Reward Shaping for User Satisfaction

Konstantina Christakopoulou[1] (✉), Can Xu[1], Sai Zhang[2], Sriraj Badam[2], Trevor Potter[2], Daniel Li[2], Hao Wan[3*], Xinyang Yi[1], Ya Le[4*], Chris Berg[2], Eric Bencomo Dixon[5*], Ed H. Chi[1], and Minmin Chen[1]

[1] Google DeepMind {konchris,canxu,xinyang,edchi,minminc}@google.com
[2] Google {saisaizhang,srirajdutt,tpotter,chrisberg}@google.com
[3] Meta
[4] OpenAI
[5] DoorDash

Abstract. How might we design Reinforcement Learning (RL)-based recommenders that encourage aligning user trajectories with the underlying user satisfaction? Three research questions are key: (1) measuring user satisfaction, (2) combatting sparsity of satisfaction signals, and (3) adapting the training of the recommender agent to maximize satisfaction. For measurement, it has been found that surveys explicitly asking users to rate their experience with consumed items can provide valuable orthogonal information to the engagement/interaction data, acting as a proxy to the underlying user satisfaction. For sparsity, i.e, only being able to observe how satisfied users are with a tiny fraction of user-item interactions, *imputation models* can be useful in predicting satisfaction level for all items users have consumed. For learning satisfying recommender policies, we postulate that reward shaping in RL recommender agents is powerful for driving satisfying user experiences. Putting everything together, we propose to jointly learn a policy network and a satisfaction imputation network: The role of the imputation network is to learn which actions are satisfying to the user; while the policy network, built on top of REINFORCE, decides which items to recommend, with the reward utilizing the imputed satisfaction. We use both offline analysis and live experiments in an industrial large-scale recommendation platform to demonstrate the promise of our approach for satisfying user experiences.

1 Introduction

Recommender systems at heart aim at creating a good user experience by surfacing users with the right content at the right time and under the right context. It is thus critical for the system to identify what defines the user experience, more specifically the underlying user utilities of the platform. Most recent advances in recommender systems have relied on implicit user feedback, such as clicks or dwell time, as proxies to capture user utilities [5,33]. Although this data measures *what users do*, it can fail to capture what users *say they want* — which are potentially

[*] Work done while at Google.

very different [14]. As a result, recommender models learned solely based on user engagement data can be misaligned with the true user utilities.

Given a specified objective, or else *reward function*, which captures the long-term user utility, recommender systems can be formulated as Reinforcement Learning (RL) agents deciding on actions to take (i.e., contents to show to users) given certain user states (i.e., latent representation at a specific time/context), with the goal of maximizing said cumulative reward [3,15,13,35]. There are several challenges especially exacerbated in industrial recommendation settings which makes the application of RL for recommendation rather unique compared to other application areas like games [21,28] and robotics [10]. The action space is extremely large and ever-changing; user preferences change over time; and data are extremely sparse for the enormous action and state space. Only recently there have been major advances addressing these challenges and showcasing RL approaches for recommendation [3,15].

Besides addressing these challenges, the key in building recommender agents lies in defining the reward function guiding the learning of the agent policy. Although we do get to observe some proxy signals indicating when a recommendation, or a series of recommendations was successful (e.g., the user clicked on the recommended content, they shared it with their social network etc.), there is a disconnect between the implicit feedback we observe, and what the user really wants. The proximity between the proxy signals we include in the agent's reward function and the true user utility, will largely determine the extent to which the RL recommender can optimize for what users want.

However, despite the importance of the reward function for building recommendation agents, there has been relatively little work on reward shaping for RL recommenders. Most works treat the reward function as a black-box, which is given, and often assume that dense engagement signals are indicative of how much users value their experience. This assumption has been recently challenged in non-RL settings, underlying that post-engagement signals and/or satisfaction survey responses together with implicit behavioral signals give a clearer picture of user utilities [11,14,31].

In this paper, we put the reward front and center, and highlight it as a key tool for optimizing for what users actually want. Satisfaction data as collected by user responses to satisfaction surveys provide an important view as to how the user felt about the recommendation, as opposed to how they behaved while interacting with it. These surveys are shown uniformly to all users, and ask users to rate on a scale how satisfying they found a sampled item from their recent engagement history. In our systems, such survey data can offer more representativeness compared to post-engagement signals, as most users tend to not engage in post-click actions such as likes, dismissals. Furthermore, optimizing for satisfaction as measured by surveys can substantially move post-engagement related metrics as well. Based on the above, both behavioral signals, and satisfaction signals should be incorporated into the reward.

However, before we can utilize satisfaction signals into the reward, we need to highlight a major challenge inherently associated with them — *sparsity*.

The volume of satisfaction data is orders of magnitude smaller compared with engagement data volume; in our case study, roughly *one out of thousands* of engagement signals will come with a satisfaction response. This is due to a number of reasons. First, it is disruptive to ask users about every item they recently consumed. Second, response rate can be very low in an environment where primary user intention is to consume content rather than providing feedback. As a result, we only have access to a small amount of survey responses covering an extremely small fraction of the user-item interaction pairs. Given this extreme sparsity, simply supplementing the existing reward signals that focus on engagement with the sparse satisfaction signals is not going to be effective in shifting towards optimizing for user satisfaction. Instead, personalized satisfaction models are required to impute for each user how they would rate their satisfaction level with each consumed item, had they responded to a survey.

Here, we propose augmenting a classic policy network trained with REINFORCE with a satisfaction imputation network to predict user satisfaction and include the prediction into the reward for the policy network, while training both networks concurrently in a multi-task learning setup.

Our contributions are threefold:

1. **Reward Shaping for User Utility Alignment:** We emphasize reward shaping as a crucial but underexplored technique for guiding RL-based recommenders to select satisfying actions (Sections 4.2, 4.3 and 6), highlighting the inherent challenges in defining, measuring, and modeling user satisfaction.
2. **Scalable Satisfaction Imputation Networks:** We introduce satisfaction imputation networks to address the sparsity of satisfaction signals, demonstrating their integration into a top-K REINFORCE recommender with vast state and action spaces (Section 4.4). We also provide offline analysis on key design considerations for effective satisfaction imputation networks (Section 5).
3. **Demonstrated Benefits in Live Experiments:** Results from A/B tests on a large-scale platform with a two-stage recommendation system show that replacing a REINFORCE nominator with a dense engagement reward (without satisfaction imputation) with our proposed architecture (Fig. 2) increases satisfying nominations and decreases low-satisfaction nominations. This leads to a statistically significant improvement in satisfying user experiences and a reduction in unsatisfying ones (Section 6).

2 Related Work

Here, we give an overview of the most closely related works.

Reinforcement Learning (RL). Problems in which an agent learns to interact with the environment, with the interactions having long-term consequences, are a natural fit to be framed as Reinforcement Learning ones [29]. Classical approaches to RL problems include value-based approaches such as Q-learning [20], and policy-based ones such as policy gradient [32]. Deep RL combines the promise of deep neural networks to help RL achieve ground-breaking success in games and robotics applications [21,19,28,10]. We build our work on top of a policy-based approach, namely REINFORCE [32], following its prior success in recommendation settings [3]. The imputation network we introduce has deep connections with value learning approaches [20], where a state-action value network is learned. The estimations of this network are utilized as *part* of our policy's reward; as a result, we still need the off-policy correction component [3]. An alternative approach we leave for future work is employing Actor-Critic or its variants [19,30,27].

Reinforcement Learning in Recommendation. Although there have been many successes in RL for applications like games [21,28] and robotics [10], only recently some successes of RL for recommendation have been demonstrated [7]. The main work we build on top of is a policy-gradient-based approach correcting for off-policy skew with importance weighting [3].This work demonstrated the value of REINFORCE and top-K off-policy correction in a large-scale industrial recommendation platform with an extremely large action space. Other recent works have demonstrated the value of deep RL approaches for recommendation, such as Actor-Critic [15], Deep Q-learning [36], and hierarchical RL [34]. Also, novel RL approaches have been proposed for the more complicated problem of slate recommendation [13], as well as for page-wise recommendation [35]. Despite the recent promise of RL for recommendation, the majority of works do not draw attention to the important aspect of reward shaping, which is key for aligning system objectives with underlying user utilities; this is the focus of our paper.

Reward Shaping. The importance of reward shaping, i.e., shaping the original sparse, delayed reward signals as in-time credit assignment for successful RL algorithms has been emphasized early on [23,16,6]. This is a general term encompassing the incorporation of domain knowledge into RL to guide the policy learning. Carefully designing the reward function is critically important as: (i) a misspecified reward leads to sub-optimal policy; (ii) an under-specified reward leads to unexpected behavior [12]. While RL for robotics and games has relied on hand-crafted reward or imitation learning [26] to effectively guide the agent to success, the perils of a misspecified reward function in the design of recommender RL agents have not received a lot of attention. Motivated by the need to bridge the RL for recommendation line of work with the recent discussions on measuring and modeling user satisfaction to properly capture user experience [17,14,18,9], we provide a reward shaping approach for imputing user satisfaction into the reward of an RL recommender, along with using the ground truth engagement proxy signals.

3 Background

3.1 Recommendation as an RL problem

The recommender system's goal is to decide which contents to recommend to the incoming user requests, given some representation of the user profile, the context, and their interaction history up to this point, as captured by the sequence of items (e.g. videos, news articles, products) they have interacted with, along with the corresponding feedback (e.g., time spent watching/reading), so as to maximize the cumulative rewards experienced by the users.

In RL terms, we formulate the recommendation problem as a Markov Decision Process (MDP), or better, a Partially Observable MDP (POMDP) as the states are unobserved:

Recommendation MDP	
Action $a \in \mathcal{A}$	item(s) available for recommendation
State $s \in \mathcal{S}$	user interests and context
State Transition $s_{t+1} \sim \mathbf{P}(\cdot\|s_t, a_t)$	unknown dynamics capturing how user state changes from t to $t+1$, conditioned on a_t and s_t
Reward $r(s, a)$	immediate reward obtained by performing action a for state s

The goal is to find a policy $\pi(a|s)$ capturing the probability distribution over the action space, i.e, items to recommend, given the current user state $s \in \mathcal{S}$, so to maximize the expected cumulative reward,

$$\max_{\pi} \mathbf{E}_{\tau \sim \pi} \left[R(\tau) \right] \tag{1}$$

where $R(\tau) = \sum_{t=0}^{|\tau|} r(s_t, a_t)$, and the expectation $\mathbf{E}$ is taken over user trajectories τ obtained by acting according to the policy: $a_t \sim \pi(\cdot|s_t)$, $s_{t+1} \sim \mathbf{P}(\cdot|s_t, a_t)$.

We build our method on top of the REINFORCE recommender introduced in [32]. Let the policy π assume a functional form, mapping states to actions, parameterized by $\theta \in \mathbf{R}^d$. Using the log-trick, the gradient of the expected cumulative reward with respect to the policy parameters θ can be derived analytically [32]:

$$\nabla_\theta \mathbf{E}_{\tau \sim \pi_\theta}[R(\tau)] = \mathbf{E}_{\tau \sim \pi_\theta} \left[R(\tau) \nabla_\theta \log \pi_\theta(\tau) \right]. \tag{2}$$

To reduce variance in the gradient estimate a common practice is to discount the future reward with a discount γ:

$$\sum_{\tau \sim \pi_\theta} \left[R(\tau) \nabla_\theta \log \pi_\theta(\tau) \right] \approx \sum_{\tau \sim \pi_\theta} \sum_{t=0}^{|\tau|} \left[R_t \nabla_\theta \log \pi_\theta(a_t|s_t) \right], \tag{3}$$

where

$$\begin{aligned} R_t = r(s_t, a_t) + \gamma r(s_{t+1}, a_{t+1}) + \gamma^2 r(s_{t+2}, a_{t+2}) + \ldots \\ + \gamma^{|\tau|-1-t} r(s_{|\tau|-1}, a_{|\tau|-1}). \end{aligned} \tag{4}$$

Eq. (3) gives an unbiased estimate of the policy gradient in *online RL*, where the gradient of the policy is computed on trajectories collected by the policy π_θ we are learning. In practice, due to infrastructure limitations or production concerns, the trajectories available for learning are collected from a different *logging* policy, or mixture of such policies, denoted by β instead. Thus, we operate in an *offline RL* setting, making the policy gradient as given by Eq. (3) no longer unbiased. To address this skew, importance weighting is adopted [22]. In this work, we also operate in batch offline RL, applying top-K off-policy correction, and we defer readers to [3] for details.

4 Imputing Satisfaction in Reward

We now turn to the main focus of this paper, i.e, shaping the reward of a REINFORCE recommender to drive user satisfaction. We start by describing how we parameterize the policy network (Section 4.1); next we highlight the role of reward in REINFORCE for capturing long-term user utility (Section 4.2); and emphasize the challenges associated with considering satisfaction as a proxy to user utility (Section 4.3). Motivated by these challenges, we propose to augment the policy network with a satisfaction imputation network (Section 4.4).

4.1 Policy Parameterization

We closely follow the setup in [1,3] to parameterize the policy. A Recurrent Neural Network (RNN) is used to encode the user's interaction history, capturing the changing user preferences. The output of the RNN is concatenated with the latent embeddings encoding context, which capture features like time of the day, device type. The concatenation of user sequential preferences and context embeddings is mapped to a lower dimensional representation via multiple Rectified Linear Units. This represents the user state $\mathbf{u}_s$. Conditioned on the user state $\mathbf{u}_s$, the policy $\pi_\theta(a|s)$ is then modeled with a softmax,

$$\pi_\theta(a|s) = \frac{\exp(\mathbf{u}_s^T \mathbf{v}_a / T)}{\sum_{a' \in \mathcal{A}} \exp(\mathbf{u}_s^T \mathbf{v}_{a'} / T)}, \tag{5}$$

where $\mathbf{v}_a$ are the action embeddings, and T is a temperature term controlling the smoothness of the learned policy.

4.2 Reward

Reward plays a paramount role in determining the final learned policy. As shown in Eq. (3), the gradient from each state-action pair is weighted by the cumulative discounted reward R_t.

As prescribed in Section 3.1, R_t depends on the immediate rewards associated with the state-action pairs $r(s, a)$ as well as the discounting factor γ. In the absence of a user utility oracle, a key design choice is which *proxy signals* to use to define the immediate reward. For each recommendation, the user could leave different signals indicating their experience with the item. Examples include implicit engagement-related signals, such as click, time spent engaging (reading/ watching/ listening), post-engagement actions, e.g., shares/ likes/ dislikes/ comments, and they could leave explicit feedback in surveys asking them about their satisfaction level with the consumed item.

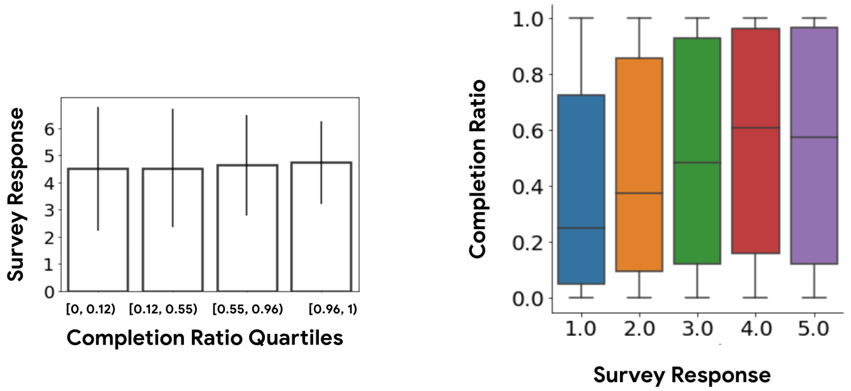

Fig. 1. Survey responses provide a different set of information compared to behavior signals. Interactions with the same completion ratio can have vastly different associated satisfaction levels.

4.3 Value of Satisfaction Signals, and Challenges

It is easy to see that if the proxy signals used in the reward are solely engagement-focused, the policy will learn to choose actions that only drive engagement. This is not ideal as what users do (engagement) can be quite different from what they say they want (satisfaction), thus neglecting other important facets of the user experience.

Fig. 1 illustrates this point. More than two million interactions with survey responses on a commercial recommendation platform were collected and analyzed. For the sake of this example, we consider completion ratio (e.g., time spent on the item out of total length of the item) as one useful behavioral signal, and study its relationship with satisfaction signals as measured by survey responses in the scale of one to five. As shown in (Fig. 1 *right*), grouping interactions by survey response rating, we find the higher the survey value, the higher the median completion ratio; however, we also see that per survey response value the range of associated completion ratios is quite large. This becomes more evident when grouping interactions with their associated survey responses based on the corresponding quantiles of completion ratios, (Fig. 1 *left*). Based on the plotted 95% confidence intervals, interactions belonging in the exact same quantile of completion ratios (i.e., same user behavior), have quite different associated satisfaction levels (Fig. 1 *left*).

It is worth pointing out that behavior signals alone fail to capture other sides of how the user felt about the interaction, e.g., did they find the content misleading, useful, did it provide some longer term value to them. It is therefore critical to consider both behavioral and satisfaction signals, and appropriately balance them when defining the reward.

Also, we opt for survey data rather than post-engagement signals as better proxies for user satisfaction as we have found that they can offer more representativeness—most users tend to not engage in post-click actions such as likes, dismissals. Having said that, although we demonstrate the effectiveness of reward shaping with imputation networks for survey signals, the same technique is equally applicable for other proxy reward signals exhibiting similar concerns, such as likes, dislikes, shares or dismissals. What is more, in our case study, we find that satisfaction as measured by survey responses highly correlates with goodness as measured by post-engagement signals. Thus, we are able to significantly increase likes, and decrease dislikes/dismissals, even without explicitly optimizing for them (Section 6).

If for each item the user interacted with in the trajectory, besides implicit engagement signals $r^e(s, a)$, we also had access to explicit satisfaction signals $r^u(s, a)$, we could define the immediate reward $r(s, a)$ as a function of the two, i.e.,

$$r(s, a) = f\left(r^u(s, a), r^e(s, a)\right), \tag{6}$$

where $f(\cdot)$ can include operators such as transformations on the raw signal (e.g., raising to a power, hinge, sigmoid) and combination functions (e.g., addition, multiplication) on the two reward signals.

While the engagement signals r^e are often dense, satisfaction signals r^u are extremely sparse, as they are derived from user-provided responses to satisfaction surveys. These surveys are shown uniformly to all users, asking them to rate on a scale how satisfying they found a sampled item from their recent engagement history. In our case study, roughly *one out of thousands of engagement signals will come with a satisfaction response*. This is because in a primarily content consumption-focused recommender platform, it would be disruptive to ask users to rate *every* item consumed. Furthermore, users tend to not respond to surveys [4]— *response rate is around 2%* in our case.

4.4 Satisfaction Imputation Model

This inherent sparsity of a subset of signals makes simply including them in the reward when present, ineffective.

To address this challenge, we propose the use of an imputation network to densify the satisfaction signals, and include the imputed satisfaction signals in the reward instead.

The role of the imputation network is to map user state and action pairs (s, a) to satisfaction scores, i.e., survey responses sr present in the satisfaction data $\mathcal{D}_{sr}$. One can imagine learning a completely separate imputation model on this data, and then utilize these imputations directly on the reward of the policy network. The alternative which we opt for is to extend the policy network with a satisfaction imputation head and have parameters shared between the two. This is quite appealing as given that the data used to train the policy head are of much higher volume compared to those used to train the imputation head, we hypothesize that transferring the learned user state and action embeddings from the dense task to the sparse one can be quite useful [24].

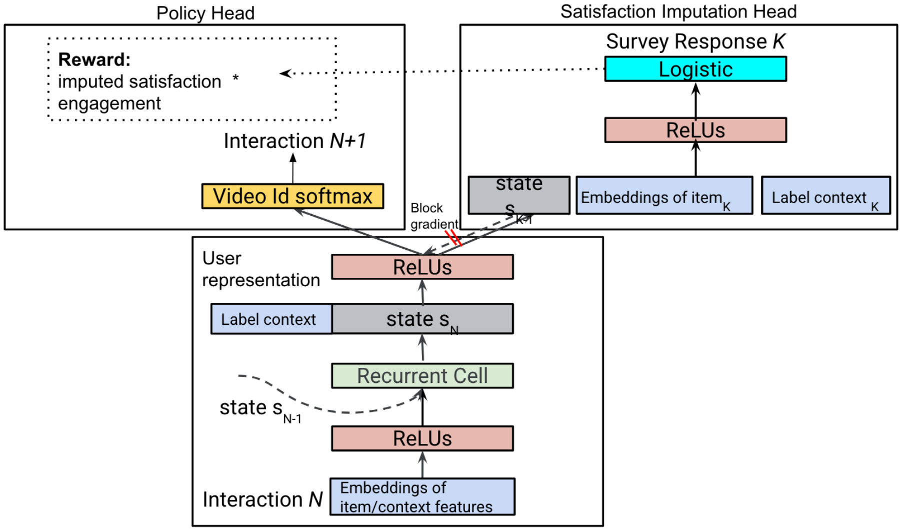

Fig. 2. Proposed architecture, where a satisfaction imputation head is added to learn goodness of actions based on sparse survey responses, conditioned on state, action and context embeddings learned from the policy. The reward utilizes the satisfaction imputations to guide the policy head to select actions that lead to satisfied engagement.

Concretely, we propose a multi-task shared-bottom architecture with two heads, the policy head and the satisfaction imputation head, each having their own task-specific parameters while sharing majorities of the state and action representations. As shown in Fig. 2 (*bottom*), the shared bottom encodes the sequential history of the user, as well as context information. The policy head in Fig. 2 (*upper left*) is identical to the standalone policy network described in Section 4.1, with the only change being in its reward. The imputation head in Fig. 2 (*upper right*) is used to infer the satisfaction score for each state-action pair (s, a) in the collected trajectories. Then, the imputed satisfaction score combined with engagement signals forms the reward used to train the policy network.

We train the imputation head by gathering the corresponding state $\mathbf{u}_s$, and action embeddings $\mathbf{v}_a$ for any state-action pair associated with a survey response in the batch, and learning a mapping $\hat{sr}_\phi$ from these embeddings to the corresponding survey response i.e., $\hat{sr}_\phi : (\mathbf{u}_s, \mathbf{v}_a) \rightarrow sr, \forall sr \in \mathcal{D}_{sr}$. As shown in Fig. 2 (upper right), we prevent the imputation head from influencing the policy parameterization by stopping its gradient from flowing to these shared-bottom embeddings. To give the imputation head its own parameters to learn the mapping $\hat{sr}_\phi$, we concatenate the embeddings, i.e., $[\mathbf{u}_s, \mathbf{v}_a]$, and send them through multiple Rectified Linear Units (ReLU), and a final dense layer to map to the ground truth survey response. The ReLU layers and the dense layer are learned by optimizing an appropriate loss function ℓ. For our case study, $\mathcal{D}_{sr}$ consists of survey responses in the scale of 1 to 5, with user studies showing

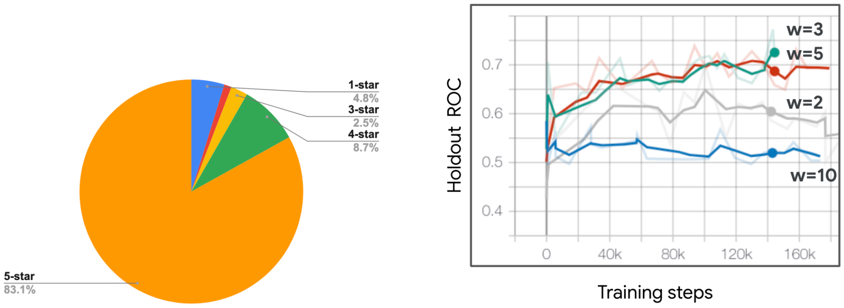

Fig. 3. Addressing class imbalance (*left*) by cost-sensitive learning (*right*). Upweighting the negative class survey responses (1/2/3 star responses) can significantly improve the holdout satisfaction AUC ROC of the imputation model. However, increasing the weight too much can lead to a satisfaction accuracy deterioration, and a negative effect on the policy head's accuracy.

that values of 4 and 5 are considered satisfying, whereas lower values show dissatisfaction. So, we considered a logistic loss, with a sigmoid for the last layer, to predict satisfying versus unsatisfying:

$$\min_{\phi} \sum_{sr \in \mathcal{D}_{sr}} \ell\left(\hat{sr}_{\phi}(\mathbf{u}_s, \mathbf{v}_a), sr\right). \tag{7}$$

The policy head is learned via REINFORCE,

$$\nabla_{\theta}\pi_{\theta} = \sum_{\tau \sim \beta} \sum_{t=0}^{|\tau|} \left[\frac{\pi_{\theta}(a_t|\mathbf{s}_t)}{\beta(a_t|\mathbf{s}_t)} \tilde{R}_t \nabla_{\theta} \log \pi_{\theta}(a_t|\mathbf{s}_t) \right] \tag{8}$$

where $\tilde{R}_t$ denotes the imputed reward, and $\frac{\pi_{\theta}(a_t|\mathbf{s}_t)}{\beta(a_t|\mathbf{s}_t)}$ does the off-policy importance weighting. We decompose $\tilde{R}_t$ as

$$\tilde{R}_t = R_t^e \times \tilde{R}_t^u, \tag{9}$$

i.e., the ground truth engagement reward R_t^e, and the satisfaction reward $\tilde{R}_t^u$ predicted by the imputation network.

The satisfaction imputation and the policy head are trained concurrently to optimize the (weighted) sum of the two losses.In practice, to prevent a poorly estimated imputation head from corrupting the policy head, we start the training of the policy head with engagement only reward R^e, and include the imputed $\tilde{R}^u$ only after the imputation head is properly trained.

5 What makes a good Satisfaction Imputation Model?

We now present some experimental findings on what makes a good satisfaction imputation model. To evaluate its predictive accuracy, we create a hold-out

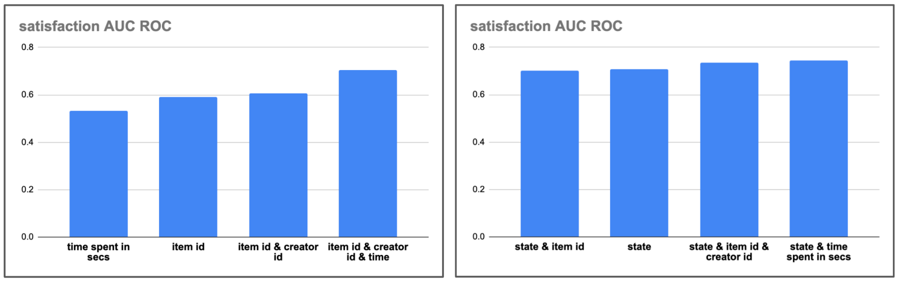

Fig. 4. *Left*: Using a set of features representing the item on the survey (i.e., item id & creator id & time spent on item in seconds), leads to an improved hold-out AUC ROC of the satisfaction imputation model, compared to using only individual action/item features. *Right*: Using only user state (as outputted by the RNN, and concatenated with context embeddings) performs equally well with when including all action feature embeddings, without user state (as shown in *top*). The satisfaction AUC ROC is further improved when including both user state and item-related features.

set consisting of user trajectories for users with at least one associated survey response. The AUC ROC achieved by the imputation model on the hold-out set is used as the offline evaluation metric.

Loss Function. One challenge associated with the survey response data is the class imbalance problem. In our case study, the majority of responses recorded are in the higher spectrum (Fig. 3, *left*). One hypothesis is that users tend to respond to surveys about items they find highly satisfying [25,4]. This creates a natural imbalance of survey values in the satisfaction data, leading the model to focus more on survey responses of higher values. An under-specified model can predict every item to be satisfying as a result. One simple approach to address this is through cost-sensitive learning [8], in which the negative class of non-satisfying state-action pairs are weighted more. We calibrate the prediction after to reflect the ground-truth distribution of satisfying vs non-satisfying survey responses [2]. Fig. 3 *right* compares the performance of the satisfaction head with different weights on the negative class. We found a weight of 3 or 5 perform the best according to the holdout AUC ROC.

Action Features. Fig. 4 *left* summarizes the predictive power of different action features, i.e., time spent, item id and creator id, on the quality of the satisfaction imputation head. We can see that when using a single feature to predict survey response, the continuous feature of the time the user spent interacting with the item is less informative compared to discrete features representing the item such as the embedding of the item id. The AUC ROC of a satisfaction imputation with item id as the only feature is further improved when including other features representing the item on the survey—we notice a slight improvement when including the creator id embedding, and a considerable improvement when also including time spent in seconds interacting with the item, on top of item id and creator id embeddings.

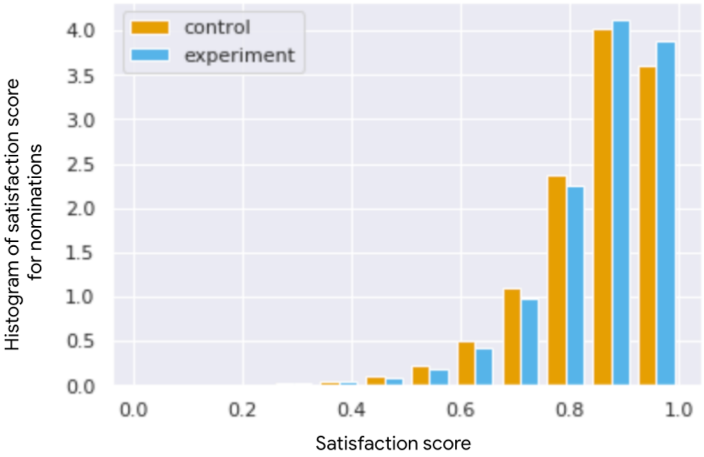

Fig. 5. Distribution of satisfaction scores for items nominated by model, for experiment arm (REINFORCE with satisfaction imputation model) versus control arm (REINFORCE without satisfaction in reward).

User State. We also evaluate the importance of including user state in learning the imputation model (Fig. 4 *right*). We can see that when using as features only the user state, as captured by the RNN over the sequence up until this point, concatenated with the label context embedding, we get the same hold out AUC ROC as the one achieved by using all features representing the action (Fig. 4 *left*, last bar). Concatenating the user state with action embeddings further improves the imputation model's predictive power. Thus, in what follows, our satisfaction imputation model will utilize both user state and action embeddings as features.

6 Live Experiments

Our case study involves a large scale two-stage recommender platform, where at the first stage multiple candidate generators retrieve potential candidates from the entire corpus; and the second stage involves a ranker model ranking the candidates and providing a final top-K recommendation list to be shown to the user [5]. To study the extent to which our approach can improve real user experiences, we apply our reward shaping approach onto a RL-based candidate generator, and conducted a series of A/B experiments. The control arm runs a REINFORCE agent learned using engagement-only reward [3]. In the experiment arm, we test our proposed approach of augmenting the policy network with a satisfaction imputation head (Fig. 2), and utilizing the imputed satisfaction reward along with the ground truth behavioral signals into the policy's reward, described in Section 4.4. Experiments are run for over a month on a fixed set of

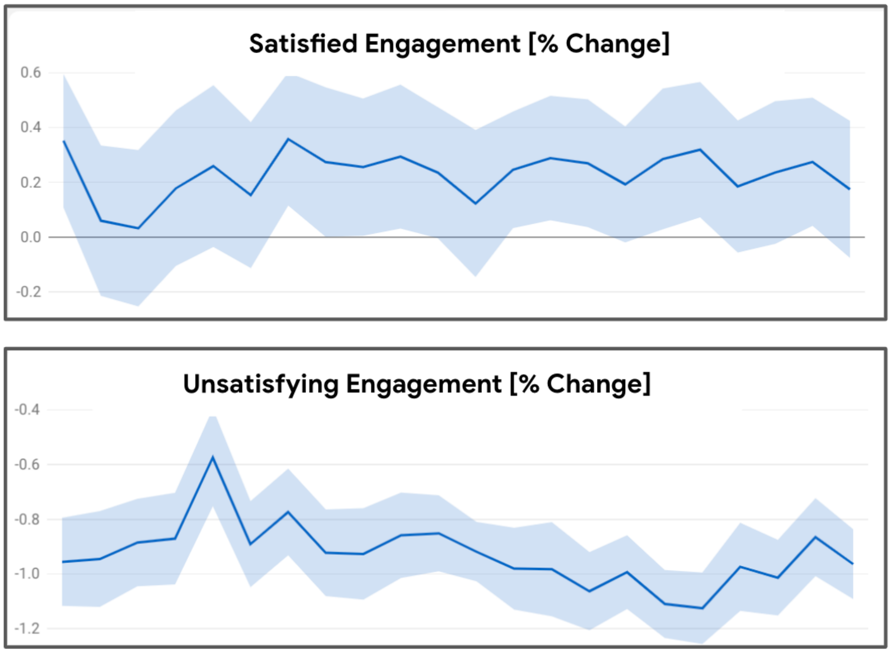

Fig. 6. Percentage improvements of online satisfaction metrics (y-axis) achieved by our proposed model over the course of a month (x-axis). On average, satisfied engagement is significantly increased by 0.23% (*top*), while unsatisfying engagement is significantly decreased by 0.93% (*bottom*), highlighting the value of our approach for driving user satisfaction.

randomly assigned user traffic to study the long-term effect. During this period, the model is trained continuously, with new interactions being used as training data with a lag under 24 hours.

Online Satisfied Engagement Metric. For evaluating whether the user experiences are improved, one could look at ground truth survey responses. An experiment which increases the average survey response over the experiment period would be considered driving more user satisfaction. In fact our experiment increases 5 star survey responses on average by **0.48%** and decreases 1 and 2 star survey responses by **1.89%**. However, we again run into the key challenge associated with survey responses which is sparsity. If we only measure on user-item pairs for which the users have responded to in a survey, we would only be looking at a very small percentage of the user interactions. To tackle this, we instead rely on a model-based metric predicting a survey response for each of the items the user has interacted with, and combining that with engagement metrics measured live. It is worth pointing out, the model used for measuring online satisfaction metric is independent of the imputation network we built, with a considerably different feature set and architecture. We cannot utilize the predictions of this

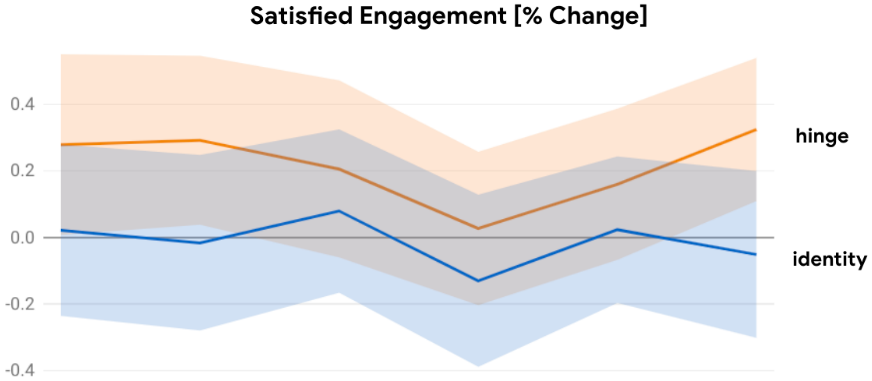

Fig. 7. Effect of choice of transformation function over the imputed satisfaction scores. Y axis denotes percentage improvements of online satisfied engagement metric, x-axis denotes the days over the course of the month, and the two lines refer to hinge and identity transformation. We find that filtering data with lower imputed satisfaction scores through the hinge function performed considerably better compared to raw imputations.

model directly into our reward, due to infrastructure complexities and freshness requirements.

Satisfaction Improvements. In Fig. 5 we show how the distribution of ground truth satisfaction scores over nominations coming from the control model (REINFORCE with engagement-reward) versus the experiment (our approach optimizing for a combination of satisfaction and engagement) compare. The x-axis represents the ground truth satisfaction probability score (used to calculate the live satisfied engagement time metric, and distinct from our imputation model predictions), with scores close to 1.0 indicating users being satisfied with their interactions, and close to 0.0 being unsatisfied. We can see that in both experiment and control arms, the majority of interactions are predicted to have a score larger than 0.5, indicating satisfying experience. Nevertheless, we can clearly see that our experiment increases the number of nominations with satisfaction scores greater or equal to 0.9, and decreases respectively nominations with a score less than 0.9. This demonstrates that our proposed imputation head is able to identify items which are satisfying to the users and shift the policy to select more satisfying items, further validating its predictive accuracy.

Fig. 6 compares the control and experiment arm on a live metric combining the model-based satisfaction metric and behavioral-based implicit engagement signals. We find that on average, satisfied engagement is increased by **0.23%**, while unsatisfying engagement is decreased by **0.93%**. Both results are statistically significant, signifying the value of reward shaping in driving user utility as specified by the reward. Furthermore, metrics orthogonal to the ground truth satisfaction scores, measuring other facets of user experience, significantly move towards the right direction: likes on items increase by **0.53%**, while dislikes are decreased by **1.11%** and dismissals decrease by **3.03%**. Note that we did not include these signals into the features or labels of our satisfaction imputation model. This further supports the point made in Section 1 that optimizing for satisfaction signals correlates well with improvements in post-engagement actions.

We found that an important design choice in the reward is the transformation function over the predicted satisfaction signal $r^u(s, a)$ by the imputation network, i.e., the probability of an item being satisfying to the user. Simply multiplying the engagement reward signals by the imputed probability of the item being satisfying (identity function) only decreases non-satisfied engagement, but did not lead to statistically significant improvements in satisfied engagement. It is critical to further differentiate highly satisfying items from less satisfying ones to allow the model to clearly prefer selecting such items. We found that in practice a simple hinge function performed the best, i.e., when imputed probability of an item being satisfying to the user is larger than a threshold, multiply with the probability, else completely zero out the engagement reward (Fig. 7). The threshold was tuned offline based on the ground truth response distribution, and the imputation network's predictions. We report results based on threshold set to 0.75.

Furthermore, we validated in live experiments some of our choices made offline. We found that predicting the probability of an item found satisfying by the user performed better than predicting the actual survey response they will give, i.e., cross-entropy loss gave better results compared to a square loss. We hypothesize that this could be the case due to better alignment with the loss used to train the model-based Satisfied Engagement live metric. Also, balancing the data to give more weight to unsatisfying survey responses and calibrating the prediction to account for the balancing (Fig. 3) was important for live improvements. Finally, we found that raising the satisfaction reward term to an exponent larger than 1 gave us slightly better results when using the identity transformation function; but for the hinge function, the improvement was not statistically significant.

7 Conclusions

This paper addresses the challenge of optimizing long-term user satisfaction in reinforcement learning-based recommenders. We advocate for reward shaping as a vital technique to align the recommender's goals with what users want. We argue that relying solely on engagement signals, as common in the current RL literature, does not adequately capture the nuances of user experience. Instead, incorporating user satisfaction signals is crucial. Recognizing the sparsity of these satisfaction signals, we propose improving a state-of-the-art REINFORCE recommender with a satisfaction imputation network to predict satisfaction scores for all interacted items. Through offline analysis and live A/B experiments on a large-scale commercial recommender platform, we demonstrate that integrating satisfaction imputation into the reward function effectively leads to more satisfying user experiences.

References

1. Beutel, A., Covington, P., Jain, S., Xu, C., Li, J., Gatto, V., Chi, E.H.: Latent cross: Making use of context in recurrent recommender systems. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining. pp. 46–54 (2018)
2. Chapelle, O., Manavoglu, E., Rosales, R.: Simple and scalable response prediction for display advertising. ACM Transactions on Intelligent Systems and Technology (TIST) **5**(4), 1–34 (2014)
3. Chen, M., Beutel, A., Covington, P., Jain, S., Belletti, F., Chi, E.H.: Top-k off-policy correction for a reinforce recommender system. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining. pp. 456–464 (2019)
4. Christakopoulou, K., Traverse, M., Potter, T., Marriott, E., Li, D., Haulk, C. Chi, E.H, and Chen, M.: Deconfounding user satisfaction estimation from response rate bias. In: Proceedings of the 14th ACM Conference on recommender systems, pp. 450–455 (2020)
5. Covington, P., Adams, J., Sargin, E.: Deep neural networks for youtube recommendations. In: Proceedings of the 10th ACM conference on recommender systems. pp. 191–198 (2016)
6. Dorigo, M., Colombetti, M.: Robot shaping: Developing autonomous agents through learning. Artificial intelligence **71**(2), 321–370 (1994)
7. Dulac-Arnold, G., Evans, R., van Hasselt, H., Sunehag, P., Lillicrap, T., Hunt, J., Mann, T., Weber, T., Degris, T., Coppin, B.: Deep reinforcement learning in large discrete action spaces. arXiv preprint arXiv:1512.07679 (2015)
8. Elkan, C.: The foundations of cost-sensitive learning. In: International joint conference on artificial intelligence. vol. 17, pp. 973–978. Lawrence Erlbaum Associates Ltd (2001)
9. Garcia-Gathright, J., St. Thomas, B., Hosey, C., Nazari, Z., Diaz, F.: Understanding and evaluating user satisfaction with music discovery. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. pp. 55–64 (2018)
10. Gu, S., Holly, E., Lillicrap, T., Levine, S.: Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In: 2017 IEEE international conference on robotics and automation (ICRA). pp. 3389–3396. IEEE (2017)
11. Guo, Q., Agichtein, E.: Beyond dwell time: estimating document relevance from cursor movements and other post-click searcher behavior. In: Proceedings of the 21st International conference on World Wide Web. pp. 569–578 (2012)
12. Hadfield-Menell, D., Milli, S., Abbeel, P., Russell, S.J., Dragan, A.: Inverse reward design. In: Advances in neural information processing systems. pp. 6765–6774 (2017)
13. Ie, E., Jain, V., Wang, J., Narvekar, S., Agarwal, R., Wu, R., Cheng, H.T., Chandra, T., Boutilier, C.: Slateq: A tractable decomposition for reinforcement learning with recommendation sets. In: Proceedings of the Twenty-eighth International Joint Conference on Artificial Intelligence (IJCAI-19). pp. 2592–2599. Macau, China (2019), see arXiv:1905.12767 for a related and expanded paper (with additional material and authors).
14. Lalmas, M.: Metrics, engagement & personalization. In: REVEAL workshop, The ACM Conference Series on Recommender Systems (2019)
15. Liu, F., Tang, R., Li, X., Zhang, W., Ye, Y., Chen, H., Guo, H., Zhang, Y.: Deep reinforcement learning based recommendation with explicit user-item interactions modeling. arXiv preprint arXiv:1810.12027 (2018)

16. Mataric, M.J.: Reward functions for accelerated learning. In: Machine learning proceedings 1994, pp. 181–189. Elsevier (1994)
17. Mehrotra, R., Lalmas, M., Kenney, D., Lim-Meng, T., Hashemian, G.: Jointly leveraging intent and interaction signals to predict user satisfaction with slate recommendations. In: Proceedings of The Web Conference 2019. pp. 1256–1267 (2019)
18. Mehrotra, R., McInerney, J., Bouchard, H., Lalmas, M., Diaz, F.: Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management. pp. 2243–2251 (2018)
19. Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., Kavukcuoglu, K.: Asynchronous methods for deep reinforcement learning. In: International conference on machine learning. pp. 1928–1937 (2016)
20. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602 (2013)
21. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. nature **518**(7540), 529–533 (2015)
22. Munos, R., Stepleton, T., Harutyunyan, A., Bellemare, M.: Safe and efficient off-policy reinforcement learning. In: Advances in Neural Information Processing Systems. pp. 1054–1062 (2016)
23. Ng, A.Y., Harada, D., Russell, S.: Policy invariance under reward transformations: Theory and application to reward shaping. In: ICML. vol. 99, pp. 278–287 (1999)
24. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on knowledge and data engineering **22**(10), 1345–1359 (2009)
25. Paulhus, D.L.: Measurement and control of response bias. (1991)
26. Schaal, S.: Is imitation learning the route to humanoid robots? Trends in cognitive sciences **3**(6), 233–242 (1999)
27. Schulman, J., Moritz, P., Levine, S., Jordan, M., Abbeel, P.: High-dimensional continuous control using generalized advantage estimation. arXiv preprint arXiv:1506.02438 (2015)
28. Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al.: A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. Science **362**(6419), 1140–1144 (2018)
29. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. MIT press (2018)
30. Sutton, R.S., McAllester, D.A., Singh, S.P., Mansour, Y.: Policy gradient methods for reinforcement learning with function approximation. In: Advances in neural information processing systems. pp. 1057–1063 (2000)
31. Wen, H., Yang, L., Estrin, D.: Leveraging post-click feedback for content recommendations. In: Proceedings of the 13th ACM Conference on Recommender Systems. pp. 278–286 (2019)
32. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning **8**(3-4), 229–256 (1992)
33. Yi, X., Hong, L., Zhong, E., Liu, N.N., Rajan, S.: Beyond clicks: dwell time for personalization. In: Proceedings of the 8th ACM Conference on Recommender systems. pp. 113–120 (2014)

34. Zhang, J., Hao, B., Chen, B., Li, C., Chen, H., Sun, J.: Hierarchical reinforcement learning for course recommendation in moocs. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 435–442 (2019)
35. Zhao, X., Xia, L., Zhang, L., Ding, Z., Yin, D., Tang, J.: Deep reinforcement learning for page-wise recommendations. In: Proceedings of the 12th ACM Conference on Recommender Systems. pp. 95–103 (2018)
36. Zheng, G., Zhang, F., Zheng, Z., Xiang, Y., Yuan, N.J., Xie, X., Li, Z.: Drn: A deep reinforcement learning framework for news recommendation. In: Proceedings of the 2018 World Wide Web Conference. pp. 167–176 (2018)