Objective-Driven Calibrated Recommendations

Anonymous Author(s)

Abstract

Calibrated recommendation (CR) is a concept that matches the proportions of categories in the recommendations with the proportions observed in the user's historical interaction. For example, if a user's interaction history consists of 60% romance and 40% action movies, CR is achieved when the recommended ratio is also 60:40. The issue that we raise in this paper is that CR is being treated as an *objective*. We argue that platforms should not aim for CR for their own sake, as CR is a heuristic objective that does not, in itself, contribute to the platform. Rather, CR is only a *method* to achieve long-term platform objectives, like subscription renewal and long-term ratings, that often align with the platform's success. However, we question whether CR is a suitable method to achieve such objectives, as CR is only heuristic. In practice, the ideal categorical proportions that maximize long-term rewards would differ by user and by platform, and the proportions of CR would not be ideal in many cases.

In this paper, we define a new concept called **O**bjective-driven **C**alibrated **R**ecommendation (**OCR**), where the ideal categorical proportion would maximize the targeted long-term reward. We argue that long-term rewards, like subscription renewal, user retention, and lifetime value (LTV), are much more suitable objectives than original CR, as they often align closely with platform success. To estimate the optimal categorical proportions and to build a recommendation algorithm based on it, we introduce a novel two-stage contextual bandit method called **O**ptimization for **OCR** (**OOCR**). OOCR is a two-stage optimization method that designs categorical proportions to maximize long-term objectives and then also maximizes immediate rewards. OOCR is implemented in an off-policy learning (OPL) framework in the contextual bandit setting, enabling training from historical interaction data without requiring online experimentation. Empirical results on synthetic datasets and simulations on real-world datasets demonstrate that OOCR can accurately design a recommendation policy that closely matches optimal category distributions, yielding significantly higher long-term outcomes than CR and other baselines.

Code and appendix (w/ additional experiment settings and results) are provided in https://github.com/cali-anon/OCR. For ease of cross-referencing from the appendix, the repository also contains the main text.

ACM Reference Format:

Anonymous Author(s). 2025. Objective-Driven Calibrated Recommendations. In . ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Modern recommendation systems often focus on maximizing immediate rewards such as clicks, watch time, or dwell time [4, 5, 24]. It has been argued that naively maximizing these immediate rewards makes the algorithms recommend only the main categories of user interest, which makes the categories where users are less interested end up under-exposed or even completely absent [2, 16]. For instance, a user with a slight preference for action movies over romance might receive a feed almost exclusively filled with action movies to maximize short-term engagement metrics.

To avoid such unbalanced recommendations, Steck [19] introduced a new concept called *calibrated recommendation* (CR), arguing that the proportions of categories in the recommendations should directly match the proportions observed in the user's historical interaction. For example, if a user's watch history consists of 60% romance and 40% action movies, CR is achieved when the recommendation has a 60:40 ratio. Since its introduction, CR has been widely used as an *objective* in recommendation scenarios [1, 6, 10–12, 17]. However, we posit that CR should not be treated as an objective. We caution that platforms should not try to achieve CR for their own sake, as it creates no benefit for the platform in itself. If platforms use CR, it should be used as a *method* in the hope that it can lead to better long-term rewards, like platform ratings, subscription renewal, and repeated visits. These long-term rewards are the genuine objectives that closely align with the platform's success, while CR *heuristically guesses* the best categorical proportions to achieve it. However, in reality, the optimal categorical proportion would not be that of CR. Some platforms or users would value unbalanced categorical proportions, while others would value more uniform proportions. Thus, we question whether CR is a suitable concept for the platform.

In this paper, we define the ideal categorical proportions using long-term rewards, which we call **O**bjective-driven **C**alibrated **R**ecommendations (**OCR**). Specifically, long-term rewards like subscription renewal and long-term ratings often align with platform success. Thus, we argue that these rewards are the only meaningful long-term objective and the original CR should not be treated as an objective. Thus, OCR defines the ideal categorical proportion as the one that maximizes these long-term rewards. Different from the original CR, the optimization of OCR directly leads to what the platform aims to maximize.

In practice, however, we need to estimate the ideal categorical proportion for each user using only historical user interactions. The difficult problem is that the user interactions may be collected under a different recommendation algorithm, resulting in a biased dataset. To deal with such bias, we consider building a recommendation policy that would maximize long-term rewards through *off-policy learning (OPL)* in contextual bandits, which learns using only historical logged data collected under previous recommendation policies [8, 20, 21, 23]. However, optimizing a recommendation policy *with ideal categorical proportions* is infeasible to do in one single step. This is because the number of visits until the long-term

rewards are received (batch length) is unknown and varies from user to user, thus unable to define a deployable policy. Even if the batch length is consistent, the action space would be exponentially large, resulting in ineffective OPL.

To overcome this challenge, we propose a novel two-stage OPL approach named **O**ptimization for **OCR** (**OOCR**). OOCR enables optimization of long-term rewards by optimizing only the categorical proportions in the first stage, which solves the problem of inconsistent batch lengths. Then, it optimizes the immediate rewards (e.g., clicks) within the optimized categorical proportions, enabling optimization of both long-term and immediate rewards.

We posit that rather than using the original CR to *guess* a categorical proportion, our *estimated* optimal proportions would better satisfy the platform objective in various scenarios. We demonstrate this empirically, where we show that OOCR closely matches optimal category proportions in synthetic experiments and also provides up to a 16% improvement in long-term outcomes in simulations on real-world datasets.

2 Problem Formulation

To formally define the problem, we model the recommendation task using the contextual bandit framework [8, 20]. This framework captures the interaction process of recommendation, where for each user visit, the system observes the user's current state (context), selects an item to show (action), and receives feedback like a click (reward).

Formally, a user arrives with context $x \in \mathcal{X} \subseteq \mathbb{R}^{d_x}$, a d_x-dimensional vector, i.i.d. from an unknown distribution $p(x)$. We let $u(x)$ denote the (possibly static) user information within x (e.g., age, location, gender), where $u \in \mathcal{U} \subseteq \mathbb{R}^{d_u}$. In practice, x can include session-level features that carry across visits (e.g., timestamps, page views) along with the user-specific features in $u(x)$. Now, given context x, a possibly stochastic policy $\pi(a|x)$ chooses an item $a \in \mathcal{A}$, where $\mathcal{A}$ is the set of all recommendable items. The immediate reward $r \in [0, r_{max}]$ (e.g., clicks) is then sampled from an unknown distribution $p(r|x, a)$ to each of the recommendations.

2.1 Immediate-Reward Maximization Policy

Platforms typically aim to learn a recommendation policy π_θ that maximizes the expected immediate reward,

$$\theta^* \in \arg\max_{\theta\in\Theta} V^r(\pi_\theta), \tag{1}$$

where the value function is given by

$$V^r(\pi) := \mathbb{E}_{p(x)\pi(a|x)p(r|x,a)}[r] = \mathbb{E}_{p(x)\pi(a|x)}[q(x,a)], \tag{2}$$

and $q(x, a) = \mathbb{E}[r|x, a]$ is the expected immediate reward given x and a.

Since the true value functions are unknown, we need to estimate the optimal policy using only the logged data $\mathcal{D}_r = \{(x_i, a_i, r_i)\}_{i=1}^n \sim p(\mathcal{D}_r)$ where the data distribution is collected by some logging policy π_0, i.e., $p(\mathcal{D}_r) = \prod_{i=1}^n p(x_i)\pi_0(a_i|x_i)p(r_i|x_i, a_i)$.

While a policy can be easily optimized via methods like IPS (refer to Section 4), these immediate-reward maximization policies typically over-expose items of a user's main categories of interest since items in these categories have high immediate rewards. Consequently, this makes items of categories where users are less interested in being under-exposed or even completely absent [2, 16, 19].

2.2 Calibrated Recommendations Policy

In order to punish such unbalanced recommendations, Steck [19] introduced the concept of calibrated recommendations (CR). The fundamental idea behind CR is to recommend categories in proportion to the user's historical interactions, ensuring that the recommended content reflects the user's interests more holistically rather than over-concentrating on a narrow subset of items. We reformulate CR within the contextual bandit framework.

All items a belong to some category $c \in C$, where C is some finite category space (e.g., movie genres and product types). Each item can only belong to one category, where $c(a)$ is the category of item a. We define the policy at the category level as $\pi(c|u) = \sum_a \sum_x \pi(a|x)\mathbb{1}\{c = c(a), u = u(x)\}$, where $\mathbb{1}$ is an indicator function that outputs 1 if the equation holds and 0 otherwise.

Now, the recommendation is said to achieve CR if categories are recommended in the proportions of their interaction history. We can define the historical interaction proportions of category c for a user u under policy π as

$$p(c|u) = \mathbb{1}\{u = u(x)\}\frac{\sum_{a\in\mathcal{A}} \pi(a|x)q(x,a)\mathbb{1}\{c = c(a)\}}{\sum_{a'\in\mathcal{A}} \pi(a'|x)q(x,a')} \tag{3}$$

If the immediate reward is a click, Eq. (3) essentially shows the expected probability that the user would have clicked on a category c, and $\sum_{c\in C} p(c|u) = 1$.

Using the historical interaction proportions of each category, Steck [19] uses Kullback-Leibler (KL) divergence to quantify how closely the recommendation policy's categorical proportions align with users' historical interaction proportions. Steck [19] then defines the calibration metric as

$$V^{CR}(\pi) = -\mathbb{E}_{p(u)}\left[\sum_c p(c|u)\log\frac{p(c|u)}{\pi(c|u)}\right]. \tag{4}$$

The expectation $\mathbb{E}_{p(u)}[\cdot]$ is computed with respect to the distribution of users $p(u)$, rather than contexts $p(x)$, because the calibration metric is defined based on each user, independent of context variations.

Finally, we find the calibrated recommendation policy by solving

$$\theta_{\text{CR}} \in \arg\max_{\theta\in\Theta}(1-w)V^r(\pi_\theta) + wV^{CR}(\pi_\theta), \tag{5}$$

where w is a parameter that balances calibration and immediate reward maximization.

CR has been widely used as a handy objective in recommendation scenarios, as optimizing for CR naturally solves the unbalanced exposure problem [1, 6, 10–12, 17]. However, we argue that CR should not be treated as an objective, as optimizing for CR in itself is not what the platform wishes to achieve. Then, CR should only be treated as a method for some platform goals, like maximizing ratings and subscriptions. If a user's history consists of 60% romance and 40% action movies, "making the recommended categorical proportion 60:40" is not what benefits the platform. It would benefit the platform only if this leads to better long-term outcomes, like increased user interactions.

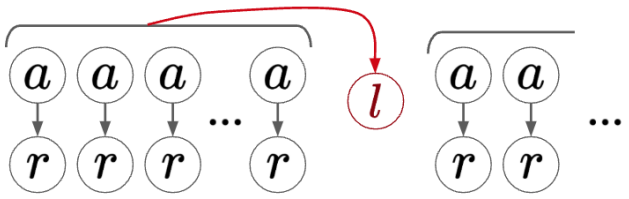

Figure 1: Illustration of how long-term rewards can be captured.

Yet, CR is not an ideal method to achieve such long-term goals. In practice, different users and platforms value different categorical proportions. Some users or platforms may prefer a completely different categorical proportion than the one consumed historically. CR is not able to adjust the targeted categorical proportions based on the platform, the user, or the platform objective. Thus, we should not assume that the ideal categorical proportion is the one proposed in CR.

3 Objective-driven Calibrated Recommendation

We propose to define a new concept of Objective-driven Calibrated Recommendations (OCR), which describes the ideal categorical proportions using long-term rewards. Unlike the original CR that targets predefined categorical proportions, in OCR, we aim to find the categorical proportion that maximizes the long-term rewards.

In various real-world recommendation platforms, we commonly observe not only immediate rewards for individual recommendations (e.g., clicks), but also long-term rewards reflecting user reactions to batches of recommendations (e.g., user retention, subscription renewals, repeated visits) [25–27, 31]. These long-term rewards directly lead to platform success and thus are the only meaningful objective. Importantly, unlike the original CR, the ideal categorical proportion of OCR is *user-dependent, reward-dependent, and platform-dependent.*

To implement this idea, we first extend the typical formulation of contextual bandits by introducing the long-term rewards denoted by $l \in [0, l_{max}]$, like subscription renewals and repeated site visits. These long-term reward signals are given against a batch of recommendations, as is described in Figure 1. The policy value for long-term rewards can be defined as

$$V^l(\pi) := \mathbb{E}_{T\sim p(T), x_t\sim p(x), a_t\sim\pi(a|x_t), l\sim p(l\,|\,\{(x_t,a_t)\}_{t=1}^T)}\left[l\right]. \quad (6)$$

Regarding that the long-term rewards are only observed every few sessions, we formulate the logged dataset for long-term rewards separately from $\mathcal{D}_r$. We formulate the long-term logged dataset as

$$\mathcal{D}_l = \left\{(x_{i,1}, a_{i,1}), \ldots, (x_{i,T_i}, a_{i,T_i}),\ l_i\right\}_{i=1}^{n_l} \sim p(\mathcal{D}_l), \quad (7)$$

where T_i is the number of visits of the user until the observation of long-term rewards. The data distribution is described

$$p(\mathcal{D}_l) = \prod_{i=1}^{n_l} p(T_i) \prod_{t=1}^{T_i} \left(p(x_{i,t})\pi_0(a_{i,t}\mid x_{i,t})\right) p\left(l_i \mid \{(x_{i,t}, a_{i,t})\}_{t=1}^{T_i}\right). \quad (8)$$

In this formulation, each sequence of visits $\{(x_{i,t}, a_{i,t})\}_{t=1}^{T_i}$ comes from a single user, which can be denoted u_i.

Using this dataset, we aim to design the policy to maximize the long-term rewards. However, typical OPL algorithms are unable to be deployed. This is because the number of visits T may vary from user to user. Moreover, even if the length of T is consistent and known, optimizing it directly would exponentially enlarge the action space, which results in large variance and an ineffective OPL [15].

Otherwise, one might consider formulating the problem as a fully fledged Offline Reinforcement Learning (RL) task, where each user's sequence $\{(x_{i,t}, a_{i,t})\}_{t=1}^{T_i}$ is treated as an episode and the long-term reward l_i is interpreted as a terminal reward [3, 27, 29]. However, this approach is often intractable in practice because the possible combinations of T_i-step action sequences can grow exponentially with T_i. As a result, performing policy optimization in such a large decision space typically suffers from extremely high variance and requires impractically large amounts of data to achieve reliable performance [15]. In real-world scenarios where each user's T_i can vary and may be quite large, this makes naive Offline RL solutions difficult to deploy at scale without additional structural assumptions or constraints.

Thus, we introduce a two-stage learning framework, **O**ptimization for **OCR** (**OOCR**), described in Figure 2, that decouples the optimization process: (i) we first optimize categorical proportions to maximize long-term rewards, and (ii) we then optimize the final recommendation policy to maximize immediate rewards while ensuring the learned category proportions.

3.1 Stage 1: Category-Level Optimization

In the first stage, we focus exclusively on optimizing the policy at the *category level* to maximize long-term rewards l, such as subscription renewals or repeated site visits. As we have discussed, directly optimizing the full item-level recommendation policy $\pi(a|u)$ to maximize these long-term rewards is practically infeasible since long-term rewards are only observed at the aggregated category level rather than for individual items. Moreover, varying batch lengths and unknown recommendation sequences further complicate the direct optimization at the item level.

To overcome these issues, we first optimize the policy at the simpler category-level space. Specifically, let $\alpha = (\alpha^c)_{c\in C}$ be a *category-proportion vector*, where each $\alpha^c \in [0, 1]$ represents the fraction of recommendations allocated to category c, with $\sum_{c\in C}\alpha^c = 1$. We introduce an intermediate categorical policy, denoted as $\mu(\alpha|u)$, which explicitly learns optimal categorical proportions to maximize long-term rewards. After learning the categorical policy μ, we use it as a constraint to guide the final recommendation policy π, as described in Section 3.2.

Now, to motivate our approach, we introduce a simplifying assumption on how l depends on the recommendations:

Assumption 1 (No Direct Effect). *The sequence of actions $a_{1:T} = \{a_1, ..., a_T\}$ and session-level user states x have no effect on the long-term rewards l, i.e., $(x, a_{1:T}) \perp\!\!\!\perp l \mid (u, \alpha)$.*

Assumption 1 requires that the long-term rewards are only affected by the user context u and the categorical proportions of the recommendations α. It is important to clarify that we do not expect Assumption 1 to hold in practice. It is simply a condition that helps to understand how we define the ideal categorical proportions and

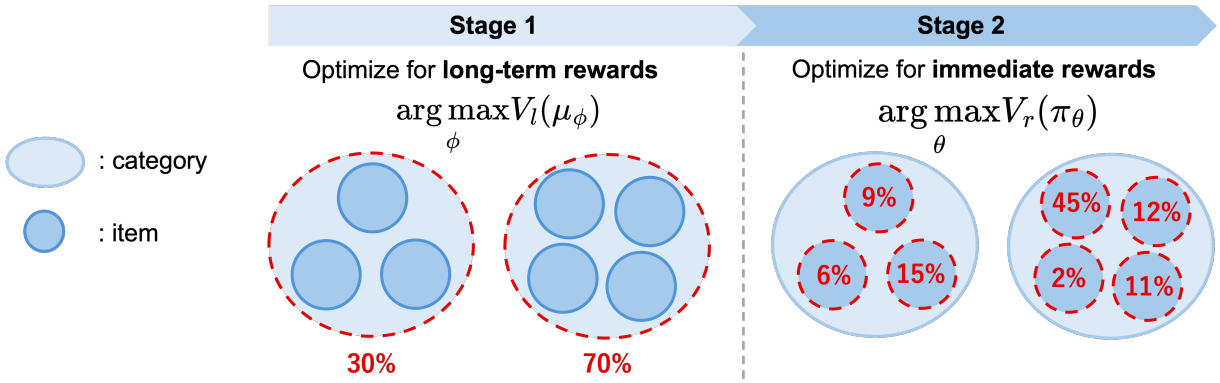

Figure 2: Illustration of our two-stage approach. Stage 1 optimizes the *category-level* policy against long-term rewards l. Stage 2 designs the final *item-level* policy to maximize the immediate rewards r while ensuring the learned categorical proportions.

how we optimize the category-level policy μ. We show in experiments that our method performs well even if this assumption is violated.

Proposition 1. *Under Assumption 1, we have*

$$V^l(\mu) = \mathbb{E}_{p(u)\mu(\alpha|u)p(l|u,\alpha)}[l], \tag{9}$$

In this first stage, we aim to find a categorical policy μ_ϕ that would maximize Eq. (9), i.e.,

$$\phi^* \in \arg\max_{\phi\in\Phi} V^l(\mu_\phi). \tag{10}$$

To optimize μ_ϕ from logged data, we rely on dataset $\mathcal{D}_l$, where each data point i contains a user u_i, a batch of T_i recommended items $(a_{i,1},\ldots,a_{i,T_i})$, and the resulting long-term reward l_i. From that batch, we can derive the *empirical* category-proportion vector

$$\alpha_i^c = \frac{1}{T_i}\sum_{t=1}^{T_i} \mathbb{1}\{c(a_{i,t}) = c\}. \tag{11}$$

Thus, $\alpha_i = (\alpha_i^c)_{c\in C}$, and each data point in $\mathcal{D}_l$ can then be summarized as (u_i, α_i, l_i).

Given Assumption 1, we postulate that the long-term reward l can be modeled as $l_i = f_\psi(u_i,\alpha_i) + \varepsilon_i$, where f_ψ is a regression function parameterized by $\psi \in \Psi$, and ε_i denotes noise with zero expectation. To find the optimal parameters ψ^*, we minimize the empirical mean squared error

$$\psi^* = \arg\min_{\psi\in\Psi} \sum_{i=1}^{n_l} \big(l_i - f_\psi(u_i,\alpha_i)\big)^2. \tag{12}$$

After obtaining the optimal parameters ψ^*, we construct the first-stage categorical policy μ_{ϕ^*} by selecting category proportions that maximize the predicted long-term reward for a given user context u:

$$\alpha^*(u) = \arg\max_{\alpha\in\Delta_C} f_{\psi^*}(u,\alpha), \tag{13}$$

where Δ_C denotes the simplex constraint $\sum_c \alpha^c = 1$, and $\alpha^c \geq 0$ for each $c \in C$. The categorical policy is set to

$$\mu_\phi(\alpha|u) = \delta\left(\alpha = \alpha^*(u)\right), \tag{14}$$

where δ represents a deterministic distribution.

3.2 Stage 2: Item-Level Optimization

In the second stage, we design the final item-selection policy π_θ by maximizing the immediate rewards in the constraints of the categorical policy μ_ϕ learned in the first stage. Let us define $\mu_\phi(c|u) = \alpha^{*c}(u)$, where $\alpha^{*c}(u)$ represents the component of the vector $\alpha^*(u)$ corresponding specifically to category c. We then impose that π_θ should, in aggregate, calibrate towards these proportions. Concretely, we solve

$$\arg\max_{\theta\in\Theta} \hat{V}^r(\pi_\theta;\mathcal{D}_r) \;\; s.t., \forall_u \;\; \mathrm{KL}(\mu_\phi(\cdot|u)||\pi_\theta(\cdot|u)) = 0, \tag{15}$$

where $\mathrm{KL}(\mu_\phi(\cdot|u)||\pi_\theta(\cdot|u)) = \sum_{c\in C}\mu_\phi(c|u)\log\frac{\mu_\phi(c|u)}{\pi_\theta(c|u)}$ is the Kullback–Leibler divergence between the target category distribution $\mu_\phi(\cdot|u)$ and the policy's induced distribution $\pi_\theta(\cdot|u)$ for user u. Also, the IPS estimator, $\hat{V}^r(\pi_\theta;\mathcal{D}_r) = \frac{1}{n}\sum_{i=1}^n \pi_\theta(a_i|x_i)/\pi_0(a_i|x_i)r_i$, can be used here.

Then, using Lagrange multipliers, an equivalent dual of Eq. (15) is

$$\begin{aligned}&\arg\max_{\theta}\min_{\lambda\geq 0} \mathcal{L}(\theta,\lambda)\\ &= \arg\max_{\theta}\min_{\lambda\geq 0} \hat{V}^r(\pi_\theta;\mathcal{D}_r) + \lambda\mathbb{E}_{p(u)}\Big[\mathrm{KL}(\mu(\cdot|u)||\pi_\theta(\cdot|u))\Big]\end{aligned} \tag{16}$$

Now, to learn the policy, we alternate between gradient ascent in θ to improve the primal objective, and gradient descent in $\lambda_{u,c}$ to enforce constraints:

$$\theta \leftarrow \theta + \eta_\theta \nabla_\theta \mathcal{L}(\theta,\lambda), \tag{17}$$

$$\lambda \leftarrow \lambda + \eta_\lambda \mathbb{E}_{p(u)}\left[\mathrm{KL}(\mu(\cdot|u)||\pi_\theta(\cdot|u))\right] \tag{18}$$

where $\eta_\theta, \eta_\lambda$ are step sizes. After convergence, π_θ is our final policy, balancing immediate reward with the category-level proportions learned from Stage 1.

In summary, OOCR explicitly optimizes categorical proportions based on long-term rewards, then constructs an item-level recommendation policy that maximizes immediate rewards under these learned proportions. We argue that OOCR is often a better option than CR at achieving the long-term platform objective. We now empirically evaluate the effectiveness of OOCR using synthetic and simulations on real-world datasets.

4 Synthetic Experiments

This section evaluates OOCR on synthetic data to identify the situations where OOCR's performance becomes particularly promising. Note that our code is available upon publication.

Data Setup. To generate synthetic data, we first define 300 users characterized by 5-dimensional context vectors (x) sampled from the standard normal distribution. Then, we randomly assign each action a to a category c. Next, we define the expected immediate reward as follows.

$$q(x,a) = x^T M_{x,x_a} x_a + \theta_x^T \cdot x + \theta_a^T \cdot x_a, \tag{19}$$

where x_a denotes action context represented by a one-hot vector and M_{x,x_a}, θ_x and θ_a are sampled from a uniform distribution within range $[-1,1]$. Based on the above synthetic reward function, we sample the immediate reward r from a binomial distribution whose mean is $q(x,a)$.

We synthesize the logging policy π_0 by applying the softmax function to the expected reward function as

$$\pi_0(a|x) = \frac{\exp\left(\beta \cdot q(x,a)\right)}{\sum_{a' \in \mathcal{A}} \exp\left(\beta \cdot q(x,a')\right)} \tag{20}$$

where β is a parameter that controls the optimality and entropy of the logging policy. A larger value of β indicates that the logging policy is near-deterministic and well-performing for immediate rewards, while a smaller β makes the logging policy less optimal. In synthetic experiments, we use $\beta = -7.0$. From this logging policy, we obtain the category-proportion vector α_x, where $\alpha_x^c = \sum_{a \in \mathcal{A}} \pi_0(a|x) \mathbb{1}\{c_a = c\}$.

We then synthesize the expected long-term reward as follows.

$$Q(x, a\alpha) = (1-\gamma) f(x,\alpha) + \gamma V_x^r(\pi), \tag{21}$$

where γ is an experiment parameter that controls the influence of immediate rewards on long-term rewards, and V_x^r denotes the immediate policy value for a user x. In particular, when $\gamma = 0$, Assumption 1 is satisfied. $f(x,\alpha)$ is defined as follows.

$$f(x,\alpha) = \exp\left(-B(x) \cdot \mathrm{KL}(\mu^*(c|x) || \pi(c|x))\right) \tag{22}$$

where $\mathrm{KL}(\cdot||\cdot)$ denotes KL divergence and μ^* is an optimal categorical proportion. $f(x,\alpha)$ takes larger values when $\pi(c|x)$ is close to $\mu^*(c|x)$. $B(x)$ controls the influence of the KL term. A larger value of $B(x)$ means that f rapidly decreases if μ^* and π are even slightly apart. We then sample the long-term reward l from a binomial distribution with mean $Q(x,\alpha)$.

To summarize, we first sample a user and define the expected immediate reward $q(x,a)$ as in Eq. (19). We then sample a discrete action a from π_0 in Eq. (20). The immediate reward r is then sampled from a binomial distribution with mean $q(x,a)$. Iterating this procedure n times generates the logged dataset $\mathcal{D}_r$ with n independent copies of (x,a,r). Additionally, given a category-proportion vector for each user, we sample the long-term reward l from a binomial distribution whose mean is $Q(x,\alpha)$. We then obtain the long-term logged dataset $\mathcal{D}_l$ whose number of (x,α,l) tuples is equal to the number of users. In this experiment, the logged data sizes for long-term rewards and immediate rewards are different, so we refer to the logged data size for immediate rewards as the training data size.

We design this synthetic experiment to simulate real-world recommendation systems such as music/video streaming platforms. The immediate reward can be interpreted as a click, like, or other engagements for each item, and the long-term reward represents a subscription renewal for each user.

Compared methods. We compare OOCR with the following baselines:

IPS-PG is a greedy approach that aims to maximize immediate rewards. Specifically, it optimizes through iterative gradient descent $\theta_{t+1} \leftarrow \theta_t + \nabla_\theta \widehat{V}^r_{\text{IPS-PG}}(\pi_\theta; \mathcal{D}_r)$, where

$$\nabla_\theta \widehat{V}^r_{\text{IPS-PG}}(\pi_\theta; \mathcal{D}_r) := \frac{1}{n} \sum_{i=1}^{n} \frac{\pi_\theta(a_i \mid x_i)}{\pi_0(a_i \mid x_i)} r_i s_\theta(x_i, a_i), \tag{23}$$

where $s_\theta(x,a) := \nabla_\theta \log \pi_\theta(a \mid x)$ is the policy score function.

CR-PG is a calibrated recommendation approach defined in Eq. (5), where we set $\lambda = 0.7$.

In order to obtain categorical proportions for OOCR, we learn an estimated long-term reward function $\hat{Q}(x,\alpha)$ by a 3-layer neural network. We then obtain categorical proportions that maximize the estimated long-term reward for each user as follows.

$$\alpha = \arg\max_{\alpha \in \Delta_\alpha} \hat{Q}(x,\alpha), \tag{24}$$

where Δ_α denotes a discrete category-proportion vector space and we set $|\Delta_\alpha| = 1000$.

4.1 Results and Discussion

Figure 3 reports categorical proportions in order to discuss whether OOCR can achieve calibration towards optimal proportions. Figures 4 to 6 show the immediate and long-term policy values, computed over 50 simulations with different random seeds to produce synthetic data instances. Unless otherwise specified, the training data size is set to $n = 1000$, the number of actions is $|\mathcal{A}| = 300$, the number of categories is $|C| = 20$, and the influence of immediate rewards on long-term rewards is $\gamma = 0.3$. Note that the shaded regions in the plots represent 95% confidence intervals.

How well can OOCR calibration towards optimal proportions? Figure 3 shows the distribution of categories for a user with the smallest value of Kullback–Leibler divergence between the optimal categorical proportion $\mu^*(c|x)$ and $\pi(c|x)$ learned by OOCR, when $n = 2000$. Optimal (black) represents the distribution of categories maximizing the long-term reward. We observe that IPS-PG tends to recommend items from some specific categories. This is because IPS-PG aims to maximize immediate rewards, and the optimal policy for maximizing immediate rewards is deterministic. Moreover, we can see that CR-PG and OOCR are well-calibrated compared to IPS-PG. However, CR-PG has a higher KL divergence compared to OOCR since CR-PG aligns the recommended proportions with the user's historical proportions. On the other hand, OOCR learns categorical distributions to maximize long-term rewards at Stage 1, resulting in achieving a smaller KL score.

Table 1 reports average values of KL divergence from the optimal categorical distribution across all users. In terms of average, OOCR achieves a lower KL score compared to IPS-PG and CR-PG. Although CR-PG can control the strength of calibration by λ in

Table 2: Comparisons of policy value improvements of each method compared to the logging policy in synthetic experiments. A larger value represents a significant improvement in performance.

	$n = 500$		$n = 32000$		$\gamma = 0.2$		$\gamma = 0.8$		$\|C\| = 10$		$\|C\| = 50$	
	immediate	long	immediate	long	immediate	long	immediate	long	immediate	long	immediate	long
$V(\pi_{IPS})/V(\pi_0)$	**1.91**	0.56	**2.08**	0.48	**1.95**	0.40	**1.95**	1.45	**1.95**	0.55	**1.95**	0.61
$V(\pi_{CR})/V(\pi_0)$	1.87	0.64	1.97	0.68	1.89	0.50	1.89	1.44	1.90	0.63	1.86	0.69
$V(\pi_{OOCR})/V(\pi_0)$	1.73	**1.22**	1.78	**1.06**	1.73	**1.14**	1.73	**1.54**	1.74	**1.17**	1.71	**1.19**

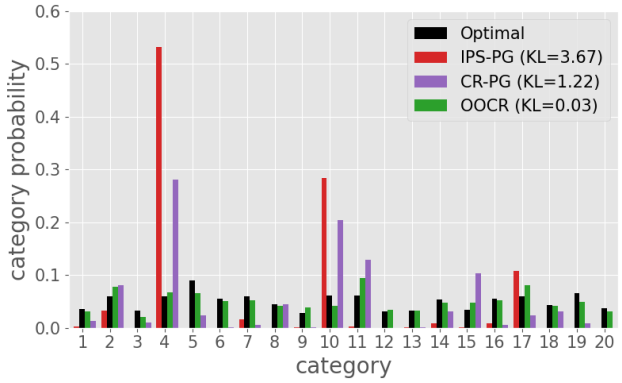

Figure 3: Categorical proportion of an optimal (black), IPS-PG (red), CR-PG (purple), and OOCR (green). The values in parentheses indicate KL divergence from the optimal.

Table 1: Comparisons of KL divergence from optimal distribution of categories. Values outside and inside the parentheses are the average and the standard deviation across all users.

	KL divergence
IPS-PG	6.57 (2.83)
CR-PG ($\lambda = 0.3$)	4.00 (2.00)
CR-PG ($\lambda = 0.7$)	2.93 (1.50)
CR-PG ($\lambda = 1.0$)	1.75 (1.19)
OOCR	**0.31 (0.13)**

Eq. (5), we can see that the KL score of CR-PG is higher than OOCR even with a larger value of λ.

How does OOCR perform with varying training data sizes? Figure 4 varies the training data sizes (n) from 500 to 32000. The results demonstrate that OOCR can improve long-term rewards compared to the logging policy across various training data sizes. This is because OOCR optimizes its categorical proportion to maximize long-term rewards using the two-stage approach. On the other hand, IPS-PG and CR-PG reduce long-term rewards compared to the logging policy. IPS-PG tends to recommend very few kinds of items in order to maximize immediate rewards, leading to decreasing long-term rewards. Although CR-PG is a calibrated recommendation approach, it does not directly maximize long-term rewards, which is why it does not significantly improve long-term rewards. Specifically, from Table 2, we see that OOCR provides approximately 5% to 20% improvement in long-term rewards compared to the logging policy, but IPS-PG and CR-PG result in decreases of approximately -50% and -40%, respectively. In addition to OOCR's improvement in long-term rewards, OOCR also provides approximately 70% improvement in immediate rewards compared to the logging policy, while its improvement is slightly smaller than IPS-PG and CR-PG. These results show that OOCR can improve both immediate and long-term rewards compared to the logging policy.

How does OOCR perform when varying the influence of immediate rewards on long-term rewards? Figure 5 reports the policy values of the methods varying the influence (γ) of immediate rewards on long-term rewards. The larger values of γ present a larger effect of immediate rewards on long-term rewards. We observe that OOCR improves long-term rewards compared to the logging policy across various values of γ. Compared to IPS-PG and CR-PG, OOCR provides substantial improvements in long-term rewards when γ is small, and the difference becomes smaller as γ increases. More specifically, compared to IPS-PG and CR-PG, OOCR provides over 200% improvement in long-term rewards when $\gamma = 0.2$, but 6% improvement when $\gamma = 0.8$. This is because maximizing immediate rewards is almost equal to maximizing long-term rewards when γ is large. Moreover, it is important to note that OOCR achieves higher long-term rewards than IPS-PG and CR-PG even when γ is set to a large value except for $\gamma = 1.0$. This result demonstrates that OOCR can perform effective OPL even when the long-term and immediate rewards are almost the same.

How does OOCR perform with varying numbers of categories? Figure 6 reports the policy values of the methods varying the number of categories from 10 to 50. The number of actions is fixed to $|\mathcal{A}| = 300$. We observe that OOCR is superior to the logging policy regarding long-term rewards. It is important to note that the long-term reward of each method gradually decreases as the number of categories increases. This is because it is more difficult to align categorical proportions with true ones when the number of categories is large. In the situation where there are many categories, the number of actions in a specific category becomes small. This means we need to optimize individual action probability, resulting in making the optimization of categorical proportions more challenging.

5 Simulation on Real-World Data

This section conducts an OPL experiment with a comprehensive simulator for the recommender system called KuaiSim [28], which simulates a reinforcement learning-based recommender system.

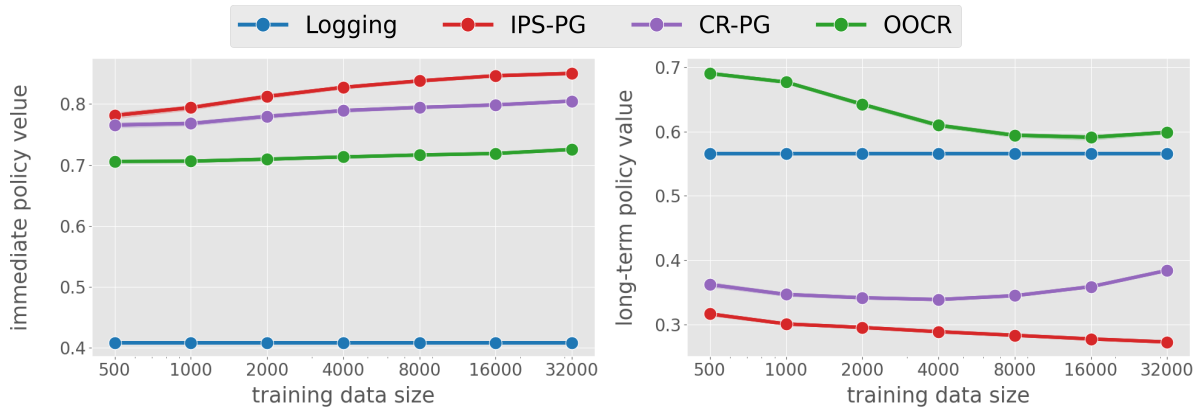

Figure 4: Comparison of the immediate (left) and long-term (right) policy values with varying training data sizes.

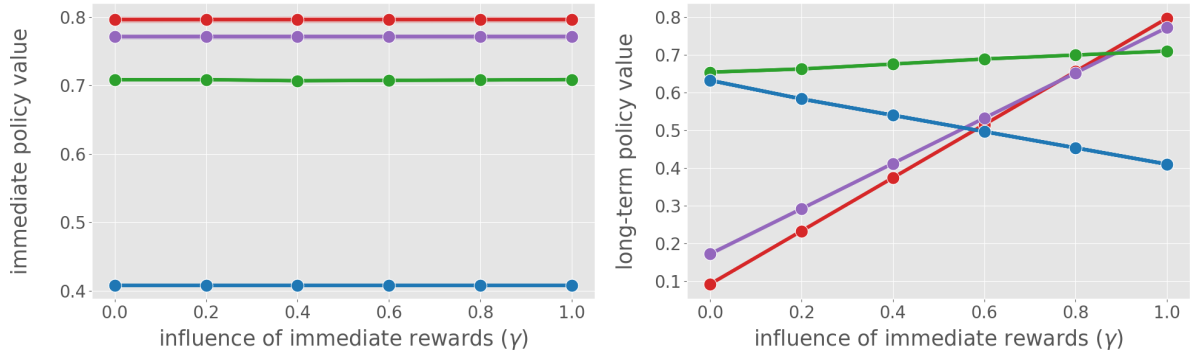

Figure 5: Comparison of the immediate (left) and long-term (right) policy values with varying values of γ in Eq. (21).

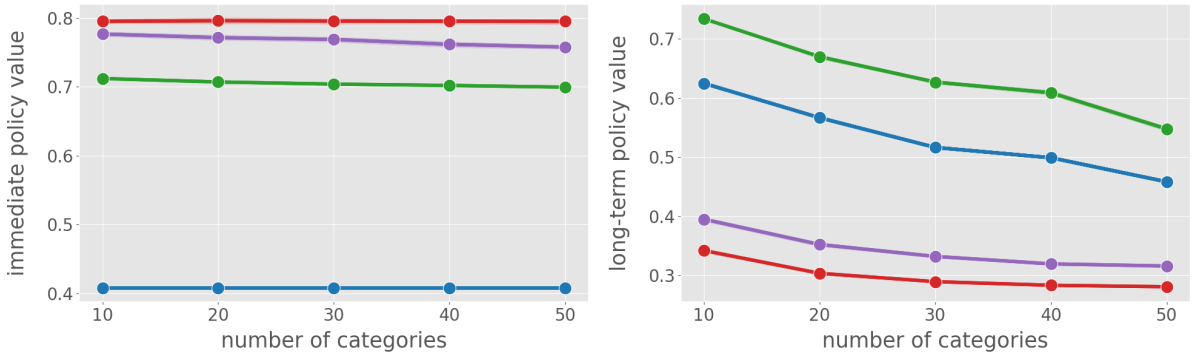

Figure 6: Comparison of the immediate (left) and long-term (right) policy values with varying number of categories.

KuaiSim is constructed on KuaiRand [7] dataset, which is a large-scale recommendation dataset collected by the video-sharing app, KuaiShou. KuaiSim simulates a process where a user arrives on the platform, receives multiple recommended items within a session, and subsequently leaves the platform. The key property of KuaiSim is that we can observe both immediate and long-term rewards. In this simulator, we observe a click signal for each recommended item as the immediate reward (r). Additionally, at the end of each session, we observe a return day signal, which indicates the number of days until the user returns. We consider this return day signal the long-term reward (l). Therefore, the ideal policy maximizes immediate rewards while minimizing long-term rewards.

To conduct an OPL experiment with this simulator, we define the logging policy as follows.

$$\pi_0(a|x) = \frac{\exp\left(\beta \cdot A(x,a)\right)}{\sum_{a' \in \mathcal{A}} \exp\left(\beta \cdot A(x,a')\right)} \tag{25}$$

where $\beta = 3.0$ and $A(x,a)$ is sampled from the standard normal distribution. Based on the above logging policy, we collect the immediate and long-term logged datasets using this simulator. In

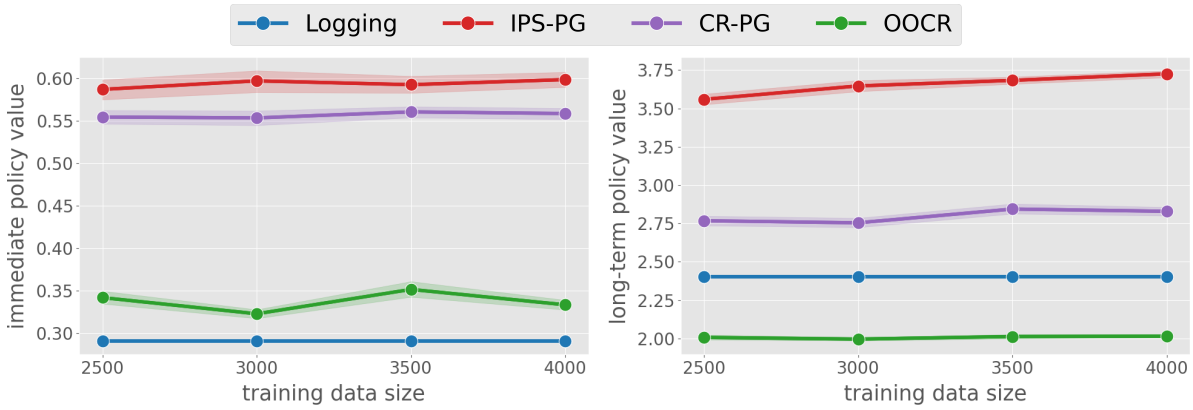

Figure 7: Comparison of the immediate (left) and long-term (right) policy values with varying training data sizes on KuaiSim. Larger immediate rewards and smaller long-term rewards represent better performance.

this experiment, the number of user visits until the end of a session is set to 5, so the immediate logged data size is five times larger than the long-term logged data size. Similar to synthetic experiments, the logged data size for immediate rewards is referred to as the training data size.

To evaluate the performance of our method in situations where calibration should be considered, we construct a setting where the longer-term rewards improve as the recommended items become more diverse. See Appendix A.2 for details of the setup.

Result. Figure 7 compares the policy learning effectiveness by the resulting policy values of each method with varying training data sizes on KuaiSim. Larger immediate policy values and smaller long-term policy values represent a greater enhancement. From Figure 7, we observe that OOCR brings in a significant improvement in long-term rewards compared to the logging policy, while IPS-PG and CR-PG reduce long-term rewards compared to the logging policy. Specifically, OOCR provides approximately 16% improvement in long-term rewards compared to the logging policy.

6 Related Works

6.1 Calibration in Recommendations

Steck [19] introduced calibrated recommendations (CR), proposing that recommended item categories should match the proportions observed in users' historical interactions. This idea has quickly gained traction [6, 9, 10, 18], as it can prevent the imbalance of recommended categories that results from algorithms maximizing immediate rewards. However, CR relies on a heuristic and unjustified assumption that aligning recommendations with historical category proportions is optimal. In practice, some users prefer a stronger focus on their dominant interests, while others desire more diverse content. Yet, the existing studies often treat CR as an objective [1, 6, 10, 11, 18].

In this work, we argue that we should not rely on such a heuristic, and we define the ideal categorical proportions by leveraging long-term rewards. We propose to learn the ideal categorical proportions by optimizing for measurable long-term outcomes, such as subscription retention or repeated platform visits, rather than relying on predefined heuristics.

6.2 Off-Policy Learning

Existing OPL methods mostly focus only on optimizing the rewards that can be observed immediately and directly against a recommendation. However, as argued in our paper and many of the existing studies [13, 19, 30], optimizing only the immediate rewards may lead to unbalanced recommendations that may lead to the dissatisfaction of the user. In this paper, we propose to also optimize the categorical proportions towards long-term rewards.

Although the long-term rewards may seem to be addressed in previous studies [14, 22], the long-term rewards that are targeted in existing studies are delayed rewards corresponding to a single action, while the long-term rewards that we present in this paper correspond to a batch of recommendations. This makes our problem difficult and crucially different from any of the existing studies.

7 Conclusion

Calibrated recommendation (CR) aims to align the categorical proportions of recommended items with those observed in users' historical interactions. However, we must recall that this is not an objective, but only a method to achieve long-term platform objectives, like retaining users and subscription renewals. Since CR is only a round guess of a good categorical proportion, we argue that CR is not a suitable concept to achieve platform objectives. Thus, we propose to explicitly define Objective-Driven CR (OCR), which sets an ideal categorical proportion using long-term rewards like user retention, which directly lead to platform success. To achieve this, we introduced a novel two-stage contextual bandit approach called OOCR. OOCR learns categorical proportions that explicitly maximize long-term batch-level rewards, and then it also optimizes immediate user interactions (e.g., clicks). Our experiments demonstrate that OOCR can accurately align the recommendation proportions to the optimal proportions and outperforms CR in achieving long-term objectives.

Ethical Considerations

This research follows responsible scientific practices and complies with all relevant ethical guidelines. Experiments were conducted using only public or institutionally authorized datasets with personally identifiable information removed. All methods were developed for beneficial applications such as improving user satisfaction and platform service quality.

References

[1] Himan Abdollahpouri, Zahra Nazari, Alex Gain, Clay Gibson, Maria Dimakopoulou, Jesse Anderton, Benjamin Carterette, Mounia Lalmas, and Tony Jebara. 2023. Calibrated Recommendations as a Minimum-Cost Flow Problem. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining* (Singapore, Singapore) *(WSDM '23)*. Association for Computing Machinery, New York, NY, USA, 571–579. https://doi.org/10.1145/3539597.3570402

[2] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. 2009. Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining* (Barcelona, Spain) *(WSDM '09)*. Association for Computing Machinery, New York, NY, USA, 5–14. https://doi.org/10.1145/1498759.1498766

[3] Qingpeng Cai, Shuchang Liu, Xueliang Wang, Tianyou Zuo, Wentao Xie, Bin Yang, Dong Zheng, Peng Jiang, and Kun Gai. 2023. Reinforcing User Retention in a Billion Scale Short Video Recommender System. arXiv:2302.01724 [cs.LG] https://arxiv.org/abs/2302.01724

[4] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep Neural Networks for YouTube Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems* (Boston, Massachusetts, USA) *(RecSys '16)*. Association for Computing Machinery, New York, NY, USA, 191–198. https://doi.org/10.1145/2959100.2959190

[5] Abhinandan S. Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. 2007. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th International Conference on World Wide Web* (Banff, Alberta, Canada) *(WWW '07)*. Association for Computing Machinery, New York, NY, USA, 271–280. https://doi.org/10.1145/1242572.1242610

[6] Farzad Eskandanian and Bamshad Mobasher. 2020. Using Stable Matching to Optimize the Balance between Accuracy and Diversity in Recommendation. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization (UMAP '20)*. ACM, 71–79. https://doi.org/10.1145/3340631.3394858

[7] 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.

[8] Thorsten Joachims, Adith Swaminathan, and Maarten de Rijke. 2018. Deep Learning with Logged Bandit Feedback. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SJaP_-xAb

[9] Jon Kleinberg, Emily Ryu, and Éva Tardos. 2024. Calibrated Recommendations for Users with Decaying Attention. arXiv:2302.03239 [cs.DS] https://arxiv.org/abs/2302.03239

[10] Kun Lin, Masoud Mansoury, Farzad Eskandanian, Milad Sabouri, and Bamshad Mobasher. 2024. Beyond Static Calibration: The Impact of User Preference Dynamics on Calibrated Recommendation. In *Adjunct Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization* (Cagliari, Italy) *(UMAP Adjunct '24)*. Association for Computing Machinery, New York, NY, USA, 86–91. https://doi.org/10.1145/3631700.3664869

[11] Mohammadmehdi Naghiaei, Mahdi Dehghan, Hossein A. Rahmani, Javad Azizi, and Mohammad Aliannejadi. 2024. Personalized Beyond-accuracy Calibration in Recommendation. In *Proceedings of the 2024 ACM SIGIR International Conference on Theory of Information Retrieval* (Washington DC, USA) *(ICTIR '24)*. Association for Computing Machinery, New York, NY, USA, 107–116. https://doi.org/10.1145/3664190.3672507

[12] Mohammadmehdi Naghiaei, Hossein A. Rahmani, Mohammad Aliannejadi, and Nasim Sonboli. 2022. Towards Confidence-aware Calibrated Recommendation. arXiv:2208.10192 [cs.IR] https://arxiv.org/abs/2208.10192

[13] Noveen Sachdeva, Yi Su, and Thorsten Joachims. 2020. Off-policy bandits with deficient support. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 965–975.

[14] Yuta Saito, Himan Abdollahpouri, Jesse Anderton, Ben Carterette, and Mounia Lalmas. 2024. Long-term Off-Policy Evaluation and Learning. arXiv:2404.15691 [cs.LG] https://arxiv.org/abs/2404.15691

[15] Yuta Saito and Thorsten Joachims. 2022. Off-Policy Evaluation for Large Action Spaces via Embeddings. arXiv:2202.06317 [cs.LG] https://arxiv.org/abs/2202.06317

[16] Tetsuya Sakai and Ruihua Song. 2011. Evaluating diversified search results using per-intent graded relevance. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Beijing, China) *(SIGIR '11)*. Association for Computing Machinery, New York, NY, USA, 1043–1052. https://doi.org/10.1145/2009916.2010055

[17] Sinan Seymen, Himan Abdollahpouri, and Edward C. Malthouse. 2021. A Constrained Optimization Approach for Calibrated Recommendations. In *Proceedings of the 15th ACM Conference on Recommender Systems* (Amsterdam, Netherlands) *(RecSys '21)*. Association for Computing Machinery, New York, NY, USA, 607–612. https://doi.org/10.1145/3460231.3478857

[18] Nasim Sonboli, Farzad Eskandanian, Robin Burke, Weiwen Liu, and Bamshad Mobasher. 2020. Opportunistic Multi-aspect Fairness through Personalized Re-ranking. arXiv:2005.12974 [cs.IR] https://arxiv.org/abs/2005.12974

[19] Harald Steck. 2018. Calibrated recommendations. In *Proceedings of the 12th ACM Conference on Recommender Systems* (Vancouver, British Columbia, Canada) *(RecSys '18)*. Association for Computing Machinery, New York, NY, USA, 154–162. https://doi.org/10.1145/3240323.3240372

[20] Yi Su, Maria Dimakopoulou, Akshay Krishnamurthy, and Miroslav Dudík. 2020. Doubly robust off-policy evaluation with shrinkage. arXiv:1907.09623 [cs.LG]

[21] Yi Su, Lequn Wang, Michele Santacatterina, and Thorsten Joachims. 2019. CAB: Continuous Adaptive Blending Estimator for Policy Evaluation and Learning. arXiv:1811.02672 [cs.LG] https://arxiv.org/abs/1811.02672

[22] Rikiya Takehi, Masahiro Asami, Kosuke Kawakami, and Yuta Saito. 2025. A General Framework for Off-Policy Learning with Partially-Observed Reward. In *The Thirteenth International Conference on Learning Representations*. https://openreview.net/forum?id=mUbYof5MKp

[23] Masatoshi Uehara, Chengchun Shi, and Nathan Kallus. 2022. A review of off-policy evaluation in reinforcement learning. *arXiv preprint arXiv:2212.06355* (2022).

[24] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative Deep Learning for Recommender Systems. arXiv:1409.2944 [cs.LG] https://arxiv.org/abs/1409.2944

[25] Yuyan Wang, Mohit Sharma, Can Xu, Sriraj Badam, Qian Sun, Lee Richardson, Lisa Chung, Ed H. Chi, and Minmin Chen. 2022. Surrogate for Long-Term User Experience in Recommender Systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Washington DC, USA) *(KDD '22)*. Association for Computing Machinery, New York, NY, USA, 4100–4109. https://doi.org/10.1145/3534678.3539073

[26] Kelly W. Zhang, Thomas Baldwin-McDonald, Kamil Ciosek, Lucas Maystre, and Daniel Russo. 2025. Impatient Bandits: Optimizing for the Long-Term Without Delay. arXiv:2501.07761 [cs.LG] https://arxiv.org/abs/2501.07761

[27] Qihua Zhang, Junning Liu, Yuzhuo Dai, Yiyan Qi, Yifan Yuan, Kunlun Zheng, Fan Huang, and Xianfeng Tan. 2022. Multi-Task Fusion via Reinforcement Learning for Long-Term User Satisfaction in Recommender Systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22)*. ACM, 4510–4520. https://doi.org/10.1145/3534678.3539040

[28] Kesen Zhao, Shuchang Liu, Qingpeng Cai, Xiangyu Zhao, Ziru Liu, Dong Zheng, Peng Jiang, and Kun Gai. 2023. KuaiSim: A comprehensive simulator for recommender systems. *Advances in Neural Information Processing Systems* 36 (2023), 44880–44897.

[29] Xiangyu Zhao, Liang Zhang, Long Xia, Zhuoye Ding, Dawei Yin, and Jiliang Tang. 2019. Deep Reinforcement Learning for List-wise Recommendations. arXiv:1801.00209 [cs.LG] https://arxiv.org/abs/1801.00209

[30] Yuying Zhao, Yu Wang, Yunchao Liu, Xueqi Cheng, Charu Aggarwal, and Tyler Derr. 2024. Fairness and Diversity in Recommender Systems: A Survey. arXiv:2307.04644 [cs.IR] https://arxiv.org/abs/2307.04644

[31] Lixin Zou, Long Xia, Zhuoye Ding, Jiaxing Song, Weidong Liu, and Dawei Yin. 2019. Reinforcement Learning to Optimize Long-term User Engagement in Recommender Systems. arXiv:1902.05570 [cs.IR] https://arxiv.org/abs/1902.05570

A Detailed Experiment Settings and Results

This section describes the detailed experiment settings and reports additional results.

A.1 Synthtic Experiments

In this section, we report additional synthetic experiments.

Additional Results. In order to evaluate OOCR at each stage, Table 3 reports the KL score between (i) the optimal categorical proportions $\mu^*(c|x)$ and $\pi(c|x)$ learned by OOCR, (ii) the optimal categorical proportions $\mu^*(c|x)$ and the estimated categorical proportion $\mu(c|x)$ learned by Stage 1, and (iii) the estimated categorical proportion $\mu(c|x)$ learned by Stage 1 and $\pi(c|x)$ learned by OOCR. Additionally, Figure 8 varies the logging policy (β) in Eq. (20). The training data size is set to $n = 1000$, the number of actions is $|\mathcal{A}| = 300$, and the influence of the immediate reward within the long-term reward is $\gamma = 0.0$ in Table 3, and $\gamma = 0.3$ in Figure 8.

Table 3: KL divergence of the pairs (i), (ii) and (iii).

	$\|C\| = 10$	$\|C\| = 20$	$\|C\| = 30$	$\|C\| = 40$	$\|C\| = 50$
(i) $\mathrm{KL}(\mu^*\|\|\pi)$	0.25	0.35	0.47	0.52	0.63
(ii) $\mathrm{KL}(\mu^*\|\|\mu)$	0.21	0.30	0.39	0.43	0.57
(iii) $\mathrm{KL}(\mu\|\|\pi)$	0.12	0.21	0.32	0.37	0.48

How is the KL divergence of (i), (ii) and (iii)? Table 3 shows that the KL scores between (i) the optimal categorical proportions $\mu^*(c|x)$ and $\pi(c|x)$ learned by OOCR, (ii) the optimal categorical proportions $\mu^*(c|x)$ and the estimated categorical proportion $\mu(c|x)$ learned by Stage 1, and (iii) the estimated categorical proportion $\mu(c|x)$ learned by Stage 1 and $\pi(c|x)$ learned by OOCR. KL scores of (i) represent final KL scores which we would like to minimize. From Table 3, we can see that KL scores of (i) gradually increase as the number of categories increases. This observation is consistent with Figure 6. Moreover, we can evaluate each stage of OOCR with KL scores of (ii) and (iii). KL scores of (ii) represent the performance of Stage 1, while KL scores of (iii) represent the performance of Stage 2. We observe that KL scores of (ii) are larger than (iii). The logged data size of long-term rewards is much smaller than that of immediate rewards, resulting in making the estimation of category proportions difficult. In addition to small training data sizes, the large category-proportion vector space makes Stage 1 of OOCR challenging.

How does OOCR perform with varying the logging policy (β) in Eq. (20)? Figure 8 varies the logging policy (β) in Eq. (20). The logging policy becomes an optimal policy for immediate rewards as the value of β increases. From Figure 8, we observe that OOCR improves long-term rewards compared to the logging policy with smaller values of β, while OOCR has slightly smaller long-term policy values with larger values of β. When β is large, the immediate policy value of the logging policy is larger than that of OOCR, resulting in the large long-term policy value of the logging policy. It is important to note that OOCR outperforms IPS-PG and CR-PG and achieves stable improvements in immediate rewards across various logging policies. These results suggest that OOCR can conduct effective OPL with any logging policies.

A.2 Real-World Experiments

In this section, we describe the detailed real-world experiment settings in the main text and conduct an additional real-world experiment.

Detailed Setup. We describe the real-world experiment settings on KuaiSim [28] in detail. KuaiSim supports the slate setting, but we consider a single-action setting, so we set the slate size to 1. KuaiSim has 19574 users, so we then define 19574 users characterized by 5-dimensional context vectors (x) sampled from the standard normal distribution. We assign each action a to a category c using k-means clustering. The number of actions is set to $|\mathcal{A}| = 300$ and the number of clusters is $|C| = 20$.

As described in the main text, the key property of KuaiSim is that we can observe both immediate and long-term rewards. In this simulator, we observe a click signal for each recommended item as the immediate reward (r). Additionally, at the end of each session, we observe a return day signal, which indicates the number of days until the user returns. We consider this return day signal the long-term reward (l). To obtain return day signals, KuaiSim calculates return probability p_{ret} as follows.

$$p_{ret} = b_u + \eta r + b_{cali} \tag{26}$$

where b_u is a retention bias depending on user states. We set $\eta = 0.01$ in the main text. Given p_{ret}, the d-th day return probability is $p_{ret} \cdot (1 - p_{ret})^{d-1}$. If the return day probability p_{ret} takes a larger value, we obtain a smaller return day signal. b_{cali} is a parameter we introduce in this experiment in order to construct situations where calibration should be considered. Note that b_{cali} is a constant and the

same for all users in the standard configuration of KuaiSim. We define b_{cali} as follows.

$$d = \frac{1}{T}\sum_{i=1}^{T-1} \| x_a^i - x_a^{i+1} \|_2 \tag{27}$$

$$b_{cali} = \tanh(d + \zeta) \tag{28}$$

where T is the length of a session and x_a^i represents i-th action context within a session. We sample x_a from the standard normal distribution, and the dimension of action context is set to 7. Note that we use the hyperbolic tangent to scale b_{cali} from -1 to 1. The length of a session is set to $T = 5$ in the main text. The calibration bias b_{cali} takes a larger value if recommended actions within a session are diverse. ζ is an experimental parameter which decides a threshold for the degree of calibration that has a positive impact on long-term rewards. We use $\zeta = -0.5$ in the main text.

Additional Results. In Figure 9 and 10, we report additional results on real-world experiments in order to demonstrate that OOCR works better than the baselines in various situations. We vary the threshold (ζ) in Eq. (28) and the influence of immediate rewards on long-term rewards (η) in Eq. (26). We set $n = 3000$ and $|\mathcal{A}| = 300$. Note that higher values represent a larger improvement regarding immediate rewards, while lower values represent a larger improvement regarding long-term rewards.

How does OOCR perform with varying the threshold (ζ) in Eq. (28) and the influence of immediate rewards on long-term rewards (η) in Eq. (26)? First, Figure 9 shows comparisons of the resulting policy values with varying the threshold (ζ) in Eq. (28). ζ decides a threshold for the degree of calibration that has a positive impact on long-term rewards. If the degree of calibration d in Eq. (27) is larger than $-\zeta$, b_{cali} takes a positive value, leading to a positive effect on long-term rewards. We observe that OOCR achieves substantial improvements in long-term rewards across various values of ζ. We can also see that all methods improve long-term policy values as the threshold ζ increases. This is because the large threshold ζ makes the calibration bias b_{cali} take larger values, leading to small long-term rewards.

Second, Figure 10 varies the influence of immediate rewards on long-term rewards (η) in Eq. (26). A larger value of η indicates that immediate rewards have a large influence on long-term rewards. We observe that OOCR improves long-term rewards compared to the logging policy. Also, OOCR outperforms IPS-PG and CR-PG across various influences of immediate rewards. This result indicates that OOCR can improve various kinds of long-term rewards.

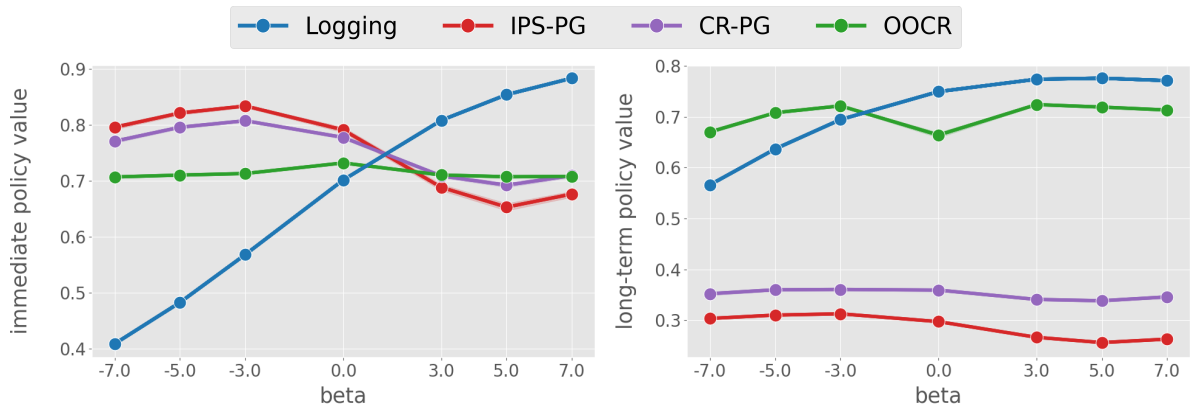

Figure 8: Comparison of the immediate (left) and long-term (right) policy values with varying the logging policy (β) in Eq. (20).

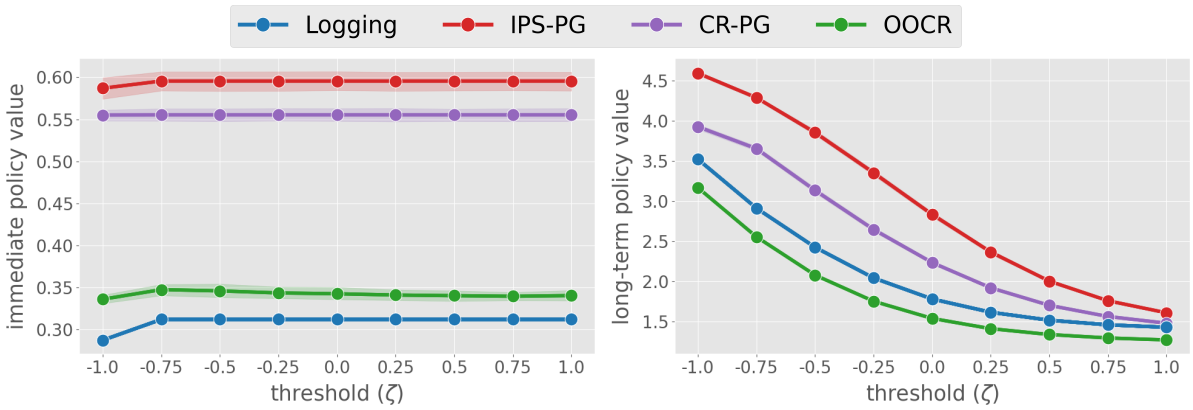

Figure 9: Comparison of the immediate (left) and long-term (right) policy values with varying threshold (ζ) in Eq. (28).

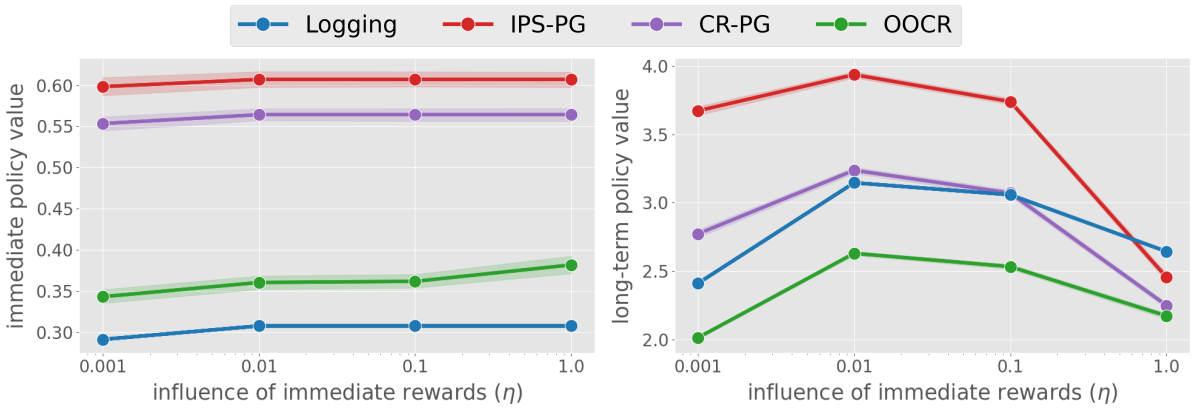

Figure 10: Comparison of the immediate (left) and long-term (right) policy values with varying influence of immediate rewards (η) in Eq. (26).