Learning and Optimization of Implicit Negative Feedback for Industrial Short-video Recommender System

Yunzhu Pan*†
University of Electronic Science and Technology of China, Chengdu, China

Nian Li*, Chen Gao‡
Beijing National Research Center for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing, China

Jianxin Chang, Yanan Niu, Yang Song
Beijing Kuaishou Technology Co., Ltd., Beijing, China

Depeng Jin, Yong Li
Beijing National Research Center for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing, China

ABSTRACT

Short-video recommendation is one of the most important recommendation applications in today's industrial information systems. Compared with other recommendation tasks, the enormous amount of feedback is the most typical characteristic. Specifically, in short-video recommendation, the easiest-to-collect user feedback is from the *skipping* behaviors, which leads to two critical challenges for the recommendation model. First, the skipping behavior reflects implicit user preferences, and thus it is challenging for interest extraction. Second, the kind of special feedback involves multiple objectives, such as total watching time, which is also very challenging. In this paper, we present our industrial solution in Kuaishou[1], which serves billion-level users every day. Specifically, we deploy a feedback-aware encoding module which well extracts user preference taking the impact of context into consideration. We further design a multi-objective prediction module which well distinguishes the relation and differences among different model objectives in the short-video recommendation. We conduct extensive online A/B testing, along with detailed and careful analysis, which verifies the effectiveness of our solution.

CCS CONCEPTS

• **Information systems** → **Recommender systems**;

KEYWORDS

Implicit Negative Feedback; Short-video Recommendation; Industrial Recommender System

ACM Reference Format:
Yunzhu Pan, Nian Li, Chen Gao, Jianxin Chang, Yanan Niu, Yang Song, Depeng Jin, Yong Li. 2023. Learning and Optimization of Implicit Negative Feedback for Industrial Short-video Recommender System. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3583780.3615482

[1]www.kuaishou.com

*Both authors contributed equally to this research.
†Work done when interning at Tsinghua University.
‡Chen Gao is the corresponding author (chgao96@gmail.com).

1 INTRODUCTION

Short-video platforms such as TikTok[2] and Kuaishou have achieved great success, attracting billions of users and spending hours watching videos every day. Distinct from traditional online recommender systems like news and e-commerce, building an effective short-video recommender system poses a new and challenging task due to its special interaction manner. In traditional recommender systems, user-item interaction feedback can be categorized into explicit feedback [29] and implicit feedback [14, 33]. Explicit feedback mainly refers to the feedback that directly conveys whether the user likes the item or not, such as movie ratings on a scale of 1-5; implicit feedback refers to binary click data in most applications and only reveals implicit user preferences, not fully describing whether the user likes or dislikes the item.

Despite the success of various recommendation models that try to teach users implicit or explicit feedback [15, 32, 34, 38, 39], the short-video recommendation is required to resolve the different feedback. Specifically, the short videos are exposed (*i.e.*, recommended) continuously, which leads to a critical challenge that the user feedback is different from existing recommenders, such as the clicks/purchases in e-commerce websites or ratings in movie websites. That is to say, in such a streaming-form interaction manner of recommender systems, existing methods are not suitable for learning preferences from user feedback. Moreover, the user-item interaction that is easiest to collect is the *implicit negative feedback*. Specifically, users can choose to skip over the recommended video, which is always the only feedback we collect for the video since users seldom choose to like or dislike a video, not to mention commenting, sharing, or other behaviors. Thus, learning from the new form of implicit negative feedback is an important but less-explored problem, which is suffering from two major challenges as follows.

- **Extracting preference signal from the feedback.** The users' behaviors will be fuzzier if most of them are just implicit skipping-over feedback, which leads to unclear user preference. While preference learning is fundamental to accurate recommendations, learning from plenty of new implicit feedback with a few other types of behaviors is challenging.
- **Complex objectives for different feedback are involved.** The implicit skipping-over feedback in short-video recommendations

[2]www.tiktok.com

makes the recommender systems involve multiple optimization goals, such as skipping rate minimization, watch time maximization, etc. Thus, the model optimization toward multiple fused objectives is also challenging.

In this paper, we introduce our industrial solution at Kuaishou to address these challenges. Serving billion-level users daily, it's a pivotal component of the entire recommendation engine. First, we deploy a feedback-aware sequential encoding module that can introduce multiple feedback, including negative feedback, into the existing context feature extraction. Given the impact of context on the users' decision-making process, the encoding module extracts the reference signal in a context-aware manner. Second, we deploy a parameter-sharing multi-objective prediction module, which can take context and user/item embeddings as input and well balance the different optimization goals during the prediction. The shared parameters reveal the relatedness among the goals, while the objective-specific parameters reveal the differences. In short, our proposed solution can address the above-mentioned critical challenges, providing accurate, personalized recommendations to users of short-video platforms. The proposed system is sufficiently evaluated in a real-world deployment.

To summarize, the contribution of this work is as follows.

- To our knowledge, we take the early step in approaching the problem of learning and optimization for the new form of implicit negative feedback in industrial short-video recommendations.
- We develop a system that addresses critical and unresolved challenges in preference learning and model optimization. The feedback-aware encoding module well extracts user preference taking the impact of context into consideration. The multi-objective prediction module well distinguishes the relation and differences among different model objectives in the short-video recommendation.
- We deployed our system for billion-level daily users, and extensive A/B testing confirmed its effectiveness.

2 THE PROPOSED SYSTEM

We present the deployed system for the learning and optimization of the new-type negative feedback in Kuaishou. Our proposed system serves in the ranking phase, producing the final recommendation list. The system includes three components as follows,

- **Feedback-aware sequential encoder**. We deploy the sequential encoder which can extract preferences from the mixed user feedback collected from the Kuaishou App in real-time.
- **Context feature embedding layer**. We develop a feature embedding layer that can utilize context information to filter useful prediction signals from sequential embeddings.
- **Multi-feedback prediction layer**. We deploy a multi-task prediction layer with shared parameters to predict multiple types of feedback, especially for the new type of negative feedback.

2.1 Feedback-aware Sequential Encoder

User feedback, encompassing both positive and negative types, is vital for modeling user preferences. In the real world, negative feedback is often implicit; users skip disliked content without explicitly expressing their aversion. This subtlety complicates user modeling. To tackle this, we've designed a feedback-aware encoder that integrates implicit negative feedback with two kinds of positive feedback. We use video watch time to define each feedback, including **E**ngaged **V**ideo **V**iewing(**EVV**), **F**ocused **V**ideo **V**iewing(**FVV**), and **G**lance **V**ideo **V**iewing(**GVV**), showed in Table 1.

Table 1: Feedback Types: EVV (positive feedback); FVV (stronger positive than EVV); GVV (negative feedback).

Feedback Type	Description
EVV	Watch time > 50% of other users.
FVV	Watch time > 60% of other users.
GVV	Watch time < 3 seconds.

2.1.1 Embedding Layer. Let $\mathcal{U}$ and $\mathcal{I}$ denote the sets of users and items, respectively. Given a user $u \in \mathcal{U}$, its history behavior sequences is denoted as $\mathbf{S}^u = \{s_1^u, s_2^u, \cdots, s_{|\mathbf{S}^u|}^u\}$. First, we build an item embedding matrix $\boldsymbol{M} \in \mathbb{R}^{|\mathcal{I}|\times d}$, where d denotes the dimension size, and the retrieved corresponding item embedding sequence is as follows,

$$\boldsymbol{E}^u = (\boldsymbol{e}_1^u, \boldsymbol{e}_2^u, \cdots, \boldsymbol{e}_{|\mathbf{S}^u|}^u). \tag{1}$$

Since each behavior belongs to multiple types of feedback, we introduce another sequence R_u^w to represent each feedback type, EVV, FVV, and GVV, respectively.

$$\boldsymbol{R}_u^w = \{r_{u,1}^w, r_{u,2}^w, \cdots, r_{u,|\mathbf{S}^u|}^w\}, w \in \{e, f, g\}, \tag{2}$$

where $r_{u,i}^w$ is either 1 or 0, corresponding to True and False.

2.1.2 Self-Attention Encoder. To capture the relationships between each user behavior in the sequence, we employ a self-attention encoder. Initially, we concatenate the feedback $\mathbf{R}_u^w$ in Eqn (2) with the item embedding $\boldsymbol{E}^u$ and integrate a learnable position embedding $\mathbf{p}$ for position information. This forms the self-attention encoder input $\hat{\boldsymbol{E}}^u \in \mathbb{R}^{L\times(d+3)}$, where L denotes the sequence length:

$$\hat{\boldsymbol{E}}^u = [\boldsymbol{E}^u; \boldsymbol{R}_u^e; \boldsymbol{R}_u^f; \boldsymbol{R}_u^g] + \mathbf{p}. \tag{3}$$

The core self-attention mechanism is defined as:

$$\text{SelfAttention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V}, \tag{4}$$

Each block of the self-attention encoder is constructed as:

$$\begin{aligned} \mathbf{H}' &= \text{LayerNorm}\Big(\text{SelfAttention}\left(\mathbf{X}\mathbf{W}^{\mathbf{q}}, \mathbf{X}\mathbf{W}^{\mathbf{k}}, \mathbf{X}\mathbf{W}^{\mathbf{v}}\right) + \mathbf{X}\Big), \\ \mathbf{H} &= \text{LayerNorm}\left(\text{FFN}\left(\mathbf{H}'\right) + \mathbf{H}'\right), \end{aligned} \tag{5}$$

Here, the projection matrices $\mathbf{W}^{\mathbf{q}}, \mathbf{W}^{\mathbf{k}}, \mathbf{W}^{\mathbf{v}} \in \mathbb{R}^{(d+3)\times(d+3)}$. $\mathbf{X}$ denotes the output from the preceding layer, with the first layer set to $\hat{\boldsymbol{E}}^u$. After processing through the self-attention encoder's final layer, we derive the outputs $\boldsymbol{e}_h$, representing the user's sequential behavior history.

2.2 Context Feature Embedding Layer

Besides the user history behavior sequence, context information (e.g., user demographics, item attributes, and platform type) also plays a significant role in learning user preferences. For example, young users (*age* attribute of user) may prefer to watch videos about electronics (*category* attributes of video) on the Web platform for a longer time (*platform* type). Hence, we utilize a feature interaction layer to incorporate the context information, including three phases: feature construction, feature transformation, and deep feature interaction.

2.2.1 Feature construction. Let $\boldsymbol{e}_u, \boldsymbol{e}_i, \boldsymbol{e}_p$ be the retrieved embedding of the user, target item, and corresponding platform. We also encode the user's profile information, e.g., age and location, and the video category feature to be the input embedding $\boldsymbol{E}_{in} \in \mathbb{R}^{D\times 1}$, where D is the size of the all concatenated features.

2.2.2 Feature transformation. Context's impact on user behaviors depends on the specific user, item, and platform. We design an embedding transform layer to learn the projection relationship between input embedding $\boldsymbol{E}_{\text{in}}$ and specific context information $\boldsymbol{e}_u$, $\boldsymbol{e}_i$, $\boldsymbol{e}_p$. Specifically, we first build a transformation layer as follows,

$$\mathbf{E}_{\text{trans}}^{(u,i,p)} = \mathbf{E}_{\mathbf{in}}\mathbf{W}_1 \cdot [\boldsymbol{e}_u; \boldsymbol{e}_i; \boldsymbol{e}_p], \tag{6}$$

where the output $\mathbf{E}_{\text{trans}}^{(u,i,p)}$ has the same size of $\mathbf{E}_{\mathbf{in}}$. Here superscript (u, i, p) means the embedding transformation depends on a specific user, item, and platform. That is, $\mathbf{E}_{\text{trans}}^{(u,i,p)}$ contains useful prediction signals, and useless ones are filtered out.

2.2.3 Feature importance alignment. After we obtain $\mathbf{E}_{\text{trans}}^{(u,i,p)}$, which has filtered out useless signals, the importance of different features still varies. For example, item category is always the most important item feature. Specifically, the feature importance of learning can be at different granularity. The finest-grained importance can be dimension-aware, *i.e.*, assigning a weight value to each dimension of $\mathbf{E}_{\text{trans}}^{(u,i,p)}$. The coarsest-grained importance actually refers to assigning a universal weight to all dimensions, not distinguishing the different feature importance.

To achieve adaptive-grained importance learning, we divide $\mathbf{E}_{\text{trans}}^{(u,i,p)}$ to S slots (the size of each slot is D/S and S is a hyper-parameter) and assign each slot different important weights (D/S dimensions inner one slot share the same weight). We then deploy a weight learning component as follows,

$$\boldsymbol{\alpha} = \text{sigmoid}(\mathbf{W}_{\mathbf{slot}}[\mathbf{E}_{\text{trans}}^{(u,i,p)}; \boldsymbol{e}_u; \boldsymbol{e}_i; \boldsymbol{e}_p]), \tag{7}$$

where $\mathbf{W}_{\mathbf{slot}} \in \mathbb{R}^{S\times 4D}$ is a learnable parameter.

With the learned slot-level feature importance, we combine it with $\mathbf{E}_{\text{trans}}^{(u,i,p)}$ to generate the final feature embeddings as follows:

$$\widetilde{\mathbf{E}}^{(u,i,p)} = \boldsymbol{\alpha} \otimes \mathbf{E}_{\text{trans}}^{(u,i,p)}, \tag{8}$$

where $\otimes$ denotes the element-wise product.

2.3 Multi-feedback Prediction Layer

In real-world recommendation engines, the presence of multiple behaviors poses a multi-objective optimization challenge. For example, a model is optimized to maximize the watch time and minimize the skip ratio at the same time. To address this, we deploy a multi-feedback prediction layer, which takes both user representation $\boldsymbol{e}_h$, *i.e.*, collaborative filtering signal, and context feature $\widetilde{\mathbf{E}}^{(u,i,p)}$ in Eqn (8) as input. That is, it helps learn the behavioral pattern in both historical behaviors and side information. For clearer presentation, here we denote it as $\mathbf{x} = [\widetilde{\mathbf{E}}^{(u,i,p)}; \boldsymbol{e}_h]$.

Since multiple behaviors are different but related, we propose a parameter-sharing framework. We first build N multiple deep

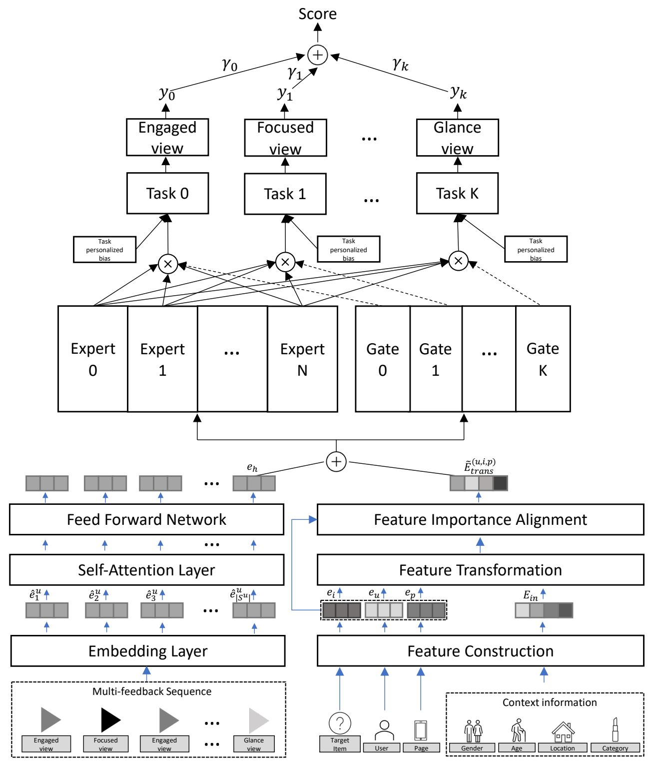


Figure 1: Illustration of our proposed system.

neural networks, inspired by [25], serving as the shared base network module, denoted as $f_i = \text{DNN}_i, i = 1, 2, \cdots, N$, where N is a hyper-parameter. Then for K objective-optimization tasks, the corresponding neural network can be a weighted combination of base neural networks. In our experimental setup, we assigned the value of K as 3, representing the three distinct feedback types: EVV, FVV, and GVV. Specifically, for the k-th task, it is formulated as $\mathbf{o}_k = \sum_{i=1}^{N} g_i^k f_i(\mathbf{x})$, where g_i^k is the i-th value of the weight vector $\mathbf{g}^k$, calculated with a gate network as $\mathbf{g}^k = \text{softmax}(\mathbf{W}_k^{\mathbf{gate}}\mathbf{x})$, where $\mathbf{W}_k^{\mathbf{gate}}$ is a trainable parameter.

Based on combining the gate network and base deep neural networks, we obtain the output of each task, $\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_K$. We then deploy a prediction tower for each task with deep neural networks to obtain the predicted values as $\mathbf{y}_k = \text{sigmoid}(\mathbf{W}_k^{\mathbf{pred}}(\mathbf{o}_k) + b_k)$, where $\mathbf{W}_k^{\mathbf{pred}}$ and b_k are learnable parameters.

2.4 Training and Serving

2.4.1 Joint training. The proposed model is optimized in an end-to-end manner, with joint training on K tasks. Specifically, the training data includes the multi-feedback data denoted as $\mathbb{Y} = \{\mathbb{Y}_1, \mathbb{Y}_2, \cdots, \mathbb{Y}_K\}$. For the training data of k-th feedback, $\mathbb{Y}_k$, it includes the collected positive samples (observed feedback), $\mathbb{Y}_k^+$, and the negative samples randomly selected from unobserved items, $\mathbb{Y}_k^-$. Then the loss function corresponding to the k-th feedback can be formulated as follows,

$$\mathbb{L}_k = -\mathbb{Y}_k^+ \log \mathbf{y}_k - (1 - \mathbb{Y}_k^-) \log(1 - \mathbf{y}_k), \tag{9}$$

where *Logloss* can also be replaced by other loss functions such as BPR loss (Bayesian Personalized Ranking) [33].

Given the importance of different feedback, in the training process, we adaptively balance tasks with weights to obtain the overall loss function as $\mathbb{L} = \sum_{k=1}^{K} \lambda_k \mathbb{L}_k$, where we have $\sum_{k=1}^{K} \lambda_k = 1$

2.4.2 Fusion strategy for online serving. Our proposed method serves in the ranking phase of the real-world recommender system, of which the final output is the ranking list exposed to the users. For the online serving in the real system, we adopt a simple yet effective fusion strategy as **score** $= \sum_{k=1}^{K} \gamma_k \mathbf{y}_k$, where the selection γ_k depends on the real-world requirements; for example, the γ_k value for negative feedback should be a negative number. With the fused score, we generate the L-length ranking list from the candidate pool (generated by the previous Recall Phase in the recommendation engine) with the L highest scores.

3 ONLINE EXPERIMENTS

In this section, we validate our user negative feedback model via online experiments on Kuaishou, a short-video platform with tens of billions of users. Despite focusing on a 5% sample, this still represented hundreds of millions of users. Through a 7-day A/B test and real-time data analysis, our approach shows promising results, with detailed findings discussed subsequently.

3.1 Overall performance of A/B test

3.1.1 Experimental Settings. The details of deployment, baseline, and metrics are presented as follows.

Deployment. We conduct an A/B test on Kuaishou's Discover and Featured pages, representing traditional double-column and emerging single-column recommendation scenarios, respectively. These popular pages let us test our model's stability and effectiveness in diverse scenarios while illuminating the role of negative feedback. Hereafter, we'll refer to the Featured and Discover pages as single-column and double-column pages.

Baseline. To evaluate the effect of negative feedback, we remove the mixed feedback encoder from the proposed model to serve as the experimental baseline model. The users are split into two equal groups for an online A/B test: the baseline group receives recommendations from the existing system, and the experimental group receives recommendations from our system that incorporates user negative feedback modeling.

Metrics. We monitor real-time user behavior in the A/B test. Metrics represent both user groups' behavior, with the improvement rate calculated as a percentage over the baseline.

3.1.2 Main results. Our results, detailed in Table 2, reveal two key insights:

- **By incorporating user negative feedback, our model outperforms the baseline in all measured metrics.** Our model successfully identifies user intent and context-aware preferences, leading to more active users, increasing video viewing time, and enhancing user engagement (evident from more likes and comments), with relative improvements of 0.055% and 0.160%, respectively.
- **Integrating implicit user negative feedback, such as Glance Video Viewing, helps reduce explicit negative behaviors.** A notable decrease of 0.049% users and 0.251% total reducing times was observed in the use of Kuaishou's 'reduce similar videos' function, which is intended to collect explicit negative feedback. Thus, our method effectively handles user feedback, enhancing the user experience.

3.1.3 User Satisfaction Questionnaire Responses. We further evaluate the efficacy of our method through online satisfaction questionnaires distributed to the experimental and baseline user groups during the A/B test. The results, presented in Table 3, provide direct user feedback on the quality of recommended videos and user experience.

After integrating multiple types of user feedback into the encoding, user satisfaction with the recommended content increased. The positive and negative feedback ratios from the questionnaire rise by 1.060% and fall by 0.089%, respectively. Simultaneously, we observe a 0.048% increase in users' likes of the video, coupled with a significant reduction in dislikes. The increase in positive feedback and the decrease in negative feedback indicates that our method infers users' intentions more accurately than the baseline.

3.2 Performance on different scenarios

3.2.1 Comparison between single-column page and double-column page. We compare performance between Kuaishou's single-column and double-column pages, which represent short videos and traditional recommendations, respectively. Key metrics analyzed include forward and comment counts, explicit negative behaviors, page visitors, video plays, and reductions in similar recommendations.

As shown in Figure 4, **our method outperforms the baseline across all the user-side and video-side metrics on both page types.** The utilization of negative feedback in our method brings an increased number of page visitors and video plays (growth rates of 0.060% and 0.079% for single-column pages; 0.041% and 0.057% for double-column pages) and higher user engagement (increases in forwarding and commenting by 0.0194% and 0.102% in single-column; 0.776% and 0.057% in double-column). This indicates users' positive reception to the recommended content. Moreover, our method reduces negative user feedback, with fewer users reporting dislikes and reducing similar recommendations. This performance across different scenarios certifies our model's stability and effectiveness.

3.2.2 Improvement trend over a week. we further study the performance improvement in a one-week time window to capture long-term performance improvement. The daily average recommendation performance for our model (bucket-A) and the base model (bucket-B) is presented in Figure 2 across key metrics, including app usage duration, video play time and count, active user and page visitor counts, and similar recommendation reductions.

- **The overall app usage duration and the number of active users show stable and smooth improvements over the baseline.** Figure 2(a) shows a steady improvement trend in app usage duration, with an average growth rate of 0.150%, and relative stability in active user count in Figure2(d), with an average improvement rate of around 0.050%.
- **Our method's improvements increase consistently over time on most metrics.** Growth users' (recently-registered users

Table 2: Main results of A/B performance on user multi-feedback modeling. Reduction refers to user reports reducing similar recommendations. ↓ denotes that the lower the value, the more satisfied the user is with the recommended video.

Method	Active Users	Play Duration	Players Number	Like Users	Like Times	Comment Times	Reduction Users ↓	Reduction Times ↓
Baseline	11,946,769	960,699,908	11,044,619	5,375,275	75,608,834	17,098,385	325,860	1,557,144
Our-Feedback	11,953,341	962,232,867	11,052,299	5,377,719	75,640,199	17,104,129	325,701	1,553,241
Difference	+6,572	+1,532,959	+7,680	+2,444	+31,365	+5,744	-159	-3,903
Improvements rate(%)	+0.055	+0.160	+0.070	+0.046	+0.042	+0.034	-0.049	-0.251

Table 3: The result of the user's response to the user satisfaction questionnaire in the system popup. ↓ represents negative metrics, in other words, the performance is improved with lower values.

Method	Questionnaire Pos / Neg feedback	Video Pos / Neg feedback	Video Likes	Questionnaire Dislikes ↓	Video Dislikes ↓
Baseline	3.916	5.216	84,019,353,073	3921	16,111,870,209
Our-Feedback	3.956	5.220	84,059,804,112	3860	16,106,497,453
Improvements rate(%)	+1.060	+0.089	+0.048	-1.552	-0.033

Table 4: Performance of our model under single-column page and double-column page scenarios. ↓ denotes that the lower the value, the better performance our model achieved.

Page	Forward	Comment	Negative	Visitors	Players	Reduction ↓
Single-Column	+0.194%	+0.102%	-0.867%	+0.060%	+0.079%	-0.093%
Double-Column	+0.776%	+0.042%	-2.045%	+0.041%	+0.057%	-2.182%

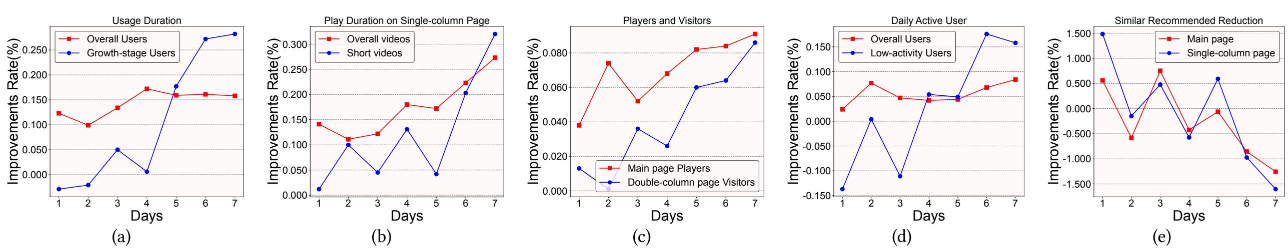


Figure 2: The performance improvement trend of our model in a one-week window. (a) The improvement trend of user's usage duration. (b) The improvement trend of video's playing duration on single-column pages. (c) The improvement trend of the main page's players and double-column page's visitors. (d) The improvement trend of the number of daily active users. (e) The improvement trend of the reduction number of similar recommendations, of which the lower number the better performance.

who are likely to continue using the app) usage time shows an upward trend (Figure 2(a)), with an improvement rate of -0.029% on the first day to 0.282% on the last day. Similarly, despite an initial lower count, active user numbers in the low-activity group gradually exceeded the baseline by over 0.15% as negative feedback modeling proceeds. Other metrics show an immediate performance improvement once our method is deployed. In addition to the improvement in positive feedback, explicit negative feedback also decreases, indicating users' growing satisfaction with the recommended videos.

The above results show our model's effectiveness and stability in enhancing both short and long-term user engagement.

3.2.3 Study on users with different engagement levels. We investigate whether the observed performance improvement is consistent across users with varying levels of engagement. Users are categorized into three engagement levels: growth stage, maturity stage, and recession stage, with the addition of a low-activity user group.

- **Our method outperforms baseline across every group,** showing improvements ranging from 0.008-0.162% in Figure 3. This consistent improvement ensures that negative feedback modeling benefits all user groups.
- **The maturity-stage group obtains the most significant improvement.** This is explained by the maturity-stage user group

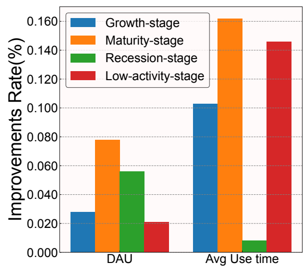
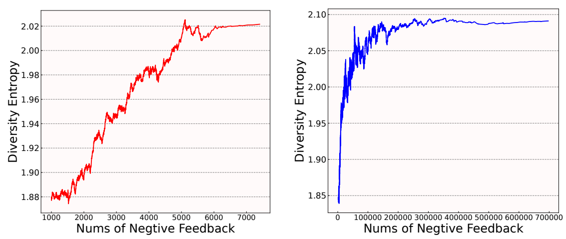


Figure 3: Daily active users (DAU) and user's average using time (Avg Use time) on different user engagement levels.



(a) Result of low exposure users. (b) Result of high exposure users.

Figure 4: Relationship between negative feedback and diversity entropy in the long-term aspect.

having the most active users and more feedback collected, which further verifies the effectiveness of the feedback modeling.

3.3 Long-term Analysis for Model Deployment

To further study the long-term influence of our method, we deploy it to the real-world recommender system, Kuaishou, over six months. We record the behaviors of over 600,000 users and investigate the correlation between the diversity of recommendations and users' negative feedback over time. We calculate the 'diversity entropy [20]' of each time window containing 100 videos from user history, which evaluates the diversity and randomness of item distribution in a set. For a balanced analysis, users are classified into low-exposure and high-exposure groups.

As presented in Figure 4, our model, which effectively incorporates users' negative feedback, promotes diverse recommendations over time. We can observe that: **Users who are more willing to express negative feedback will result in a more diverse set of recommended videos by our model.**

This extensive six-month experiment validates the long-term efficacy of our negative feedback modeling approach, demonstrating its potential for steady and significant improvements.

4 RELATED WORK

4.1 Implicit Feedback Learning

Early recommender systems relied on explicit feedback like ratings [17, 18]. However, with the increasing availability of binary interaction data, such as click behavior, implicit feedback learning has become a more common setting for recommendation methods, including early matrix factorization-based methods [10, 11, 14, 31, 33, 40], neural network-based ones [1–3, 8, 13, 21, 22, 30, 35], advanced graph neural network (GNN) approaches [7, 12, 36, 41], and even to recent automated machine learning ones [22, 23]. Other works [6, 9] proposed a negative sampling strategy to utilize implicit feedback as negative candidates. However, new and important feedback caused by the streaming-manner recommendation widely exists in the short-video recommendation, such as skipping behavior. While previous works in music streaming [26, 27] studied the types and reasons for skipping behavior, they didn't provide effective solutions to application problems. However, our system in Kuaishou has well addressed these, which cannot be well handled by existing works since the skipping behavior does not directly reveal user preference signals and involves multiple fused objectives.

4.2 Video Recommendation

Video recommendation, especially for short-video apps, is a critical application of recommender models. Existing works [5, 16, 28, 37]mainly focus on how to better exploit the video features into preference learning. For example, Deldjoo et al. [5] proposed to automatically extract visual features of videos and provided users with similar videos based on these video features. Lee et al. [19] utilized raw visual and auditory content to learn a compact representation of videos. There are some works that approach the problem by studying the optimization watching time [42]. Zheng et al. [42] proposed an unbiased evaluation metric watch time gain to learn unbiased user preferences and alleviate the duration bias in previous evaluation metrics. Some other works considered both user and video content attributes to improve recommendation accuracy. [4, 24] For example, Cui et al. [4] combined social and content attributes and leveraged both social and content information to address the sparsity problem in the micro-video recommendation. Liu et al. [24] studied the multi-modal information from both user and micro-video sides using the attention mechanism. However, the challenge of feedback learning, particularly implicit negative feedback for video recommendation, has not been well studied.

5 CONCLUSION AND FUTURE WORK

In this paper, we present the deployed solution for utilizing implicit negative feedback, which is far easier to collect than other behaviors, in the short-video recommendation engine of Kuaishou. Our solution well combines context-aware feedback learning and multi-objective prediction/optimization, well addressing the two corresponding critical challenges. The sufficient evaluation through billion-level users' A/B testing demonstrates the effectiveness of the proposed solution. As for future work, we plan to further combine the sequential modeling into negative feedback learning, further improving preference learning.

ACKNOWLEDGMENTS

This work is partially supported by the National Key Research and Development Program of China under 2022YFB3104702, and the National Natural Science Foundation of China under 62272262, 61972223, U1936217, and U20B2060, and also supported by Kuaishou.

REFERENCES

[1] Wanyu Chen, Fei Cai, Honghui Chen, and Maarten De Rijke. 2019. Joint neural collaborative filtering for recommender systems. *ACM Transactions on Information Systems (TOIS)* 37, 4 (2019), 1–30.

[2] Heng Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, and Mustafa Ispir. 2016. Wide & Deep Learning for Recommender Systems. (2016).

[3] Weiyu Cheng, Yanyan Shen, and Linpeng Huang. 2020. Adaptive Factorization Network: Learning Adaptive-Order Feature Interactions. (2020).

[4] Peng Cui, Zhiyu Wang, and Zhou Su. 2014. What videos are similar with you? learning a common attributed representation for video recommendation. In *Proceedings of the 22nd ACM international conference on Multimedia*. 597–606.

[5] Yashar Deldjoo, Mehdi Elahi, Paolo Cremonesi, Franca Garzotto, Pietro Piazzolla, and Massimo Quadrana. 2016. Content-based video recommendation system based on stylistic visual features. *Journal on Data Semantics* 5 (2016), 99–113.

[6] Jingtao Ding, Yuhan Quan, Quanming Yao, Yong Li, and Depeng Jin. 2020. Simplify and robustify negative sampling for implicit collaborative filtering. *Advances in Neural Information Processing Systems* 33 (2020), 1094–1105.

[7] Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, et al. 2023. A survey of graph neural networks for recommender systems: challenges, methods, and directions. *ACM Transactions on Recommender Systems* 1, 1 (2023), 1–51.

[8] Chen Gao, Yu Zheng, Wenjie Wang, Fuli Feng, Xiangnan He, and Yong Li. 2022. Causal Inference in Recommender Systems: A Survey and Future Directions. *arXiv preprint arXiv:2208.12397* (2022).

[9] Shansan Gong and Kenny Q Zhu. 2022. Positive, negative and neutral: Modeling implicit feedback in session-based news recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1185–1195.

[10] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).

[11] 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.

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

[13] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[14] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. 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*. 549–558.

[15] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE international conference on data mining*. Ieee, 263–272.

[16] Yanxiang Huang, Bin Cui, Jie Jiang, Kunqian Hong, Wenyu Zhang, and Yiran Xie. 2016. Real-time video recommendation exploration. In *Proceedings of the 2016 International Conference on Management of Data*. 35–46.

[17] Zahid Younas Khan, Zhendong Niu, Sulis Sandiwarno, and Rukundo Prince. 2021. Deep learning techniques for rating prediction: a survey of the state-of-the-art. *Artificial Intelligence Review* 54 (2021), 95–135.

[18] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[19] Joonseok Lee and Sami Abu-El-Haija. 2017. Large-scale content-only video recommendation. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 987–995.

[20] Nian Li, Chen Gao, Jinghua Piao, Xin Huang, Aizhen Yue, Liang Zhou, Qingmin Liao, and Yong Li. 2022. An Exploratory Study of Information Cocoon on Short-form Video Platform. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 4178–4182.

[21] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1754–1763.

[22] Bin Liu, Chenxu Zhu, Guilin Li, Weinan Zhang, Jincai Lai, Ruiming Tang, Xiuqiang He, Zhenguo Li, and Yong Yu. 2020. AutoFIS: Automatic Feature Interaction Selection in Factorization Models for Click-Through Rate Prediction. (2020).

[23] Hanxiao Liu, Karen Simonyan, and Yiming Yang. 2019. Darts: Differentiable architecture search. (2019).

[24] 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.

[25] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1930–1939.

[26] Francesco Meggetto, Crawford Revie, John Levine, and Yashar Moshfeghi. 2021. On skipping behaviour types in music streaming sessions. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 3333–3337.

[27] Francesco Meggetto, Crawford Revie, John Levine, and Yashar Moshfeghi. 2023. Why people skip music? On predicting music skips using deep reinforcement learning. In *Proceedings of the 2023 Conference on Human Information Interaction and Retrieval*. 95–106.

[28] Tao Mei, Bo Yang, Xian-Sheng Hua, and Shipeng Li. 2011. Contextual video recommendation by multimodal relevance and user feedback. *ACM Transactions on Information Systems (TOIS)* 29, 2 (2011), 1–24.

[29] Andriy Mnih and Russ R Salakhutdinov. 2007. Probabilistic matrix factorization. *Advances in neural information processing systems* 20 (2007).

[30] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. 2016. Product-based neural networks for user response prediction. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 1149–1154.

[31] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International Conference on Data Mining*. IEEE, 995–1000.

[32] Steffen Rendle. 2021. Item recommendation from implicit feedback. In *Recommender Systems Handbook*. Springer, 143–171.

[33] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. 452–461.

[34] Steffen Rendle, Walid Krichene, Li Zhang, and John Anderson. 2020. Neural collaborative filtering vs. matrix factorization revisited. In *Proceedings of the 14th ACM Conference on Recommender Systems*. 240–248.

[35] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1161–1170.

[36] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 165–174.

[37] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2019. MMGCN: Multi-modal graph convolution network for personalized recommendation of micro-video. In *Proceedings of the 27th ACM international conference on multimedia*. 1437–1445.

[38] Le Wu, Xiangnan He, Xiang Wang, Kun Zhang, and Meng Wang. 2022. A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[39] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2022. Graph neural networks in recommender systems: a survey. *Comput. Surveys* 55, 5 (2022), 1–37.

[40] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional factorization machines: Learning the weight of feature interactions via attention networks. (2017).

[41] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 974–983.

[42] Yu Zheng, Chen Gao, Jingtao Ding, Lingling Yi, Depeng Jin, Yong Li, and Meng Wang. 2022. DVR: Micro-Video Recommendation Optimizing Watch-Time-Gain under Duration Bias. In *Proceedings of the 30th ACM International Conference on Multimedia*. 334–345.