Values of User Exploration in Recommender Systems

Minmin Chen, Yuyan Wang, Can Xu, Elaine Le, Mohit Sharma, Lee Richardson, Su-Lin Wu, Ed Chi
minminc,yuyanw,canxu,elainele,mohitsharma,leerich,sulin,edchi@corporation.com
Google
Mountain View, California, USA

ABSTRACT

Reinforcement Learning (RL) has been sought after to bring next-generation recommender systems to further improve user experience on recommendation platforms. While the exploration-exploitation tradeoff is the foundation of RL research, the value of exploration in (RL-based) recommender systems is less well understood. Exploration, commonly seen as a tool to reduce model uncertainty in regions of sparse user interaction/feedback, is believed to cost user experience in the short term, while the indirect benefit of better model quality arrives at a later time. We focus on another aspect of exploration, which we refer to as user exploration to help discover new user interests, and argue it can improve user experience even in the more imminent term.

We examine the role of user exploration in changing different facets of recommendation quality that more directly impact user experience. To do so, we introduce a series of methods inspired by exploration research in RL to increase user exploration in an RL-based recommender system, and study their effect on the end recommendation quality, more specifically, on *accuracy, diversity, novelty and serendipity*. We propose a set of metrics to measure (RL based) recommender systems in these four aspects and evaluate the impact of exploration-induced methods against these metrics. In addition to the offline measurements, we conduct live experiments on an industrial recommendation platform serving billions of users to showcase the benefit of user exploration. Moreover, we use conversion of casual users to core users as an indicator of the holistic long-term user experience and study the values of user exploration in helping platforms convert users. Through offline analyses and live experiments, we study the correlation between these four facets of recommendation quality and long term user experience, and connect serendipity to improved long term user experience.

CCS CONCEPTS

• **Theory of computation → Reinforcement learning**; • **Information systems → Personalization**.

KEYWORDS

reinforcement learning, exploration, serendipity, recommender systems

RecSys '21, September 27-October 1, 2021, Amsterdam, Netherlands

ACM ISBN 978-1-4503-8458-2/21/09.
https://doi.org/10.1145/3460231.3474236

ACM Reference Format:
Minmin Chen, Yuyan Wang, Can Xu, Elaine Le, Mohit Sharma, Lee Richardson, Su-Lin Wu, Ed Chi. 2021. Values of User Exploration in Recommender Systems. In *Fifteenth ACM Conference on Recommender Systems (RecSys '21), September 27-October 1, 2021, Amsterdam, Netherlands.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3460231.3474236

1 INTRODUCTION

In the era of increasing choices, recommender systems are becoming indispensable in helping users navigate the million or billion pieces of contents available on recommendation platforms. These systems are built to satisfy users' information needs by anticipating what they would be interested in consuming next. Collaborative filtering [28, 47] and supervised learning based approaches predicting users' immediate response toward recommendations [12, 65] such as clicks, dwell time, likes, have had enormous successes. Researchers however are becoming increasingly aware of the limitations of such approaches. First, focus on driving short-term engagements such as user clicks fails to account for the long term impact of a recommendation. Second, lack of exploration causes these systems to increasingly concentrate on the known user interests and create satiation effect, i.e., reduced enjoyment of the content.

Reinforcement learning (and bandits) techniques have emerged as appealing alternatives [11, 23, 67, 68] over the years. Compared with supervised learning based approaches, RL offers two advantages: 1) Exploration. (Online) RL algorithms inherently explore regions they are less certain about. This provides a natural mechanism to deviate from the current system behavior, and introduce previously unseen contents to users; 2) Long-term user experience optimization. As the planning horizon of these RL agents extends, the recommender naturally shifts its focus from short-term user engagement toward optimizing the long-term user experience on the platform. We focus our discussion on exploration, though as we show in the analyses it innately connects to the long-term user experience.

The tradeoff between exploration and exploitation is central to the design of RL agents [17, 57]. An agent learns to form a policy to maximize returns in a changing environment by taking actions and receiving reward/feedback from the environment. The agent is incentivized to exploit, repeating actions taken in the past that produced higher rewards, to maximize reward. On the other hand, the agent needs to explore previously unseen actions in order to discover potentially better options. Exploration in RL based recommender systems serves a similar goal, that is to expose users to previously unseen items to discover contents the user is potentially interested in. The benefit of exploration to counter the selection bias of existing systems and generate training data to reduce model uncertainty has been established [11]. Here we focus on another

aspect of exploration that we refer to as user exploration, i.e., exploration for discovering something new for the user.

As exploration innately leads to recommending something less pertinent to the *known* user interests, it is often seen as a cost to user experience, especially in the short term. Here we argue that recommender systems have an inherent need for exploration as users perceive other factors of recommendation quality besides accuracy [5, 66]. We dissect the values of user exploration by examining its role in changing different aspects of recommendation quality that impact the user experience on recommendation platforms. Together, we make the following contributions:

- **Methods to Introduce User Exploration**: We introduce a collection of methods, inspired by exploration research in RL, to improve user exploration in recommender systems.
- **Metrics**: We propose a set of metrics measuring the different aspects of recommendation quality, that is accuracy, diversity, novelty and serendipity for RL based recommender systems.
- **Offline Analyses**: We conduct an extensive set of offline analyses to understand the values of user exploration in changing the four aspects of recommendation quality.
- **Live Experiments**: We conduct live experiments of the proposed methods on a commercial recommendation platform serving billions of users and millions of items, and showcase the value of user exploration in improving long-term user experience on the platform.
- **Serendipity for Long Term User Experience**: Through offline analyses and live experiments, we study the correlation between these four aspects of recommendation quality and the long term user experience. Using conversion of casual users to core users as an indicator of the holistic long term user experience, we connect serendipity to improved long term user experience.

2 RELATED WORK

Reinforcement Learning for Recommender Systems. Deep reinforcement learning, combining high-capacity function approximators, i.e., deep neural networks, with the mathematical formulations in classic reinforcement learning [57], has achieved enormous success in various domains such as games, robotics and hardware design [18, 33, 36, 52]. It has attracted a lot of attention from the recommeder system research community as well. Shani et al. [51] were among the first to formally formulate recommendation as a Markov decision process (MDP) and experiment with model-based RL approaches for book recommendation. Zheng et al. [70] applied DQN for news recommendation. Dulac-Arnold et al. [14] enabled RL in problem spaces with a large number of discrete actions and showcased its performance on various recommendation tasks with tens of thousands of actions. Liu et al. [34] tested actor-critic approaches on recommendation datasets such as MovieLens, Yahoo Music and Jester. Set recommendation using RL has been studied in [11, 23, 69]. In recent years, we also start seeing success of RL in real-world recommendation applications. Chen et al. [11] scaled a batch RL algorithm, i.e., REINFORCE with off-policy correction to a commercial platform serving billions of users and tens of millions of contents. Hu et al. [22] tested an extension of the deep deterministic policy gradient (DDPG) method for learning to rank on *Taobao*, a commercial search platform.

Exploration in Reinforcement Learning. The exploration/exploitation dilemma has long been studied in multi-armed bandits and classic reinforcement learning [17, 57]. Exploration methods are concerned with reducing agents' uncertainty of the environment reward and/or the dynamics. For the stochastic bandits problems, Upper Confidence Bound (UCB) [30] and Thompson Sampling (TS) [4, 10, 59] are among the most well known techniques with both theoretical guarantees and empirical successes. In classic reinforcement learning with tabular settings, count-based exploration techniques quantifying the uncertainty based on the inverse square root of the state-action visit count, can be seen as extension of these techniques to Markov Decision Processes (MDPs). Tang et al. [58] further generalizes counted-based methods to deep RL with high-dimensional state spaces. Another camp of methods, commonly referred to as intrinsic motivation [21, 24, 48, 56], encourages the agents to explore regions leading up to surprises. The surprise factor is often measured by the agents' predictive errors on environment reward or dynamics, or equivalently information gain the agents can acquire by taking an action under the current state. Bellemare et al. [6] unifies count-based exploration and intrinsic motivation through the lens of information gain or learning progress. Our work takes inspiration from these existing works, and re-designs the algorithms to fit more closely with the recommendation setup.

Diversity, Novelty and Serendipity of Recommender Systems. While early recommendation research has focused almost exclusively on improving recommendation accuracy, it has become increasingly recognized that there are other factors of recommendation quality contributing to the overall user experience on the platform. Herlocker et al. [19] in their seminal work of evaluating collaborative filtering based recommender systems defined various metrics to measure recommendation accuracy, coverage, novelty as well as serendipity. Diversity is another important aspect that has been extensively studied [3, 5]. Diversification algorithms are used to increase coverage of the full range of user interests, and to counter the saturation effect of consuming similar contents [72]. Zhou et al. [71] studied the dilemma between accuracy and diversity, and proposed a hybrid approach to balance the two. Novelty [8] is closely related to long tail recommendation [62], measuring the capacity of the recommender systems to make predictions and reach the full inventory of contents available on the platforms. One of the early definitions of serendipity was introduced in [19], which captures the degree to which a recommendation is both relevant and surprising to users. Zhang et al. [66] proposed a hybrid rank-interpolation approach to combine outputs of three LDA algorithms [7] focusing on either accuracy, diversity or serendipity to achieve a balance between these factors in the end recommendations. Oku and Hattori [41] proposed a fusion based technique to mix items users expressed interest on based on item attributes in order to introduce serendipitous contents. Our work measures the effect of exploration on recommendation accuracy, diversity, novelty and serendipity, and connects these factors to long term user experience.

3 BACKGROUND

We base our work on the REINFORCE recommender system introduced in [11], in which the authors framed a set recommendation problem as a Markov Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, \mathbf{P}, R, \rho_0, \gamma)$. Here $\mathcal{S}$ is the state space capturing the user interests and context, $\mathcal{A}$ is the discrete action space containing items available for recommendation, $\mathbf{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the state transition probability, and $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, with $r(s, a)$ note the immediate reward of action a under state s. ρ_0 is the initial state distribution, and γ the discount for future rewards.

Let $\mathcal{H}_t = \{(A_0, a_0, r_0), \cdots, (A_{t-1}, a_{t-1}, r_{t-1})\}$ denote an user's historical activities on the platform up to time t, where $A_{t'}$ stands for the set of items recommended to the user at time t', $a_{t'}$ denotes the item the user interacted with at t' ($a_{t'}$ can be null), and $r_{t'}$ captures the user feedback (reward) on $a_{t'}$ ($r_{t'} = 0$ if the user did not interact with any item in $A_{t'}$). The history $\mathcal{H}_t$ is encoded through a recurrent neural network to capture the latent user state, that is, $\mathbf{u}_{s_t} = \text{RNN}_\theta(\mathcal{H}_t)$. Given the latent user state, a softmax policy over the item corpus $\mathcal{A}$ is parameterized as

$$\pi_\theta(a|s_t) = \frac{\exp(\mathbf{u}_{s_t}^\top \mathbf{v}_a)}{\sum_{a' \in \mathcal{A}} \exp(\mathbf{u}_{s_t}^\top \mathbf{v}_{a'})}, \quad \forall a \in \mathcal{A} \tag{1}$$

which defines a distribution over the item corpus $\mathcal{A}$ conditioning on the user state s_t at time t. Here $\mathbf{v}_a$ stands for the embedding of the item a. The agent then generates a set of recommendation A_t to user at time t according to the learned softmax policy $\pi_\theta(\cdot|s_t)$. The policy parameters θ are learned using REINFORCE [60] so as to maximize the expected cumulative reward over the user trajectories,

$$\begin{aligned} \max_\theta \mathcal{J}(\pi_\theta) &= \mathbb{E}_{s_0 \sim \rho_0, A_t \sim \pi_\theta(\cdot|s_t), s_{t+1} \sim \mathbf{P}(s_t, A_t)} \left[\sum_{t=0}^{T} r(s_t, a_t) \right] \\ &\approx \mathbb{E}_{s_t \sim d_t^{\pi_\theta}(s), a_t \sim \pi_\theta(\cdot|s_t)} \left[R_t(s_t, a_t) \right]. \end{aligned} \tag{2}$$

where $R_t = \mathbb{I}_{r(s_t, a_t) > 0} \cdot \sum_{t'=t}^{T} \gamma^{t'-t} r(s_{t'}, a_{t'})$ is the discounted cumulative reward starting from time t.

RL was designed as an online learning paradigm in the first place [57]. Note that the expectation in eq. 2 is taken over the trajectories generated according to the learned policy, and $d_t^{\pi_\theta}(s)$ is the discounted state visitation probability under π_θ [32]. One of the main contribution of [11] is bringing the REINFORCE algorithm to the offline batch learning setup commonly deployed in industrial recommender systems. The authors applied a first-order approximation [2] of importance sampling to address the distribution shift caused by offline training, resulting in a gradient of the following:

$$\nabla_\theta \mathcal{J}(\pi_\theta) = \sum_{s_t \sim d_t^\beta(s), a_t \sim \beta(\cdot|s_t)} \left[\frac{\pi_\theta(a_t|s_t)}{\beta(a_t|s_t)} R_t(s_t, a_t) \nabla_\theta \log \pi_\theta(a_t|s_t) \right]. \tag{3}$$

Here $\beta(\cdot|s)$ denotes the behavior policy, i.e., the action distribution conditioning on state s in the batch collected trajectories. $d_t^\beta(s)$ is the discounted state visitation probability under β. This importance weight is further adapted to accommodate the set recommendation setup. We refer interested readers to [11] for more details.

To balance exploration and exploitation, a hybrid approach that returns the top K' most probable items, while sampling the rest $K - K'$ items according to π_θ (Boltzmann exploration [13]), is employed *during serving*.

4 METHOD

Here we introduce three simple methods inspired by exploration research in RL to increase user exploration in the REINFORCE recommender system *during training*. That is, to recommend content less pertinent to the known user interests, and to discover new user interests.

4.1 Entropy Regularization

The first method promotes recommending contents less pertinent to the known user interests by encouraging the policy $\pi_\theta(\cdot|s)$ to have an output distribution with high entropy [61]. Mnih et al. [38] observed that adding entropy of the policy to the objective function discourages premature convergence to sub-optimal deterministic policies and leads to better performance. Pereyra et al. [46] conducted a systemic study of entropy regularization and found it to improve a wide range of state-of-the-art models.

We add of the entropy to the RL learning objective as defined in eq. 2 during training. That is,

$$\max_\theta \mathcal{J}(\pi_\theta) + \alpha \sum_{s_t \sim d_t^\beta(s)} H\left(\pi_\theta(\cdot|s_t)\right). \tag{4}$$

where the entropy of the conditional distribution $\pi_\theta(\cdot|s)$ is defined as $H\left(\pi_\theta(\cdot|s)\right) = -\sum_{a \in \mathcal{A}} \pi_\theta(a|s) \log \pi_\theta(a|s)$. Here α controls the strength of the regularization. The entropy is equivalent to the negative reverse KL divergence of the conditional distribution $\pi_\theta(\cdot|s)$ to the uniform distribution. That is, $H\left(\pi_\theta(\cdot|s)\right) = -D_{KL}(\pi_\theta(\cdot|s)||U) + const$, where U stands for a uniform distribution across the action space $\mathcal{A}$. As we increase this regularization, it pushes the learned policy to be closer to a uniform distribution, thus promoting exploration.

4.2 Intrinsic Motivation and Reward Shaping

The second method helps discovering new user interests through reward shaping. The reward function $r(s, a)$ as defined in eq. 2, describes the (immediate) value of a recommendation a to a user s. It plays a critical role in deciding the learned policy π_θ. Reward shaping, transforming or supplying additional rewards beyond those provided by the MDP, is very effective in guiding the learning of RL agents to produce policies desired by the algorithm designers [1, 27, 40].

Exploration has been extensively studied in RL [6, 42–44, 55], and has been shown to be extremely useful in solving hard tasks, e.g., tasks with sparse reward and/or long horizons, and . These works can be roughly grouped into two categories. One concerns quantifying the uncertainty of the value function of the state-action pairs so the agent can direct its exploration on regions where it is most uncertain. The other uses a qualitative notion of curiosity or intrinsic motivation to encourage the agent to explore its environment and learn skills that might be useful later. Both camps of methods later adds an intrinsic reward $r^i(s, a)$, either capturing the uncertainty or curiosity to the extrinsic reward $r^e(s, a)$ that is emitted by the environment directly, to help the agent explore the unknown or learn new skills. That is, transforming the reward

function to

$$r(s,a) = c \cdot r^i(s,a) + r^e(s,a), \tag{5}$$

where c controls the relative importance of the intrinsic reward w.r.t. the extrinsic reward emitted by the environment.

Schmidhuber [49] formally captures the theory of creativity, fun and curiosity as an intrinsic desire to discover surprising patterns of the environment, and argues that a curiosity-driven agent can learn even in the absence of external reward. Our proposal bears the same principle by rewarding the agent more when it discovers some previously unknown patterns of the environment, that is the user. Let $R^e_t(s_t, a_t) = \mathbb{I}_{r^e(s_t,a_t)>0} \cdot \sum_{t'=t}^{T} \gamma^{t'-t} r^e(s_{t'}, a_{t'})$ be the discounted cumulation of the extrinsic reward on the state-action pair (s_t, a_t) observed on the trajectory. We then define the cumulative reward $R_t(s_t, a_t)$ used for the gradient update in eq. 3 as

$$R_t(s_t,a_t) = \begin{cases} c \cdot R^e_t(s_t,a_t) & \text{if recommending } a_t \text{ under } s_t \\ & \text{leads to discovery of previously} \\ & \text{unknown user interests;} \\ R^e_t(s_t,a_t) & \text{otherwise.} \end{cases} \tag{6}$$

Here $c > 1$ is a constant multiplier.

As explained in Section 3, the agent perceives the environment, that is the user interests and context, through encoding user's historical activities $\mathcal{H}_t = \{(A_0, a_0, r_0), \cdots, (A_{t-1}, a_{t-1}, r_{t-1})\}$. One can imagine a large update (surprise) to the agent's modeling of the environment if an item a_t recommended given the state s_t is 1) drastically different from any of the items the user interacted with in the past; 2) enjoyed by the user, i.e., $r^e(s_t, a_t)$ or $R^e(s_t, a_t)$ is high. These two conditions, *surprise* and *relevance*, align with the serendipity metrics we are going to detail in Section 5.5.

To measure the surprise of a_t, we define $\mathcal{I}_t = \{a_{t'}, \forall t' < t \text{ and } r_{t'} > 0\}$ as the set of items the user interacted with up to time t. As recommendation items are often associated with various attributes as described in Section 5.1, we use these attributes to measure the similarity (or difference) of a candidate action a_t towards $\mathcal{I}_t$. For example, we consider an item a_t surprising (different) if its topic cluster is different from any of the items in $\mathcal{I}_t$.

The multiplicative design in eq. 6 naturally accomplishes the second condition, that is, relevance. Comparing with the additive form (eq. 5), the multiplicative design results in: 1) a candidate action a_t with zero extrinsic reward, i.e., $R^e_t(s_t, a_t) = 0$ will NOT receive any additional reward even if being under-surfaced; 2) an action a_t receiving higher extrinsic reward $R^e_t(s_t, a_t)$ will be rewarded even more compared with those that are equally surprising but received lower extrinsic reward. This contrasts with the additive form where the extrinsic rewards observed does not influence the intrinsic reward. In other words, the additive design gives a uniform boost to actions based entirely on surprise. The multiplicative design on the other end, favors surprising actions that actually lead to improved user experience, indicated by higher extrinsic reward.

4.3 Actionable Representation for Exploration

The third method reinforces the newly discovered user interest through representation learning. Learning effective representation is critical to improve the sample efficiency of many machine learning algorithms, and RL is no exception. Most prior work on representation learning for RL has focused on generative approaches, learning representations that capture all underlying factors of variation in the observation space in a more disentangled or well-ordered manner. Self-supervised learning [20, 25, 50, 54] to capture the full dynamics of the environment has also attracted a lot of attentions lately. Ghosh et al. [16] instead argue to learn functionally salient representations: representations that are not necessarily complete in terms of capturing all factors of variation in the observation space, but rather aim to capture those factors of variation that are important for decision making – that are "actionable."

The REINFORCE agent introduced in Section 3 describes the environment, i.e., the user, through encoding his/her historical activities $\mathcal{H}_t$. That is, $\mathbf{u}_{s_t} = \text{RNN}_\theta(\mathcal{H}_t)$. When an user interacted with a surprising item a_t (to the agent) and gave high reward, the user state $\mathbf{u}_{s_t}$ should be updated to capture the new information so the agent can act differently next. That is, to make recommendations according to the newly acquired information about the new interest of the user. To aid the agent in capturing this information in its state, we extend $\mathcal{H}_t$ with an additional bit, indicating whether or not an item the user interacts with is surprising and relevant. That is, we expand $\mathcal{H}_t = \{(A_0, a_0, r_0, i_0), \cdots, (A_{t-1}, a_{t-1}, r_{t-1}, i_{t-1})\}$, where $i_{t'} = 1$ if 1) the attribute of a_t (such as topic cluster) is different from that of any items in $\mathcal{I}_{t'}$ (being a surprise) and; 2) $r_t > 0$ (being relevant). Here $\mathcal{I}_{t'}$ is the list of items the user has interacted with up to time t'. This feature is then embedded and consumed by the RNN along with other features describing the item a_t.

5 MEASUREMENT

Personalization has been the cornerstone of modern recommender systems. It aims to produce targeted and accurate recommendations based on user historical activities. Overly focusing on the accuracy aspect of recommendation, however, runs the risk of exposing users only to a concentrated set of contents. This could attract user attention in the near term, but likely hurt user experience in the long run. There has been a growing body of work examining factors other than accuracy in shaping user's perception of recommendation quality [9, 19, 35, 66, 72, 72]. In particular, aspects such as diversity, novelty and serendipity of recommendations have been studied. Here we design metrics to measure these four aspects for a RL based recommender system. Some of the metrics measure directly on the learned policy π_θ, and thus apply only to systems producing a distribution over the content vocabulary. Others measure on the recommendation set A^{π_θ} generated by acting according to π_θ (taking most probable items) [1], which are generic for any types of recommender systems [2]. These metrics bear similarity to many prior works in quantifying the four factors of recommendation quality [5, 8, 15, 19, 26, 29, 66].

5.1 Attributes

We first introduce two item attributes that are used to define both the surprise factor in eq (6) as well as the metrics:

[1] We employed the hybrid policy, i.e., choosing top K' and sample $K - K'$ according to π_θ as explained at the end of Section 3 for all the live experiments. For offline comparisons, we focus on the top K' items to reduce randomness introduced through sampling.

[2] Note that for all the metrics we defined, it is straight-forward to define the policy-based or set-based counterpart, and the trends observed in offline analyses are consistent between these two sets of metrics, we thus only report one to save space.

Topic cluster. A topic cluster for each item is produced by: 1) taking the item co-occurrence matrix, where entry (i, j) counts the number of times item i and j were interacted by the same user consecutively; 2) performing matrix factorization to generate one embedding for each item; 3) using k-means to cluster the learned embeddings into $10K$ clusters; 4) assigning the nearest cluster to each item.

Content provider. Content provider is another attribute of interest as: 1) we observed consistency between contents produced by the same provider, e.g., a food blogger often writes about specific cuisines; 2) we are interested in understanding the importance of content-provider diversity/novelty [37, 64] in influencing long term user experience.

5.2 Accuracy

Arguably the most important property of a recommender is to be able to retrieve contents the user is interested in consuming. We compute the mean average precision at $K = 50$ ($mAP@50$) [63] on the recommended set A^{π_θ} to measure the accuracy, that is the average precision of identifying an item the user is interested in consuming among A^{π_θ}.

5.3 Diversity

Diversity measures the number of distinct faucets the recommendation set contains. Many measurements of set diversity have been proposed [39, 45, 53]. Among them, the average dissimilarity of all pairs of items in the set is a popular choice.

$$Diversity(A^{\pi_\theta}) = \mathbb{E}_{s\in d^\beta}\left[1 - \frac{1}{|A^{\pi_\theta}|(|A^{\pi_\theta}|-1)} \sum_{i,j\in A^{\pi_\theta}, i\neq j} sim(i,j)\right] \tag{7}$$

We define the similarity between two items i and j both on topic level and on content provider level. That is, $Sim(i, j) = 1$ if i and j belongs to the same topic cluster, and 0 otherwise. Similarly for content provider.

5.4 Novelty

The two terms of novelty and serendipity have been used interchangeably in the literature. In this work, we use novelty to focus on the *global* popularity-based measurements and serendipity to capture the unexpectedness/surprise of the recommendation to *a specific user*. That is, novelty concerns the recommender system's capacity to suggest something a user is unlikely to know about already or discover by themselves. Zhou et al. [71] first introduced the notion of self-information of a recommended item, which measures the unexpectedness of a recommended item relative to its global popularity.

$$\begin{aligned} I(a) &= -\log p(a) = -\log \frac{\#\text{ users consumed item } a}{\#\text{ users}} \\ &= -\log(\#\text{ users consumed item } a) + const \end{aligned} \tag{8}$$

Here $p(a)$ measures the chance a random user would have consumed item a. By definition, a globally "under-explored" item (tail content) will have higher self-information. With the definition of item-level self-information, we can then measure novelty of the learned policy π_θ as

$$Novelty(\pi_\theta) = \mathbb{E}_{s_t\in d_t^\beta}\left[\sum_{a\in\mathcal{A}} \pi_\theta(a|s_t) I(a)\right], \text{ [3]} \tag{9}$$

A learned policy π_θ that casts more mass on items with higher self-information, being able to recommend "under-explored" items, is deemed more novel. We can define the novelty metrics for attributes similarly by looking at the self-information of the attribute instead, e.g., popularity of the content provider.

5.5 Serendipity

Serendipity captures the unexpectedness/surprise of a recommendation to a specific user. It measures the capability of the recommender system to recommend relevant contents outside of the user's normal interests. There are two important factors in play here: 1) unexpectedness/surprise: as a counter example, a recommendation of John Lenon to listeners of The Beatles will not constitutes a surprising recommendation; 2) relevance: the surprising contents should be of interest to the user. In other words, serendipity measure the ability of the recommender to discover previously unknown (to the recommender) interests of the user.

We define the serendipity value of a recommendation a_t w.r.t. a user with interaction history of $\mathcal{I}_t$ as

$$S^{topic}(a_t|s_t, \mathcal{I}_t) = \begin{cases} 1 & \text{if } r^e(s_t, a_t) > 0 \text{ and } a_t \text{ belongs to} \\ & \text{a different topic cluster than any} \\ & \text{item in } \mathcal{I}_t; \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

Again we can define the content-provider level serendipity value similarly. With the serendipity value of an item defined, we can then quantify the serendipity of the recommendation set S_{π_θ} as.

$$Serendipity(A^{\pi_\theta}) = \mathbb{E}_{s_t\in d_t^\beta}\left[\frac{1}{|A^{\pi_\theta}|}\sum_{a\in A^{\pi_\theta}} S^{topic}(a_t|s_t, H_t)\right], \tag{11}$$

5.6 Long Term User Experience

Past work has suggested connections between these recommendation qualities toward long term user experience, either through surveys or interviews [5, 66]. We use user returning to the platform, and user moving from a low-activity bucket to a highly-active one on the platform as the holistic measurement of improved long term user experience, and establish the connection between these measurements and long term user experience.

6 OFFLINE ANALYSES

We conducted an extensive set of offline experiments comparing the exploration strategies introduced in Section 4. Specifically, we built these exploration approaches onto the baseline REINFORCE recommender described in Section 3. We evaluate them by computing the set of metrics defined in Section 5 and compare the metric movements between different hyper-parameter settings and different exploration methods.

[3] Note that for the novelty metric, we ignore the constant during evaluation, as a result the numbers reported in Section 6 are negative.

	Accuracy	Diversity		Novelty		Serendipity	
	mAP@50	topic	provider	item	provider	topic	provider
baseline	0.070 ± 0.002	0.784 ± 0.003	0.903 ± 0.003	-10.160 ± 0.036	-12.690 ± 0.049	0.037 ± 0.002	0.078 ± 0.003
$\alpha = 0.1$	0.064± 0.002	0.817 ± 0.003	0.915± 0.004	-9.612 ± 0.020	-12.403 ± 0.009	0.038 ± 0.002	0.072 ± 0.006
$\alpha = 0.5$	0.053 ±0.002	0.861 ± 0.004	0.940 ± 0.004	-9.130 ± 0.040	-12.120 ± 0.089	0.033 ± 0.002	0.056 ± 0.004
$\alpha = 1.0$	0.047 ± 0.003	0.871 ± 0.009	0.942 ± 0.002	-8.913 ± 0.158	-11.990 ± 0.096	0.029 ± 0.003	0.053 ± 0.002
$\alpha = 10.0$	0.033 ± 0.008	0.909 ± 0.023	0.965 ± 0.013	-8.653 ± 0.192	-11.850 ± 0.114	0.021 ± 0.005	0.037 ± 0.010

Table 1: Effect of entropy regularization with regularization coefficient in $[0.1, 0.5, 1.0, 10.0]$.

	Accuracy	Diversity		Novelty		Serendipity	
	mAP@50	topic	provider	item	provider	topic	provider
baseline	0.070 ± 0.002	0.784 ± 0.003	0.903 ± 0.003	-10.160 ± 0.036	-12.690 ± 0.049	0.037 ± 0.002	0.078 ± 0.003
topic, $d = 1$	0.061± 0.002	0.864 ± 0.003	0.925 ± 0.002	-10.247 ± 0.017	-12.647 ± 0.097	0.042 ± 0.001	0.077 ± 0.002
topic, $d = 7$	0.063 ± 0.002	0.860 ± 0.004	0.923 ± 0.004	-10.253 ± 0.037	-12.753 ± 0.041	0.044 ± 0.000	0.075 ± 0.002
topic, $d = 365$	0.062 ± 0.001	0.855 ± 0.005	0.923 ± 0.002	-10.237 ± 0.053	-12.713 ± 0.057	0.043 ± 0.001	0.075 ± 0.001
provider, $d = 7$	0.059 ± 0.002	0.807 ± 0.003	0.954 ± 0.001	-10.213 ± 0.048	-12.560 ± 0.057	0.039 ± 0.001	0.087 ± 0.004

Table 2: Effect of intrinsic motivation with different definitions of surprise.

	Accuracy	Diversity		Novelty		Serendipity	
	mAP@50	topic	provider	item	provider	topic	provider
baseline	0.070 ± 0.002	0.784 ± 0.003	0.903 ± 0.003	-10.160 ± 0.036	-12.690 ± 0.049	0.037 ± 0.002	0.078 ± 0.003
repre. alone	0.073 ± 0.002	0.785 ± 0.004	0.905 ± 0.004	-10.110 ± 0.042	-12.620 ± 0.061	0.038 ± 0.001	0.085 ± 0.003
intrinsic alone	0.063 ± 0.002	0.860 ± 0.004	0.923 ± 0.004	-10.253 ± 0.037	-12.753 ± 0.041	0.044 ± 0.000	0.075 ± 0.002
repre. + intrinsic	0.063 ± 0.001	0.859 ± 0.003	0.920 ± 0.004	-10.200 ± 0.025	-12.690 ± 0.012	0.046 ± 0.001	0.077 ± 0.002

Table 3: Effect of the actionable representation when combined with intrinsic motivation.

6.1 Dataset

We conducted 3 runs of experiments for each comparison and report the mean and standard deviation of the metrics. For each experiment run, we extracted close to a billion user trajectories from a commercial recommendation platform. Each trajectory $\mathcal{H}_T = \{(s_t, A_t, a_t, r_t) : t = 0, \ldots, T\}$, as described in Section 3, contains user historical events on the platform. The lengths of trajectories between users can vary depending on their activity level. We keep at most 500 historical pages with at least one positive user interaction (nonzero r_t) for each user. Among the collected trajectories, we hold out 1% for evaluation. We restrict our action space (item corpus) to the most popular 10 million items in the past 48 hours on the platform. Our goal is to build a recommender agent that can choose among the 10 million corpus the next set of items for users to consume so as to maximize the cumulative long-term reward.

6.2 Entropy Regularization

The most straightforward knob to tune up and down the exploration strength for entropy regularization is the regularization coefficient α as defined in eq. 4. We compare the baseline method, a REINFORCE agent maximizing only the expected return as defined in eq. 2, with added entropy regularization with α in $[0.1, 0.5, 1.0, 10.0]$.

As shown in Table 1, entropy regularization is an extremely efficient method to introduce diversity and novelty to the system, at the cost of reduced accuracy. When the regularization strength is large, it also significantly drops the system's capability to introduce serendipitous contents to users because of the loss of relevance. For example, a regularization strength of $\alpha = 1.0$ drops the topic serendipity value by -21.6% ($0.037 \rightarrow 0.029$).

6.3 Intrinsic Motivation

One of the obvious hyper-parameters to adjust the exploration strength for the intrinsic motivation approach is to tune the boosting factor c defined in eq. 6. Here we study the impact of the more interesting variants.

First, on which attribute to use to define surprise. We experimented with defining surprise by topic cluster (denoted as "topic" in Table 2) and content provider (denoted as "provider" in Table 2). Second, on the length of the user historical events used to define surprise. As explained in [66], users' perception of surprise of contents can drift over time. Contents that the user interacted in the past, but has not been served and interacted for a long time, can be deemed surprising when being resurfaced again. We experimented with having $\mathcal{I}_t$ contain all the items the user interacted with in the past one day, one week and one year (denoted as $d = 1$, $d = 7$ and $d = 365$ respectively in Table 2).

Table 2 summarizes the comparison between different variants of the intrinsic motivation proposal. Similar to entropy regularization, all variants improve on diversity at the cost of lower accuracy. This method does not change the novelty metrics significantly, neither on the item level nor content provider level. We thus conclude that tail contents are not necessarily more serendipitous (relevant and surprising) than popular ones. We do see a significant improvement in the serendipity metrics, even though the overall accuracy of these methods turn out unfavorable comparing with the baseline. As an example, the variant which uses topic cluster and a historical window size of 7 days, improves the serendipity level by +18.9% ($0.037 \rightarrow 0.044$) even though the overall accuracy measured by $mAP@50$ was dropped by -13.7% ($0.070 \rightarrow 0.063$). **Attributes.** Offline analyses showed both definitions of surprise based on topic

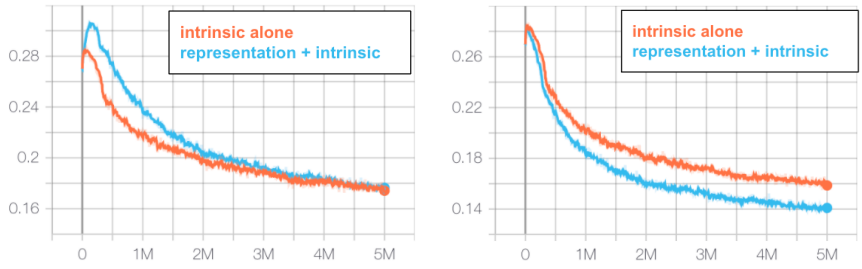


Figure 1: Mean input gate activation on historical events that are surprising (left) vs not (right). Adding the representation helps RNN differentiate better between historical events that are surprising and those that are not.

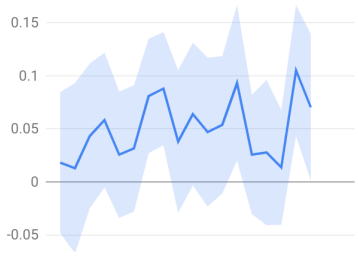


Figure 2: Improvement on user returning to platform.

cluster and content provider are equally effective in optimizing different angles of serendipity. That is topic cluster definition improves offline topic serendipity metrics by +18.9% from 0.037 to 0.044, and content provider definition improves content provider serendipity for +11.5% from 0.078 to 0.087. We however do see very different performance in user metrics in live experiments as shown in Section 7.1 below, suggesting one angle (topic serendipity) is more important than the other (content provider serendipity) in optimizing the overall user experience.

Window sizes. As we extend the historical window used to define surprise, i.e., having $\mathcal{I}_t$ contain longer user history, the definition of surprise becomes stricter. An item is less likely to be surprising/different when comparing with a longer history than a shorter one. As a result the percentage of state-action pairs receiving the extra multiplier of $c > 1$ is reduced. In the datasets, the percentage is reduced from 36% $\rightarrow$ 19% $\rightarrow$ 12% when the window size is extended from 1 $\rightarrow$ 7 $\rightarrow$ 365 days. The intrinsic motivation boost is applied to a smaller and smaller set of state-action pairs. The relative change on diversity related metrics is marginal between these variants. The variant with window size of $d = 7$ scored the highest on the topic serendipity metric, which is defined using a window size of one year.

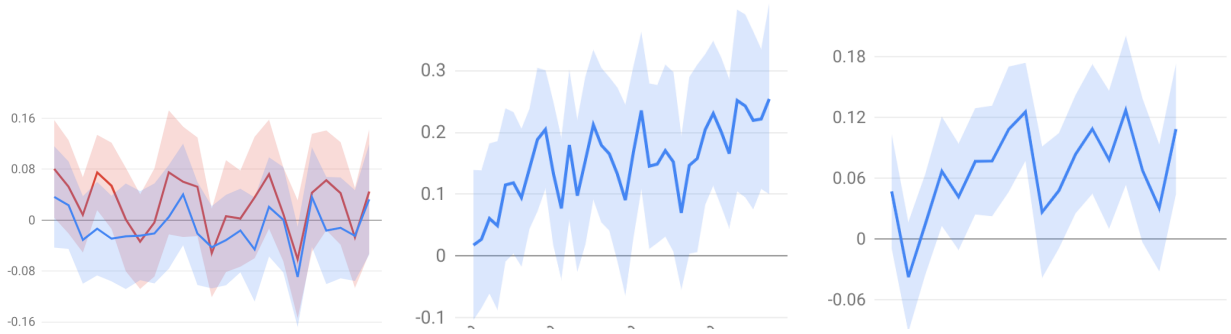
6.4 Actionable Representation

In this set of experiments, we compare four setups: 1) baseline: the baseline REINFORCE algorithm; 2) repre. alone: the baseline REINFORCE with the actionable representation, i.e., the additional bit indicating if the item a_t is serendipitous at state s_t according to user historical interactions $\mathcal{I}_t$; 3) intrinsic alone: the baseline REINFORCE with intrinsic motivation for reward shaping; 4) repre. + intrinsic: the baseline REINFORCE adding both the intrinsic motivation and the actionable representation. As shown in Table 3, adding the indicator alone (row 2) and adding the indicator along with the intrinsic motivation (row 4) resulted in very different metrics. Adding the indicator alone without the reward shaping performs very similarly to the baseline method, suggesting the representation is more useful when combined with the reward shaping. We see +24.3% improvement in the serendipity value comparing (row 4) to (row 1) (0.037 $\rightarrow$ 0.046), and +4.5% improvement comparing to (row 3) (0.044 $\rightarrow$ 0.046). This suggests that the added representation is indeed helpful for decision making when the intrinsic motivation is rewarding serendipitous actions, i.e. actions that discover previously unknown user interests.

To gain more insight into how the agent utilizes the additional bit indicating whether or not a historical event is surprising when provided, we compare the learning of the baseline REINFORCE algorithm with intrinsic motivation alone (shown in orange in Figure 1) vs the one combined with both the intrinsic motivation and the actionable representation (shown in cyan in Figure 1). The RNN [31] Chen et al. [11] used to encode the user history $\mathcal{H}_t$ has an important gate named input gate. This gate controls how much the RNN is updating its hidden state to take into account a new input (event). We take the activation values of the input gates across the user trajectory, and separate the values in two groups: the ones on historical events that are considered surprising and relevant (shown in Figure 1 left), and the ones on historical events that are not (shown in Figure 1 right). Comparing the left and right figures, we can see that by adding this additional information, the RNN is able to differentiate better between historical events that are serendipitous and those that are not. At the end of training, the mean activation for events that are surprising and relevant (left) is at 0.1765 (+1.4% higher) for intrinsic motivation + actionable representation compared with 0.1741 for intrinsic motivation alone. The mean activation for events that are NOT serendipitous (right) is at 0.1409 (−11.2% lower) for intrinsic motivation + actionable representation compared with 0.1586 for intrinsic motivation alone. This suggests that relying on the reward alone, RNN can still recognize the difference between these two groups of events and perform slightly larger update when the historical event is considered surprising. Adding the feature helps RNN differentiate the two groups better.

7 LIVE EXPERIMENTS AND LONG TERM USER EXPERIENCE

We conduct a series of live A/B tests on a industrial recommendation platform serving billions of users to evaluate the impact of the proposed exploration approaches. The control serves the base REINFORCE agent as described in Section 3. The agent selects hundreds of candidates from a corpus of 10 million. The returned candidates A^{π_θ}, along with others, are ranked by a separate ranking system before showing to the users. We ran three separate experiments: 1) Entropy regularization: serving the REINFORCE agent



(a) Entropy Regularization **(b) Intrinsic motivation** **(c) Intrinsic Motiv. + Actionable Repre.**

Figure 3: Overall user enjoyment improvement by comparing (a) Entropy regularization vs base REINFORCE; (b) Intrinsic motivation vs base REINFORCE; (c) Intrinsic motivation + Actionable representation vs Intrinsic motivation.

with entropy regularization as explained in Section 4.1; 2) Intrinsic motivation: serving the REINFORCE agent with intrinsic motivation to discover new user interest (using topic cluster attributes with a history window of 7 days and a serendipity boost $c = 4$) as explained in Section 4.2; 3) Intrinsic Motivation + Actionable Representation: serving the REINFORCE agent with both the intrinsic motivation and the actionable representation as introduced in Section 4.3. We compare 1) and 2) to the baseline REINFORCE system as described in Section 3 as control to measure the effect of entropy regularization and intrinsic motivation respectively, and 3) to 2) as control to measure the additional value of introducing the actionable representation on top of intrinsic motivation. We first summarize the live experiment results of these experiments in Section 7.1, and later measure several aspects of long term user experience in Section 7.2. In the end, we establish the connection between exploration and different aspects of recommendation quality toward improving long term user experience.

7.1 Results

Figure 3 summarizes the performances of these exploration approaches on the top-line metric capturing user overall enjoyment of the platform. As shown in Figure 3a ($\alpha = 0.1$ in red, and $\alpha = 0.5$ in blue), although entropy regularization increases diversity and novelty in both offline and live experiments, it does not lead to significant improvement on the user enjoyment. In other words, increased diversity or novelty alone does not necessarily lead to better user experience. When we increase the regularization strength to $\alpha = 0.5$, we see slightly worse live metrics.

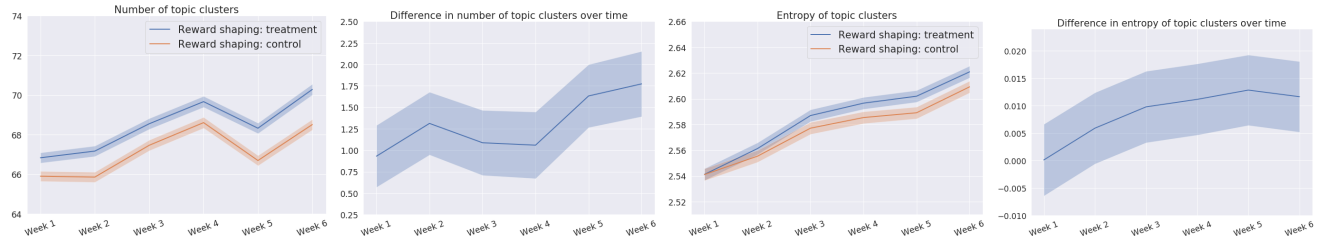
Comparing with entropy regularization (Figure 3a), intrinsic motivation (Figure 3b) and its combination with actionable representation (Figure 3c), not only significantly improve on the top-line metric, but also exhibit a strong learning effect over the course of the experiments [4]. We compare the offline measurement on accuracy, diversity, novelty and serendipity between entropy regularization with $\alpha = 0.5$ (Table 1 row 3) and intrinsic motivation (Table 3 row 3) and its combination with actionable representation (Table 3 row 4) and make the following observations: 1) the entropy regularization method with $\alpha = 0.5$ achieves very similar diversity metrics comparing to intrinsic motivation or its combination with actionable representation. All three methods reach a topic diversity around 0.86, and content provider diversity around 0.93; 2) The entropy regularization method achieved slightly higher novelty metric, both in item level and content provider level; 3) The metrics that entropy regularization loses is on accuracy and serendipity. 4) Intrinsic motivation method and its combination with actionable representation have favorable improvement on serendipity comparing with the baseline REINFORCE algorithm even though their accuracy numbers are worse. In conclusion, intrinsic motivation and its combination with actionable representation compare favorably to the baseline REINFORCE and entropy regularization only in the serendipity metrics offline. In live experiments, the intrinsic motivation and its combination with actionable representation were shown to significantly improve over the baseline REINFORCE and entropy regularization, as shown in Figure 3 (middle and right). Combining the offline and live experiment observations, we hypothesize that serendipity is an important faucet of recommendation quality that leads to improved long term user experience. We also conducted another group of live experiments defining surprise for optimization using content provider rather than topic. The experiment turns out neutral on the top-line metric, which suggest topic serendipity is more connected with long term user experience than content provider.

Another top-line metric that we keep track of is the number of days users returning to the platform. For both the intrinsic motivation and actionable representation treatment, we observed significant improvement on this metric as well, suggesting users are encouraged to return to the platform due to better recommendation quality. Figure 2 shows the improvement of user returning in the actionable representation experiment, comparing with the base REINFORCE with intrinsic motivation as control, suggesting that aiding the representation learning with the serendipity information further improves the learned policy, leading to better overall user experience.

7.2 Long Term User Experience

Learning Effect of Intrinsic Motivation. To better understand the effect of intrinsic motivation and reward shaping in the long term, we examine the temporal trend of the live metrics in addition to the aggregated metrics reported above. For the 6-week experiment on intrinsic motivation, we look at week-over-week metrics

[4]The intrinsic motivation experiment had been running for 6 weeks, while the actionable representation experiment 2 weeks.



(a) Number of topic clusters. **(b) Difference in (a) over time.** **(c) Entropy of topic clusters.** **(d) Difference in (c) over time.**

Figure 4: Learning effect: Temporal trend in number and entropy of topic clusters users interacted with during the experiment.

Days Active Last 14 days	User State
[0]	None
[1, 4]	Casual
[5, 11]	Emerging
[12, 14]	Core

(a) User activity level definition.

Pre Period	None	Casual	Emerging	Core
None	0.40% [-0.28, 1.08] %	-0.01% [-0.12, 0.09] %	0.02% [-0.24, 0.28] %	-0.49% [-1.47, 0.49] %
Casual	0.02% [-0.14, 0.18] %	-0.03% [-0.13, 0.06] %	-0.00% [-0.15, 0.15] %	0.53% [0.15, 0.91] %
Emerging	0.04% [-0.31, 0.39] %	-0.05% [-0.20, 0.10] %	-0.01% [-0.06, 0.04] %	0.06% [-0.07, 0.19] %
Core	-0.56% [-2.04, 0.92] %	-0.06% [-0.72, 0.61] %	-0.03% [-0.20, 0.14] %	0.01% [-0.02, 0.04] %

(b) State transition matrix change between treatment and control.

Figure 5: User activity level transition for *actionable representation.*

by aggregating user activities within each week. Specifically, we track the number of unique topic clusters the user has interacted with over every week, as well as the entropy of those topic clusters. Suppose the user has interacted with N_i items from topic cluster i, then the entropy of his/her history is computed as $-\sum_i \hat{p}_i log(\hat{p}_i)$, where $\hat{p}_i = N_i / \sum_i N_i$ is the proportion of items interacted with that are from topic cluster i.

Figure 4 shows the comparison between control and treatment, where the treatment group has a boosting multiplier of 4 for unknown user interests as in Eq. (6). Compared with users in the control group which does not have the reward shaping, users in the treatment group have consistently interacted with more topic clusters (Fig 4a) and generated a higher entropy over cluster distributions (Fig 4c) over the whole experiment period. More interestingly, the amount of improvements over control is increasing over time (Fig 4b and 4d). This suggests a learning effect over time from exploration, which enables users to continuously find and engage with new topics.

User Activity Levels. Users who come to the recommendation platform are heterogeneous in terms of activity levels. Some users visit the platform occasionally, while others visit the platform more regularly and consistently. The long-term goal of a recommendation platform is to not only satisfy the user's need in the current session, but ideally to see them return to the recommendation platform more often in the future.

We would like to see if adding exploration in the recommendation has any effect on moving user activity levels. We define four user activity levels in terms of how many days they are active on the platform in a 2-week period, which is shown in Fig 5a. For example, a user being casual means that he/she has been active for 1 to 4 days in the last 14 days. Users can become more active or less active depending their experience on the platform as well as exogenous factors not control by recommendation. Suppose the goal of a recommendation platform is moving causal users to become core users. An intuitive way to measure the conversion is by counting the number of users who start off casual, and end up core. This can be realized with a *user activity level transition matrix*, which measures the movement between different user activity levels.

We examine user activity level before the experiment start date and at the end of the experiment for every treatment group to compute the transition matrix, and compare with control. Figure 5b shows the percentage difference of the transition matrices between the actionable representation treatment group and control. We see that there is a significant increase in casual-to-core conversion rate. This suggests that a successful exploration strategy can result in a desired user movement as less active users are becoming more engaged on the platform.

8 CONCLUSION

We present a systemic study to understand the values of exploration in recommender systems beyond reducing model uncertainty. We examine different user exploration strategies in affecting the four facets of recommendation quality, i.e., accuracy, diversity, novelty and serendipity, that contribute directly to user experience on the platform. We showcase exploration strategies that oriented toward discovering unknown user interests in positively influencing user experience on recommendation platforms. Using conversion of casual users to core users as an indicator of the holistic long term

user experience, we connects serendipity to improved long term user experience. We believe these are important first steps in understanding and improving exploration and serendipity in (RL based) recommender systems, and providing foundation for future effort in this direction.

REFERENCES

[1] Pieter Abbeel and Andrew Y Ng. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*. 1.

[2] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. 2017. Constrained policy optimization. *arXiv preprint arXiv:1705.10528* (2017).

[3] Gediminas Adomavicius and YoungOk Kwon. 2011. Maximizing aggregate recommendation diversity: A graph-theoretic approach. In *Proc. of the 1st International Workshop on Novelty and Diversity in Recommender Systems (DiveRS 2011)*. Citeseer, 3–10.

[4] Shipra Agrawal and Navin Goyal. 2012. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on learning theory*. JMLR Workshop and Conference Proceedings, 39–1.

[5] Ashton Anderson, Lucas Maystre, Ian Anderson, Rishabh Mehrotra, and Mounia Lalmas. 2020. Algorithmic effects on the diversity of consumption on spotify. In *Proceedings of The Web Conference 2020*. 2155–2165.

[6] Marc G Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. 2016. Unifying count-based exploration and intrinsic motivation. *arXiv preprint arXiv:1606.01868* (2016).

[7] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research* 3 (2003), 993–1022.

[8] Pablo Castells, Saúl Vargas, and Jun Wang. 2011. Novelty and diversity metrics for recommender systems: choice, discovery and relevance. (2011).

[9] Òscar Celma and Perfecto Herrera. 2008. A new approach to evaluating novel recommendations. In *Proceedings of the 2008 ACM conference on Recommender systems*. 179–186.

[10] Olivier Chapelle and Lihong Li. 2011. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*. 2249–2257.

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

[12] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 191–198.

[13] Nathaniel D Daw, John P O'doherty, Peter Dayan, Ben Seymour, and Raymond J Dolan. 2006. Cortical substrates for exploratory decisions in humans. *Nature* 441, 7095 (2006), 876.

[14] Gabriel Dulac-Arnold, Richard Evans, Hado van Hasselt, Peter Sunehag, Timothy Lillicrap, Jonathan Hunt, Timothy Mann, Theophane Weber, Thomas Degris, and Ben Coppin. 2015. Deep reinforcement learning in large discrete action spaces. *arXiv preprint arXiv:1512.07679* (2015).

[15] Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach. 2010. Beyond accuracy: evaluating recommender systems by coverage and serendipity. In *Proceedings of the fourth ACM conference on Recommender systems*. 257–260.

[16] Dibya Ghosh, Abhishek Gupta, and Sergey Levine. 2018. Learning actionable representations with goal-conditioned policies. *arXiv preprint arXiv:1811.07819* (2018).

[17] John Gittins, Kevin Glazebrook, and Richard Weber. 2011. *Multi-armed bandit allocation indices*. John Wiley & Sons.

[18] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. 2019. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*. 2555–2565.

[19] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)* 22, 1 (2004), 5–53.

[20] Irina Higgins, Arka Pal, Andrei A Rusu, Loic Matthey, Christopher P Burgess, Alexander Pritzel, Matthew Botvinick, Charles Blundell, and Alexander Lerchner. 2017. Darla: Improving zero-shot transfer in reinforcement learning. *arXiv preprint arXiv:1707.08475* (2017).

[21] Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. 2016. Vime: Variational information maximizing exploration. *arXiv preprint arXiv:1605.09674* (2016).

[22] Yujing Hu, Qing Da, Anxiang Zeng, Yang Yu, and Yinghui Xu. 2018. Reinforcement learning to rank in e-commerce search engine: Formalization, analysis, and application. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 368–377.

[23] Eugene Ie, Vihan Jain, Jing Wang, Sanmit Narvekar, Ritesh Agarwal, Rui Wu, Heng-Tze Cheng, Tushar Chandra, and Craig Boutilier. 2019. SlateQ: A tractable decomposition for reinforcement learning with recommendation sets. (2019).

[24] Laurent Itti and Pierre Baldi. 2009. Bayesian surprise attracts human attention. *Vision research* 49, 10 (2009), 1295–1306.

[25] Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. 2017. Reinforcement learning with unsupervised auxiliary tasks. In *ICLR*.

[26] Marius Kaminskas and Derek Bridge. 2016. Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 7, 1 (2016), 1–42.

[27] Jens Kober, J Andrew Bagnell, and Jan Peters. 2013. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research* 32, 11 (2013), 1238–1274.

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

[29] Denis Kotkov, Shuaiqiang Wang, and Jari Veijalainen. 2016. A survey of serendipity in recommender systems. *Knowledge-Based Systems* 111 (2016), 180–192.

[30] Tze Leung Lai and Herbert Robbins. 1985. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics* 6, 1 (1985), 4–22.

[31] Thomas Laurent and James von Brecht. 2016. A recurrent neural network without chaos. *arXiv preprint arXiv:1612.06212* (2016).

[32] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. 2020. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643* (2020).

[33] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. 2018. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research* 37, 4-5 (2018), 421–436.

[34] Feng Liu, Ruiming Tang, Xutao Li, Weinan Zhang, Yunming Ye, Haokun Chen, Huifeng Guo, and Yuzhou Zhang. 2018. Deep reinforcement learning based recommendation with explicit user-item interactions modeling. *arXiv preprint arXiv:1810.12027* (2018).

[35] Sean M McNee, John Riedl, and Joseph A Konstan. 2006. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI'06 extended abstracts on Human factors in computing systems*. 1097–1101.

[36] Azalia Mirhoseini, Hieu Pham, Quoc V Le, Benoit Steiner, Rasmus Larsen, Yuefeng Zhou, Naveen Kumar, Mohammad Norouzi, Samy Bengio, and Jeff Dean. 2017. Device placement optimization with reinforcement learning. In *International Conference on Machine Learning*. PMLR, 2430–2439.

[37] Martin Mladenov, Elliot Creager, Omer Ben-Porat, Kevin Swersky, Richard Zemel, and Craig Boutilier. 2020. Optimizing long-term social welfare in recommender systems: A constrained matching approach. In *International Conference on Machine Learning*. PMLR, 6987–6998.

[38] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*. 1928–1937.

[39] Klaus Nehring and Clemens Puppe. 2002. A theory of diversity. *Econometrica* 70, 3 (2002), 1155–1198.

[40] Andrew Y Ng, Daishi Harada, and Stuart Russell. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, Vol. 99. 278–287.

[41] Kenta Oku and Fumio Hattori. 2011. Fusion-based Recommender System for Improving Serendipity. *DiveRS@ RecSys* 816 (2011), 19–26.

[42] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. 2016. Deep exploration via bootstrapped DQN. *arXiv preprint arXiv:1602.04621* (2016).

[43] Georg Ostrovski, Marc G Bellemare, Aäron Oord, and Rémi Munos. 2017. Count-based exploration with neural density models. In *International conference on machine learning*. PMLR, 2721–2730.

[44] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. 2017. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning*. PMLR, 2778–2787.

[45] GP Patil and Charles Taillie. 1982. Diversity as a concept and its measurement. *Journal of the American statistical Association* 77, 379 (1982), 548–561.

[46] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548* (2017).

[47] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. 285–295.

[48] Jürgen Schmidhuber. 1991. Curious model-building control systems. In *Proc. international joint conference on neural networks*. 1458–1463.

[49] Jürgen Schmidhuber. 2010. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development* 2, 3 (2010), 230–247.

[50] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. 2018. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE International Conference on Robotics*

and Automation (ICRA). IEEE, 1134–1141.

[51] Guy Shani, David Heckerman, and Ronen I Brafman. 2005. An MDP-based recommender system. *Journal of Machine Learning Research* 6, Sep (2005).

[52] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. 2017. Mastering the game of go without human knowledge. *nature* 550, 7676 (2017), 354–359.

[53] Barry Smyth and Paul McClave. 2001. Similarity vs. diversity. In *International conference on case-based reasoning*. Springer, 347–361.

[54] Aravind Srinivas, Michael Laskin, and Pieter Abbeel. 2020. Curl: Contrastive unsupervised representations for reinforcement learning. *arXiv preprint arXiv:2004.04136* (2020).

[55] Bradly C Stadie, Sergey Levine, and Pieter Abbeel. 2015. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814* (2015).

[56] Jan Storck, Sepp Hochreiter, and Jürgen Schmidhuber. 1995. Reinforcement driven information acquisition in non-deterministic environments. In *Proceedings of the international conference on artificial neural networks, Paris*, Vol. 2. Citeseer, 159–164.

[57] Richard S Sutton, Andrew G Barto, et al. 1998. *Reinforcement learning: An introduction*. MIT press.

[58] Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. 2017. # exploration: A study of count-based exploration for deep reinforcement learning. In *31st Conference on Neural Information Processing Systems (NIPS)*, Vol. 30. 1–18.

[59] William R Thompson. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 25, 3/4 (1933), 285–294.

[60] Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8, 3-4 (1992), 229–256.

[61] Ronald J Williams and Jing Peng. 1991. Function optimization using connectionist reinforcement learning algorithms. *Connection Science* 3, 3 (1991), 241–268.

[62] Hongzhi Yin, Bin Cui, Jing Li, Junjie Yao, and Chen Chen. 2012. Challenging the long tail recommendation. *arXiv preprint arXiv:1205.6700* (2012).

[63] Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims. 2007. A support vector method for optimizing average precision. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. 271–278.

[64] Ruohan Zhan, Konstantina Christakopoulou, Ya Le, Jayden Ooi, Martin Mladenov, Alex Beutel, Craig Boutilier, Ed Chi, and Minmin Chen. 2021. Towards Content Provider Aware Recommender Systems: A Simulation Study on the Interplay between User and Provider Utilities. In *Proceedings of the Web Conference 2021*. 3872–3883.

[65] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)* 52, 1 (2019), 1–38.

[66] Yuan Cao Zhang, Diarmuid Ó Séaghdha, Daniele Quercia, and Tamas Jambor. 2012. Auralist: introducing serendipity into music recommendation. In *Proceedings of the fifth ACM international conference on Web search and data mining*. 13–22.

[67] Xiangyu Zhao, Long Xia, Jiliang Tang, and Dawei Yin. 2019. " Deep reinforcement learning for search, recommendation, and online advertising: a survey" by Xiangyu Zhao, Long Xia, Jiliang Tang, and Dawei Yin with Martin Vesely as coordinator. *ACM SIGWEB Newsletter* Spring (2019), 1–15.

[68] Xiangyu Zhao, Long Xia, Dawei Yin, and Jiliang Tang. 2019. Model-based reinforcement learning for whole-chain recommendations. *arXiv preprint arXiv:1902.03987* (2019).

[69] Xiangyu Zhao, Liang Zhang, Long Xia, Zhuoye Ding, Dawei Yin, and Jiliang Tang. 2017. Deep reinforcement learning for list-wise recommendations. *arXiv preprint arXiv:1801.00209* (2017).

[70] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. 2018. DRN: A deep reinforcement learning framework for news recommendation. (2018), 167–176.

[71] Tao Zhou, Zoltán Kuscsik, Jian-Guo Liu, Matúš Medo, Joseph Rushton Wakeling, and Yi-Cheng Zhang. 2010. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences* 107, 10 (2010), 4511–4515.

[72] Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*. 22–32.