SAQRec: Aligning Recommender Systems to User Satisfaction via Questionnaire Feedback

Kepu Zhang
Teng Shi
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
{kepuzhang,shiteng}@ruc.edu.cn

Sunhao Dai
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
sunhaodai@ruc.edu.cn

Xiao Zhang*
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
zhangx89@ruc.edu.cn

Yinfeng Li
Jing Lu
Kuaishou Technology Co., Ltd.
Beijing, China
{liyinfeng,lvjing06}@kuaishou.com

Xiaoxue Zang
Yang Song
Kuaishou Technology Co., Ltd.
Beijing, China
{zangxiaoxue,yangsong}@kuaishou.com

Jun Xu
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
junxu@ruc.edu.cn

ABSTRACT

In real-world recommender systems, user engagement and subjective feedback play pivotal roles in shaping the content distribution mechanism of the platform. When platforms reach a certain scale, they often gather valuable questionnaire feedback data from users to evaluate their satisfaction with recommended items. Compared to traditional user feedback such as likes, questionnaires explicitly capture both satisfaction and dissatisfaction and are unaffected by other users' questionnaires, thus better expressing users' true preferences. In this paper, we aim to leverage the questionnaire feedback to align the recommendation model with users' true preferences. However, due to the platform distribution mechanism and divergent user attitudes toward questionnaires, the questionnaire feedback data frequently becomes sparse and exhibits selection biases, resulting in challenges in feature integration and training process. To address these issues, we introduce a novel user **S**atisfaction **A**lignment framework that effectively leverages **Q**uestionnaire feedback to enhance **Rec**ommendation, named **SAQRec**. SAQRec begins by training an unbiased satisfaction model to impute satisfaction, addressing selection bias and data sparsity. Then, SAQRec aligns features with users' true preferences by disentangling satisfaction and dissatisfaction from click history and categorizing clicked items into multiple satisfaction levels through the imputed satisfactions. Additionally, the imputed satisfactions from the pre-trained unbiased satisfaction model serve as pseudo-labels to align the model's outputs with users' true preferences. Extensive experiments on both public and commercial datasets demonstrate SAQRec's superior integration of questionnaire feedback in recommendation models. Online A/B testing on a short video platform confirms its effectiveness in boosting user watch time and positive-to-negative feedback ratio, enhancing overall performance and user satisfaction.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Computing methodologies** → **Learning from implicit feedback**.

KEYWORDS

alignment, unbiased learning, questionnaire feedback, recommendation

ACM Reference Format:
Kepu Zhang, Teng Shi, Sunhao Dai, Xiao Zhang, Yinfeng Li, Jing Lu, Xiaoxue Zang, Yang Song, and Jun Xu. 2024. SAQRec: Aligning Recommender Systems to User Satisfaction via Questionnaire Feedback. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3627673.3679643

1 INTRODUCTION

Recommender systems have seen extensive utilization across a number of platforms such as e-commerce [4, 25], music [7, 8, 41], and video [19, 23, 43]. Existing recommendation models primarily rely on user click feedback to learn user preferences [5, 9, 13, 16, 20, 31]. Nevertheless, as identified in the previous work [2], various biases influencing user clicks, the click feedback may fail to accurately reflect users' true preferences. In response to this limitation, online platforms such as e-commerce and video-watching applications have elected to use a more interactive approach. Specifically, they incorporate questionnaires presented with the interacted items during user browsing sessions, as shown in Figure 1(a).

Unlike user feedback such as likes and favorites, questionnaire feedback explicitly conveys users' true preferences [28, 30]. The specific reasons are: (1) Existing user feedback is either positive or negative, such as "like", with no explicit negative option. Questionnaires, however, offer both positive and negative options, thus

*Xiao Zhang is the corresponding author. Work partially done at Engineering Research Center of Next-Generation Intelligent Search and Recommendation, Ministry of Education. Work done when Kepu Zhang and Teng Shi were interns at Kuaishou.

CIKM '24, October 21–25, 2024, Boise, ID, USA

ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3679643

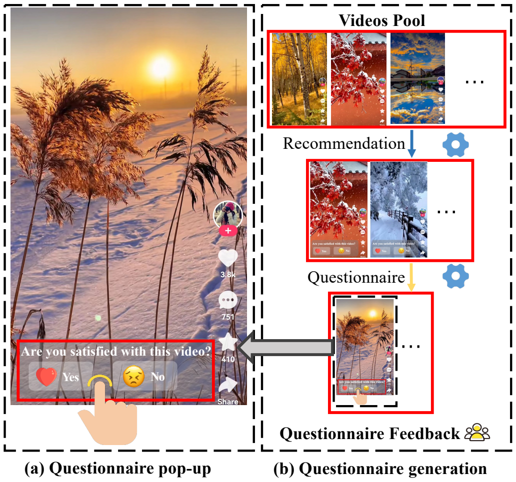


Figure 1: Questionnaire in the short-video scenario. (a) An example of a video with a questionnaire pop-up. (b) The generation process of the questionnaire feedback.

allowing users to express both positive and negative sentiments. (2) Although likes and favorites can be considered as a form of active feedback, this feedback is often built upon the choices of others. For instance, users clicking "like" may be influenced by herd mentality, as they add to existing likes. In contrast, when users fill out questionnaires, because they do not know the opinions of other users, their choices can better reflect their true thoughts. In real-world recommender systems, the primary goal of recommendation models is to boost user engagement, specifically by increasing user clicks. Leveraging questionnaire feedback can significantly assist recommendation models in aligning recommended results with users' true preferences and potentially increase user clicks.

However, aligning with users' true preferences using questionnaire data faces two challenges: (1) *Data sparsity and selection bias*: The questionnaire interactions, comprising about 1% of total interactions (refer to Table 1), exhibit extreme sparsity. Figure 1(b) illustrates the platform's questionnaire exposure mechanism, limiting questionnaires to a small subset of specific items. Users may click on only a portion of the questionnaires for various reasons, introducing notable selection bias during data collection. The data sparsity and selection bias pose challenges in enhancing recommendations using user questionnaire feedback. (2) *Clicks despite dissatisfaction*: A user may click on an item that is not satisfactory. For instance, as depicted in Figure 1, a user's watch time exceeding a certain threshold is considered a click, but the user may be dissatisfied with the video. Moreover, from the data statistics in Table 1, it is evident that there are many instances of user dissatisfaction within the click data. How to leverage questionnaire data to help disentangle the information about user satisfaction and dissatisfaction in clicks is crucial for aligning with users' true preferences.

A natural way to leverage users' questionnaire feedback is to formulate the problem as multi-behavior recommendations since they also utilize feedback beyond just clicks, such as likes and others. These methods can be mainly categorized into two types: (1) The first type separately inputs different user behaviors into distinct encoders for modeling, such as DMT [11] and DIPN [12]. (2) The second type combines different behaviors into a single sequence after sorting them by time, aiming to model the relationships between different behaviors, including DFN [38], FeedRec [36]. However, these approaches cannot be directly applied to address the two challenges commonly faced when incorporating questionnaire feedback: (1) These approaches typically depend on positive user feedback, such as "like", which is less sparse compared to questionnaire feedback, while treating unselected user feedback heuristically as negative feedback. (2) They model various user behaviors but do not pay particular attention to that click sequences may carry information about both user satisfaction and dissatisfaction. Therefore, further research is needed to refine the recommendation model to better align with users' true preferences through questionnaire feedback.

To better *leverage the questionnaire feedback to align with users' true preferences*, we propose a novel user **S**atisfaction **A**lignment framework using users' **Q**uestionnaire feedback for enhancing **Rec**ommendation, called **SAQRec**. Overall, we employ user questionnaire feedback to pre-train a satisfaction model and leverage its output to guide the feature representation and training process of the recommendation model, ensuring alignment with users' true preferences. Specifically, SAQRec mainly consists of two stages: (1) *Unbiased satisfaction model pre-training* aims to address the first challenge. Specifically, we train a satisfaction model on data with questionnaire feedback to provide imputed satisfactions for user-item pairs without questionnaire feedback, thus mitigating the issue of data sparsity. Meanwhile, given the selection bias in questionnaire feedback, we employ the IPS [29] method for the training of the satisfaction model, making it unbiased to address this concern. (2) *Satisfaction feature alignment* distinguishes satisfaction and dissatisfaction information in user clicks from representation perspectives. On one hand, we represent satisfaction and dissatisfaction histories separately, using attention mechanisms to extract them from the overall click history. On the other hand, we use the trained unbiased satisfaction model to assign scores to these items, group items based on their scores and perform a weighted aggregation within each group.

In the training process, we employ a multi-task learning approach to predict click and satisfaction simultaneously. Imputed satisfactions from the pre-trained satisfaction model serve as supervision signals to align the model outputs with users' true preferences. In summary, we make the following contributions:

- Our work represents a pioneering effort in utilizing questionnaire feedback to refine recommendation models, where the questionnaire feedback captures users' true preferences but is sparse and subject to selection bias.
- We introduce SAQRec, a novel framework that leverages questionnaire feedback to aligns recommendation models with users' true preferences. SAQRec tackles data sparsity and selection bias by pre-training an unbiased satisfaction model and aligns the recommendation model with user true preferences in feature representation and model training.
- Offline experiments on commercial and public datasets confirm SAQRec's superior performance over traditional sequential recommendation models and methods directly transferred from multi-behavior recommendation models.
- Online A/B testing on Kuaishou APP across 7 consecutive days shows that SAQRec consistently increases users' watch time and

the positive-to-negative feedback ratio, demonstrating enhanced performance and user satisfaction.

2 RELATED WORK

Sequential Recommendation is a pivotal domain within recommender systems, aiming to capture user interests from their interaction history. Existing works leverage various neural network architectures for sequential recommendation. GRU4Rec [14] employs GRU units to capture the dynamic changes in user interests through recurrent neural networks. Caser [32] combines CNN by treating user interaction history as an "image" in the time-latent space. It utilizes convolutional filters to capture sequential patterns. SASRec [16] is based on the Transformer architecture, utilizing self-attention mechanisms to capture global dependencies within the sequences. There are also researches [1, 37] employing GNN to enhance sequential recommendation. FMLP-Rec [46] employs learnable filters to reduce noise information and utilizes MLP structures to encode sequential data. In addition, STAMP [22] introduces memory network technology, while HGN [24] utilizes hierarchical gating techniques to capture user interests. In contrast to existing works, we utilize questionnaire feedback to align the recommendation model with users' true preferences.

Multi-Behavior Recommendation aims to effectively combine and utilize diverse user behavior data to improve the performance of recommendations. Existing works employ various approaches to integrate data from different behavior sequences. DMT [11] utilizes multiple Transformers to concurrently model different user behavior sequences, capturing diverse aspects of user interests. DIPN [12] leverages hierarchical attention mechanisms to model relationships within and between different sequences in a layered manner. Nevertheless, these two approaches lack explicit differentiation in the significance of modeling the user interests across distinct behavior sequences. In contrast to the methods mentioned above, DFN [38] designs a feedback interaction mechanism for presenting explicit and implicit feedback interactions. FeedRec [36] utilizes heterogeneous and homogeneous Transformers for sequence modeling. It employs an attention mechanism to capture transition patterns between different feedback types. However, in the two aforementioned works, explicit feedback is not as sparse as questionnaire feedback, and unselected user feedback is heuristically treated as negative. They did not take into account that user click sequences may involve interactions with varying levels of satisfaction. In this work, we employ a satisfaction model to handle data sparsity through imputing satisfaction. By incorporating questionnaire feedback, we align the model's features with users' true preferences, enhancing multi-task learning in the recommendation model for better alignment during training.

3 PROBLEM FORMULATION

Let $\mathcal{U}$ and $\mathcal{I}$ denote the sets of users and items, respectively. The recommendation dataset with questionnaire feedback is denoted as $\mathcal{D} = \{(u, i, T_r, T_{s+}, T_{s-}, y, o, s)_k\}_{k=1}^{|\mathcal{D}|}$, where $u \in \mathcal{U}$ and $i \in \mathcal{I}$ denote a user and an item, respectively. $T_r = \{i_1, i_2, \ldots, i_{N_r}\}$, $T_{s+} = \{i_1, i_2, \ldots, i_{N_{s+}}\}$, and $T_{s-} = \{i_1, i_2, \ldots, i_{N_{s-}}\}$ represent the user u's clicked, satisfied, and dissatisfied histories, respectively. There are N_r, N_{s+}, and N_{s-} interacted items corresponding to T_r, T_{s+} and T_{s-}. $y \in \{0, 1\}$ denotes whether user u clicks on item i, where 1 represents a click, and 0 represents no click. $o \in \{0, 1\}$ indicates whether the platform exposed a questionnaire while user u was browsing item i, where 1 represents exposure, and 0 represents no exposure. $s \in \{0, 1\}$ represents whether user u is satisfied with item i, where 1 indicates satisfaction, and 0 indicates dissatisfaction. Then, given the contextual sequences T_r, T_{s+} and T_{s-} representing the user u's clicked, satisfied, and dissatisfied histories, our goal is to improve the performance of the recommendation model.

4 SAQREC: THE PROPOSED FRAMEWORK

In this section, we introduce SAQRec, which is a user **S**atisfaction **A**lignment framework that effectively leverages sparse and biased **Q**uestionnaire feedback to enhance **Rec**ommendation.

4.1 Overall Framework

SAQRec initially pre-trains a satisfaction model to impute unknown questionnaire feedback from users, as illustrated in Figure 2 (a). In this process, a propensity model is utilized to mitigate selection bias and an unbiased satisfaction model is obtained. Subsequently, as shown in Figure 2 (b), SAQRec aligns the recommendation model with users' true preferences through the pre-trained satisfaction model, guiding the feature representation and the training process of the recommendation model. For a more detailed illustration, refer to Figure 2 (c): (1) The satisfaction feature alignment includes a Satisfaction Information Disentanglement (SID) module that utilizes attention mechanisms to disentangle users' satisfaction and dissatisfaction information from the click history, and a Multi-level Satisfaction Enhancement (MLSE) module that extracts multiple satisfaction levels from the user's clicked history using the satisfaction model; (2) During training, SAQRec outputs predicted clicks and satisfactions, with imputed satisfactions from the pre-trained unbiased satisfaction model acting as pseudo-labels to align the model's outputs with users' true preferences.

4.2 Unbiased Satisfaction Model Pre-training

In this section, we present an unbiased satisfaction model to provide imputations for the large amounts of samples lacking user questionnaire feedback.

4.2.1 Ideal and Naive Learning Objectives for Satisfaction Model. First, we define the satisfaction model g with parameters Θ_g, which takes an (u, i) pair as input and outputs the corresponding imputed satisfaction $\hat{s}$:

$$\hat{s} = g(u, i; \Theta_g). \tag{1}$$

Ideally, the empirical loss for learning the satisfaction model can be defined as follows:

$$\mathcal{L}_{\text{ideal}}(\hat{s}) = -\frac{1}{|\mathcal{D}|} \sum_{\mathcal{D}} s\log(\hat{s}) + (1 - s)\log(1 - \hat{s}), \tag{2}$$

where the widely used cross-entropy loss [44, 45] is adopted.

As mentioned above, we only have limited data O with questionnaire feedback from a small number of users and items, i.e., $O = \{(u, i)_k | o_k = 1\}_{k=1}^{|\mathcal{D}|}$. Thus, it is impossible to directly train the satisfaction model with the ideal loss in practice.

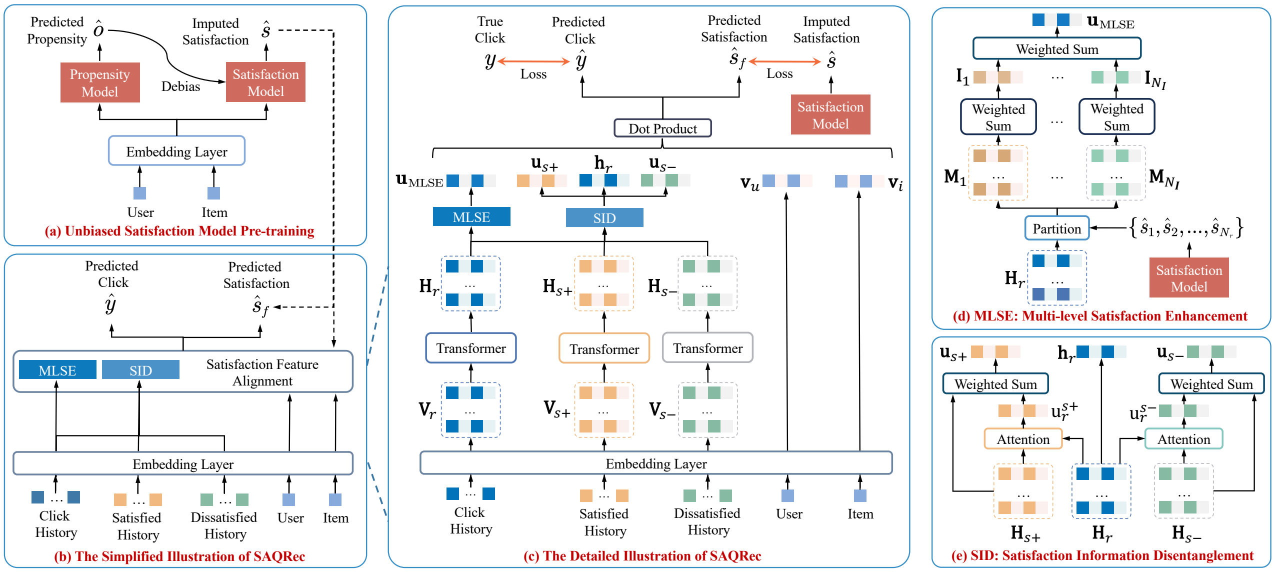


Figure 2: The architecture of SAQRec framework. (a) The structure of the unbiased satisfaction model pre-training in the first stage; (b) A simplified illustration depicting the SAQRec framework in the second stage and its relationship with the satisfaction model outputs; (c) A detailed illustration of the second stage of the SAQRec framework; (d) The structure of Multi-level Satisfaction Enhancement (MLSE); (e) The structure of Satisfaction Information Disentanglement (SID).

A naive method is to optimize directly on the dataset $\mathcal{O}$:

$$\mathcal{L}_{\text{naive}}(\hat{s}) = -\frac{1}{|\mathcal{O}|}\sum_{\mathcal{O}} s\log(\hat{s}) + (1-s)\log(1-\hat{s}). \tag{3}$$

However, due to the platform selectively distributing questionnaires to specific users and items, and users may choose not to click when receiving a questionnaire for various reasons, there exists a challenge of **selection bias** in the observed questionnaire feedback dataset $\mathcal{O}$. Consequently, the naive loss in Eq. (3) is biased against the ideal loss Eq. (2), i.e., $\mathbb{E}_{\mathcal{O}}[\mathcal{L}_{\text{naive}}(\hat{s})] \neq \mathcal{L}_{\text{ideal}}(\hat{s})$. The experiment using the naive loss to train a satisfaction model for the recommendation model in Sec. 5.3 also shows that this bias leads to a decline in performance.

4.2.2 Unbiased Learning Objective for Satisfaction Model. To achieve an unbiased learning objective on the observed questionnaire feedback data, we adopt the Inverse Propensity Score (IPS) [29, 42] method. The propensity is represented as $\Pr(o=1)$, which is the probability of observing questionnaire feedback from user u for item i. Then, the proposed unbiased learning objective for the satisfaction model can be defined as

$$\mathcal{L}_{\text{unbias}}(\hat{s}) = -\frac{1}{|\mathcal{O}|}\sum_{(u,i)\in\mathcal{O}} \frac{s\log(\hat{s}) + (1-s)\log(1-\hat{s})}{\Pr(o=1)}. \tag{4}$$

The unbiasedness of the above loss is shown in the following theorem. Proofs are provided in the Appendix.

Theorem 4.1 (Unbiased Satisfaction Estimation). *The proposed loss in Eq.* (4) *is unbiased in terms of the ideal loss in Eq.* (2)*:*

$$\mathbb{E}_{\mathcal{O}}\left[\mathcal{L}_{\text{unbias}}(\hat{s})\right] = \mathcal{L}_{\text{ideal}}(\hat{s}).$$

To estimate the propensity $\hat{o} = \Pr(o=1)$, we train a propensity model ρ with parameters Θ_ρ using the following learning objective on the full dataset $\mathcal{D}$:

$$\mathcal{L}_{\text{propensity}}(\hat{o}) = -\frac{1}{|\mathcal{D}|}\sum_{\mathcal{D}} o\log(\hat{o}) + (1-o)\log(1-\hat{o}). \tag{5}$$

Following the common practice [3, 27, 35], we adopt propensity clipping techniques in experiments to reduce the high variance of unbiased estimates. Therefore, the final loss for training the satisfaction model is as follows:

$$\mathcal{L}_{\text{final}}(\hat{s}) = -\frac{1}{|\mathcal{O}|}\sum_{\mathcal{O}} \frac{s\log(\hat{s}) + (1-s)\log(1-\hat{s})}{\max(\hat{o}, M)}, \tag{6}$$

where $M \in [0,1]$ is a hyper-parameter to control the clip value.

During the training of the satisfaction model, we utilize the user and item embeddings from a pre-trained recommendation model as initialization. We keep the embedding parameters frozen during training. This approach allows us to train on data with questionnaire feedback and generalize the model to interactions without questionnaires.

Motivated by RLHF [15, 26], questionnaire feedback can be regarded as genuine human feedback that reveals users' true preferences in recommender systems. The satisfaction model, trained based on questionnaire feedback, serves as a reward model, guiding the recommendation model to align with users' true preferences.

Specifically, leveraging the imputed satisfactions provided by the satisfaction model, we introduce the Satisfaction Feature Alignment module in Section 4.3.3 to better model users' true preferences at the feature level. Additionally, in Section 4.4.2, we use the imputed satisfactions as supervision signals for the multi-task learning of the recommendation model, enhancing alignment with users' true preferences during training.

4.3 Satisfaction Feature Alignment

In this section, we use questionnaire feedback and the pre-trained satisfaction model to align feature representations with users' true preferences. In Section 4.3.2, we disentangle user satisfaction and dissatisfaction preferences from the click history using questionnaire feedback. In Section 4.3.3, using the trained satisfaction model, we assign scores to clicked items, creating muti-level satisfaction groups to enhance the model's alignment with user preferences.

4.3.1 Embedding Layer. We maintain separate lookup tables for the IDs and attributes of users and items. The user (item) embeddings, denoted as $\mathbf{v}_u$ and $\mathbf{v}_i$, are obtained by concatenating the ID and attribute embeddings: $\mathbf{v}_u = \mathbf{v}_{\mathrm{ID}_u} \oplus \mathbf{v}_{a_1} \oplus \cdots \oplus \mathbf{v}_{a_n}$ ($\mathbf{v}_i = \mathbf{v}_{\mathrm{ID}_i} \oplus \mathbf{v}_{b_1} \oplus \cdots \oplus \mathbf{v}_{b_m}$), where $\oplus$ denotes concatenation, $a_1, \ldots, a_n$ and $b_1, \ldots, b_m$ are the attributes of users and items, respectively. For convenience, both $\mathbf{v}_u$ and $\mathbf{v}_i$ are set to be d-dimensional, i.e., $\mathbf{v}_u, \mathbf{v}_i \in \mathbb{R}^d$. Given the user u's histories T_r, T_{s+} and T_{s-}, we can get their embeddings: $\mathbf{V}_r = [\mathbf{v}_1^i, \mathbf{v}_2^i, \ldots, \mathbf{v}_{N_r}^i]^\top \in \mathbb{R}^{N_r \times d}$, $\mathbf{V}_{s+} = [\mathbf{v}_1^i, \mathbf{v}_2^i, \ldots, \mathbf{v}_{N_{s+}}^i]^\top \in \mathbb{R}^{N_{s+} \times d}$ and $\mathbf{V}_{s-} = [\mathbf{v}_1^i, \mathbf{v}_2^i, \ldots, \mathbf{v}_{N_{s-}}^i]^\top \in \mathbb{R}^{N_{s-} \times d}$ through the look-up operation. We further incorporate positional embeddings $\mathbf{P}_r \in \mathbb{R}^{N_r \times d}$, $\mathbf{P}_{s+} \in \mathbb{R}^{N_{s+} \times d}$, and $\mathbf{P}_{s-} \in \mathbb{R}^{N_{s-} \times d}$ for the three histories:

$$\widehat{\mathbf{V}}_r = \mathbf{V}_r + \mathbf{P}_r, \quad \widehat{\mathbf{V}}_{s+} = \mathbf{V}_{s+} + \mathbf{P}_{s+}, \quad \widehat{\mathbf{V}}_{s-} = \mathbf{V}_{s-} + \mathbf{P}_{s-}. \tag{7}$$

To model the dependencies between user histories, we feed each of the three histories into three transformers [34]:

$$\mathbf{H}_r = \mathrm{Trm}_r(\widehat{\mathbf{V}}_r), \quad \mathbf{H}_{s+} = \mathrm{Trm}_{s+}(\widehat{\mathbf{V}}_{s+}), \quad \mathbf{H}_{s-} = \mathrm{Trm}_{s-}(\widehat{\mathbf{V}}_{s-}), \tag{8}$$

where $\mathbf{H}_r \in \mathbb{R}^{N_r \times d}$, $\mathbf{H}_{s+} \in \mathbb{R}^{N_{s+} \times d}$ and $\mathbf{H}_{s-} \in \mathbb{R}^{N_{s-} \times d}$. "Trm" denotes the transformer encoder.

4.3.2 Satisfaction Information Disentanglement. Users may not always be satisfied with clicked items, as clicks can result from accidental touches or other biases. The click history includes both satisfied and dissatisfied information. Our goal is to disentangle the satisfied and dissatisfied information from the click history using historical questionnaire feedback data.

Firstly, we use the representations of the last positions: $\mathbf{h}_r$, $\mathbf{h}_{s+}$, and $\mathbf{h}_{s-}$, from $\mathbf{H}_r$, $\mathbf{H}_{s+}$, and $\mathbf{H}_{s-}$, as the representations of the overall click information, satisfaction information, and dissatisfaction information for the user, respectively. Then, we use an attention mechanism, treating satisfied and dissatisfied histories as queries, to disentangle satisfied and dissatisfied information from the click history. The detailed process is as follows:

$$\mathbf{u}_r^{s+} = \mathbf{H}_r^\top \mathrm{Softmax}(\mathbf{H}_r \mathbf{h}_{s+}), \quad \mathbf{u}_r^{s-} = \mathbf{H}_r^\top \mathrm{Softmax}(\mathbf{H}_r \mathbf{h}_{s-}), \tag{9}$$

where $\mathbf{u}_r^{s+} \in \mathbb{R}^d$ and $\mathbf{u}_r^{s-} \in \mathbb{R}^d$ are the disentangled satisfaction and dissatisfaction information from click history. The attention mechanism computes the similarity between each item in the click history and the satisfied (dissatisfied) representation, allowing us to identify the items that users are more likely to be satisfied (dissatisfied) with. In this way, we can use the similarity as the weight to disentangle satisfied (dissatisfied) information from the click history, instead of treating all items equally.

After disentangling satisfied (dissatisfied) information from the click history, we fuse it with the information expressed in the user's satisfied (dissatisfied) history using MLPs (Multi-layer Perceptrons) to obtain the final user satisfaction (dissatisfaction) information:

$$\begin{aligned} \mathbf{u}_{s+} &= \left[\mathbf{h}_{s+}, \mathbf{u}_r^{s+}\right] \cdot \mathrm{MLP}_{s+}(\mathbf{h}_{s+} \oplus \mathbf{u}_r^{s+}), \\ \mathbf{u}_{s-} &= \left[\mathbf{h}_{s-}, \mathbf{u}_r^{s-}\right] \cdot \mathrm{MLP}_{s-}(\mathbf{h}_{s-} \oplus \mathbf{u}_r^{s-}), \end{aligned} \tag{10}$$

where $\mathrm{MLP}_{s+}(\mathrm{MLP}_{s-}) : \mathbb{R}^{2d} \to \mathbb{R}^2$ computes the weight of $\mathbf{h}_{s+}$ and $\mathbf{u}_r^{s+}$ ($\mathbf{h}_{s-}$ and $\mathbf{u}_r^{s-}$). $\mathbf{u}_{s+} \in \mathbb{R}^d$ and $\mathbf{u}_{s-} \in \mathbb{R}^d$ are the disentangled user satisfaction and dissatisfaction information, which will be utilized along with $\mathbf{h}_r$ for subsequent predictions.

4.3.3 Multi-level Satisfaction Enhancement. User interests naturally manifest at multiple levels [33, 40]. Satisfaction levels for clicked items may vary. For instance, using a short video platform as an example, a user may frequently browse videos in certain categories, indicating stronger interest and elevated satisfaction. However, there might also be videos the user clicked without true interest due to misleading titles, thumbnails, positional bias, or other reasons, resulting in dissatisfaction. For other videos, the user might only occasionally browse, representing a middle ground between satisfaction and dissatisfaction, which implies various levels of satisfaction. This suggests that exploring different levels of user satisfaction with clicked items can help us model users' true preferences better. As depicted in Figure 2 (d), we employ the pre-trained satisfaction model to assign scores to clicked items, thereby modeling users' multi-level satisfaction.

Specifically, we input the user and the clicked history into the satisfaction model g, obtaining satisfaction scores $\{\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_{N_r}\}$ for each item in the history. We divide the user's click history into N_I groups according to the satisfaction scores, with each group containing about $k = N_r / N_I$ items. N_I is a hyper-parameter. To do this, we sort the satisfaction scores of clicked items in descending order, making it easier for us to group items with close satisfaction scores together. The top k items form the first group, the items from $k+1$ to $2k$ with the next highest scores form the second group, and so on. After this partitioning, we obtain N_I groups. $M_i = \{\mathbf{h}_{i,j}^r\}_{j=1}^{N_i}$ represents the i-th group, comprising N_i items with corresponding satisfaction scores $S_i = \{\hat{s}_{i,j}\}_{j=1}^{N_i}$.

Then, within each group, we aggregate the items with weights based on the their satisfaction scores. This helps us to obtain the representation of the current satisfaction level:

$$\mathbf{I}_i = \sum_{j=1}^{N_i} \frac{\exp(\hat{s}_{i,j})}{\sum_{m=1}^{N_i} \exp(\hat{s}_{i,m})} \mathbf{h}_{i,j}^r, \tag{11}$$

where $\mathbf{I}_i \in \mathbb{R}^d$ is the representation of the i-th level satisfaction. Next, we perform a weighted aggregation of all groups:

$$\mathbf{u}_{\mathrm{MLSE}} = \left[\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_{N_I}\right] \cdot \mathrm{MLP}_{\mathrm{MLSE}}\left(\mathbf{I}_1 \oplus \mathbf{I}_2 \oplus \cdots \oplus \mathbf{I}_{N_I}\right), \tag{12}$$

where $\mathrm{MLP}_{\mathrm{MLSE}} : \mathbb{R}^{d_I} \to \mathbb{R}^{N_I}$ calculates the weights for each satisfaction level, and $\mathbf{u}_{\mathrm{MLSE}} \in \mathbb{R}^d$ is the aggregation of all the multi-level satisfaction groups of the user. Here $d_I = N_I \times d$.

4.4 Prediction and Training

4.4.1 Prediction. We obtain the final aggregated user representation by computing a weighted sum of the outputs $\mathbf{u}_{s+}$, $\mathbf{u}_{s-}$, and $\mathbf{h}_r$

Algorithm 1 The Optimization Process of SAQRec

Input: Recommendation dataset with questionnaire feedback $\mathcal{D}$;
Output: Propensity model ρ, satisfaction model g, main model f.

```
Initialization the parameters of the propensity model Θ_ρ, the satisfaction model Θ_g, the main model Θ_f.
while Θ_ρ not converged do                  ▷ Propensity Model Training
    Update Θ_ρ based on Eq. (5)
end while
while Θ_g not converged do                  ▷ Satisfaction Model Training
    Update Θ_g based on Eq. (6)
end while
while Θ_f not converged do                  ▷ Main Model Training
    Update Θ_f based on Eq. (19)
end while
```

from Section 4.3.2, along with $\mathbf{u}_{\text{MLSE}}$ obtained from Section 4.3.3, and the user embedding $\mathbf{v}_u$:

$$\begin{aligned}\mathbf{u}_f = \{ [\mathbf{u}_{s+}, \mathbf{u}_{s-}, \mathbf{h}_r, \mathbf{u}_{\text{MLSE}}, \mathbf{v}_u] \cdot \\ \text{MLP}_f \left(\mathbf{u}_{s+} \oplus \mathbf{u}_{s-} \oplus \mathbf{h}_r \oplus \mathbf{u}_{\text{MLSE}} \oplus \mathbf{v}_u \right) \},\end{aligned} \tag{13}$$

where $\text{MLP}_f : \mathbb{R}^{5d} \rightarrow \mathbb{R}^5$ computes the weight of each vector. $\mathbf{u}_f \in \mathbb{R}^d$ is the final aggregated user representation.

Our model consists of two outputs, $\hat{y}$ and $\hat{s}_f$, which denote the predicted click and satisfaction, respectively. Following previous works [36], $\hat{y}$ and $\hat{s}_f$ are computed as follows:

$$\hat{y} = \mathbf{u}_f \cdot \mathbf{v}_i, \quad \hat{s}_f = \mathbf{u}_f \cdot \text{MLP}_s(\mathbf{v}_i), \tag{14}$$

where $\text{MLP}_s : \mathbb{R}^d \rightarrow \mathbb{R}^d$ maps $\mathbf{v}_i$ to the space where the satisfaction is calculated.

4.4.2 Training. In the training process, the imputed satisfactions provided by the satisfaction model act as pseudo-labels for multi-task learning. These pseudo-labels assist the recommendation model in better understanding and capturing users' satisfaction with items, thereby improving its ability to align with users' true preferences.

For the predicted click, we utilize the widely used binary cross-entropy loss [44, 45] as the optimization objective function:

$$\mathcal{L}_C = -\frac{1}{|\mathcal{D}|} \sum_{\mathcal{D}} y \log(\hat{y}) + (1-y) \log(1-\hat{y}). \tag{15}$$

For the predicted satisfaction, as large amounts of interactions lack questionnaire feedback, we use imputed satisfaction provided by the pre-trained satisfaction model g as the pseudo-label to supervise the training process:

$$\mathcal{L}_S = -\frac{1}{|\mathcal{D}|} \sum_{\mathcal{D}} \hat{s} \log(\hat{s}_f) + (1-\hat{s}) \log(1-\hat{s}_f). \tag{16}$$

Furthermore, considering that SAQRec takes into account user historical information, as the model trains, the SAQRec's output $\hat{s}_f$ may become more accurate than the satisfaction model's output $\hat{s}$. Inspired by previous works [6, 21, 39], we employ the adaptive label correction technique, mixing $\hat{s}_f$ and $\hat{s}$ with a weighted sum to form a new satisfaction label $\hat{s}_{\text{mix}}$ that is more accurate at this point during the training process:

$$\begin{aligned}&\hat{s}_{\text{mix}} = \lambda \hat{s} + (1-\lambda)\hat{s}_f, \quad \lambda = \text{Beta}(\mathcal{L}_S), \\ &\mathcal{L}_S^{\text{mix}} = -\frac{1}{|\mathcal{D}|} \sum_{\mathcal{D}} \hat{s}_{\text{mix}} \log(\hat{s}_f) + (1-\hat{s}_{\text{mix}}) \log(1-\hat{s}_f),\end{aligned} \tag{17}$$

where $\text{Beta}(\cdot)$ represents the beta function. When $\mathcal{L}_S$ is smaller, indicating better satisfaction prediction by SAQRec, λ becomes smaller, and the weight of SAQRec's output $\hat{s}_f$ is larger. In the early training stages of SAQRec, we use $\mathcal{L}_S$, and after a certain number of epochs, we switch to $\mathcal{L}_S^{\text{mix}}$. The final loss $\mathcal{L}_S^{\text{final}}$ for the satisfaction output is:

$$\mathcal{L}_S^{\text{final}} = \begin{cases} \mathcal{L}_S, & \text{if } t < N_{\text{mix}}, \\ \mathcal{L}_S^{\text{mix}}, & \text{if } t \geq N_{\text{mix}}, \end{cases} \tag{18}$$

where N_{mix} is a hyper-parameter and t represents the current epoch step.

Considering the click loss in Eq. (15) and the satisfaction loss in Eq. (18), we employ multi-task learning to train our model end-to-end. The total loss for training our model is:

$$\mathcal{L} = \mathcal{L}_C + \beta \mathcal{L}_S^{\text{final}} + \gamma \|\Theta_f\|_2, \tag{19}$$

where β and γ are two hyper-parameters to control the weights of the satisfaction loss and the regularization term, respectively. And Θ_f denotes all of the parameters of SAQRec. The optimization process of SAQRec is summarized in Algorithm 1.

Table 1: Statistics of datasets used in this paper.

Dataset	#Users	#Items	#Interactions	#Questionnaires	
				Positive	Negative
Commercial	9,756	381,150	6,483,498	44,172	17,854
KuaiRand	7,270	142,255	1,795,008	53,343	10,064

5 EXPERIMENT

5.1 Experimental Setup

5.1.1 Dataset. We conduct experiments on the following two datasets: one is collected from a popular short video platform, and the other is based on the publicly available KuaiRand [10] dataset. The statistics of these datasets are shown in Table 1.

Commercial: The dataset is constructed based on behavior logs of 9,756 users who have questionnaire interactions on a popular short video platform over one week in 2023. It includes user historical clicks and questionnaire interactions. For data pre-processing, following common practices [16, 31], we group interaction records by the user, sorting them in ascending order based on timestamps, and filter out unpopular items and users with limited interactions using a 5-core approach.

KuaiRand[1] [10]: Due to the absence of a publicly available recommendation dataset containing both user clicks and questionnaire feedback, we modify the KuaiRand dataset to serve as a dataset with questionnaire feedback. We treat "is_like" and "is_hate" feedback in the KuaiRand dataset, which is user-selected, as questionnaire positive and negative feedback, respectively. We pre-process this dataset according to the method used for the Commercial dataset.

[1] https://kuairand.com/

Table 2: Performance comparisons between SAQRec and the baselines on two datasets. The best and second-best performance methods are denoted in bold and underlined fonts, respectively. * represents improvements over the second-best methods that are significant (p-value < 0.05 with t-test).

Dataset		Commercial						KuaiRand					
Category	Model	NDCG@5	NDCG@10	HR@1	HR@5	HR@10	MRR	NDCG@5	NDCG@10	HR@1	HR@5	HR@10	MRR
Sequential	STAMP	0.3386	0.3881	0.2045	0.4637	0.6168	0.3348	0.6122	0.6374	0.4330	0.7655	0.8422	0.5776
	Caser	0.4584	0.5015	0.2882	0.6126	0.7456	0.4371	0.6351	0.6570	0.4326	0.8072	0.8730	0.5919
	HGN	0.4048	0.4532	0.2453	0.5514	0.7011	0.3903	0.6164	0.6415	0.4186	0.7836	0.8604	0.5767
	NARM	0.4564	0.5027	0.2758	0.6196	0.7622	0.4334	0.6706	0.6897	0.4923	0.8175	0.8758	0.6342
	GRU4Rec	0.4743	0.5192	0.2944	0.6364	0.7746	0.4502	0.6665	0.6862	0.4868	0.8143	0.8740	0.6302
	SASRec	0.4895	0.5309	0.3205	0.6428	0.7706	0.4667	0.6837	0.7023	0.5171	0.8223	0.8791	0.6495
	FMLP-Rec	0.4560	0.5009	0.2790	0.6159	0.7544	0.4331	0.6726	0.6905	0.4890	0.8257	0.8802	0.6334
Muti-behavior (Questionnaire-aware)	DIPN	0.4166	0.4631	0.2624	0.5575	0.7012	0.4035	0.5811	0.6137	0.3956	0.7447	0.8436	0.5471
	DMT	0.4213	0.4654	0.2676	0.5616	0.6985	0.4072	0.6455	0.6756	0.4809	0.7919	0.8832	0.6145
	DFN	0.4209	0.4650	0.2669	0.5606	0.6969	0.4074	0.6449	0.6720	0.4637	0.8010	0.8834	0.6094
	FeedRec	0.4914	0.5327	0.3191	0.6449	0.7717	0.4685	0.6404	0.6592	0.4717	0.7806	0.8377	0.6067
	GRU4Rec$_M$	0.4800	0.5248	0.3047	0.6372	0.7745	0.4576	0.6615	0.6809	0.4751	0.8166	0.8763	0.6224
	SASRec$_M$	0.4960	0.5374	0.3198	0.6540	0.7813	0.4712	0.6860	0.7043	0.5147	0.8286	0.8842	0.6504
	FMLP-Rec$_M$	0.5114	0.5532	0.3368	0.6665	0.7951	0.4870	0.6866	0.7026	0.5066	0.8340	0.8828	0.6481
	SAQRec (Ours)	**0.5334***	**0.5718***	**0.3638***	**0.6839***	**0.8015***	**0.5091***	**0.7071***	**0.7221***	**0.5396***	**0.8444***	**0.8902***	**0.6714***

Following [16, 31], we utilize the *leave-one-out* strategy to partition both datasets for training the main model. For each user, we reserve the most recent click behavior for testing, the second most recent click behavior for validation, and all the remaining click behaviors for training.

Dataset for training the satisfaction model: We pre-train the unbiased satisfaction model in Section 4.2 using interactions with users who have both positive and negative questionnaire feedback.

5.1.2 Evaluation Metrics. We adopt several widely used metrics [16, 31], including Hit Ratio (HR), Normalized Discounted Cumulative Gain (NDCG), and Mean Reciprocal Rank(MRR). We report results using HR@ {1, 5, 10}, NDCG@{5, 10}, and MRR metrics. We pair the ground-truth items with 99 randomly sampled negative items that the user has not interacted with for evaluation. When calculating all metrics, we conduct statistics based on item rankings and report average results.

5.1.3 Baselines. To demonstrate the effectiveness of SAQRec, we compare it with two types of baselines as follows:

Sequential Recommendation Models: (1) **STAMP** [22]: It introduces short-term attention and long-term memory priority mechanisms for recommendation. (2) **Caser** [32]: It embeds the sequence of items recently interacted with by the user into an image, which is subsequently unfolded in both temporal and latent space. (3) **HGN** [24]: It utilizes a hierarchical gated network, incorporating feature gating and instance gating modules to capture both long-term and short-term user interests. (4) **NARM** [18]: It combines global and local encoders along with attention mechanisms to simultaneously capture the sequential behavior of users. (5) **GRU4Rec** [14]: It utilizes Recurrent Neural Networks to process user interaction sequences for recommendation. (6) **SASRec** [16]: It proposes a sequential recommendation model based on a self-attention mechanism. (7) **FMLP-Rec** [46]: It is an all-MLP model that enhances the performance of sequential recommendation by integrating learnable filters to reduce noise in user behavior data.

Multi-Behavior Sequential Recommendation Models: These models simultaneously consider user questionnaire satisfied behaviors, dissatisfied behaviors, and clicked behaviors. (8) **DIPN** [12]: It effectively combines data from different behaviors through a hierarchical attention mechanism to improve the accuracy of recommendations. (9) **DMT** [11]: It leverages multiple Transformers to model diverse user behavior sequences, capturing the behavioral diversity of users for recommendation. (10) **DFN** [38]: It enhances recommendations by combining explicit and implicit, positive and negative user feedback through a feedback interaction mechanism. (11) **FeedRec** [36]: It achieves more precise recommendations by integrating explicit and implicit user feedback using an attention network to refine user interests. Additionally, we enhance GRU4Rec, SASRec, and FMLP-Rec by sorting and mixing different behavior sequences based on timestamp, resulting in (12) **GRU4Rec$_M$**, (13) **SASRec$_M$**, and (14) **FMLP-Rec$_M$**.

5.1.4 Implementation Details. For all models, the maximum lengths for click history, satisfied history, and dissatisfied history are set to 100 (50), 25 (20), and 5 (5) on the Commercial (KuaiRand) dataset, respectively. The batch size is set to 1024 and 512 for the Commercial and KuaiRand datasets, respectively. We use the Adam [17] optimizer with a learning rate of 0.001 and employ early stopping during training to prevent over-fitting. The source code and other details can be found at https://github.com/ke-01/SAQRec.

5.2 Overall Performance

We compare SAQRec with the aforementioned baselines on Commercial and KuaiRand datasets, as shown in Table 2. From the results, we can observe that:

(1) Firstly, SAQRec achieves the best results across the two datasets. It outperforms existing state-of-the-art models and passes statistical significance tests. This indicates the effectiveness of SAQRec in leveraging the pre-trained satisfaction model to address data sparsity and selection bias issues and further demonstrates the effectiveness of satisfaction feature alignment and utilizing imputed satisfactions from the satisfaction model for multi-task learning in modeling users' true preferences.

(2) Secondly, compared to traditional sequential recommendation models, SAQRec and some multi-behavior methods obtain better results. This demonstrates the effectiveness of incorporating questionnaire feedback in recommendation models. Questionnaire feedback can reflect users' true preferences, aiding in more accurate modeling of user behaviors.

Table 3: Ablation studies of SAQRec on the Commercial dataset. Unbiased is short for Unbiased Satisfaction Model Pre-Training. SID is short for Satisfaction Information Disentanglement. MLSE is short for Multi-Level Satisfaction Enhancement. MTL is short for Multi-Task Learning.

Model	N@5	N@10	H@1	H@5	H@10	MRR
SAQRec	**0.5334**	**0.5718**	**0.3638**	**0.6839**	**0.8015**	**0.5091**
w/o Unbiased	0.5197	0.5597	0.3515	0.6707	0.7937	0.4963
w/o SID	0.5141	0.5541	0.3439	0.6659	0.7886	0.4903
w/o MLSE	0.5238	0.5621	0.3563	0.6736	0.7917	0.4998
w/o MTL	0.5248	0.5638	0.3566	0.6754	0.7954	0.5007

(3) Finally, we can also observe that not all multi-behavior models surpass sequential recommendation models. And among multi-behavior models, models that directly integrate questionnaire and click feedback into a strong sequential framework outperform those that treat questionnaire feedback as regular behavior. This indicates that simply treating questionnaire feedback as regular behavior in multi-behavior modeling is not only suboptimal but could also lead to adverse effects. Instead, it is crucial to consider the challenges associated with questionnaire feedback, such as data sparsity and selection bias issues. And effectively leveraging the characteristics of questionnaire feedback that can reflect users' true preferences is crucial for recommendation models.

5.3 Ablation Study

SAQRec comprises four key components. To investigate how different components influence the performance of SAQRec, we conduct experiments by removing each module at a time from the overall model. Table 3 shows the ablation results on the commercial dataset. We will delve into a detailed discussion of each component:

Unbiased Satisfaction Model Pre-Training (Unbiased): We employ Eq. (3) rather than Eq. (5) for training the satisfaction model to validate unbiased training. It is evident that the absence of unbiased training significantly impacts model performance due to selection bias. Therefore, adopting an unbiased training approach helps improve the generalization ability of the satisfaction model, leading to more reliable satisfaction estimates and guiding the recommendation model's feature representation and training process for better alignment.

Satisfaction Information Disentanglement (SID): This module disentangles satisfaction and dissatisfaction information from the users' click history using historical questionnaire feedback. Observations indicate that SID significantly contributes to the final predictions, emphasizing the benefit of introducing historical questionnaire feedback to uncover users' preferences.

Multi-level Satisfaction Enhancement (MLSE): This module categorizes click history into distinct satisfaction groups based on imputed satisfaction provided by the pre-trained satisfaction model. The performance decline emphasizes the importance of leveraging imputed satisfaction for a more nuanced understanding of users' preferences. Despite SID already disentangles user click history into satisfaction and dissatisfaction, the model can further capture users' preferences in detail through MLSE.

Multi-task Learning (MTL): As shown in Eq. (19), SAQRec employs imputed satisfaction provided by the pre-trained satisfaction model as the pseudo-label for multi-task learning. It can be observed that without multi-task learning, optimizing only the click loss $\mathcal{L}_C$ in Eq. (15) leads to a decrease in performance. This indicates that multi-task learning enables the model to leverage imputed satisfaction information, enhancing click predictions and better aligning the model's output with users' true preferences.

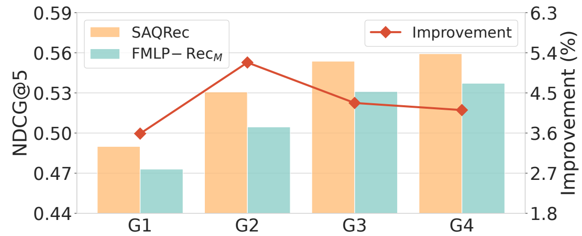


Figure 3: Performance comparison of SAQRec over user groups with different activity levels. The histograms represent the percentage of improvement (%) over FMLP-Rec$_M$ and the lines denote the performance. The X-axis represents grouping by user activity levels from low to high, which are G1, G2, G3, and G4, respectively.

5.4 Experimental Analysis

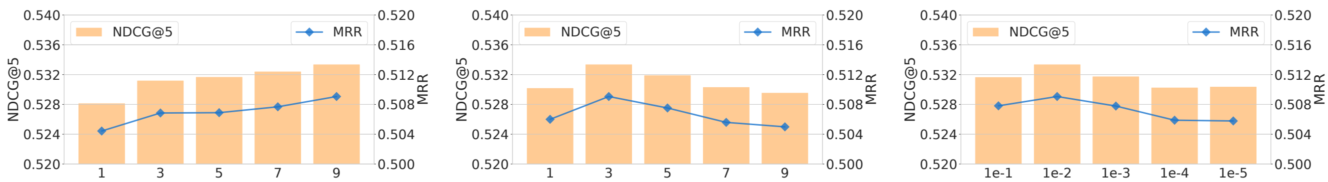
In this section, we conduct an experimental analysis on the commercial dataset for SAQRec.

5.4.1 Performance Improvement w.r.t. Different User Groups. In this section, we explore the performance improvements of SAQRec relative to FMLP-Rec$_M$ for users with different levels of activity. Specifically, we categorize users into four groups, G1, G2, G3, and G4, based on the number of their interactions, ranging from fewer to more. The results are shown in Figure 3. We can draw the following conclusions: SAQRec consistently outperforms the best-performing baseline, FMLP-Rec$_M$, across users with different levels of activity. Additionally, it can be observed that SAQRec achieves more significant performance improvements for users with midium to high activity levels. This is because when the historical interaction length is shorter, users may exhibit random behavior patterns that are difficult to capture. As the historical length increases, SAQRec can effectively capture users' true preferences with more interactions, thus achieving relatively stable performance gains.

5.4.2 Impact of Hyper-parameters. In this section, we analyze the impact of different hyper-parameters on the model performance.

Performance Comparison w.r.t. the Numbers of Satisfaction Levels. We analyze how the number of user satisfaction levels (N_I) in MLSE impacts SAQRec's performance (see Figure 4(a)). The model performs better as satisfaction levels increase from 1 to 9, indicating an improved understanding of users' preferences at a finer granularity. Considering the potential for increased complexity, we choose to use only nine levels.

Performance Comparison w.r.t. Different Epoch for Label Correction. We enhance the model's learning of satisfaction by introducing adaptive label correction at epoch N_{mix}, as defined in Eq. (18). Figure 4(b) shows that introducing label correction at the third epoch achieves optimal effects. Performance gradually improves before this point but declines afterward as the model tends to be overfitted, relying excessively on the imputed satisfaction.



(a) Performance comparison w.r.t. different N_I. (b) Performance comparison w.r.t. different N_{mix}. (c) Performance comparison w.r.t. different β.

Figure 4: Effects of hyper-parameters N_I, N_{mix} and β. NDCG@5 is on the left axis, and MRR is on the right axis.

Performance Comparison w.r.t. Different Weight of the Satisfaction Loss. We implement multi-task learning with the hyper-parameter β (Eq. (19)) to balance click and satisfaction prediction tasks. Figure 4(c) illustrates the impact of varying β from 1e-5 to 1e-1 on model performance. A smaller β reduces the significance of the satisfaction prediction task, weakening its influence on guiding the click prediction task. This leads to a decline in performance, highlighting the importance of leveraging imputed satisfactions as pseudo-labels to better align with users' true preferences.

5.5 Online A/B Testing

To comprehensively validate SAQRec's performance in our scenario, we conduct a consecutive 7-day online A/B testing on Kuaishou APP. The test, utilizing 10% of the primary traffic through random bucketing, includes a control group using the platform's advanced base model and an experimental group combining the base model with SAQRec. Specific results are presented in Table 4, demonstrating positive gains in watch time and positive-to-negative feedback ratio across all seven days compared to the base model. Specifically, (1) The average user watch time increases by 0.124%, and the average video watch time increases by 0.509%, signifying the improved effectiveness of the recommendation model in understanding users' preferences. This enables the delivery of more personalized and user-preference-aligned recommendations, thereby enhancing platform satisfaction and encouraging users to spend more time on the platform. (2) The positive-to-negative feedback ratio for video content increases by 0.333%, indicating a relative rise in users' positive feedback compared to negative feedback. This enhancement indicates user preference for positive behaviors (e.g., sharing, liking, and commenting), reflecting an increase in user satisfaction.

Table 4: Online A/B results on Kuaishou APP. AUT is short for the average user watch time. AVT is short for average video watch time. P2N is short for positive-to-negative feedback ratio. "↑" denotes that a higher value of the corresponding metric is better. In Kuaishou's scenario, a 0.1% increase in watch time is considered a significant improvement, potentially yielding substantial business gain.

Metrics	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
AUT ↑	+0.085%	+0.107%	+0.119%	+0.136%	+0.098%	+0.157%	+0.164%
AVT ↑	+0.509%	+0.594%	+0.328%	+0.556%	+0.512%	+0.564%	+0.499%
P2N ↑	+0.344%	+0.405%	+0.238%	+0.395%	+0.336%	+0.314%	+0.300%

6 CONCLUSION

In this paper, we propose SAQRec, which enhances recommendations by leveraging questionnaire feedback to align with users' true preferences. SAQRec has two stages: (1) The first stage pre-trains an unbiased satisfaction model to address the issues of data sparsity and selection bias; (2) The second stage aligns the recommendation model with user preferences through the satisfaction feature alignment module, which includes satisfaction information disentanglement and multi-level satisfaction enhancement. The imputed satisfactions are further used as pseudo-labels for multi-task learning. The results of offline and online experiments validate the effectiveness of SAQRec in enhancing user satisfaction.

APPENDIX

This section provides the proofs of Theorem 4.1.

Proof.

$$
\begin{aligned}
&\mathbb{E}_O\left[\mathcal{L}_{\text{unbias}}(\hat{s})\right] \\
&= -\mathbb{E}_O\left[\frac{1}{|O|}\sum_{O}\frac{s\log(\hat{s})+(1-s)\log(1-\hat{s})}{\Pr(o=1)}\right] \\
&= -\mathbb{E}_O\left[\frac{1}{|\mathcal{D}|}\sum_{\mathcal{D}}\frac{s\log(\hat{s})+(1-s)\log(1-\hat{s})}{\Pr(o=1)}o\right] \\
&= -\frac{1}{|\mathcal{D}|}\sum_{\mathcal{D}}\frac{s\log(\hat{s})+(1-s)\log(1-\hat{s})}{\Pr(o=1)}\mathbb{E}_O\left[o\right] \\
&= -\frac{1}{|\mathcal{D}|}\sum_{\mathcal{D}}\frac{s\log(\hat{s})+(1-s)\log(1-\hat{s})}{\Pr(o=1)}\Pr(o=1) = \mathcal{L}_{\text{ideal}}(\hat{s})\,.
\end{aligned}
$$

□

ACKNOWLEDGMENTS

This work was funded by the National Science and Technology Major Project (2022ZD0114802), the National Natural Science Foundation of China (No. 62376275, 62377044), Intelligent Social Governance Interdisciplinary Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China. Supported by fund for building world-class universities (disciplines) of Renmin University of China. Supported by Public Computing Cloud, Renmin University of China. Supported by the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China (23XNKJ13). Supported by Kuaishou Technology.

REFERENCES

[1] Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. 2021. Sequential recommendation with graph neural networks. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 378–387.

[2] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2023. Bias and debias in recommender system: A survey and future directions. *ACM Transactions on Information Systems* 41, 3 (2023), 1–39.

[3] Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed H Chi. 2019. Top-k off-policy correction for a REINFORCE recommender system. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 456–464.

[4] Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, and Wenwu Ou. 2019. Behavior sequence transformer for e-commerce recommendation in alibaba. In *Proceedings of the 1st international workshop on deep learning practice for high-dimensional sparse data*. 1–4.

[5] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.

[6] Quanyu Dai, Yalei Lv, Jieming Zhu, Junjie Ye, Zhenhua Dong, Rui Zhang, Shu-Tao Xia, and Ruiming Tang. 2022. LCD: Adaptive Label Correction for Denoising Music Recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 3903–3907.

[7] Sunhao Dai, Changle Qu, Sirui Chen, Xiao Zhang, and Jun Xu. 2024. ReCODE: Modeling Repeat Consumption with Neural ODE. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2599–2603.

[8] Sunhao Dai, Ninglu Shao, Jieming Zhu, Xiao Zhang, Zhenhua Dong, Jun Xu, Quanyu Dai, and Ji-Rong Wen. 2024. Modeling User Attention in Music Recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*.

[9] Sunhao Dai, Yuqi Zhou, Jun Xu, and Ji-Rong Wen. 2023. Dually Enhanced Delayed Feedback Modeling for Streaming Conversion Rate Prediction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 390–399.

[10] Chongming Gao, Shijun Li, Yuan Zhang, Jiawei Chen, Biao Li, Wenqiang Lei, Peng Jiang, and Xiangnan He. 2022. KuaiRand: An Unbiased Sequential Recommendation Dataset with Randomly Exposed Videos. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 3953–3957.

[11] Yulong Gu, Zhuoye Ding, Shuaiqiang Wang, Lixin Zou, Yiding Liu, and Dawei Yin. 2020. Deep multifaceted transformers for multi-objective ranking in large-scale e-commerce recommender systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2493–2500.

[12] Long Guo, Lifeng Hua, Rongfei Jia, Binqiang Zhao, Xiaobo Wang, and Bin Cui. 2019. Buying or browsing?: Predicting real-time purchasing intent using attention-based deep network with multiple behavior. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 1984–1992.

[13] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 355–364.

[14] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[15] Jiaming Ji, Tianyi Qiu, Boyuan Chen, Borong Zhang, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, Jiayi Zhou, Zhaowei Zhang, et al. 2023. Ai alignment: A comprehensive survey. *arXiv preprint arXiv:2310.19852* (2023).

[16] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.

[17] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. http://arxiv.org/abs/1412.6980

[18] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1419–1428.

[19] Yongqi Li, Meng Liu, Jianhua Yin, Chaoran Cui, Xin-Shun Xu, and Liqiang Nie. 2019. Routing micro-videos via a temporal graph-guided recommendation system. In *Proceedings of the 27th ACM International Conference on Multimedia*. 1464–1472.

[20] Zeyu Li, Wei Cheng, Yang Chen, Haifeng Chen, and Wei Wang. 2020. Interpretable click-through rate prediction through hierarchical attention. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 313–321.

[21] Dugang Liu, Pengxiang Cheng, Zhenhua Dong, Xiuqiang He, Weike Pan, and Zhong Ming. 2020. A general knowledge distillation framework for counterfactual recommendation via uniform data. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 831–840.

[22] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: short-term attention/memory priority model for session-based recommendation. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1831–1839.

[23] Shang Liu, Zhenzhong Chen, Hongyi Liu, and Xinghai Hu. 2019. User-video co-attention network for personalized micro-video recommendation. In *The world wide web conference*. 3020–3026.

[24] Chen Ma, Peng Kang, and Xue Liu. 2019. Hierarchical gating networks for sequential recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 825–833.

[25] Yabo Ni, Dan Ou, Shichen Liu, Xiang Li, Wenwu Ou, Anxiang Zeng, and Luo Si. 2018. Perceive your users in depth: Learning universal user representations from multiple e-commerce tasks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 596–605.

[26] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems* 35 (2022), 27730–27744.

[27] Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. 2020. Unbiased recommender learning from missing-not-at-random implicit feedback. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 501–509.

[28] Harry B Santoso, Martin Schrepp, Lintang M Hasani, Rian Fitriansyah, and Arief Setyanto. 2022. The use of User Experience Questionnaire Plus (UEQ+) for cross-cultural UX research: evaluating Zoom and Learn Quran Tajwid as online learning tools. *Heliyon* 8, 11 (2022).

[29] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. In *international conference on machine learning*. PMLR, 1670–1679.

[30] Hyorim Shin, Changhoon Oh, Na Young Kim, Hobin Choi, Boyoung Kim, and Yong Gu Ji. 2024. Evaluating and eliciting design requirements for an improved user experience in live-streaming commerce interfaces. *Computers in Human Behavior* 150 (2024), 107990.

[31] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.

[32] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 565–573.

[33] Yu Tian, Jianxin Chang, Yanan Niu, Yang Song, and Chenliang Li. 2022. When multi-level meets multi-interest: A multi-grained neural model for sequential recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1632–1641.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[35] Zhenlei Wang, Shiqi Shen, Zhipeng Wang, Bo Chen, Xu Chen, and Ji-Rong Wen. 2022. Unbiased Sequential Recommendation with Latent Confounders. In *Proceedings of the ACM Web Conference 2022*. 2195–2204.

[36] Chuhan Wu, Fangzhao Wu, Tao Qi, Qi Liu, Xuan Tian, Jie Li, Wei He, Yongfeng Huang, and Xing Xie. 2022. Feedrec: News feed recommendation with various user feedbacks. In *Proceedings of the ACM Web Conference 2022*. 2088–2097.

[37] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 346–353.

[38] Ruobing Xie, Cheng Ling, Yalong Wang, Rui Wang, Feng Xia, and Leyu Lin. 2021. Deep feedback network for recommendation. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*. 2519–2525.

[39] Zixuan Xu, Penghui Wei, Weimin Zhang, Shaoguo Liu, Liang Wang, and Bo Zheng. 2022. Ukd: Debiasing conversion rate estimation via uncertainty-regularized knowledge distillation. In *Proceedings of the ACM Web Conference 2022*. 2078–2087.

[40] Shengyu Zhang, Lingxiao Yang, Dong Yao, Yujie Lu, Fuli Feng, Zhou Zhao, Tat-seng Chua, and Fei Wu. 2022. Re4: Learning to re-contrast, re-attend, re-construct for multi-interest recommendation. In *Proceedings of the ACM Web Conference 2022*. 2216–2226.

[41] Xiao Zhang, Sunhao Dai, Jun Xu, Zhenhua Dong, Quanyu Dai, and Ji-Rong Wen. 2022. Counteracting user attention bias in music streaming recommendation via reward modification. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2504–2514.

[42] Xiao Zhang, Haonan Jia, Hanjing Su, Wenhan Wang, Jun Xu, and Ji-Rong Wen. 2021. Counterfactual reward modification for streaming recommendation with delayed feedback. In *Proceedings of the 44th International ACM SIGIR Conference*

on Research and Development in Information Retrieval. 41–50.

[43] Haiyuan Zhao, Lei Zhang, Jun Xu, Guohao Cai, Zhenhua Dong, and Ji-Rong Wen. 2023. Uncovering User Interest from Biased and Noised Watch Time in Video Recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 528–539.

[44] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep interest evolution network for click-through rate prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 5941–5948.

[45] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1059–1068.

[46] Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Filter-enhanced MLP is all you need for sequential recommendation. In *Proceedings of the ACM web conference 2022*. 2388–2399.