning models and are not adapted for the online setting. Thus, we set a batch update criterion for these models as follows: the

[3] https://grouplens.org/datasets/hetrec-2011/

dataset for experiments are equally split into five smaller datasets chronologically, and each of these models performs batch update based on all observed feedback every time the dataset is switched.

- **EFM** [48] is a conventional matrix factorization baseline for tag-based explainable recommendation, which considers user-feature attention and item-feature quality to learn the latent vectors of users, items, and tags.
- **EXPLORE** [50] is a state-of-the-art matrix factorization model that utilizes a topic modeling algorithm to learn the semantic meaning of tags and recommended items attached with tags for explanations.
- **Bart** [26] is a state-of-the-art model that pre-defined a threshold for exploration given the context without any theoretical guide to guarantee the performance. Notably, this model is retrained in batch modes periodically in practice, as pointed out in Section 2.1.2

Among the aforementioned explainable recommendation models, Bart has a different reward assumption from our proposed models, whereas EFM and EXPLORE have the same reward function as ours.

#### *4.1.3. Evaluation Metrics*

In general, we adopt two widely used metrics, namely, Cumulative Regret and Cumulative Reward, to evaluate the performance of the online recommendation task [8, 38, 39] and three metrics, namely, Cumulative Recall and Cumulative Precision, and Cumulative NDCG, to evaluate the performance of the online explanation task in the ERSs [5, 27, 42, 46, 50]. We provide a detailed description of each metric as follows.

**Cumulative Regret:** This metric is defined in Equation(11) and is widely used in the online recommendation task to evaluate the performance of a model in the simulated environment.

**Cumulative Reward:** This metric is widely used in online recommendations tasks to evaluate the performance of a model on the offline real-world dataset [21, 40, 45]. Based on the reward setting for each real-world dataset in Section 4.1.1, we defined cumulative reward as the accumulation of the reward until each round and define *relative* cumulative reward as the cumulative reward of each model divided by a random strategy's cumulative reward. The term 'Cumulative Reward' refers to 'relative cumulative reward' for convenience.

**Cumulative Precision:** This metric evaluates the effectiveness of explanations from items' perspectives, that is, it evaluates whether the predicted tag cloud can describe the item correctly. First, we define $Precision@10$ in Equation (15), where $T^{10}$ and $T_{t_{a_t}}$ denote the top 10 tags in the predicted tag cloud and the true tag set for item $a_t$ respectively. Then, we define cumulative precision as the cumulation of $Precision@10$ until every round and define *relative* cumulative precision as the the cumulative precision of each model divided by a random strategy's cumulative precision. The term 'Cumulative Precision' refers to 'relative cumulative precision' for convenience.

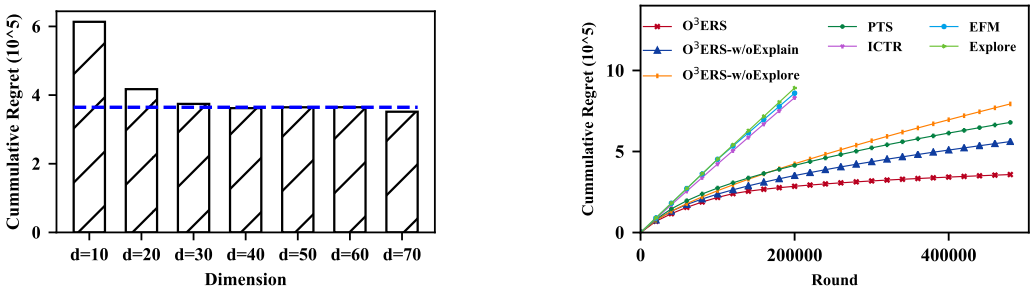
$$Precision@10 = |T^{10} \cap T_{t_{a_t}}|/|T^{10}|. \tag{15}$$

**Cumulative Recall:** This metric is widely used to evaluate the persuasiveness of explanations from users' perspectives, namely, it indicates whether the predicted tags best fit the user's needs [27, 46]. First, we define $Recall@10$ in Equation (16), where $T^{10}$ and $T_{t_{u_t}}$ denote the top 10 tags in the predicted tag cloud and true tags set for user $u_t$ respectively. Then, we define cumulative recall as the cumulation of $Recall@10$ until every round and define *relative* cumulative recall as the the cumulative recall of each model divided by a random strategy's cumulative recall. The term 'Cumulative Recall' refers to 'relative cumulative recall' for convenience.

$$Recall@10 = |T^{10} \cap T_{t_{u_t}}|/|T_{t_{u_t}}|. \tag{16}$$

**Cumulative NDCG:** This metric is widely used to evaluate whether the explanations can improve users' satisfaction degree of the the ERSs, namely, it indicates whether the predicted tags correctly fit the user's needs and the items' features in an integrated view [5, 42]. First, we define $NDCG@10$ in Equation (17) where $T^{10}$ and $T_{t_{u_t a_t}}$ denote the top 10 tags in the predicted tag cloud and the intersection of the true tag sets for user $u_t$ and item $a_t$ respectively. Then, we define cumulative NDCG as the cumulation of $NDCG@10$ until every round and define *relative* cumulative NDCG as the the cumulative NDCG of each model divided by a random strategy's cumulative NDCG. The term 'Cumulative NDCG' refers to 'relative cumulative NDCG' for convenience.

$$DCG@T^{10} = \frac{|T^{10} \cap T_{t_{u_t a_t}}|}{\sum_{i=1}^{10} log_2(i+1)}, \qquad NDCG@10 = \frac{DCG@T^{10}}{IDCG@T_{t_{u_t a_t}}}. \tag{17}$$




(a) Cumulated Regret with different dimension settings

(b) Cumulated Regret on simulated dataset

**Figure 3**: Results on dimension sensitivity and performance on simulated dataset. (a) Result under different dimensions. (b) Cumulative Regret of all comparison models which have the same recommendation reward assumption.

## 4.2. Evaluation of Recommendation Task

In this section, we aim to test if O$^3$ERS can guarantee the long-term performance of the recommendation task with the partial and insufficient recommendation feedback. Extensive experiments are conducted to answer three questions:

- **Q1:** How is O$^3$ERS's recommendation performance affected by the dimension of latent vectors?
- **Q2:** How does O$^3$ERS's recommendation task perform empirically compared with the theoretical analysis?
- **Q3:** How does O$^3$ERS's recommendation task perform compared with the state-of-the-art models in terms of the Cumulative Reward?

We first conduct experiments on the simulated dataset to answer **Q1** and **Q2**. Then we test O$^3$ERS's performance on two real-world datasets to answer **Q3**.

### *4.2.1. Evaluations of Dimension Sensitivity (for Q1)*

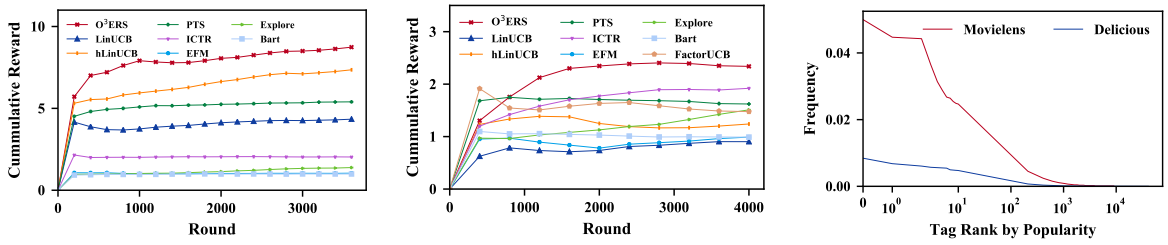
In our proposed O$^3$ERS model, the only parameter we need to tune is the dimension $d$ of all the latent vectors. To be consistent with the canonical factorization bandit models [21, 40, 41], we set the regularization hyperparameters of our proposed model to be one. In this experiment, the dimension of latent vectors $d$ used in O$^3$ERS varies from 10 to 70. In Figure 3 (a), each column of the bar plot shows the Cumulative Regret of O$^3$ERS with varying $d$. The blue line presents the result when $d$ is set to be 50, which is the ground truth dimension.

**Results & Analysis:** When the dimension we set is lower than the ground truth dimension, the Cumulative Regret of O$^3$ERS increases. Interestingly, when the dimension is higher, the regret is even slightly better. This reason is that when $d$ is smaller than the ground truth dimension, the dimension ignored by O$^3$ERS will produce a linear regret. Conversely, when the dimension is higher than the ground truth dimension, learning the extra dimension is equivalent to learning the residual after learning the first $d$ dimensions. To sum up, setting $d$ to be reasonably large makes O$^3$ERS insensitive to the ground truth settings. As a result, we manually set the latent dimension $d$ to 50 in the following experiments.

### *4.2.2. Evaluations of Cumulative Regret (for Q2)*

Now we test O$^3$ERS's cumulative regret on the simulated dataset. We did not compare the Cumulative Regret with the contextual models including LinUCB, hLinUCB, FactorUCB, and Bart because our simulated dataset does not have the context setting. Moreover, comparing the Cumulative Regret with models under different reward assumptions is meaningless. Additionally, to investigate the properties of O$^3$ERS, we introduce two simplified versions of our proposed O$^3$ERS as two additional comparison models: (1) O$^3$ERS without using explanation feedback, named **O$^3$ERS-w/oExplain**, and (2) O$^3$ERS without exploration, named **O$^3$ERS-w/oExplore**. The Cumulative Regrets of different models are shown in Figure 3 (b). Note that we did not plot the complete results of ICTR and the offline learning models over all rounds to increase visibility because they incur relatively large linear regret.

**Results & Analysis:** In Figure 3 (b), we can make three conclusions. First, O$^3$ERS achieves a sub-linear cumulative regret, which is consistent with the theoretical analysis in Section 3.3. Second, O$^3$ERS outperforms the two simplified versions of our proposed model ( i.e., **O$^3$ERS-w/oExplain** and **O$^3$ERS-w/oExplore**). **O$^3$ERS-w/oExplain**




(a) Cumulative Precision (b) Cumulative Recall (c) Distributions of explanation feedback

**Figure 4**: Evaluation on recommendation task on two real-world datasets. (a) Cumulative Reward on Movielens. (b) Cumulative Reward on Delicious. (c) Tags distributions on two real-world datasets.

performs worse without leveraging explanation feedback compared with O$^3$ERS, and **O$^3$ERS-w/oExplore** incurs a linear cumulative regret without proper exploration. Therefore, this result demonstrates the importance of incorporating explanation feedback and exploration in O$^3$ERS. Third, O$^3$ERS achieves the lowest cumulative regret among the the comparison models. The reason is that O$^3$ERS has a better exploitation–exploration strategy. More specifically, O$^3$ERS outperforms the factorization bandit model PTS because the latter does not have a closed-form solution and requires resorting to approximate sampling methods, thereby potentially introducing additional errors to the matrix factorization. ICTR performs even worse because it uses latent topic modeling to learn the relations among items, which does not have a theoretical guarantee on the performance. Although EFM and Explore have closed-form solutions compared with PTS and ICTR, they are offline learning models without any exploration. Thus, they can only perform batch updates based on stored data and will incur a linear regret when deployed in the online environment.

#### *4.2.3. Evaluation on Cumulative Reward (for Q3)*

We compare O$^3$ERS with all the comparison models introduced in 4.1.2 on two real-world datasets. For all comparison models, we use the parameter settings recommended in their relevant studies. FactorUCB cannot be run on Movielens because this dataset does not have the context of social relationship. The Cumulative Reward on both datasets are plotted in Figures 4(a) and 4(b).

**Results & Analysis:** O$^3$ERS outperforms all models by an average of 278.10%, which ranges from 18.46% to 784.67%. It obtains the highest Cumulative Reward in both datasets consistently because it utilizes the recommendations' feedback and the explanations' feedback robustly, that is, it updates the latent vectors for users and items in an online manner. Contextual models with pre-trained vectors (i.e., LinUCB, hLinUCB, FactorUCB, and Bart) obtain a higher average Cumulative Reward on Movielens but lower ones on Delicious compared with context-free models (i.e., PTS, ICTR, EFM, and Explore). More accurately, the average relative Cumulative Reward of contextual models is 3.18 on Movielens and 1.14 on Delicious, whereas those of context-free models is 1.27 on Movielens and 1.32 on Delicious. This finding indicates that contextual models are not robust and depend on whether the pre-trained vectors can learn the users' preference and items' features correctly. On the contrary, updating latent vectors to learn both users' preference and items' features from users' feedback continuously is a more robust choice.

In addition, O$^3$ERS gets higher improvement against the comparison models on Movielens (by average of 480.27 %) than on Delicious (by average of 101.20%) because O$^3$ERS can recognize the quality of the explanation feedback better in a long-tailed distribution. In Figure 4(c), we calculate the frequency of each explanation feedback in both datasets and rank them by their popularity in a descending order to visualize the different distributions of both datasets. The popularity of the explanation feedback on these two datasets differ significantly: there are a lot more popular explanation feedback on Movielens than those on Delicious. Similar to the analysis in [45], the highly skewed distribution of explanation feedback in Delicious increases the difficulty for O$^3$ERS to recognize the quality of the explanation feedback because identifying the difference among explanation feedback when more explanation feedback appears is difficult. By contrast, learning the explanation feedback is easier on Movielens. This characteristic promotes the learning of users' preference and items' features indirectly, thereby facilitating the recommendation task on Movielens further.

**Table 3**
Performance on Movielens and Delicious under different sparsity.

| Metric | Sparsity | Movielens | | | | | Delicious | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | EFM | Bart | Explore | O$^3$ERS | Improve | EFM | Bart | Explore | O$^3$ERS | Improve |
| Cumulative Precision | 10% | 1.78 | 2.88 | 2.73 | 2.98 | 26.68% | 1.21 | 1.23 | 1.58 | 1.82 | 37.86% |
| | 30% | 1.59 | 2.59 | 2.57 | 2.72 | 27.31% | 1.09 | 1.21 | 1.44 | 1.72 | 39.80% |
| | 50% | 1.44 | 2.21 | 2.19 | 2.46 | 31.49% | 1.06 | 1.17 | 1.33 | 1.65 | 40.25% |
| | 70% | 1.31 | 2.12 | 1.94 | 2.29 | 33.62% | 0.98 | 1.14 | 1.31 | 1.59 | 41.03% |
| | 90% | 1.3 | 1.89 | 1.76 | 2.18 | 35.63% | 0.98 | 1.01 | 1.31 | 1.56 | 44.24% |
| Cumulative Recall | 10% | 1.13 | 1.76 | 1.92 | 5.48 | 260.58% | 1.33 | 1.16 | 1.42 | 1.38 | 6.64% |
| | 30% | 1.07 | 1.84 | 1.71 | 5.36 | 268.56% | 1.36 | 1.24 | 1.71 | 1.52 | 7.74% |
| | 50% | 1.03 | 1.76 | 1.77 | 5.26 | 268.91% | 1.38 | 1.26 | 1.77 | 1.69 | 17.36% |
| | 70% | 1.01 | 1.28 | 1.56 | 4.61 | 270.70% | 1.4 | 1.33 | 1.56 | 1.81 | 27.13% |
| | 90% | 0.88 | 0.97 | 1.43 | 3.92 | 274.57% | 1.41 | 1.38 | 1.39 | 1.93 | 38.53% |
| Cumulative NDCG | 10% | 4.12 | 4.82 | 3.99 | 17.55 | 309.98% | 2.03 | 2.02 | 2.56 | 4.06 | 86.53% |
| | 30% | 4.12 | 3.77 | 3.44 | 16.97 | 351.78% | 2.07 | 2.03 | 2.61 | 4.12 | 86.61% |
| | 50% | 2.72 | 2.89 | 2.15 | 14.53 | 470.92% | 2.08 | 2.02 | 2.52 | 4.35 | 99.03% |
| | 70% | 1.77 | 2.87 | 2.01 | 13.12 | 517.04% | 2.13 | 2.08 | 2.01 | 4.81 | 132.13% |
| | 90% | 1.69 | 1.65 | 1.67 | 12.07 | 622.82% | 2.19 | 2.17 | 2.16 | 5.17 | 137.89% |

## 4.3. Evaluation on Explanation Task

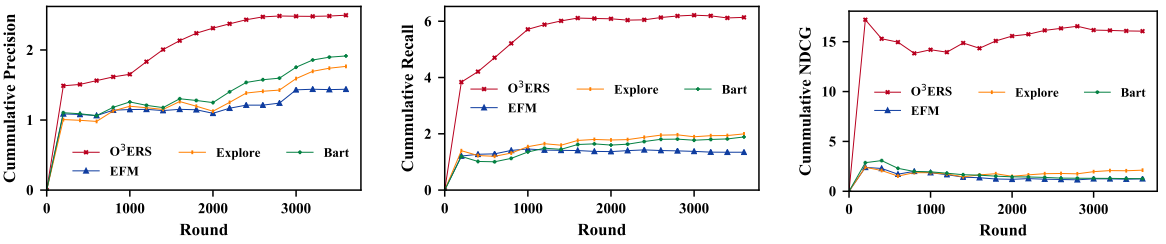
We only compare O$^3$ERS's explanation performance with those offline learning explainable recommendation models introduced in Section 4.1.2 because the existing online learning recommendation models cannot perform the explanation task. Extensive experiments are conducted to answer the following questions:

- **Q4:** How does O$^3$ERS perform compared with the state-of-the-art models in terms of effectiveness, persuasiveness, and satisfaction?
- **Q5:** How is O$^3$ERS's ability to deal with the sparse and delayed online explainables' feedback under different data sparsity levels?
- **Q6:** How does O$^3$ERS work to provide explanations in an ERS via a tag-cloud interface?

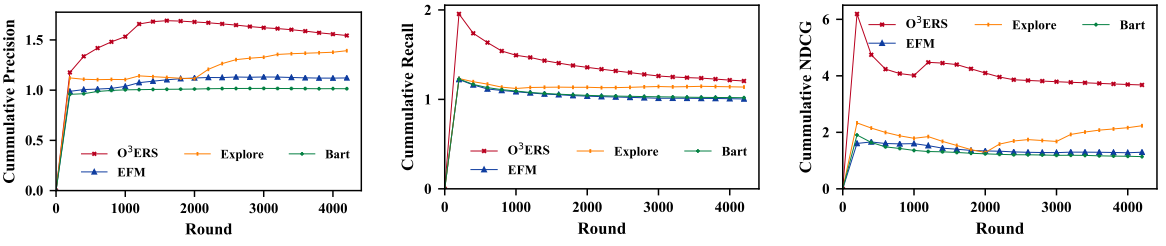
### *4.3.1. Evaluation on Effectiveness, Persuasiveness, and Satisfaction (for Q4)*

This experiment aims to measure whether the explanations provided by O$^3$ERS improves the effectiveness, persuasiveness, and satisfaction of users in real online scenarios of the ERSs, which indicate whether the explanations can help users determine the actual features of items correctly, attract users' attention, and match both users' interest and items' features accurately, respectively [12]. However, because of the interactive nature of this task, it would seem that the performance of the explanation task needs to be evaluated on a real-world ERS, which we lack. As an alternative, we turn to Cumulative Precision, Cumulative Recall, and Cumulative NDCG for evaluation, as described in Section 4.1.3. The results for both datasets in terms of different evaluation metrics for the explanation task are shown in Figure 5.

**Results & Analysis:** On both datasets, O$^3$ERS achieves the most promising performance on all three metrics and provides effective, persuasive, and satisfactory explanation because it can utilize the recommendation feedback to make up for the sparse and delayed explanations' feedback. Precisely, the average improvement of O$^3$ERS against the comparison models on Cumulative Precision, Cumulative Recall, and Cumulative NDCG are 61.52% (ranging from 49.59% to 72.97%), 273.83% (ranging from 236.80% to 310.08%), and 859.28% (ranging from 745.09% to 973.38%) on Movielens and 41.47% (ranging from 27.03% to 54.80%), 28.67% (ranging from 22.40% to 31.45%), and 179.63% (ranging from 125.51% to 217.75%) on Delicious. Therefore, the improvement of O$^3$ERS on Movielens is higher than that on Delicious. Having a deep observation of the characteristics of the two datasets, we found that the explanation feedback sparsity is 99.83% on Movielens and 99.66% on Delicious. The sparse and delayed feedback degenerates the performance of all explainable recommendation models. However, the performance of O$^3$ERS degenerate less because it can transfer high-quality knowledge learned from the recommendation feedback to the explanation task. Moreover, the difference in degeneration becomes more apparent when the dataset become sparser.




(a) Cumulative Precision on Movielens (b) Cumulative Recall on Movielens (c) Cumulative NDCG on Movielens

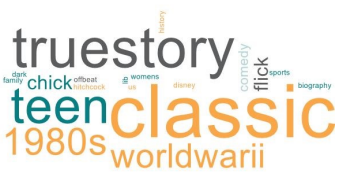



(d) Cumulative Precision on Delicious (e) Cumulative Recall on Delicious (f) Cumulative NDCG on Delicious

**Figure 5**: Evaluation on the explanation task on two real-world datasets. (a), (b), and (c) are results on Movielens. (c), (d), and (e) are results on Delicious.

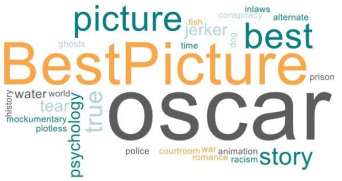
### *4.3.2. Evaluation of Model's Performance under Different Data Sparsity (for Q5)*

In this experiment, we further evaluate the performance of the explainable recommendation models under different sparsity. For this purpose, data are separated chronologically into two groups. The first group contains 80% of the data and is used to train all explainable recommendation models, which is called the sparsity training set. The second group contains 20% of the data and is reserved as our sparsity testing set. Then, we filter the sparsity training set into five sub-datasets randomly based on the different degrees of sparsity. For example, a sparsity of 10% suggests that 10% of




(a) True tag cloud for User 1


(b) Predicted tag cloud for User 1




(c) True tag cloud for User 2


(d) Predicted tag cloud for User 2

**Figure 6**: Tag clouds for two users with different preferences when recommending the movie 'Kill Bill'. (a) Accumulative tags that User 1 has labeled. (b) Tag cloud predicted by O$^3$ERS for User 1. (c) Accumulative tags that User 2 has labeled. (d) Tag cloud predicted by O$^3$ERS for User 2.

the data is filtered out in the sparsity training set. The result is shown in Table 3, in which the last column represents O$^3$ERS's average improvement over the remaining explainable recommendation models.

**Results & Analysis:** In Table 3, O$^3$ERS outperforms the comparison models under different sparsity and the average improvement of O$^3$ERS against the comparison models increases when the sparsity training set becomes sparser. This result further demonstrates the superiority of the structure of O$^3$ERS, that is, it transfers the promising knowledge of users and items learned from the recommendation task to the explanation task. When the dataset becomes sparser, EFM, Explore, and Bart collect sparser recommendations' and explanations' feedback. In other words, a large amount of zero-reward feedback is contained in their training datasets, which increases the difficulty to perform matrix factorization in their models, thereby affecting the performance of their explanation task negatively. By contrast, O$^3$ERS can collect denser recommendation feedback based on its effective exploration–exploitation strategy. Therefore, a larger amount of positive-reward recommendation feedback promotes the learning of users' preference and items' features in O$^3$ERS's recommendation task, which are transferred to the explanation task subsequently to make up for the sparse and delayed feedback, thereby facilitating the performance of O$^3$ERS's explanation task.

#### *4.3.3. Case Study (for Q6)*

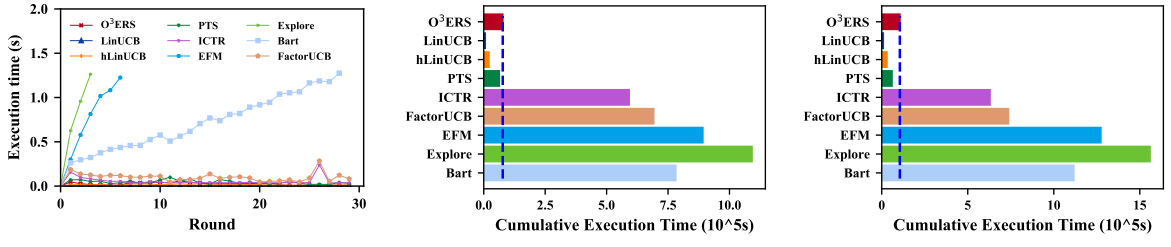
To have a more intuitive understanding of the O$^3$ERS's mechanism to provide explanations via a tag-cloud interface, we conduct two case studies on Movielens. We randomly select two different users who have been recommended the well-known movie 'Titanic' during the online recommendation task and visualize the corresponding tag-cloud explanations when 'Titanic' is recommended to each of the users. 'Titanic' is a romantic movie that based on a historical event of the sinking of the Titanic. It won the awards for Best Picture and Best Director in Oscar. Figure 6 depicts two true tag clouds and two predicted tag clouds for User 1 and User 2.

From Figure 6 (a), we can identify that User 1 loves classic movies, and true stories and our tag-cloud for explanation of this movie in 6 (b) can attract User 1 because it emphasizes these two features. As for User 2, the tag cloud in 6 (c) suggests that the user enjoys Oscar movie, and the predicted tag cloud in 6 (d) is persuasive and effective to User 2. In summary, the tag cloud-based explanations of O$^3$ERS are effective, persuasive, and satisfactory. Such explanations are intuitive, automatic, and personalized.

### 4.4. Evaluation on Execution Time

In this section, we conduct two experiments on two real-world datasets (i.e., Movielens and Delicious) to evaluate O$^3$ERS's performance from the perspective of efficiency. First, we compare the execution time of O$^3$ERS with that of the comparison models for ERSs (i.e., EFM, Explore, and Bart), which are the offline learning models to perform both the recommendation and explanation tasks. Then, to further evaluate the efficiency of O$^3$ERS, we compare its execution time with that of the online learning recommendation models (i.e., LinUCB, hLinUCB, PTS, ICTR, and FactorUCB), which perform the recommendation task only, rather than both the recommendation and explanation tasks. The results are plotted in Figure 7. In Figure 7(a), we update all the models at each round and plot their execution time. However, the execution time of the offline learning models increases dramatically over time, thereby making them impractical to be updated at each round. Thus, each of the offline learning models performs only five updates over all rounds using the batch update criterion in Section 4.1.2, and their cumulative execution time are plotted in Figures 7(b) and 7(c).

**Results & Analysis:** The results in Figure 7 indicate that O$^3$ERS is more scalable and efficient than the state-of-the-




(a) Execution time at each round (b) Cumulative execution time on Movielens (c) Cumulative execution time on Delicious

**Figure 7**: Execution time of different models. (a) Execution time at each round. (b) Cumulative execution time on Movielens. (c) Cumulative execution time on Delicious.

art models for ERSs. First, in Figure 7(a), the execution time of O$^3$ERS at each round is the shortest and does not increase over time compared with that of EFM, Explore, and Bart. The reason is that O$^3$ERS only needs to slightly adjust its parameters based on the instant feedback, whereas the offline learning recommendation models for ERSs are updated on the basis of recalculation from scratch when feedback accumulates. This result indicates that O$^3$ERS is more scalable than the state-of-the-art models for ERS. Second, in Figures 7(b) and 7(c), the cumulative execution time of O$^3$ERS is the shortest compared with that of EFM, Explore, and Bart. On average, the cumulative execution time of O$^3$ERS is only 8.1% (ranging from 6.74% to 9.90%) of that of EFM, Explore, and Bart, which indicates that O$^3$ERS is more efficient than the state-of-the-art models for ERSs.

Moreover, the results in Figures 7(b) and 7(c) show that the cumulative execution time of O$^3$ERS is comparable to that of the state-of-the-art online learning models, which perform the recommendation task only, rather than both the recommendation and explanation tasks. On the one hand, O$^3$ERS is more efficient than ICTR and FactorUCB. The reason is that ICTR and FactorUCB adopt some time-consuming techniques (i.e., item clustering technique and social influence modeling technique) in addition to the factorization technique to learn extra latent vectors, whereas O$^3$ERS adopts the efficient factorization technique only. On the other hand, O$^3$ERS is less efficient than LinUCB, hLinUCB, and PTS. The reason is that O$^3$ERS needs to learn the latent vectors of tags additionally, whereas LinUCB, hLinUCB, and PTS focus on the latent vectors of users and items only. O$^3$ERS, however, outperforms LinUCB, hLinUCB, and PTS in the recommendation task thanks to its superiority to learn the latent vectors of tags. To sum up, O$^3$ERS provides comparable efficiency to the online learning models, which perform the recommendation task without the explanation task.

In conclusion, our proposed O$^3$ERS is not only more scalable and efficient than the state-of-the-art models for ERSs that perform both the recommendation and explanation tasks but also provides comparable efficiency to the online learning models that perform the recommendation task without the explanation task.

## 5. Conclusion and Future Work

In this paper, we have proposed a novel online setting for ERSs and devised an effective model called O$^3$ERS in this setting, which can perform online learning using instant feedback at each round to provide online recommendations attached with online explanations. One of the main novelties of O$^3$ERS is that it has better scalability and better theoretical guarantee on performance compared with existing models for ERSs. Another main novelty of O$^3$ERS is its ability to address the two challenging problems of ERSs with the online settings. Specifically, targeting the challenging problem caused by the sparsity and delay of online explanations' feedback, O$^3$ERS transfers the knowledge learned from the recommendations' feedback instantly to the explanation task for better explanations. Targeting the challenging problem caused by the partialness and insufficiency of online recommendations' feedback, O$^3$ERS utilizes a novel exploitation–exploration strategy, which incorporates the explanations' feedback in the recommendation task to provide better recommendations.

In addition, we have provided theoretical and empirical studies to validate the superiority of O$^3$ERS. Theoretically, we have proven that O$^3$ERS has better long-term performance with a sub-linear cumulative regret upper bound than existing models for ERSs because of its superiority to unify the recommendation and the explanation tasks in an online learning manner. Empirically, we have conducted extensive experiments on one simulated and two real-world datasets, which show that O$^3$ERS not only outperforms the state-of-the-art recommendation models remarkably in terms of cumulative regret and cumulative reward but also provides effective, persuasive, and satisfactory explanations. Moreover, O$^3$ERS has better efficiency in terms of execution time than the state-of-the-art models for ERSs.

In the future, we intend to extend our work in three potential directions. First, due to the limitation that our model is not specifically designed for the cold-start problem, it performs similarly to a random strategy when historical feedback about users or items are unavailable. Therefore, our first direction is to consider more side information, such as social networks or context, into our future work to address the cold-start problem. Second, for the convenience of theoretical development, the reward assumptions among users, items, and tags are assumed to be linear in our model. In the future work, we plan to extend our model to exploit more reward assumptions, which may be closer to reality. Finally, given the fact that more kinds of explanation feedback in addition to tags are available in the real world, devising a mechanism to convert other kinds of explanations' feedback into tags is interesting, such that more explanations' feedback can be incorporated into our model and more forms of explanation can be displayed to users.

## 6. Acknowledgement

This work was supported in part by the National Key R&D Program of China (No. 2018YFB1403001), the National Natural Science Foundation of China (No. U1509221), and the Zhejiang Provincial Key R&D Program, China (No. 2017C03044).

## References

[1] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, pages 2312–2320, 2011.

[2] Jacob Abernethy, Kevin Canini, John Langford, and Alex Simma. Online collaborative filtering. *University of California at Berkeley, Tech. Rep*, 2007.

[3] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3:397–422, 2002.

[4] Marko Balabanovic and Yoav Shoham. Content-based, collaborative recommendation. *Communications of theACM*, 40(3):66–72, 1997.

[5] Krisztian Balog, Filip Radlinski, and Shushan Arakelyan. Transparent, scrutable and explainable user models for personalized recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 265–274. ACM, 2019.

[6] Dale Borowiak. Linear models, least squares and alternatives. *Technometrics*, 43(1):99, 2001.

[7] Giuseppe Burtini, Jason L. Loeppky, and Ramon Lawrence. A survey of online experiment design with the stochastic multi-armed bandit. *CoRR*, abs/1510.00757, 2015.

[8] Nicolò Cesa-Bianchi, Claudio Gentile, and Giovanni Zappella. A gang of bandits. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, page 737–745, 2013.

[9] Shuo Chang, F. Maxwell Harper, and Loren Gilbert Terveen. Crowd-based personalized natural language explanations for recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 175–182. ACM, 2016.

[10] Sergio Cleger-Tamayo, Juan M. Fernández-Luna, and Juan F. Huete. Learning from explanations in recommender systems. *information sciences*, 287:90–108, 2014.

[11] Henriette S. M. Cramer, Vanessa Evers, Satyan Ramlal, Maarten van Someren, Lloyd Rutledge, Natalia Stash, Lora Aroyo, and Bob J. Wielinga. The effects of transparency on trust in and acceptance of a content-based art recommender. *User Modeling and User-Adapted Interaction*, 18(5):455–496, 2008.

[12] Julie Daher, Armelle Brun, and Anne Boyer. A review on explanations in recommender systems. Technical report, Université de Lorraine, 2017. hal-01836639f.

[13] Aminu Da'u, Naomie Salim, Idris Rabiu, and Akram Osman. Recommendation system exploiting aspect-based opinion mining with deep learning method. *Information Sciences*, 512:1279–1292, 2020.

[14] Jorge Díez, Pablo Pérez-Núñez, Oscar Luaces, Beatriz Remeseiro, and Antonio Bahamonde. Towards explainable personalized recommendations by learning from users' photos. *Information Sciences*, 520:416–430, 2020.

[15] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval,*, pages 549–558, 2016.

[16] Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *Proceeding on the ACM Conference on Computer Supported Cooperative Work*, pages 241–250, 2000.

[17] Steven C. H. Hoi, Doyen Sahoo, Jing Lu, and Peilin Zhao. Online learning: A comprehensive survey. *CoRR*, abs/1802.02871, 2018.

[18] Yunfeng Hou, Ning Yang, Yi Wu, and Philip S. Yu. Explainable recommendation with fusion of aspect information. *World Wide Web*, 22(1):221–240, 2019.

[19] Jaya Kawale, Hung Hai Bui, Branislav Kveton, Long Tran-Thanh, and Sanjay Chawla. Efficient thompson sampling for online matrix-factorization recommendation. In *International Conference on Neural Information Processing Systems*, pages 1297–1305, 2015.

[20] Lihong Li, Wei Chu, John Langford, Taesup Moon, and Xuanhui Wang. An unbiased offline evaluation of contextual bandit algorithms with generalized linear models. *Journal of Machine Learning Research*, 26:19–36, 2012.

[21] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, pages 661–670, 2010.

[22] Xueqi Li, Wenjun Jiang, Weiguang Chen, Jie Wu, Guojun Wang, and Kenli Li. Directional and explainable serendipity recommendation. In *The Web Conference*, pages 122–132, 2020.

[23] Yuanxiang Li, Zhijie Li, Feng Wang, and Li Kuang. Accelerated online learning for collaborative filtering and recommender systems. In *IEEE International Conference on Data Mining Workshops*, pages 879–885, 2014.

[24] Bo Liu, Ying Wei, Yu Zhang, Zhixian Yan, and Qiang Yang. Transferable contextual bandit for cross-domain recommendation. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 3619–3626, 2018.

[25] Nathan Nan Liu, Min Zhao, Evan Wei Xiang, and Qiang Yang. Online evolutionary collaborative filtering. In *Proceedings of the ACM Conference on Recommender Systems*, pages 95–102. ACM, 2010.

[26] James McInerney, Benjamin Lacker, Samantha Hansen, Karl Higley, Hugues Bouchard, Alois Gruson, and Rishabh Mehrotra. Explore, exploit, and explain: personalizing explainable recommendations with bandits. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 31–39, 2018.

[27] Deng Pan, Xiangrui Li, Xin Li, and Dongxiao Zhu. Explainable recommendation via interpretable feature mapping and evaluation of explainability. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, pages 2690–2696, 2020.

[28] Weike Pan, Shanchuan Xia, Zhuode Liu, Xiaogang Peng, and Zhong Ming. Mixed factorization for collaborative recommendation with heterogeneous explicit feedbacks. *information sciences*, 332:84–93, 2016.

[29] Michael J. Pazzani and Daniel Billsus. Content-based recommendation systems. In *The Adaptive Web, Methods and Strategies of Web Personalization*.

[30] Michael J. Pazzani and Daniel Billsus. Content-based recommendation systems. In *The Adaptive Web, Methods and Strategies of Web Personalization*, pages 325–341, 2007.

[31] Badrul Munir Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International World Wide Web Conference 2001*, pages 285–295, 2001.

[32] Amit Sharma and Dan Cosley. Do social explanations work?: studying and modeling the effects of social explanations in recommender systems. In *22nd International World Wide Web Conference*, pages 1133–1144, 2013.

[33] Aleksandrs Slivkins. Introduction to multi-armed bandits. *Foundations and Trends in Machine Learning*, 12(1-2):1–286, 2019.

[34] Kirsten Swearingen and Rashmi Sinha. Beyond algorithms: An hci perspective on recommender systems. In *ACM SIGIR 2001 workshop on recommender systems*, volume 13, pages 1–11. Citeseer, 2001.

[35] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. Providing justifications in recommender systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 38(6):1262–1272, 2008.

[36] André Uschmajew. Local convergence of the alternating least squares algorithm for canonical tensor approximation. *SIAM Journal on Matrix Analysis and Applications*, 33(2):639–652, 2012.

[37] Jesse Vig, Shilad Sen, and John Riedl. Tagsplanations: explaining recommendations using tags. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, pages 47–56. ACM, 2009.

[38] Hao Wang, Binyi Chen, and Wu-Jun Li. Collaborative topic regression with social regularization for tag recommendation. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pages 2719–2725, 2013.

[39] Huazheng Wang, Qingyun Wu, and Hongning Wang. Learning hidden features for contextual bandits. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, pages 1633–1642. ACM, 2016.

[40] Huazheng Wang, Qingyun Wu, and Hongning Wang. Factorization bandits for interactive recommendation. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 2695–2702. AAAI Press, 2017.

[41] Jialei Wang, Steven C. H. Hoi, Peilin Zhao, and Zhiyong Liu. Online multi-task collaborative filtering for on-the-fly recommender systems. In *7th ACM Conference on Recommender Systems*, pages 237–244, 2013.

[42] Nan Wang, Hongning Wang, Yiling Jia, and Yue Yin. Explainable recommendation via multi-task learning in opinionated text data. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 165–174, 2018.

[43] Qing Wang, Chunqiu Zeng, Wubai Zhou, Tao Li, S. S. Iyengar, Larisa Shwartz, and Genady Ya. Grabarnik. Online interactive collaborative filtering using multi-armed bandit with dependent arms. *IEEE Transactions on Knowledge and Data Engineering*, 31(8):1569–1580, 2019.

[44] Qingyun Wu, Zhige Li, Huazheng Wang, Wei Chen, and Hongning Wang. Factorization bandits for online influence maximization. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 636–646. ACM, 2019.

[45] Qingyun Wu, Huazheng Wang, Quanquan Gu, and Hongning Wang. Contextual bandits in a collaborative environment. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 529–538. ACM, 2016.

[46] Ning Yang, Yuchi Ma, Li Chen, and Philip S. Yu. A meta-feature based unified framework for both cold-start and warm-start explainable recommendations. *World Wide Web*, 23(1):241–265, 2020.

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

[48] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 83–92, 2014.

[49] Xiaoxue Zhao, Weinan Zhang, and Jun Wang. Interactive collaborative filtering. In *22nd ACM International Conference on Information and Knowledge Management*, pages 1411–1420, 2013.

[50] Xiaolin Zheng, Menghan Wang, Chaochao Chen, Yan Wang, and Zhehao Cheng. EXPLORE: explainable item-tag co-recommendation. *Information Sciences*, 474:170–186, 2019.

## A. Proof of Lemma 1 (Upper Confidence Bound of Latent Vectors)

We first prove the upper confidence bound of $\boldsymbol{m}_{u,t}$ at time $t$. By taking the gradient of the objective function defined in Equation (3) and plugging the reward definition in Equation (1) and (2), we can derive the closed form of $\boldsymbol{m}_{u,t}$ via the alternating least square method. Let $\mathbf{A}_{u,t} = \left[\lambda_u \mathbf{I} + \sum_{\tau=1}^{t} \left(\boldsymbol{n}_{a_\tau,\tau} \boldsymbol{n}_{a_\tau,\tau}^\top + \sum_{v \in \mathcal{T}_\tau} \boldsymbol{s}_{v,\tau}^u \boldsymbol{s}_{v,\tau}^{u\top}\right)\right]$, and $\mathcal{G} = \sum_{\tau=1}^{t} \boldsymbol{m}_u^{*\top} \left(\boldsymbol{n}_{a_\tau}^* \boldsymbol{n}_{a_\tau,\tau}^\top + \sum_{v \in \mathcal{T}_\tau} \boldsymbol{s}_v^{u*} \boldsymbol{s}_{v,\tau}^{u\top}\right)$, we have

$$\boldsymbol{m}_{u,t}^\top = \mathbf{A}_{u,t}^{-1} \sum_{\tau=1}^{t} \left[ \left( \boldsymbol{m}_u^{*\top} \boldsymbol{n}_{a_\tau}^* + \boldsymbol{p}_u^{*\top} \boldsymbol{q}_{a_\tau}^* + \eta_\tau^{rec} - \boldsymbol{p}_{u,\tau}^\top \boldsymbol{q}_{a_\tau,\tau} \right) \boldsymbol{n}_{a_\tau,\tau}^\top + \sum_{v \in \mathcal{T}_\tau} \left( \left( \boldsymbol{m}_u^*, \boldsymbol{n}_{a_\tau}^* \right)^\top \left( \boldsymbol{s}_v^{u*}, \boldsymbol{s}_v^{a*} \right) + \eta_\tau^{exp} - \boldsymbol{n}_{a_\tau,\tau}^\top \boldsymbol{s}_{v,\tau}^a \right) \boldsymbol{s}_{v,\tau}^{u\top} \right], \tag{18}$$

$$\mathbf{A}_{u,t} \boldsymbol{m}_{u,t}^\top = \mathcal{G} + \sum_{\tau=1}^{t} \left[ \eta_\tau^{rec} \boldsymbol{n}_{a_\tau,\tau}^\top + \eta_\tau^{exp} \sum_{v \in \mathcal{T}_\tau} \boldsymbol{s}_{v,\tau}^{u\top} + \left( \boldsymbol{p}_u^{*\top} \boldsymbol{q}_{a_\tau}^* - \boldsymbol{p}_{u,\tau}^\top \boldsymbol{q}_{a_\tau,\tau} \right) \boldsymbol{n}_{a_\tau,\tau}^\top + \sum_{v \in \mathcal{T}_\tau} \left( \boldsymbol{n}_{a_\tau}^{*\top} \boldsymbol{s}_v^{a*} - \boldsymbol{n}_{a_\tau,\tau}^\top \boldsymbol{s}_{v,\tau}^a \right) \boldsymbol{s}_{v,\tau}^{u\top} \right]. \tag{19}$$

For the first term, we have

$$\begin{aligned}\mathcal{G} =& \boldsymbol{m}_u^{*\top} \sum_{\tau=1}^{t} \left[ \lambda_u \mathbf{I} + \boldsymbol{n}_{a_\tau,\tau} \boldsymbol{n}_{a_\tau,\tau}^\top + \sum_{v \in \mathcal{T}_\tau} \boldsymbol{s}_{v,\tau}^u \boldsymbol{s}_{v,\tau}^{u\top} + \left( \boldsymbol{n}_{a_\tau}^* - \boldsymbol{n}_{a_\tau,\tau} \right) \boldsymbol{n}_{a_\tau,\tau}^\top + \sum_{v \in \mathcal{T}_\tau} \left( \boldsymbol{s}_v^{u*} - \boldsymbol{s}_{v,\tau}^u \right) \boldsymbol{s}_{v,\tau}^{u\top} - \lambda_u \mathbf{I} \right] \\ =& \boldsymbol{m}_u^{*\top} \mathbf{A}_{u,t} + \boldsymbol{m}_u^{*\top} \sum_{\tau=1}^{t} \left[ \left( \boldsymbol{n}_{a_\tau}^* - \boldsymbol{n}_{a_\tau,\tau} \right) \boldsymbol{n}_{a_\tau,\tau}^\top + \sum_{v \in \mathcal{T}_\tau} \left( \boldsymbol{s}_v^{u*} - \boldsymbol{s}_{v,\tau}^u \right) \boldsymbol{s}_{v,\tau}^{u\top} - \lambda_u \mathbf{I} \right]\end{aligned} \tag{20}$$

Putting Equation (19) and (20) togeteher, we have

$$\begin{aligned}\mathbf{A}_{u,t} \left( \boldsymbol{m}_{u,t}^\top - \boldsymbol{m}_u^{*\top} \right) = \sum_{\tau=1}^{t} \Bigg[ & \boldsymbol{m}_u^{*\top} \left( \boldsymbol{n}_{a_\tau}^* - \boldsymbol{n}_{a_\tau,\tau} \right) \boldsymbol{n}_{a_\tau,\tau}^\top + \boldsymbol{m}_u^{*\top} \sum_{v \in \mathcal{T}_\tau} \left( \boldsymbol{s}_v^{u*} - \boldsymbol{s}_{v,\tau}^u \right) \boldsymbol{s}_{v,\tau}^{u\top} - \lambda_u \boldsymbol{m}_u^* + \eta_\tau^{rec} \boldsymbol{n}_{a_\tau,\tau}^\top \\ & + \eta_\tau^{exp} \sum_{v \in \mathcal{T}_\tau} \boldsymbol{s}_{v,\tau}^{u\top} + \left( \boldsymbol{p}_u^{*\top} \boldsymbol{q}_{a_\tau}^* - \boldsymbol{p}_{u,\tau}^\top \boldsymbol{q}_{a_\tau,\tau} \right) \boldsymbol{n}_{a_\tau,\tau}^\top + \sum_{v \in \mathcal{T}_\tau} \left( \boldsymbol{n}_{a_\tau}^{*\top} \boldsymbol{s}_v^{a*} - \boldsymbol{n}_{a_\tau,\tau}^\top \boldsymbol{s}_{v,\tau}^a \right) \boldsymbol{s}_{v,\tau}^{u\top} \Bigg].\end{aligned} \tag{21}$$

Then,

$$\begin{aligned}& \left\| \boldsymbol{m}_{u,t}^\top - \boldsymbol{m}_u^{*\top} \right\|_{\mathbf{A}_{u,t}} = \left\| \mathbf{A}_{u,t} \left( \boldsymbol{m}_{u,t}^\top - \boldsymbol{m}_u^{*\top} \right) \right\|_{\mathbf{A}_{u,t}^{-1}} \\ \leq & \sum_{\tau=1}^{t} \|\boldsymbol{m}_u^*\|_2 \|\boldsymbol{n}_{a_\tau,\tau}\|_2 \|\boldsymbol{n}_{a_\tau}^* - \boldsymbol{n}_{a_\tau,\tau}\|_{\mathbf{A}_{u,t}^{-1}} + \sum_{\tau=1}^{t} \sum_{v \in \mathcal{T}_\tau} \|\boldsymbol{m}_u^*\|_2 \|\boldsymbol{s}_{v,\tau}^u\|_2 \|\boldsymbol{s}_v^{u*} - \boldsymbol{s}_{v,\tau}^u\|_{\mathbf{A}_{u,t}^{-1}} \\ & + \| \sum_{\tau=1}^{t} \eta_\tau^{rec} \boldsymbol{n}_{a_\tau,\tau}^\top \|_{\mathbf{A}_{u,t}^{-1}} + \| \sum_{\tau=1}^{t} \eta_\tau^{exp} \sum_{v \in \mathcal{T}_\tau} \boldsymbol{s}_{v,\tau}^{u\top} \|_{\mathbf{A}_{u,t}^{-1}} + \| \lambda_m \boldsymbol{m}_u^{*\top} \|_2 \\ \leq & \frac{L_m L_n^2}{\sqrt{\lambda_m}} Q_n + \frac{L_m L_s^2}{\sqrt{\lambda_m}} Q_s + \sqrt{2 \ln \left( \frac{det\left( \mathbf{A}_{u,t} \right)^{1/2}}{det\left( \mathbf{S}_{u,t}^u \mathbf{S}_{u,t}^{u\top} + \lambda_m \mathbf{I} \right)^{1/2} \delta} \right)} + \sqrt{2 \ln \left( \frac{det\left( \mathbf{A}_{u,t} \right)^{1/2}}{det\left( \mathbf{N}_{u,t} \mathbf{N}_{u,t}^\top + \lambda_m \mathbf{I} \right)^{1/2} \delta} \right)} + \sqrt{\lambda_m} L_m \\ = & \alpha^u,\end{aligned} \tag{22}$$

where the first inequality are derived similarly as the proof of Lemma 1 in [39]. The second terms in the last inequality are derived based on the property of $q$-linear convergence and the third and fourth terms are bounded by the property of self-normalized vector-valued martingales [1] if the noise $\eta_t^{rec}$ and $\eta_t^{exp}$ are $1/\sqrt{2}$-sub-Gaussian.

The upper bound of $\left\| \boldsymbol{n}_a^* - \boldsymbol{n}_{a,t} \right\|_{\mathbf{B}_{a,t}}$, $\left\| \boldsymbol{p}_u^* - \boldsymbol{p}_{u,t} \right\|_{\mathbf{C}_{u,t}}$ and $\left\| \boldsymbol{q}_a^* - \boldsymbol{q}_{a,t} \right\|_{\mathbf{D}_{a,t}}$ are similarly proved.

## B. Proof of Lemma 2 (Upper Bound of the True Reward)

First, let $\mathcal{Q}_t = 2L_m L_n \left( q_m + \epsilon_m \right)^t \left( q_n + \epsilon_n \right)^t + 2L_p L_q \left( q_p + \epsilon_p \right)^t \left( q_q + \epsilon_q \right)^t$. According to arm selection strategy, if arm $a_t$ is chosen at time $t$, we have,

$$\boldsymbol{m}_{u,t}^\top \boldsymbol{n}_{a_t,t} + \boldsymbol{p}_{u,t}^\top \boldsymbol{q}_{a_t,t} + \mathcal{B}_{a_t,t} + \mathcal{Q}_t \geq \boldsymbol{m}_{u,t}^\top \boldsymbol{n}_{a_t^*,t} + \boldsymbol{p}_{u,t}^\top \boldsymbol{q}_{a_t^*,t} + \mathcal{B}_{a_t^*,t} + \mathcal{Q}_t \geq \boldsymbol{m}_u^{*\top} \boldsymbol{n}_{a_t^*}^* + \boldsymbol{p}_u^{*\top} \boldsymbol{q}_{a_t^*}^* - \mathcal{F}_t + \mathcal{Q}_t, \tag{23}$$

where the second inequality is derived by proving that $\boldsymbol{m}_{u,t}^\top \boldsymbol{n}_{a_t^*,t} + \boldsymbol{p}_{u,t}^\top \boldsymbol{q}_{a_t^*,t} + \mathcal{B}_{a_t^*,t} - \boldsymbol{m}_u^{*\top} \boldsymbol{n}_{a_t^*}^* - \boldsymbol{p}_u^{*\top} \boldsymbol{q}_{a_t^*}^* \geq -\mathcal{F}$ as follows.

$$
\begin{aligned}
&\boldsymbol{m}_{u,t}^\top \boldsymbol{n}_{a_t^*,t} + \boldsymbol{p}_{u,t}^\top \boldsymbol{q}_{a_t^*,t} + \mathcal{B}_{a_t^*,t} - \boldsymbol{m}_u^{*\top} \boldsymbol{n}_{a_t^*}^* - \boldsymbol{p}_u^{*\top} \boldsymbol{q}_{a_t^*}^* \\
=&\boldsymbol{n}_{a_t^*,t}^\top \left(\boldsymbol{m}_{u,t} - \boldsymbol{m}_u^*\right) + \boldsymbol{m}_u^{*\top} \left(\boldsymbol{n}_{a_t^*,t} - \boldsymbol{n}_{a_t^*}^*\right) + \boldsymbol{q}_{a_t^*,t}^\top \left(\boldsymbol{p}_{u,t} - \boldsymbol{p}_u^*\right) + \boldsymbol{p}_u^{*\top} \left(\boldsymbol{q}_{a_t^*,t} - \boldsymbol{q}_{a_t^*}^*\right) + \mathcal{B}_{a_t^*,t} \\
\geq& - \left\|\boldsymbol{n}_{a_t^*,t}\right\|_{\mathbf{A}_{u,t}^{-1}} \left\|\boldsymbol{m}_{u,t} - \boldsymbol{m}_u^*\right\|_{\mathbf{A}_{u,t}} - \left\|\boldsymbol{m}_u^*\right\|_{\mathbf{B}_{a_t^*,t}^{-1}} \left\|\boldsymbol{n}_{a_t^*,t} - \boldsymbol{n}_{a_t^*}^*\right\|_{\mathbf{B}_{a_t^*,t}} \\
&- \left\|\boldsymbol{q}_{a_t^*,t}\right\|_{\mathbf{C}_{u,t}^{-1}} \left\|\boldsymbol{p}_{u,t} - \boldsymbol{p}_u^*\right\|_{\mathbf{C}_{u,t}} - \left\|\boldsymbol{p}_u^*\right\|_{\mathbf{D}_{a_t^*,t}^{-1}} \left\|\boldsymbol{q}_{a_t^*} - \boldsymbol{q}_{a_t^*}^*\right\|_{\mathbf{D}_{a_t^*,t}} + \mathcal{B}_{a_t^*,t} \\
\geq& - \alpha_t^{m_u} \left\|\boldsymbol{n}_{a_t^*,t}\right\|_{\mathbf{A}_{u,t}^{-1}} - \alpha_t^{n_{a_t^*}} \left\|\boldsymbol{m}_u^*\right\|_{\mathbf{B}_{a_t^*,t}^{-1}} - \alpha_t^{p_u} \left\|\boldsymbol{q}_{a_t^*,t}\right\|_{\mathbf{C}_{u,t}^{-1}} - \alpha_t^{q_{a_t^*}} \left\|\boldsymbol{p}_u^*\right\|_{\mathbf{D}_{a_t^*,t}^{-1}} \\
&+ \alpha_t^{p_u} \left\|\boldsymbol{q}_{a_t^*,t}\right\|_{\mathbf{C}_{u,t}^{-1}} + \alpha_t^{q_{a_t^*}} \left\|\boldsymbol{p}_{u,t}\right\|_{\mathbf{D}_{a_t^*,t}^{-1}} + \alpha_t^{n_{a_t^*}} \left\|\boldsymbol{m}_{u,t}\right\|_{\mathbf{B}_{a_t^*,t}^{-1}} + \alpha_t^{m_u} \left\|\boldsymbol{n}_{a_t^*,t}\right\|_{\mathbf{A}_{u,t}^{-1}} \\
\geq& - \alpha_t^{n_{a_t^*}} \left\|\boldsymbol{m}_u^* - \boldsymbol{m}_{u,t}\right\|_{\mathbf{B}_{a_t^*,t}^{-1}} - \alpha_t^{q_{a_t^*}} \left\|\boldsymbol{p}_u^* - \boldsymbol{p}_{u,t}\right\|_{\mathbf{D}_{a_t^*,t}^{-1}} \\
=& - \mathcal{F}.
\end{aligned}
\tag{24}
$$

The first inequality is derived using the Cauchy-Schwartz inequality. The second inequality is derived using the upper confidence bound of the latent vectors in Lemma 1. The third inequality is derived similarly as in [39, 40, 44].

## C. Proof of Lemma 3 (Regret at Time $t$)

The regret at time $t$, i.e., $\mathcal{R}_t$ can be derived as:

$$
\begin{aligned}
\mathcal{R}_t =& r_{u_t,a_t^*} - r_{u_t,a_t} \\
=& \boldsymbol{m}_{u_t}^{*\top} \boldsymbol{n}_{a_t^*}^* + \boldsymbol{p}_{u_t}^{*\top} \boldsymbol{q}_{a_t^*}^* - \boldsymbol{m}_{u_t}^{*\top} \boldsymbol{n}_{a_t}^* - \boldsymbol{p}_{u_t}^{*\top} \boldsymbol{q}_{a_t}^* \\
\leq& \boldsymbol{m}_{u_t,t}^\top \boldsymbol{n}_{a_t,t} + \boldsymbol{p}_{u_t,t}^\top \boldsymbol{q}_{a_t,t} + \mathcal{B}_{a_t,t} + \mathcal{F}_t - \boldsymbol{m}_{u_t}^{*\top} \boldsymbol{n}_{a_t}^* - \boldsymbol{p}_{u_t}^{*\top} \boldsymbol{q}_{a_t}^* \\
=& \boldsymbol{n}_{a_t,t}^\top \left(\boldsymbol{m}_{u_t,t} - \boldsymbol{m}_{u_t}^*\right) + \boldsymbol{m}_{u_t}^{*\top} \left(\boldsymbol{n}_{a_t,t} - \boldsymbol{n}_{a_t}^*\right) + \boldsymbol{q}_{a_t,t}^\top \left(\boldsymbol{p}_{u_t,t} - \boldsymbol{p}_{u_t}^*\right) + \boldsymbol{p}_{u_t}^{*\top} \left(\boldsymbol{q}_{a_t,t} - \boldsymbol{q}_{a_t}^*\right) + \mathcal{B}_{a_t,t} + \mathcal{F}_t \\
\leq& \left\|\boldsymbol{n}_{a_t,t}\right\|_{\mathbf{A}_{u_t,t}^{-1}} \left\|\boldsymbol{m}_{u_t,t} - \boldsymbol{m}_{u_t}^*\right\|_{\mathbf{A}_{u_t,t}} + \left\|\boldsymbol{m}_{u_t}^*\right\|_{\mathbf{B}_{a_t,t}^{-1}} \left\|\boldsymbol{n}_{a_t,t} - \boldsymbol{n}_{a_t}^*\right\|_{\mathbf{B}_{a_t,t}} + \left\|\boldsymbol{q}_{a_t,t}\right\|_{\mathbf{C}_{u_t,t}^{-1}} \left\|\boldsymbol{p}_{u_t,t} - \boldsymbol{p}_{u_t}^*\right\|_{\mathbf{C}_{u_t,t}} \\
&+ \left\|\boldsymbol{p}_{u_t}^*\right\|_{\mathbf{D}_{a_t,t}^{-1}} \left\|\boldsymbol{q}_{a_t,t} - \boldsymbol{q}_{a_t}^*\right\|_{\mathbf{D}_{a_t,t}} + \mathcal{B}_{a_t,t} + \mathcal{F}_t \\
\leq& \alpha_t^{m_{u_t}} \left\|\boldsymbol{n}_{a_t,t}\right\|_{\mathbf{A}_{u_t,t}^{-1}} + \alpha_t^{n_{a_t}} \left\|\boldsymbol{m}_{u_t}^*\right\|_{\mathbf{B}_{a_t,t}^{-1}} + \alpha_t^{p_{u_t}} \left\|\boldsymbol{q}_{a_t,t}\right\|_{\mathbf{C}_{u_t,t}^{-1}} + \alpha_t^{q_{a_t}} \left\|\boldsymbol{p}_{u_t}^*\right\|_{\mathbf{D}_{a_t,t}^{-1}} + \mathcal{B}_{a_t,t} + \mathcal{F}_t \\
\leq& 2\alpha_t^{n_{a_t}} \left\|\boldsymbol{m}_{u_t,t}\right\|_{\mathbf{B}_{a_t,t}^{-1}} + 2\alpha_t^{q_{a_t}} \left\|\boldsymbol{p}_{u_t,t}\right\|_{\mathbf{D}_{a_t,t}^{-1}} + 2\alpha_t^{m_{u_t}} \left\|\boldsymbol{n}_{a_t,t}\right\|_{\mathbf{A}_{u_t,t}^{-1}} + 2\alpha_t^{p_{u_t}} \left\|\boldsymbol{q}_{a_t,t}\right\|_{\mathbf{C}_{u_t,t}^{-1}} \\
&+ \alpha_t^{n_{a_t}} \left\|\boldsymbol{m}_{u_t}^* - \boldsymbol{m}_{u_t,t}\right\|_{\mathbf{B}_{a_t,t}^{-1}} + \alpha_t^{q_{a_t}} \left\|\boldsymbol{p}_{u_t}^* - \boldsymbol{p}_{u_t,t}\right\|_{\mathbf{D}_{a_t,t}^{-1}} + \alpha_t^{n_{a_t^*}} \left\|\boldsymbol{m}_{u_t}^* - \boldsymbol{m}_{u_t,t}\right\|_{\mathbf{B}_{a_t^*,t}^{-1}} + \alpha_t^{q_{a_t^*}} \left\|\boldsymbol{p}_{u_t}^* - \boldsymbol{p}_{u_t,t}\right\|_{\mathbf{D}_{a_t^*,t}^{-1}},
\end{aligned}
\tag{25}
$$

where the first inequality is derived according to Lemma 2 and the other inequalities are derived similar to the inequalities in the proof of Lemma 2.

## D. Proof of Theorem 1 (Cumulative Regret until Time $T$)

$$
\begin{aligned}
R(T) = \sum_{t=1}^{T} \mathcal{R}_t \leq & \sqrt{T \sum_{t=1}^{T} \mathcal{R}_t^2} \\
\leq & 2\alpha_T^n \sqrt{T \sum_{t=1}^{T} \left\| \boldsymbol{m}_{u_t,t} \right\|^2_{\mathbf{B}^{-1}_{a_t,t}}} + 2\alpha_T^m \sqrt{T \sum_{t=1}^{T} \left\| \boldsymbol{n}_{a_t,t} \right\|^2_{\mathbf{A}^{-1}_{u_t,t}}} + 2\alpha_T^h \sqrt{T \sum_{t=1}^{T} \left\| \boldsymbol{g}_{u_t,t} \right\|^2_{\mathbf{D}^{-1}_{a_t,t}}} + 2\alpha_T^g \sqrt{T \sum_{t=1}^{T} \left\| \boldsymbol{h}_{a_t,t} \right\|^2_{\mathbf{C}^{-1}_{u_t,t}}} \\
& + 2\alpha_T^n \frac{1}{\sqrt{\lambda_n}} \sum_{t=1}^{T} \left\| \boldsymbol{m}^*_{u_t} - \boldsymbol{m}_{u_t,t} \right\|_2 + 2\alpha_T^h \frac{1}{\sqrt{\lambda_g}} \sum_{t=1}^{T} \left\| \boldsymbol{g}^*_{u_t} - \boldsymbol{g}_{u_t,t} \right\|_2 \\
\leq & 2\alpha_T^n \sqrt{2T \ln \sum_{a \in \mathcal{I}} \frac{\det \left( \mathbf{B}_{a,T} \right)^{1/2}}{\det \left( \mathbf{S}^a_{a,T} \mathbf{S}^{a\top}_{a,T} + \lambda_n \mathbf{I} \right)^{1/2} \delta}} + 2\alpha_T^m \sqrt{2T \ln \sum_{u \in \mathcal{U}} \frac{\det \left( \mathbf{A}_{u,T} \right)^{1/2}}{\det \left( \mathbf{S}^u_{u,T} \mathbf{S}^{u\top}_{u,T} + \lambda_m \mathbf{I} \right) \delta}} \\
& + 2\alpha_T^h \sqrt{2dT \ln \left( 1 + \frac{TL_g}{\lambda_h d} \right)} + 2\alpha_T^g \sqrt{2dT \ln \left( 1 + \frac{TL_h}{\lambda_g d} \right)} + 2\alpha_T^n \frac{L_m}{\sqrt{\lambda_n}} Q_m + 2\alpha_T^h \frac{L_g}{\sqrt{\lambda_g}} Q_g,
\end{aligned}
\tag{26}
$$

where the first four terms in the inequality are derived according to the Lemma 11 in [1] and the last two terms are derived according to the $q$-linear convergence property similarly as in [39, 40].