Marketing Science Institute Working Paper Series 2022

Report No. 22-103

Ensembling Experiments to Optimize Interventions Along Customer Journey: A Reinforcement Learning Approach

Yicheng Song and Tianshu Sun

Ensembling Experiments to Optimize Interventions Along Customer Journey: A Reinforcement Learning Approach

Yicheng Song
Clarson School of Management, University of Minnesota, ycsong@umn.edu

Tianshu Sun
Marshall Business School, University of Southern California, tianshus@marshall.usc.edu

Firms adopt randomized experiment to evaluate various interventions (e.g., new website design, creative content, and price promotion). However, most of randomized experiments are designed to identify the impact of one specific intervention in the customer journey. The literature on randomized experiments lacks a holistic approach to optimize a sequence of interventions along the customer journey. Specifically, locally optimal interventions unveiled by one-shot experiments might be globally sub-optimal when considering their interdependence as well as the long-term rewards. Fortunately, the accumulation of large number of historical experiments creates and tests various exogenous interventions at different stages of the customer journey and provides a new opportunity. This study integrates multiple experiments with the Reinforcement Learning (RL) framework in order to tackle the questions that cannot be answered by standalone one-shot experiments: How can we learn optimal policy with sequence of interventions along the customer journey by ensembling exogenous interventions from multiple historical experiments? And how can we utilize multiple historical experiments to guide future intervention trials to further improve the learnt policy? We propose the Bayesian Deep Recurrent Q Network (BDRQN) model that can leverage the exogenous interventions from multiple experiments to learn effectiveness of interventions at different stages of the customer journey and optimize them for long-term rewards. The Bayesian approach enables the proposed model not only identifying the long-term reward of various interventions but also estimating the distribution of those rewards. Beyond optimization within the existing interventions, the estimated distribution of rewards can guide the allocation of participants in future intervention trials to balance exploitation and exploration. In summary, the proposed model creates a two-way complementarity between RL and randomized experiment, and thus provides a holistic approach to learn and optimize interventions along the customer journey.

Key words: Reinforcement Learning, Customer Journey, Long-Term Reward Optimization, Bayesian Deep Recurrent Q Network Model (BDRQN), Randomized Experiment, Experiment Design

1. Introduction

Firms have long wished to optimize the sequence of interventions at various touch points along the customer journey. With more granular data (e.g., browsing, click, add-to-cart, promotion redemption) on customers' digital traits across public domains (e.g., social media, content websites, search engines) and the firms' websites, firms can increasingly intervene with the *right customers* at the *right time* with the *right policy*. With the help of digital tools, brands often have a large amount of

observational data on customers' online journey as well as their responses to various online interventions. Such fine-grained observational data facilitate the modeling of online consumer journey and the design of interventions to optimize key performance indicators like the click-through rate (Ghose and Yang 2009), conversion rate (Ghose et al. 2019) and user engagement (Zhang et al. 2019). However, observational data generated from naturally occurring situations faces two major challenges: first, the interventions ("actions") are often endogenously determined by customers' own state ("state") (Li and Kannan 2014); second and importantly, such data lack exogenous and diverse interventions at various touchpoints of the customer journey and they explore only a small subset of the policy space. Thus, firms can only infer and optimize the interventions ("actions") within a subset of action-state pairs that are endogenously generated in naturally occurring data. Using targeting models built on such observational data, firms may obtain suboptimal interventions and miss the opportunity to achieve a global optimum as they exploit only the endogenously limited action-state pairs.

On the other hand, the randomized experiment (A/B test) has been increasingly adopted by firms to evaluate various interventions (Anderson and Simester 2011), ranging from designing new features on the website (Huang et al. 2019), price promotion (Zhang et al. 2020), customer trust building (Gu and Zhu 2021) and the impact of recommender systems (Lee and Hosanagar 2021). Therefore, firms could learn and adopt optimal intervention from the treated/control policies via the experiment. However, the standalone experiment is designed with the goal to identify the impact of one specific intervention. Current randomized experiment analyze each of intervention separately without considering other interventions, therefore cannot be used to optimize interventions across the entire customer journey. This is especially true when different interventions have strong interdependencies (e.g., sequential promotions). Therefore, locally optimal interventions unveiled by the one-shot experiments could become globally sub-optimal when considering the sequence of intervention along the customer journey. In addition, even when an experiment's goal is to find the optimal sequence of interventions, the policy space would grow exponentially. It is extremely hard to overcome the "curse of dimensionality" and identify the optimal policy using a one-shot experiment with a large number of interventions. Moreover, preparing a randomized experiment on multiple touch points and coordinating them across channels may involve major costs and are also time-consuming: months are needed before any finding can be reached. Lastly, these standalone experiments are rarely being utilized to guide future intervention trials to balance the exploitation of learnt optimal interventions and the exploration of new/under-explored interventions. Thus, a disciplined, model-based approach is required to complement randomized experiments (Cui et al. 2021) and address the above limitations.

Table 1 Comparison of the RL+A/B Model with Other Methods

	One-Shot Experiment	Models with Observational Data	Proposed: RL+A/B
Endogeneity Bias	No	Yes	No
Diversity of Action-State Space	Low	Low	High
Time Cost of Learn Optimal Policy	Long	Short	Short
Monetary Cost of Learn Optimal Policy	High	Low	Low
Holistic Optimization of Customer Journey	No	Possible	Yes
Long-Term Reward Optimization	No	Possible	Yes
Guide Future Intervention Trials	No	No	Yes

Fortunately, the accumulation of a large number of experiments create and test various exogenous interventions at different stages of the customer journey. Our study explores and answers two research questions: **How can firms ensemble a large number of historical experiments to design a model-based algorithm to learn optimal policy with sequence of interventions along the customer journey? How to utilize multiple historical experiments to guide future intervention trials to further improve the learnt policy?** In this paper, we propose a Reinforcement Learning (RL) approach that ensembles those historical experiments: i.e., integrating multiple historical experiments (A/B test) using the RL model (RL+AB). We comprehensively compare RL+AB with models built on observational data and one-shot experiments in Table 1. Our proposed RL+AB model offers four major advantages: (1) the experiments allow firms to create exogenous variation in the intervention at different stages of the customer journey, and therefore increase the diversity of interventions at different stages of the customer journey, (i.e., to fully explore the potential state-action pairs). Such exogenous exploration in the state-action space is missing in endogenously formed observational data and presents the value of leveraging randomized experiments so as to optimize interventions along the customer journey. (2) In one experiment, there is only one pair of treated/control actions. But with n experiments, we are not limited to only n pairs of treated/control actions: the maximal possible combinations of different actions is 2^n. As the number of experiments grow, our RL+AB model enables firms to keep up with the exponential growth of the action space and take advantage of the inter-dependency between the sequence of actions; that is, firms can go beyond the focus on isolated treated/control actions for a locally optimal and short-term goal, and take a holistic perspective that utilizes the sequence of all possible actions to optimize sequence of interventions for long-term reward. (3) RL models are concerned with how intelligent agents should take actions in an environment in order to maximize their long-term rewards, and the key challenge is that the return reward of the action is only partially unveiled, and may become clear as more interaction happens. Such a model could help policy makers navigate through the often overwhelming task of designing optimal sequential polices in marketplaces such as the smart electronic market (Peters et al. 2013), music

playlist recommendation (Liebman et al. 2019), traffic forecasting (Zhou et al. 2020), career path planning (Kokkodis and Ipeirotis 2021) and sequential marketing promotions (Wang et al. 2019). By ensembling historical experiments, we can construct diverse and exogenous state-action pairs, which is ideal data to help RL models learn optimal interventions at different stages of the customer journey. (4) As firms will continue exploring new interventions in the future, the experimental method requires only randomly allocating participants among treated and control groups and rarely provides additional guidance. The firms might face a dilemma in allocating participants between the "exploiting learnt interventions to gain revenue" group and "exploring new interventions with uncertain outcomes" group. This offers the opportunity to apply the Bayesian Optimization (Frazier 2018) to balance exploring uncertain interventions, which could unexpectedly result in higher reward, against focusing on learnt policy with stable outcome.

In Section 3, we introduce a model that integrates RL and randomized experiments—the Bayesian Deep Recurrent Q-Network (BDRQN)—to learn optimal sequences of interventions along the customer journey. Built on multiple sets of experimental data, BDRQN first constructs diverse and exogenous state-action pairs at each touchpoint along the consumer journey, and then estimates the distribution of the long-term reward for each pair. Partnering with a national e-commerce platform in the US, we built and estimated the models using a detailed dataset involving 149,913 users across 10 randomized experiments (Section 4.1). We then evaluate the model with rejection sampling on holdout data (Section 4.2). The results show that adopting our model to exploit the learnt policy from these experiments leads to 7.3% to 43% improvement in terms of reward (i.e., profit) per episode (i.e., a sequence of user interactions) for the firm. To show the superiority of ensemble multiple experiments, we train the model with data from all 10 experiments versus just using fewer experiments. The results show that the increased exogenous action-state pairs empowered by multiple experiments lead to 7.1% to 36.2% performance improvement, as compared only exploring a smaller action-state space with fewer experiments. Using state-of-the-art RL algorithms as benchmarks, we compare their performance in long-term reward optimization and find that our method outperforms these baselines. Beyond optimizing interventions within the existing experiments, the model's results could also guide future intervention trials to balance the exploitation of learnt policy for immediate revenue and the exploration of new/under-explored interventions for future potential (Section 4.3). The experiments show that the model could effectively further improve the learnt policies to increase long-term rewards and efficiently allocate participants to avoid expensive and unnecessary trials. Therefore, the proposed RL+A/B approach creates a two-way complementarity between reinforcement learning and experiment. That is, by ensembling experiments enabled the RL model to learn optimal interventions along the customer journey, and the results of the RL model could guide the allocation of participants in new intervention trials

to balance exploitation and exploration. They jointly lead to a holistic approach of intervention learning and optimization along the customer journey.

2. Literature Review

Our research closely relates to two streams of literature. One is analysis interventions along customer journey, which contextually relates to this study. The second is reinforcement learning research in the management science literature that methodologically relates to our work. Our paper extends the two streams of literature that offer a new solution for sequential intervention optimization along customer journey by customizing a RL model to learn from multiple experiments.

Interventions along Customer Journey: Observational Data versus Randomized Experiments. Fine-grained observational data like clickstream data contains rich information about user behaviors (Ghose et al. 2019). Moe and Fader (2004) utilized detailed clickstream data to formulate a dynamic model on customers' online shopping behavior. Bronnenberg et al. (2016) studied customers' search behavior in detail using customer browsing histories and product search queries. Song et al. (2021) modeled customers' session-to-session transition and extracted characteristic paths that end with key conversions (i.e., purchase). However, many of these consumer journey modeling approaches based on clickstream data are challenged by endogeneity issues (Li and Kannan 2014), where interventions are often endogenously determined by customers' own states. On the other hand, a large body of literature leverages one-shot experiments to examine and optimize various interventions on the consumer journey such as website design (Huang et al. 2019), price promotions (Zhang et al. 2020) and the recommender system (Lee and Hosanagar 2021). Our study utilizes extensive clickstream data and diverse exogenous interventions at different stages of consumers' journeys by ensemble multiple experiments to build a reinforcement learning model that optimizes the long-term reward. Our study contributes to the literature on customers journey analysis by taking a holistic approach that fully leveraging historical experiments.

Reinforcement Learning in Management Science: Management science studies that develop and apply reinforcement learning algorithms focus primarily on policy planning and resource allocation problems. By casting the decision-making problem as the Multi-Armed Bandit (MAB), Katehakis and Veinott Jr (1987) aim to allocate limited resources between different actions so as to maximize their expected gains. The key challenge is that each action's properties are only partially unveiled, and can be better understood over time, as more interactions occur. Hauser et al. (2009, 2014) adopted MAB to optimize limited marketing resources in personalized advertisement targeting, which balance exploration-exploitation when evaluating the action pool. One major limitation of the MAB model is that its concerned about immediate feedback/reward but doesn't explicitly model future rewards. Therefore, another research stream aims to build

reinforcement learning models to optimize the long-term reward directly. Built on the Q-Learning framework (Watkins and Dayan 1992), Kokkodis and Ipeirotis (2021) adopted the Markov Decision Process (MDP) to operate on a knowledge graph of skill sets and dynamically recommend profitable career paths for users to optimize their long-term rewards. Also build on the Q-Learning framework, Wang et al. (2019) proposed a deep reinforcement learning model for the sequential targeting problem. They first built a predictive model to capture consumers' responses to various marketing actions. Then, they trained a deep reinforcement learning agent to interact with the consumer response predictive model to learn optimal sequential marketing strategies. Such model considers the dynamic sequential behavior of consumers aimed to optimize the long-term revenues for the firm.

Three major differences between our proposed model and previous RL studies in management science literature are: (1) we build an RL model to learn from multiple experiments so as to learn optimal interventions along the customer's journey. This enables us to explore exogenous and diverse state-action space than research that relies either on endogenous observational data or on one-shot experiments; (2) we develop a Bayesian Deep Recurrent Q-Network (BDRQN) to learn the expected reward and the uncertainty when adopting different interventions along the customer journey. This Bayesian analysis not only allows us to exploit the learnt policy to better target customers, and also enables a systematic solution to guide future intervention trial; (3) most previous works train and evaluate reinforcement learning models using a simulator (a predictive model that simulate the response and reward from the environment), whereas we directly train the model using the experimental data and adopt rejection sampling to evaluate the model. The rejection sampling results ensure the sampled data draw from the same distribution as applying such RL model online and the estimation of the long-term reward is unbiased.

3. Model

A typical reinforcement learning model has five core components (Sutton and Barto 2018): agent, environment, states, actions and rewards, all of which we'll concretize below in an e-commerce setting. In our research setting, the intelligent agent (i.e. reinforcement learning model) interacting with the environment (i.e., heterogeneous customers), when receiving the state of customer (i.e., a summary/abstraction of observed customer behavior until the current moment), the intelligent agent will execute the learnt optimal action policy (the action space is the combination of interventions from all experiments explored earlier, and each combination is treated as a unique action). The agent will receive immediate rewards (i.e., feedback from the customer measured as the immediate observed outcome of an agent's actions, can be represented as monetary reward)

from the customer. The reinforcement learning model aims to learn an optimal policy to maximize the cumulative reward from the customer in the long term. However, when applying classical reinforcement learning models to our problem, we face two main challenges:

- Many RL algorithms rely on the assumption of Markov Decision Processes where the future reward and the state are based solely on the most recent observation/state. Such Markov property rarely holds in real world environments especially like customer state cannot be fully described only by the most recently observations. A Partially Observable Markov Decision Process (POMDP) better captures the dynamics of many real-world environments by explicitly acknowledging that the sensations received by the agent are only partial glimpses of the underlying state.
- Most RL algorithms focus on the precise estimation of the expected long-term reward for different state-action pairs (Sutton and Barto 2018). Beyond estimation of a single number, managers also care about the uncertainty of long-term rewards if they adopt such an intelligent agent for decision making. The expected rewards from executing certain actions could be relatively high but they are not the best choice due to the high variance. However, such actions could be good candidates to explore in the future as they might lead to better policies for certain cases. From an exploitation perspective, mangers may prefer actions that lead to high expected rewards but low variance, but they will need to bear the risk of not exploring these high potential but uncertain actions. Thus, it is essential to enable the model to learn the uncertainty of the long-term reward.

In terms of the first challenge of POMDP, Hausknecht and Stone (2015) found that the performance of the Deep Q-Network (DQN) (Mnih et al. 2015) declines when given incomplete state (only those with most recent observations) and hypothesize that the DQN may be modified to better accommodate POMDP by leveraging Recurrent Neural Networks (RNNs) to learn state representation not only from recent but also from historical observations. In a similar vein, we adopt the Deep Recurrent Q-Network (DRQN), a combination of RNN and Deep Q-Network to learn the state of users from both recent observations and historical interactions. To tackle the second challenge, we aim to learn the distribution of long-term rewards for different state-action pairs. We add a Bayesian regression layer (Azizzadenesheli et al. 2018) on top of DRQN and name the proposed model Bayesian Deep Recurrent Q Network (BDRQN). The Bayesian regression layer will update the posterior distribution of parameters by synthesizing the prior distributions and observed data, and allow us to measure the uncertainty of long-term rewards for each state-action pair. With these adjustments, BDRQN can address the two challenges discussed above. The structure of the proposed BDRQN is illustrated in Fig 1. Next, we discuss state-representation learning via RNN to accommodate the POMDP, in Section 3.1, followed by introduce Bayesian regression into the reinforcement learning structure to estimate the distribution of long-term rewards in Section 3.2. Finally, we integrate these two modules into a unified model and discuss the model estimation in Section 3.3.

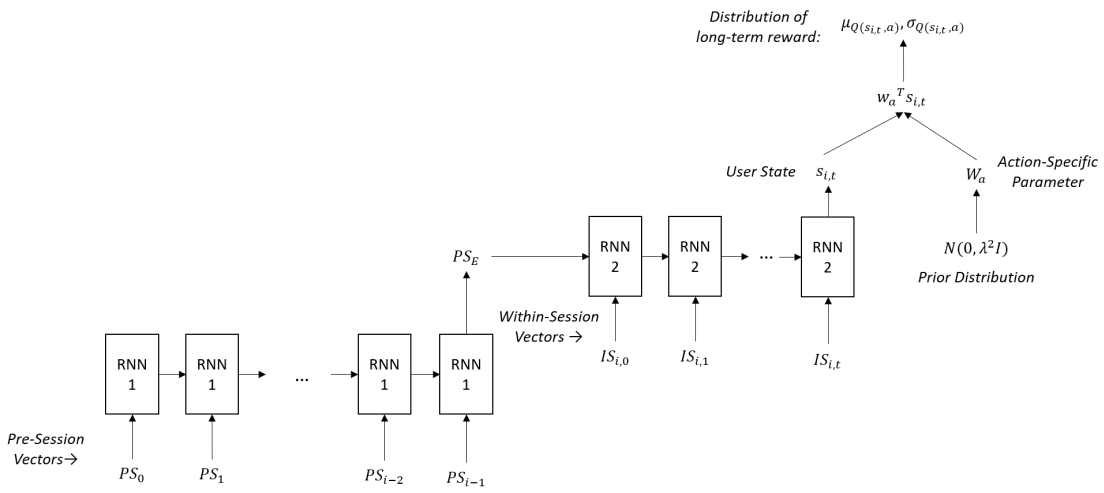

Figure 1 Outline of Bayesian Deep Recurrent Q-Network (BDRQN)

Note: Two-layer RNN structure is designed to deal with two types of data when facing with the tth page visit of the ith session from the user. The first RNN dealing with Pre-session vectors to summarize the user behavior before session i. The outcome of first RNN, PS_E, is the embedding for pre-session activity, will be used as initial embedding for the second RNN. The second RNN will process within-session vectors and generate state s_{it}, which is the summary of user state until to the tth page in the ith session. Meanwhile, W_a represents the Bayesian regression coefficients for action a, while different W_a share the same prior distribution $N(0, \lambda^2 I)$. With the s_{it} and action specific W_a, we can infer the expectation and variance of the long-term reward by taking action a under state s_{it}.

3.1. State Representation Learning via RNNs

Following Hausknecht and Stone (2015), to accommodate POMDP in optimizing interventions along the customer journey, we use RNNs to process customers' interaction history to generate the customer's state/embedding. Specifically, two types of time series are necessary when faced with the user's tth page visit in the ith session: (1) the pre-session vector, PS_j, summarizes the activities in session j, where $j < i$; this vector describes session-level summarization with coarse granularity, and (2) the within-session vector, $IS_{it'}$, which summarizes the activity on the t'th page within the ith session, describing the finer granularity activities within the focal session.

Correspondingly, as shown in Figure 1, two RNNs deal with the two types of timeseries above. The first RNN process the pre-session vectors to summarize the user state before session i. The outcome of this first RNN, PS_E, which is the embedding for pre-session activities, is used as the initial embedding for the second RNN. The second RNN then process the within-session vectors and generates state s_{it}, which is the summary of the user state until it reaches the tth page in the ith session. We adopt this two-layer RNN architecture because most experimental interventions (actions in the RL framework) are associated with a particular webpage, therefore we need a fine granularity state representation at the webpage level, which is captured via the second RNN. But if we also use such fine granulated data for previous session activities as well, we have too

many details for the pre-sessions, rendering the time series too long, and raising vanishing gradient concern (Ribeiro et al. 2020) for the RNN to summarize user state [1]. Of note, the two RNNs have their own parameter to estimate. For simplicity, we denote the parameter set of two RNNs as θ.

3.2. Q-Value Estimation via Bayesian Regression

Once the user state $s_{i,t}$ is ready, we can set the parameter w_a for each action a. Correspondingly, the cumulative reward of executing action a under $s_{i,t}$, Q-value $Q(s_{i,t}, a)$, can be simply estimated as $w_a^T s_{i,t}$. We can then obtain a Q-value for each action and the firm can adopt the action that leads to the highest Q-value estimation for exploitation purpose. However, such a single-number estimation suffers from the lack of uncertainty limitation mentioned before. Thus, we adopt Bayesian regression to estimate the distribution of the Q-value. Specifically, we model the prior distribution of w_a following a normal distribution with a prior mean zero along with prior variance $\lambda^2 I$:

$$w_a \backsim N(0, \lambda^2 I) \tag{1}$$

The estimated Q value then can be modeled as a normal distribution with prior variance of σ^2:

$$P(Q|w_a^T, s_{i,t}) \backsim N(w_a^T s_{i,t}, \sigma^2) \tag{2}$$

For all d observations that are associated with action a, we construct the user state matrix $S_a \in \mathbb{R}^{|s|} \times d$ and reward vector $r_a \in R^d$ from the observed data. Then, we update the posterior of $w_a \backsim N(\mu, \Sigma)$ by integrating the prior distribution and observed data as:

$$\Sigma = (\frac{S_a S_a^T}{\sigma^2} + \frac{1}{\lambda^2} I)^{-1} \tag{3}$$

$$\mu = \frac{1}{\sigma^2} \Sigma S_a r_a \tag{4}$$

Correspondingly, the closed-form posterior distribution of the Q value of taking action a under state $s_{i,t}$ can also be represented as normal distribution $N(\mu_Q, \sigma_Q)$ with

$$\mu_Q = {s_{i,t}}^T \mu \tag{5}$$

$$\sigma_Q = {s_{i,t}}^T \Sigma \; s_{i,t} \tag{6}$$

The posterior distribution of the Q value tells us the mean μ_Q and variance σ_Q for each state-action pair, which not only enables firms to learn a stable high-reward policy but also can guide future intervention trials to explore high-potential interventions as we discuss in Section 3.4.

[1] We also try a single RNN to process all the fine granularity data, the performance is inferior to the two-layer RNN.

3.3. Bayesian Deep Recurrent Q Network

With above module (1) state representation learning via RNNs and (2) Q-value estimation via Bayesian regression, we integrate them into a unified model, BDRQN, and estimate all parameters via end-to-end learning. To show the estimation process of the proposed model, we start with the derivation of Q value, $Q(s_t, a)$, which defines the cumulative reward obtained by executing action a under the customer state s_t. Generally, the customer with state s_t exposed to action a at page t, the immediate reaction (e.g., product order) of this customer is defined as the numerical reward r_t, and leads to customer state s_{t+1} in the next period $t+1$. Thus, we express the Q function as:

$$Q(s_t, a) = r_t + \gamma \max_{a'} Q\left(s_{t+1}, a'\right) \tag{7}$$

where γ is the predefined discounted factor (e.g. $\gamma = 0.99$) that balances between the current and future rewards as $\gamma = 1$ whereby all future rewards are fully considered during interaction t and $\gamma = 0$ only the immediate reward is counted. It is well established that parameters in the Q-Learning framework can be estimated via regression on temporal differences (Sutton and Barto 2018). Thus, Q-learning can update its parameter by minimizing the temporal difference between the left and right sides of Eq 7. We also adopt Double-Q-Learning (Van Hasselt et al. 2016) and Dueling Architectures (Wang et al. 2016). The main benefits of these variants are improving the learning stability and generalize learning across actions without imposing any change to the underlying reinforcement learning algorithm. Thus, there is Q Network of the focal training model and another Q_{target} Network for the target model. The target model is the copy of the focal model at beginning and get updated every M steps. The estimation process to learn the parameters θ of two RNNs in the Q Network is:

1. Find the action a' leads the maximal Q under state s_{t+1}: $\max_{a'} Q\left(s_{t+1}, a' | \theta\right) = \max_{a'} \left({w_{a'}}^T s_{t+1}\right)$.
2. Get the Q value of taking action a' at state s_{t+1} via target Q Network: $Q_{target}\left(s_{t+1}, a' | \theta_{target}\right)$.
3. The temporal difference error is:

$$Err = \frac{1}{2}[Q(s_t, a|\theta) - (r_t + \gamma Q_{target}(s_{t+1}, a'|\theta_{target}))]^2 \tag{8}$$

4. Parameter θ in RNNs can be updated by gradient descending with learning rate τ:

$$\theta = \theta - \tau \frac{dErr}{d\theta} \tag{9}$$

Therefore, given w_a, parameter θ can be estimated separately. Thus, the full model estimation process consists of the Bayesian update of w_a via Eq. 3 and 4, along with the estimation of θ via Eq. 9. The complete estimation process is shown in Algorithm 1.

Algorithm 1 Bayesian Deep Recurrent Q Network Estimation

Set the empty Reply Buffer $RB()$ and load the historical experiment data into the buffer.
Initialize RNN parameter θ, and Bayesian parameter W_a for each action.
Set N as the interval for Bayesian Sampling, and M as the interval for Target Model Update.
Copy RNN parameter θ to target RNN with θ_{target}, regression parameter $W_{a,target} = W_a$.
while $step <= num_{steps}$ **do**
    Load a minibatch data from RB()
    Get s_{t-1} based on RNN
    Get s_t based on RNN_{target}
    Get the temporal difference regression error between state s_{t-1} and s_t based on EQ 8
    Update the RNN parameter θ based on EQ 9
    Bayesian Posterior Update based on EQ 3 and 4
    For every N steps: Thompson Sampling $w_a \backsim N(\mu, \Sigma)$
    For every M steps: $\theta_{target} = \theta$ and $W_{a,target} = W_a$
    $step = step + 1$
end while

3.4. Guide Future Intervention Trial

Beyond learning optimal intervention policy using historical experiments, the result of the proposed model can also guide future intervention trial to further improve the learnt policy, especially for (1) allocating new sample to trial existing interventions and (2) explore new interventions. In the first scenario, the firm has extra resources to recruit additional subjects into existing interventions to refine the policy learnt from existing treatments/controls, but must decide how to allocate their limited sample. In the second scenario, the firm needs to design an entirely new intervention to further improve the targeting model but must decide which state-action pair to explore.

Eqs. 5 and 6 define the distribution of the Q-value estimation for any state-action pair, which can be used to guide future intervention trial. It is straightforward to deal with those state-action pairs that lead to Q-value of low variance: we can exploit the state-action that leads to a Q-value with a high mean, while the state-action with a low mean Q-values will be weeded from the consideration set. But for state-action pairs with high variance, we need a systematical solution to balance the promise of explore those actions and the costs of running such expensive and uncertain trials. We adopt the Bayesian optimization approach to balance the need to explore uncertain actions (exploration), which might unexpectedly bring high rewards, against focusing on learnt actions we know have stable rewards (exploitation). This solution can be used to guide future intervention trial in the above two scenarios.

For every incoming traffic/state, we determine which action to evaluate based on the distribution of the Q values and acquisition functions. Acquisition functions are heuristics for how desirable it is to evaluate an action given a state. We explore three options of acquisition functions below, and detail the implementation of each acquisition function in Appendix B.

1. Expected Improvement (Frazier 2018): This acquisition function will choose the next action as the one with the highest expected improvement over the current maximum reward.

2. Probability of Improvement (Kushner 1964): This acquisition function will choose the next action that has the highest probability of improvement over the current maximum reward.

3. Thompson Sampling (Thompson 1933): For every recommendation, we sample a distribution from the posterior distribution of the cumulative reward and choose the optimal action directly from the sampled distribution.

After adopting the actions recommended from these acquisition functions, we evaluate how these Bayesian optimization approaches can balance exploration and exploitation in Section 4.3.

4. Empirical Analysis

4.1. Context and Data

We partner with an e-commerce platform in US to evaluate the performance of the proposed BDRQN model. The platform is a pioneer in the industry that uses randomized filed experiments to evaluate various policies and interventions. The platform has conducted hundreds of experiments and their trials include experiments with new web page design, price promotion, manipulating reviews on product page, email campaigns, and widgets redesign, to name a few. The tremendous amount of randomized experimental data on this platform makes it an ideal testbed for this study.

Table 2 Data Summary

Entity	Summary
Num of Experiments	10
Num of Users	149,913
Time Span	07/01/2017 - 10/31/2017
Num of Distinct Actions	22
Num of Sessions	348,072
Total Sales	$3,261,158.78

We aimed to find a period during which multiple experiments were conducted (we had no requirement for an exact alignment of start and end dates) with overlaps of treated users from different experiments to ensure numbers of distinct action combinations. Based on this criterion, we chose 10 experiments that took place from July 1 to Oct 31, 2017. Specifically, our dataset includes 75,913 treated users and some are being treated in multiple experiments. As each experiment is designed

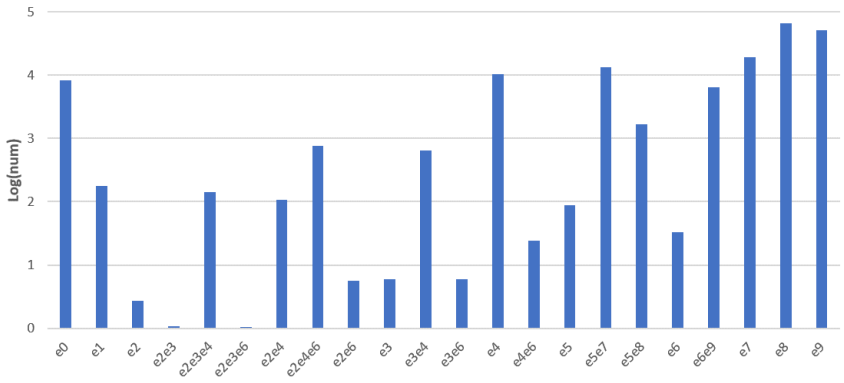

Figure 2 Number of users (log10() scale) with different treatment. e2 means users are only treated under experiment 2, e2e3 means the users are treated under both experiment 2 and 3.

on a specific web page and some of them are hosted on the same page, each combination of treatment that located on the same page should be considered as a unique action in the reinforcement learning framework. Meanwhile, we should also treat the control policy in any experiment as a separate action. Similarly, the control policies of experiments that were designed on the same webpage will be grouped into one action. We also randomly chose 74,000 users not being treated in any experiment in that period. In total, Our dataset includes 149,913 users across control and treated group across the selected 10 experiments. For all the selected users, we collected their clickstream data from Jan 1 (6 months before the first experiment, to construct pre-session vectors) to Oct 31 in 2017. Table 2 shows some high-level statistics of the dataset and Figure 2 shows the number of users under different action combinations in log10() scale.

Next, we show how to materialize state s_t and reward r_t in our empirical setting. We first construct the time series feature as pre-session vectors and within-session vectors to feed two RNNs. To construct pre-session vectors, we temporally segment the sequence of click-streams, as any two consecutive sequences that are more than 12 hours apart will be treated as two sessions. Then we summarize the users' time spent on different web pages in that session, the number of purchases of different types of products in that session, along with the time gap from the previous session as the pre-session vector, PS. To construct within-session vectors, we temporally segment a session into within-session segments based on users' web page visit(s) to any specific webpage where the experiment(s) take place. For instance, if experiment 1 is conducted on product page while experiment 2 is associated with builder page, then a customer's visit "homepage→category page→product page→order page→builder page" can be divided into two within-session segments ("homepage→category page→product page" and "product page→order page→builder page"). For each within-session segment, we construct a within-session vector IS similar to that of PS. Finally, the reward associated with the within-session segment is set as total money spent (cents) on all the products in the focal within-session segment.

4.2. Off-Policy Training and Evaluation

Many RL algorithms adopt the "on-policy" style whereby an agent actively interacts with an environment to learn from its own collected experience and evaluate the learnt model in the same setting. Historically, these algorithms have been evaluated on simple hand-designed problems with a small number of states. Recently, several studies employed simulator (e.g., Atari video games) as a testbed (Mnih et al. 2015), or directly play with human experts on those well-designed games (Silver et al. 2016), to evaluate reinforcement learning algorithms. These on-policy algorithms are challenging to apply to complex real-world problems (e.g., medical diagnosis and business decisions) because using an unjustified RL system to interact with the real-world environment can be extremely expensive and risky due to unintended (negative) outcomes. On the other hand, high-fidelity simulators where these applications operate are challenging to build as well. Fortunately, pre-collected real-world data can be utilized to make RL training/testing feasible, which is known as "off-policy" RL.

Off-policy RL uses a fixed offline dataset of logged interactions (with no further interactions with the environment), which is an important tool for real-world applications. Off-policy RL can help (1) train RL models using existing data, (2) and empirically evaluate RL models based on their ability to utilize a fixed dataset of interactions. Previous research has shown that off-policy trained RL will achieve comparable or even better performance than on-policy RL in video game settings (Agarwal et al. 2020). From a model training perspective, the Q-learning framework upon which BDRQN is built is a well-known off-policy RL algorithm (Sutton and Barto 2018); we train our model using 80% of our dataset. For evaluation, we adopt Fixed-M per-episode rejection sampling (Fixed-M PERS) (Mandel et al. 2016) to evaluate the performance of the proposed model using the holdout dataset (20% of the data). Fixed-M PERS samples from the achieved data and more favorably select those episodes (ordered trajectory of action, reward and state) that follow the policy of the learnt RL model. Fixed-M PERS is a well-established RL off-policy evaluation method with several nice properties: (1) Given the interaction history and the fact that the algorithm accepts episode samples of observations and rewards, Fixed-M PERS has proved that the accepted episode is drawn from a distribution identical to the distribution that the algorithm would encounter if it were run online, which is known as *samples true property*. (2) Based on the episode samples accepted by the Fixed-M PERS, we derive an unbiased estimation of the reward obtained by the RL algorithm in episodes as it were run online. This is known as *unbiased estimation of episode performance property*.

These features make Fixed-M PERS an ideal choice when adopting archived experimental data to evaluate a proposed model. As Fixed-M PERS offers the flexibility to set the length of episode H to sample episodes with different lengths, we evaluate set $H = 1, 4, 8$. When $H = 1$, we sample

only the next observation as an episode. When $H = 4$ or 8, we aim to sample the next consecutive 4 or 8 observations as an episode. For all sampled episodes, we calculate the average reward per episode and use the average reward to compare the performance of different models. Specifically, we compare BDRQN with the following algorithms:

1. **Original Website Intervention**. For all the records in the holdout dataset, we collect episodes with different lengths and calculate the average reward per episode. This original intervention policy provides a good baseline for the comparison of other model-based benchmarks.

2. **Optimal Intervention from Experiments**. The firm chooses the optimal interventions between treated and control interventions for all 10 experiments after comparing their performances. As these experiments are designed as standalone experiments without considering dependency between interventions, they become a useful benchmark with which to compare with model-based benchmarks. We sample the episodes from holdout data that matches with the optimal interventions derived from standalone experiments and obtain the average episode reward.

3. **Deep Q-Network (DQN)** (Mnih et al. 2015). We adopt the classical DQN model, which uses a deep feed-forward Neural Network to process only the most recent observation (no historical data) to summarize the state of the user. Then, the user state is connected with a regular dense layer (a regression layer) to predict the Q value for different actions. To ensure learning stability and generalizability, we also adopt Double-Q-Learning and dueling architectures in DQN.

4. **Deep Recurrent Q-Network (DRQN)** (Hausknecht and Stone 2015). Similar to BDRQN, DRQN also adopts two RNNs to process pre-session and within-session vectors to obtain the state representation of the user using both recent and historical observations. The difference is that DRQN uses a regular dense layer (regression layer) to predict rewards based on the state, while BDRQN utilizes Bayesian regression. To ensure learning stability and generalizability, we also adopt Double-Q-Learning and Dueling architectures in DRQN.

Table 3 Average Reward per Episode (cents) Across Different Models

Setting	Original Intervention	Optimal Intervention	DQN	DRQN	BDRQN
Fixed-M PERS @1	498.5	502.9	510.6	527.9	534.9 (7.30%↑)
Fixed-M PERS @4	2001.3	2058.0	2165.6	2366.9	2421.6 (21.00%↑)
Fixed-M PERS @8	3989.5	4250.2	4697.6	5521.7	5704.9(42.99%↑)

Note: The Original Intervention provides the baseline of the average reward per episode with lengths of 1, 4 and 8. We also collect episodes that match with the Optimal Intervention derived from standalone experiments and obtain the average reward. For all RL models (DQN, DRQN, BDRQN), we use Fixed-M PERS to sample episodes of different lengths. Therefore, we can obtain the average reward of sampled episode from DQN, DRQN and proposed BDRQN model. The percentage of improvement for the average reward compared with the original intervention is shown in parentheses in the last column.

Table 3 shows the average reward per episode across different models under different episode lengths. First, it is not surprising that the average rewards of the Optimal Intervention derived from experiments are greater than that of the Original Website Intervention, which indicates that it is beneficial to adopt the Learnt Optimal intervention derived from randomized experiments. But we also notice that all the RL models that comprehensively model the interdependency of interventions from all 10 experiments achieve higher average reward than those of the Optimal Intervention from one-shot experiments. Such comparisons demonstrate that the locally optimal interventions unveiled by the one-shot experiments become globally sub-optimal. Thus, it is necessary to ensemble experiments using RL models, to take a holistic approach to optimize interventions along the customer journey. Second, by comparing the performance among the three RL models, we find that both the DRQN and BDRQN models lead to higher average reward compared with the DQN model. The superiority of the DRQN and BDRQN models over the DQN confirms the need to adopt RNNs so as to incorporate the historical data to accommodate the POMDP. Otherwise, the most recent observations unveil only part of the customer state and lead to inferior decision making. Compared to the DRQN, the BDRQN model, equipped with Bayesian regression, is superior in improving the average reward in the range of 1.3% to 3.3%. Rather than only estimating the expectation of the Q value, the BDRQN model draws a distribution for each state-action Q value estimation, which allows the model to directly incorporate the uncertainty over the Q-function and learn posterior distribution, resulting in efficient exploration/exploitation. Finally, by comparing the differences of average rewards across different episode lengths, we find that the improvement of the average reward for the RL models is inconsistent. When $H = 1$, the improvement of BDRQN is 7.3% higher than the current policy. With $H = 4$, the average reward improvement of BDRQN increases to 21%. The improvement from 7.3% to 21% confirms that the reinforcement learning model optimizes the long-term reward rather than focusing myopically on the immediate reward. Such improvement becomes even more significant as the BDRQN's average reward improvement surges to 42.99% when set $H = 8$. These comparisons support adopting RL models, which could help firms optimize their long-term rewards.

Table 4 Average Reward per Episode (cents) of BDRQN with Different Experimental Data

Setting	1 Experiment	5 Experiments	10 Experiments
Fixed-M PERS @1	499.6	511.9	534.9
Fixed-M PERS @4	2023.5	2162.1	2421.6
Fixed-M PERS @8	4188.77	4712.0	5704.9

Note: We train BDRQN model with data from only one experiment (the one with maximal number of treated users), five experiments (the top five with the greatest number of treated users) and all 10 experiments. Then we adopt the Fixed-M PERS to sample episodes with length of 1, 4 and 8. The average reward of the sampled episode across different settings shows promise of fueling RL with multiple experiments.

To evaluate the value of diverse exogenous interventions derived from multiple experiments, we train the BDRQN model using data from all 10 experiments and contrast this with using data from only one experiment (the one with a maximal number of treated users) and then from five experiments (the top five experiments with the most treated users). Then, we apply the Fixed-M PERS to sample the episode under the different settings of H from the holdout dataset. As shown in Table 4, using data from additional experiments leads to a significantly higher average reward per episode. When we train the BDRQN model with data from five experiments, we improve the average reward per episode in the range of 2.4% to 12.5% compared with the model that has learnt from only one experiment. This improvement soared to the range of 7% to 36.1% after training the model with data from all 10 experiments, indicating a nonlinear but exponential growth trajectory. With data from only one experiment, we can only explore the action space with two possible actions (treated and control). Integrating data from multiple experiments exponentially increases the exploitable action space and enables the BDRQN model to explore much more diverse state-action pairs to learn optimal interventions to target different users, therefore resulting in a significant higher reward.

In summary, our findings from inter-model (Table 3) and inter-data comparisons (Table 4) illustrate a clear advantage of fueling RL with multiple experiments to optimize interventions along the customer journey.

4.3. Future Intervention Trial Evaluation

We now evaluate the outcomes by adopting the intervention trial design from different acquisition functions. Specifically, we are interested in whether our model can guide the sample allocation in future intervention trials so as to balance the exploitation of known promising actions and the exploration of actions with potential to further improve the model.

Previously, after training the model with 80% of the data, we use the remaining 20% as testing data to evaluate the model. In this task, we use the remaining data for a different purpose to simulate how the reinforcement learning agent will evolve when following the future intervention trials as recommended by different acquisition functions. Specifically, we treat 20% of the data as incoming traffic after the model is trained. For a incoming traffic with state s, the acquisition function will recommend action a. If the recommended action matches the action in the actual data, we use this record to update the posterior distribution of w_a via Eqs. 3 and 4. We simulate the future trials by allocating incoming data as new samples in the following two scenarios: (1) Allocate new samples to trial existing interventions to refine the policies, (2) Run new interventions trials to further improve the policies. The detailed simulation process is shown in Algorithm 2.

Algorithm 2 Simulate Future Intervention Trials via Hold-Out Data

Input: Holdout dataset D, d represents a record in D contains the state and action
$iteration = 1$
while $iteration \leq num_{iterations}$ **do**
  for $d \in D$ **do**
    Get s, start state of d
    Get a, action of d
    Get the recommended action $ra(s)$ via the chosen acquisition functions for state s
    if $ra(s) == a$ **then**
      Update BDRQN model with record d via EQ 3 and 4
      $num_{update} = num_{update} + 1$
    end if
  end for
  $iteration = iteration + 1$
end while

Table 5 Q-Value (Mean and Standard Deviation) Comparison Before and After Model Update for Scenario 1

		Before Update		After Update		P-value
AF	Rank	Mean	S.D	Mean	S.D	
Expected Improvement	Top 1	2885.36	68.62	2951.77(2.30%↑)	63.13(8.01%↓)	$< 0.001\%$
	Top 2	2876.54	70.76	2928.31(1.79%↑)	66.22(6.41%↓)	$< 0.001\%$
	Top 3	2853.81	79.39	2886.16(1.13%↑)	74.36(6.33%↓)	$< 0.001\%$
Probability of Improvement	Top 1	2885.36	68.62	2949.13(2.21%↑)	64.08(6.61%↓)	$< 0.001\%$
	Top 2	2876.54	70.76	2925.66(1.70%↑)	65.39(7.58%↓)	$< 0.001\%$
	Top 3	2853.81	79.39	2892.84(1.36%↑)	75.27(5.18%↓)	$< 0.001\%$
Thompson Sampling	Top 1	2885.36	68.62	2946.96(2.13%↑)	64.73(5.56%↓)	$< 0.001\%$
	Top 2	2876.54	70.76	2927.75(1.78%↑)	67.91(4.02%↓)	$< 0.001\%$
	Top 3	2853.81	79.39	2890.56(1.28%↑)	76.28(3.91%↓)	$< 0.001\%$
Random Select	Top 1	2885.36	68.62	2886.62(0.25%↑)	67.58(1.51%↓)	3.21%
	Top 2	2876.54	70.76	2877.59(0.21%↑)	69.86(1.27%↓)	6.76%
	Top 3	2853.81	79.39	2854.71(0.10%↑)	77.65(2.19%↓)	12.58%

Evaluation of Future Intervention Trial for Scenario 1: For each state, we first choose the actions that lead to the top-three expected Q-values as they represent the learnt optimal policies. Then we summarize the mean and standard deviation (S.D) of the top three Q-values by aggregating all of the states. We compare the mean and S.D of Q-value estimation from the BDRQN model before and after the model is updated via the above simulation, the results are shown in Table 5. We also add a naive benchmark that *randomly selects* the same amount of data (rather than selecting actions that match the acquisition function recommendation) to update the model. The comparisons clearly show the superiority of adopting the acquisition function: (1) We

Table 6 Q-Value (Mean and Standard Deviation) Comparison After Model Update between Scenario 1 and 2

		Scenario 1 After Update		Scenario 2 After Update		P-value	Number of Impression Trials for New Experiment
AF	Rank	Mean	S.D	Mean	S.D		
Expected Improvement	Top 1	2951.77	63.13	2950.33	63.39	1.14%	121k(77.83%↓)
	Top 2	2928.31	66.22	2925.25	67.31	$< 0.001\%$	
	Top 3	2886.16	74.36	2883.57	75.07	0.02%	
Probability of Improvement	Top 1	2949.13	64.08	2949.75	64.58	16.76%	125k(77.10%↓)
	Top 2	2925.66	65.39	2922.32	66.03	0.01%	
	Top 3	2892.84	75.27	2888.46	75.15	$< 0.001\%$	
Thompson Sampling	Top 1	2946.96	64.73	2945.92	64.22	5.53%	186k(65.93%↓)
	Top 2	2927.75	67.91	2926.28	67.35	1.48%	
	Top 3	2890.56	76.28	2885.92	76.92	$< 0.001\%$	

find that the mean values for the top three average Q-Value increase significantly after the model update, which shows that the recommended policy could not only exploit the learnt existing policy but also utilize additional samples to explore state-action space to optimize the intervention to further improve the policy. (2) The S.D of the top three Q-Values decreases after the model update, which indicates that adopting the mechanism could decrease the uncertainty of the learnt optimal policies. (3) All three acquisition functions achieve the above two goals, and both goals are much higher than those of the random selection.

Evaluation of Future Intervention Trial for Scenario 2: Different from Scenario 1, which uses all 10 experiments to train and update the model, Scenario 2 simulates how to best allocate incoming traffic with new intervention. Thus, we first use experiments 1–9 from 80% of the original training data to train the model, and then update the model using the original holdout 20% of the data that contains experiments 1– 10 to simulate experiment 10 as new intervention. We compare the Q-value estimation from the BDRQN model after the model is updated in Scenarios 1 and 2. We are interested in how much incoming data (new samples) is needed to ensure that the top average Q-values in Scenario 2 are statistically insignificant different from that in Scenario 1, which indicating that the best learnt policies in the two scenarios are almost identical. As shown in Table 6, the number of trials for experiment 10 (the number of impressions associated with the treated action in experiment 10, not the number of treated users) must exceed 100K to ensure that the best policies derived from the two scenarios are indifferent. But this number remains much lower than that of the original experiment, with 60%–80% fewer samples. This shows that our model could guide the allocation of participants on new trials to improve policy with greater efficiency. Considering that the cost of recruiting subjects for intervention trials is always high, the proposed model is a promising tool to guide future intervention trials.

5. Conclusion

Randomized experiments (A/B testing) have been widely adopted by firms to evaluate various interventions. However, locally optimal interventions unveiled by one-shot experiments might be globally sub-optimal when considering their inter-dependency as well as the long-term rewards along the customer journey. The literature on randomized experiments lacks a holistic approach to optimize interventions along the customer journey. Fortunately, the accumulation of a massive number of historical A/B tests assesses various exogenous interventions at different stages of customers' journey and provides a new opportunity. This study proposed a Bayesian Deep Recurrent Q Network model to utilize diverse and exogenous interventions from multiple experiments to learn interventions at different stages of customers' journey to optimize the long-term reward. The evaluation results show that adopting our model to learn policy from historical experiments leads to a 7.3% to 43% improvement in terms of reward (i.e., profits) per episode for the platform. In addition, we also identify the source of performance lift by (1) inter-model comparison: comparing the performance of our model with benchmark models and (2) inter-data comparison: utilizing data from multiple experiments versus using fewer experiments. The results from our inter-model comparison show that the proposed model outperforms all benchmarks due to the model design which better addresses POMDPs problem and captures the distribution of reward estimation. The results from our inter-data comparison show that the additional exploration in action-state space enabled by multiple experiments leads to a significant performance gain. Moreover, our model can also be used to guide the design of future intervention trials. The empirical results show the proposed model could well balance the exploitation of learnt policy to gain revenues and explore new/under-explored interventions with potential to further improve the learnt model.

In summary, the findings based on our model evaluation illustrate a clear advantage of fueling RL with multiple experiments. The proposed RL+A/B approach creates a two-way complementarity between reinforcement learning and experiment, and thus provides a holistic approach to intervention learning and optimization along the customer journey.

References

Agarwal R, Schuurmans D, Norouzi M (2020) An optimistic perspective on offline reinforcement learning. *International Conference on Machine Learning*, 104–114 (PMLR).

Anderson ET, Simester D (2011) A step-by-step guide to smart business experiments. *Harvard business review* 89(3):98–105.

Azizzadenesheli K, Brunskill E, Anandkumar A (2018) Efficient exploration through bayesian deep q-networks. *2018 Information Theory and Applications Workshop, ITA 2018, San Diego, CA, USA, February 11-16, 2018*, 1–9 (IEEE), URL `http://dx.doi.org/10.1109/ITA.2018.8503252`.

Bronnenberg BJ, Kim JB, Mela CF (2016) Zooming in on choice: How do consumers search for cameras online? *Marketing science* 35(5):693–712.

Cui TH, Ghose A, Halaburda H, Iyengar R, Pauwels K, Sriram S, Tucker C, Venkataraman S (2021) Informational challenges in omnichannel marketing: remedies and future research. *Journal of Marketing* 85(1):103–120.

Frazier PI (2018) Bayesian optimization. *Recent Advances in Optimization and Modeling of Contemporary Problems*, 255–278 (INFORMS).

Ghose A, Ipeirotis PG, Li B (2019) Modeling consumer footprints on search engines: An interplay with social media. *Management Science* 65(3):1363–1385.

Ghose A, Yang S (2009) An empirical analysis of search engine advertising: Sponsored search in electronic markets. *Management science* 55(10):1605–1622.

Gu G, Zhu F (2021) Trust and disintermediation: Evidence from an online freelance marketplace. *Management Science* 67(2):794–807.

Hauser JR, Liberali G, Urban GL (2014) Website morphing 2.0: Switching costs, partial exposure, random exit, and when to morph. *Management science* 60(6):1594–1616.

Hauser JR, Urban GL, Liberali G, Braun M (2009) Website morphing. *Marketing Science* 28(2):202–223.

Hausknecht MJ, Stone P (2015) Deep recurrent q-learning for partially observable mdps. *AAAI 2015*, 29–37 (AAAI Press).

Huang N, Sun T, Chen P, Golden JM (2019) Word-of-mouth system implementation and customer conversion: A randomized field experiment. *Information Systems Research* 30(3):805–818.

Katehakis MN, Veinott Jr AF (1987) The multi-armed bandit problem: decomposition and computation. *Mathematics of Operations Research* 12(2):262–268.

Kokkodis M, Ipeirotis PG (2021) Demand-aware career path recommendations: A reinforcement learning approach. *Management Science* 67(7):4362–4383.

Kushner HJ (1964) A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Fluids Engineering* .

Lee D, Hosanagar K (2021) How do product attributes and reviews moderate the impact of recommender systems through purchase stages? *Management Science* 67(1):524–546.

Li H, Kannan P (2014) Attributing conversions in a multichannel online marketing environment: An empirical model and a field experiment. *Journal of Marketing Research* 51(1):40–56.

Liebman E, Saar-Tsechansky M, Stone P (2019) The right music at the right time: Adaptive personalized playlists based on sequence modeling. *MIS Quarterly* 43(3).

Mandel T, Liu YE, Brunskill E, Popović Z (2016) Offline evaluation of online reinforcement learning algorithms. *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, et al. (2015) Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533.

Moe WW, Fader PS (2004) Dynamic conversion behavior at e-commerce sites. *Management Science* 50(3):326–335.

Peters M, Ketter W, Saar-Tsechansky M, Collins J (2013) A reinforcement learning approach to autonomous decision-making in smart electricity markets. *Machine learning* 92(1):5–39.

Ribeiro AH, Tiels K, Aguirre LA, Schön T (2020) Beyond exploding and vanishing gradients: analysing rnn training using attractors and smoothness. 2370–2380 (PMLR 2020).

Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, et al. (2016) Mastering the game of go with deep neural networks and tree search. *Nature* 529(7587):484–489.

Song Y, Sahoo N, Srinivasan S, Dellarocas C (2021) Uncovering characteristic response paths of a population. *Informs Journal on Computing* (Forthcoming).

Sutton RS, Barto AG (2018) *Reinforcement learning: An introduction* (MIT press).

Thompson WR (1933) On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 25(3/4):285–294.

Van Hasselt H, Guez A, Silver D (2016) Deep reinforcement learning with double q-learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Wang W, Li B, Luo X (2019) Deep reinforcement learning for sequential targeting. *Available at SSRN 3487145* .

Wang Z, Schaul T, Hessel M, Hasselt H, Lanctot M, Freitas N (2016) Dueling network architectures for deep reinforcement learning. *International conference on machine learning*.

Watkins CJ, Dayan P (1992) Q-learning. *Machine learning* 8(3-4):279–292.

Zhang DJ, Dai H, Dong L, Qi F, Zhang N, Liu X, Liu Z, Yang J (2020) The long-term and spillover effects of price promotions on retailing platforms: Evidence from a large randomized experiment on alibaba. *Management Science* 66(6):2589–2609.

Zhang Y, Li B, Luo X, Wang X (2019) Personalized mobile targeting with user engagement stages: Combining a structural hidden markov model and field experiment. *Information Systems Research* 30(3):787–804.

Zhou F, Yang Q, Zhang K, Trajcevski G, Zhong T, Khokhar A (2020) Reinforced spatiotemporal attentive graph neural networks for traffic forecasting. *IEEE Internet of Things Journal* 7(7):6414–6428.

Appendix A: Rejection Sampling

In this section, we show the detailed implementation of Fixed-M Per-Episode Rejection Sampler (PERS) evaluator (Mandel et al. 2016) that evaluates RL models. Fixed-M PERS aims to sampling from the dataset D that contains episodes, where each episode d represents an ordered trajectory of actions, rewards and states, $(s_0, a_0, r_0, s_1, a_1, r_1, ..., r_H)$ obtained by executing existing policy e for H steps in the environment. Fixed-M PERS performs rejection sampling at the episode level. It will first compare evaluated policy b (RL model need to evaluate) against the executing policy e to get the ratio of the probability of executing a action under a state between two policies, then accept or reject the episode according to whether a random variable sampled from the uniform distribution is lower than the computed ratio. In order to ensure that rejection sampling returns a sample from the candidate distribution that resemble the distribution as applying evaluated policy b online, it is critical to set ratio normalization constant M correctly. Define the probability of executing action a under state s via policy e as $\pi_e(a|s)$, the probability of executing action a under state s via policy b as $\pi_b(a|s)$. The ratio $\frac{\pi_b(a|s)}{\pi_e(a|s)}$ can grow extremely large, we need an M such that $\frac{\pi_b(a|s)}{M\pi_e(a|s)}$ is a probability between 0 and 1. Therefore, M should be assigned as a constant that represents the maximum possible ratio. The detailed implementation of Fixed-M PERS is shown in Algorithm 3, where the for-loop between line 5-10 is use to construct M, and the for-loop in line 11-24 for episode-level rejection sampling.

Appendix B: Acquisition Functions

Acquisition functions are crucial to select which intervention to evaluate given the state. In this sections, we will go through details of three options we explored in this paper.

B.1. Probability of Improvement

Give the current state s of the customer, this acquisition function chooses the next action as the one which has the highest probability of improvement over the current max reward $MR(s)$ achieved by the current state s. Mathematically, we write the selection of next action a using the surrogate function

$$\arg\max_a(P(Q(s,a) > MR(s) + \epsilon) \tag{10}$$

where ϵ is a small positive number that controls the balance between exploration and exploitation. Increasing ϵ results in querying actions with a larger variance as their reward distribution density is spread, which will encourage the agent to explore more. While decreasing ϵ will restrict the agent to exploits actions with a high promise. After trial several settings, we set $\epsilon = 20$. Due to the fact that $Q(s,a)$ was modeled as normal distribution with the mean $\mu_{Q(s,a)}$ and variance $\sigma_{Q(s,a)}$ defined in EQ 5 and 6, the above surrogate function can be written as

$$\arg\max_a(1 - \Phi(\frac{MR(s) + \epsilon - \mu_{Q(s,a)}}{\sqrt{\sigma_{Q(s,a)}}})) \tag{11}$$

where Φ is the CDF of standard normal distribution.

Algorithm 3 Fixed-M Per-Episode Rejection Sampling Evaluator

Input: Executing policy e, evaluated policy b, state space S, binary transition matrix T denoting whether a nonzero transition probability from one state to another, maximum horizon H, and Dataset of episodes D with each episodes length of H.
Output: AER, where $AER(i)$ is the ith Accepted Episode cumulated Reward
Initialize $M_s = 1.0$ for all $s \in S$
for h=1 to H **do**
  for $s \in \mathcal{S}$ **do**
    Update $M_s^{'} = max_a(\frac{\pi_b(a|s)}{\pi_e(a|s)} T(s,a,s')M_{s'})$
  end for
  $M = M^{'}$
end for
$i = 1$
for $d \in \mathcal{D}$ **do**
  $p = 1.0, R = 0, s = []$
  Get start state st of the episode d
  for $(o,a,r) \in d$ **do**
    $s = (s,o)$
    $p = p\frac{\pi_b(a|s)}{\pi_e(a|s)}$
    $R = r + \gamma R$
  end for
  Sample $\mu \sim Uniform(0,1)$
  if $\mu \leq \frac{p}{M_{st}}$ **then**
    $AER(i) = R$
    $i = i + 1$
  end if
end for
return AER

B.2. Expected Improvement

Different from probability of improvement that focus on how likely an action leads to an improvement, Expected Improvement consider how much the reward can improve directly. Intuitively, Expected Improvement aim to choose the next action as the one which has the highest expected improvement over the current max reward $MR(s)$. Mathematically, expected improvement function is defined as:

$$EI(s,a) = \mathbb{E}[max(0, Q(s,a) - MR(s))] \tag{12}$$

Maximizing this function with $\arg\max_a$ will lead us to the action that, in expectation, improves upon current maximal reward most. Such surrogate function can re-write as analytical expression as:

$$EI(s,a) = \begin{cases} (\mu_{Q(s,a)} - MR(s))\Phi(Z) + \sqrt{\sigma_{Q(s,a)}}\phi(Z) & \text{if } \sigma_{Q(s,a)} > 0 \\ 0 & \text{if } \sigma_{Q(s,a)} = 0 \end{cases} \tag{13}$$

where $Z = \frac{\mu_{Q(s,a)} - MR(s)}{\sqrt{\sigma_{Q(s,a)}}}$, Φ and ϕ is CDF and PDF of the standard normal distribution. Such closed form expression gives us insights into what sort of actions will result in a higher expected improvement. EI is high when the posterior mean $\mu_Q(s,a)$ is higher than the current best $MR(s)$, or the variance $\sigma_Q(s,a)$ associate with the action is high. In another word, the mechanism will either sample from actions with expected high value of reward, or actions we are uncertain at, when we aim to maximize the expected improvement.

B.3. Thompson Sampling

The last option we explored for the acquisition function is Thompson Sampling. At every step, for the customer with state s, we first sample a reward from the posterior distribution of Q function for each action. Based on the sampled reward, we choose the action leads with the highest reward and adopt this action to evaluate. The intuition behind Thompson Sampling can be explained by two observations. (1) actions with high uncertainty will show a large variance in the functional values sampled from the posterior distribution. Thus, there is a non-trivial probability that a sample can take high value in a highly uncertain region. Optimizing such samples can aid exploration. (2) The sampled functions must pass through the actions with the highest expected reward with smaller uncertainty at the evaluated state. Thus, optimizing samples from the surrogate posterior will ensure exploitation as well.