https://pubsonline.informs.org/journal/mnsc

MANAGEMENT SCIENCE
Vol. 70, No. 8, August 2024, pp. 5115–5130
ISSN 0025-1909 (print), ISSN 1526-5501 (online)

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

Yicheng Song,[a] **Tianshu Sun**[b,c,*]

[a] Carlson School of Management, University of Minnesota, Minneapolis, Minnesota 55455; [b] Center for Digital Transformation, Cheung Kong Graduate School of Business, Beijing 100006, China; [c] Marshall School of Business, University of Southern California, Los Angeles, California 90089

*Corresponding author

Contact: ycsong@umn.edu, https://orcid.org/0000-0002-9107-814X (YS); tianshusun@ckgsb.edu.cn, https://orcid.org/0000-0002-9786-044X (TS)

Received: December 6, 2021
Revised: July 6, 2022; November 22, 2022
Accepted: January 19, 2023
Published Online in Articles in Advance: October 4, 2023

https://doi.org/10.1287/mnsc.2023.4914

Abstract. Firms adopt randomized experiments to evaluate various interventions (e.g., website design, creative content, and pricing). However, most randomized experiments are designed to identify the impact of one specific intervention. 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 randomized experiments might be globally suboptimal when considering their interdependence as well as the long-term rewards. Fortunately, the accumulation of a large number of historical experiments creates exogenous interventions at different stages along the customer journey and provides a new opportunity. This study integrates multiple experiments within the reinforcement learning (RL) framework to tackle the questions that cannot be answered by stand-alone randomized experiments. How can we learn optimal policy with a sequence of interventions along the customer journey based on an ensemble of historical experiments? Additionally, how can we learn from multiple historical experiments to guide future intervention trials? We propose a Bayesian recurrent *Q*-network model that leverages the exogenous interventions from multiple experiments to learn their effectiveness at different stages of the customer journey and optimize them for long-term rewards. Beyond optimization within the existing interventions, the Bayesian model also estimates the distribution of rewards, which can guide subject allocation in the design of future experiments to optimally balance exploration and exploitation. In summary, the proposed model creates a two-way complementarity between RL and randomized experiments, and thus, it provides a holistic approach to learning and optimizing interventions along the customer journey.

History: Accepted by Anindya Ghose, information systems.

Funding: This work was supported by Adobe Faculty Research Award and the Marketing Science Institute Research Grant.

Supplemental Material: The data files and online appendix are available at https://doi.org/10.1287/mnsc.2023.4914.

Keywords: reinforcement learning • customer journey • long-term reward optimization • Bayesian recurrent *Q*-network model (BRQN) • randomized experiment • experiment design

1. Introduction

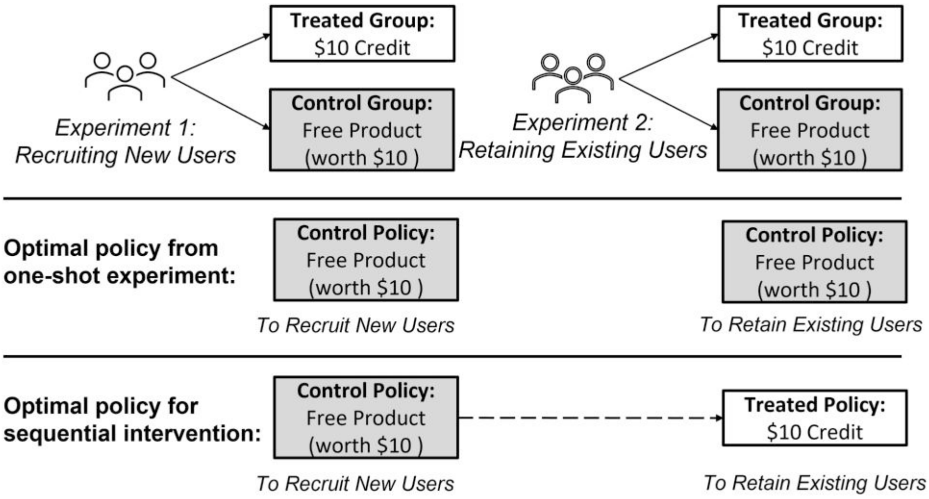
With more granular data on customers' digital traits across public domains (e.g., social media, content websites, search engines) and the firms' websites, firms can analyze each touch point (e.g., any interaction between the firm and the customer) along the customer journey to learn to intervene with the *right customers* at the *right time* with the *right policy*. With the help of digital tools, firms often have a large amount of observational data on customers' online journey as well as their responses to various interventions. Such fine-grained observational data facilitate the modeling of the 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 face 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 touch points on the customer journey, and they explore only a small subset of the action space. Thus, firms can only infer and optimize the interventions within a subset of state-action space that are endogenously generated in naturally occurring data. Using targeting models built on such data, firms may obtain suboptimal intervention policies as they exploit only the endogenously limited action-state space.

On the other hand, the randomized experiment (A/B test) has been increasingly adopted by firms to evaluate various interventions, ranging from designing new features on the website (Huang et al. 2019) to price promotion (Zhang et al. 2020) and the impact of recommender systems (Lee and Hosanagar 2021). Therefore, firms could learn and adopt the optimal intervention from the treated/control policies via the experiment. However, the stand-alone experiment is designed to identify the impact of one specific intervention. Current randomized experiments analyze each intervention separately without considering other interventions and therefore, cannot be used to optimize interventions across the entire customer journey. This is especially true when different interventions have strong dependencies (e.g., sequential promotions). Therefore, locally optimal interventions unveiled by the one-shot experiments could become globally suboptimal when considering the sequence of the intervention along the customer journey (Gallo 2017). In the conceptual example shown in Figure 1, the firm gets a budget to recruit new users, and they run a randomized experiment to evaluate two policies: a $10 credit as the treated policy and offering a free product worth $10 as the control policy. As the after-tax price for the product is higher than $10, the control policy is more effective to recruit new users. After that, the firm runs another randomized experiment to retain existing users by offering the two incentives again. They also find out that the control policy is more effective. However, if the optimal policies from each experiment are directly applied without considering interdependencies, those users who join the platform with the free product will be offered the same product again, which is less attractive from the user perspective because of decreasing marginal utility (Mankiw 2020). Thus, the optimal sequential intervention should be offering a free product to recruit new users, followed by a $10 credit to keep those users engaged. Such an example shows the necessity of learning sequential interventions beyond stand-alone randomized experiments.

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 experiments with a large number of sequential interventions. Moreover, preparing such an 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 stand-alone experiments are rarely utilized to guide future intervention trials to balance the exploitation of learned optimal interventions and the exploration of new/underexplored interventions. Thus, a disciplined, model-based approach is required to complement randomized experiments and address the limitations.

Fortunately, the accumulation of a large number of historical experiments creates and tests various exogenous interventions at different stages of the customer journey. Utilizing such data, this paper aims to answer

Figure 1. Conceptual Example of Learning Sequential Interventions from Randomized Experiments



Notes. Randomized experiment 1 is designed to evaluate the effectiveness of two policies for recruiting new users. Randomized experiment 2 evaluates the same policies for retaining existing users. Because of the after-tax price of the product being higher than $10, the control policy, when evaluated independently, is more effective in both randomized experiments. However, if the optimal policies from each experiment are directly applied without considering interdependencies, those users who join the platform with the free product will be offered the same product again, which is less attractive from the user perspective because of decreasing marginal utility. Thus, the better sequential intervention should be offering a free product to recruit new users, followed by $10 credit to keep those users engaged. Such an example shows the possibility that optimal intervention from one-shot randomized experiments might become globally suboptimal and the necessity to learn personalized sequential intervention policy from randomized experiments to optimize long-term reward.

two complementary research questions. How can we learn the optimal policy with a sequence of interventions along the customer journey by utilizing exogenous interventions from multiple historical experiments? Additionally, how can we learn from multiple historical experiments to guide future intervention trials to balance exploration and exploitation? In this paper, we propose a reinforcement learning (RL) model of ensemble learning from historical experiments (i.e., integrating multiple historical experiments (A/B test) using the RL model) to answer the two questions. We comprehensively compare RL + A/B with models built on observational data and one-shot experiments in Table 1.

The proposed RL + A/B model offers four major advantages. (1) The randomized experiments create exogenous variation in the intervention at different stages of the customer journey, and therefore, they increase the diversity of interventions (i.e., expanding exploration of state-action space). Such exogenous exploration in the state-action space is missing in endogenously formed observational data and presents the value of leveraging randomized experiments. (2) In one experiment, there is only one pair of treated/control actions. However, with n historical experiments, we might not be limited to n pairs of treated/control actions; the theoretical maximal combinations of different actions are 2^n.[1] With more experiments, our RL + A/B model could learn the interdependency between sequential actions, which will go beyond the focus of identifying the causal effect of a single intervention, and take a holistic perspective to learn optimal sequential interventions. Compared with the high time and monetary cost of deploying randomized experiments, the proposed model could turn “unmined” historical experimental data into treasured intervention policies with real-world business value in a shorter time. (3) RL models are concerned with how intelligent agents should take action in an environment to optimize the long-term reward. Therefore, RL models have been adopted in many marketplaces, such as the smart electronic market (Peters et al. 2013), music playlist recommendations (Liebman et al. 2019), career path planning (Kokkodis and Ipeirotis 2021), and sequential marketing promotions (Wang et al. 2022), to learn optimal sequential policies. In this study, by utilizing historical experiments, we can construct diverse and exogenous state-action pairs, which are ideal data to help RL models learn sequential interventions to optimize long-term reward. (4) As firms will continue exploring new interventions, the experimental method only requires 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 learned interventions to gain revenue group and the exploring new interventions with uncertain outcomes group. This offers the opportunity to apply the Bayesian optimization (Frazier 2018) on top of the RL + A/B model to balance exploring new interventions, which might result in higher rewards against focusing on a learned policy with a stable outcome.

In Section 3, we introduce an RL model—the Bayesian recurrent Q network (BRQN)—to learn sequences of interventions from randomized experiments. Built on multiple historical experimental data, BRQN first constructs diverse and exogenous state-action pairs 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 United States, we built and estimated the model using a detailed data set 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). Using state-of-the-art RL algorithms as benchmarks, we compare their performances in long-term reward optimization and find that the proposed model outperforms these benchmarks. The results show that adopting BRQN to exploit the sequential intervention policy leads to 7.6%–43% improvement in terms of reward (i.e., profit) for the firm (Section 4.4). To show the potential of ensemble learning from 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 performance improvement

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

	One-shot experiment	Models with observational data	Proposed: RL + A/B
Exogenous intervention	Yes	No	Yes
Diversity of action-state space	High	Low	High
Time cost of learn optimal policy	High	Low	Low
Monetary cost of learn optimal policy	High	Low	Low[a]
Holistic optimization of customer journey	No	Possible	Yes
Long-term reward optimization	No	Possible	Yes
Guide future intervention trials	No	Possible[b]	Yes

[a] RL + A/B learns from existing historical experiments and turns archived historical experiments into policies with real-world business value. However, if there is no historical experiment, the monetary and time costs could be high.

[b] The structural econometric model could also evaluate new interventions via policy simulation to guide future trials.

as compared with only exploring a smaller action-state space with fewer experiments. However, not all additional experiments are helpful. We also evaluate various combinations of historical experiments and unveil what types of historical experiments will boost the model performance (Section 4.5). Beyond optimizing interventions within the existing experiments, the model's results could also guide future intervention trials to balance the exploitation of learned policy for stable revenue and the exploration of new/underexplored interventions for future potential (Section 4.7). The experiments show that the model could effectively further improve the learned policies to increase long-term rewards and efficiently allocate participants to avoid expensive and unnecessary trials. Therefore, the proposed RL + A/B model creates a two-way complementarity between reinforcement learning and experiment. That is, ensemble learning from 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 create a holistic approach to intervention learning and optimization along the customer journey.

2. Literature

Our research closely relates to two streams of literature. One is the studies of interventions along the customer journey, which contextually relate to this paper. The second is reinforcement learning research in the management science literature that is methodologically related. This research extends the two streams of literature and offers a new solution for sequential intervention optimization along the customer journey by building an RL model to learn from multiple experiments.

2.1. Interventions Along the Customer Journey: Observational Data vs. Randomized Experiments

Fine-grained observational data, like clickstream data, contain rich information about user behaviors (Ghose et al. 2019). Moe and Fader (2004) utilized detailed clickstream data to formulate a dynamic model of 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. (2022) 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' states. On the other hand, a large body of literature leverages randomized 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 from multiple experiments to build a reinforcement learning model that aims to optimize the long-term reward. Our study contributes to the literature on customer journey analysis by taking a holistic approach that fully leverages historical experiments.

2.2. 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 multiarmed bandit (MAB), Katehakis and Veinott (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 balances exploration-exploitation when evaluating the action pool. One major limitation of the MAB model is that it is concerned about immediate feedback/reward but does not explicitly model future rewards. Therefore, another research stream aims to build reinforcement learning models to explicitly optimize the long-term reward. Built on the Q-learning framework (Watkins and Dayan 1992), Kokkodis and Ipeirotis (2021) adopted the Markov decision process to operate on a knowledge graph of skill sets and dynamically recommend profitable career paths for users to optimize their long-term rewards. Also built on the Q-learning framework, Wang et al. (2022) proposed a deep reinforcement learning model for sequential targeting problems. 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 a model considers the dynamic sequential behavior of consumers to optimize long-term revenues.

Three major differences between the proposed model and previous RL studies in the management science literature are as follows. (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 more exogenous and diverse state-action space than research that relies either on endogenous observational data or on one-shot experiments. (2) We develop a BRQN 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 learned policy to better target customers, but it also enables a systematic solution to guide future intervention trials. (3) Most previous works train and evaluate reinforcement learning models using a simulator (a predictive model that simulates 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 that the sampled data draw from the same distribution as applying the RL model online, and the estimation of the long-term reward is unbiased.

3. Model

A standard reinforcement learning model has five components (Sutton and Barto 2018): agent, environment, states, actions, and rewards, all of which we will concretize in an e-commerce setting; the intelligent agent (i.e., reinforcement learning model) interacts with the environment (i.e., heterogeneous customers), and when receiving the state of the customer (i.e., a summary/abstraction of observed customer behavior), the intelligent agent will execute the learned optimal action policy (the action space consists of the interventions from all experiments explored earlier). The agent will receive immediate rewards (i.e., the immediate reward is set based on the immediate feedback from the customer; positive feedback, like product order, leads to a positive monetary reward, whereas other feedback leads to zero rewards) from the customer. The reinforcement learning model aims to learn an optimal policy to maximize the cumulative reward from the customers. However, when applying classical reinforcement learning models to this setting, we face two main challenges.

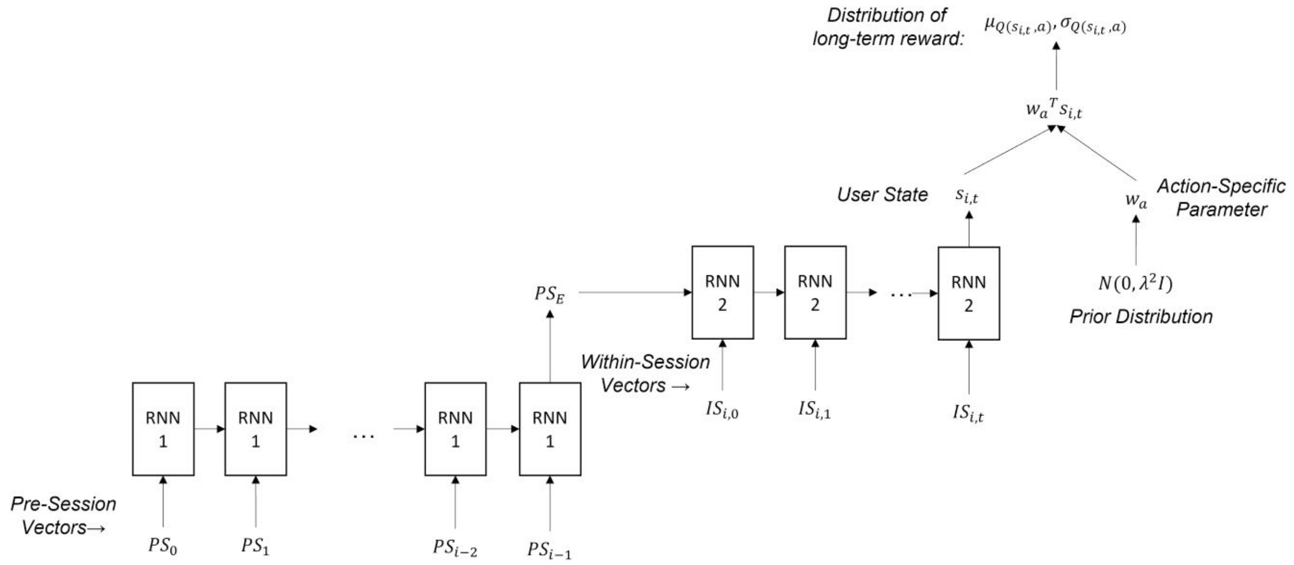
- Many RL algorithms rely on the assumption of Markov decision processes where the future reward and the state are based solely on the most recent observation/state. Such a Markov property rarely holds in real-world environments, especially as customer state cannot be fully described only by the most recent observations. A partially observable Markov decision process (POMDP) (Cassandra 1998) better captures the dynamics of many 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 the estimation of a singular number, managers also care about the uncertainty of long-term rewards if they adopt the model. The expected rewards from executing certain actions could be relatively high, but they are not the best choice because of 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, managers may prefer actions that lead to high expected rewards but low variance (Avalos et al. 2022), but they will need to bear the risk of not exploring these 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 an incomplete state (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 deep recurrent Q network (DRQN), a combination of RNN and deep Q network to learn the state of the user from recent observations as well as 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 linear regression layer (Azizzadenesheli et al. 2018) on top of DRQN and name the proposed model BRQN. The Bayesian regression layer will update the posterior distribution of parameters by synthesizing the prior distributions and observed data, and this will allow us to measure the uncertainty of long-term rewards for each state-action pair. With these adjustments, BRQN can address the two challenges discussed. The structure of the proposed BRQN is illustrated in Figure 2. Next, we introduce state-representation learning via RNN to accommodate the POMDP in Section 3.1, followed by introducing Bayesian linear 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 in Section 3.3.

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 learn the customer's state. As we have detailed page visit clickstream data,[2] we first segment such clickstream data into sessions, where any consecutive page visits that are more than 12 hours apart will segment into two sessions. Thus, two types of time series are necessary when faced with the user's tth page visit in the ith session. (1) The presession vector, PS_j, summarizes the activities in session j, where $j < i$; this vector describes session-level summarization with coarse granularity. (2) The within-session vector, $IS_{it'}$, summarizes the activity on the t'th page visit within the ith session, describing the finer-granularity activities within the focal session.

Figure 2. Outline of BRQN



Notes. Two-layer RNN structure is designed to deal with two types of data when facing with user's tth page visit within the ith session. The first RNN deals with presession vectors to summarize the user interaction history before session i. The outcome of the first RNN, PS_E, is the embedding for presession interactions, which 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 the user state until the tth page in the ith session. Meanwhile, w_a represents the Bayesian linear regression coefficients for action a (different w_a's share the same prior distribution $N(0,\lambda^2 I)$). With s_{it} and action-specific w_a, we can estimate the distribution of the long-term reward by taking action a under state s_{it}.

Correspondingly, as shown in Figure 2, two RNNs process two types of sequential data. The first RNN processes the presession vectors to summarize the user state before session i. The outcome of this first RNN PS_E, which is the embedding for presession activities, is used as the initial embedding for the second RNN. The second RNN then processes the within-session sequential vectors and generates state s_{it}, which is the summary of the user state as she or he reaches the tth page in the ith session. We adopt such two-layer RNN architecture because most experimental interventions (actions in the RL framework) are associated with a particular web page; therefore, we need a fine granularity state representation at the page visit level, which is captured via the second RNN. However, if we also adopt such a fine granulated format for all interactions in previous sessions, we have too many details for presession vectors, rendering a long time series and raising vanishing gradient concern (Ribeiro et al. 2020) for RNN estimation.[3] 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 Linear Regression

With user state representation $s_{i,t}$, next we 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 simply be estimated by $w_a^T s_{i,t}$ (Mnih et al. 2015). We can obtain a Q value for each action, and the firm can adopt the action that leads to the highest Q-value estimation for exploitation purposes. However, such a singular number estimation suffers from the lack of uncertainty limitation mentioned before. Thus, we adopt Bayesian linear regression (Hartigan 1969) to estimate the distribution of Q value. Specifically, we model the prior distribution of w_a following a normal distribution with a prior mean zero along with prior variance $\lambda^2 I$:

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

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

$$P(Q \mid w_a^T, s_{i,t}) \sim N(w_a^T s_{i,t}, \sigma^2). \tag{2}$$

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 the corresponding long-term reward vector $Q_a \in \mathbb{R}^d$ (derived from Equation (7)). Then, we can update the posterior of $w_a \sim N(\mu,\Sigma)$ by consolidating the prior distribution and observed data as

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

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

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 is equipped with mean μ_Q and variance σ_Q for each state-action pair, which not only enable the firm to learn a high-reward policy but could also guide future intervention trials to balance exploitation and exploration (Section 3.4).

3.3. Bayesian Recurrent *Q* Network

With the modules (1) state representation learning via RNNs and (2) Q-value estimation via Bayesian linear regression, we integrate them into a unified model, BRQN, and estimate all parameters via end-to-end learning. We start with the derivation of Q value, $Q(s_t, a)$, which defines the expected cumulative reward obtained by executing action a under the customer state s_t. In such a context, the customer with state s_t is exposed to action a at the tth page visit, the immediate reaction (e.g., product order) of this customer will set the numerical immediate reward r_t, and this will lead to customer state s_{t+1} in the next period $t+1$. Thus, we write Q as

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

where $\gamma \in [0, 1]$ is the predefined discounted factor that balances between the current and future rewards, as $\gamma = 1$ whereby all future rewards are fully considered during interaction t and $\gamma = 0$ whereby only the immediate reward is counted. It is well established that parameters in the Q-learning model can be estimated via regression on temporal differences (TDs) (Sutton and Barto 2018). Thus, Q learning can update its parameter by minimizing the temporal difference between the left and right sides of Equation (7). We also adopt improvements, like double-Q learning (Van Hasselt et al. 2016) and prioritized replay (Schaul et al. 2015),[4] to improving the learning stability and generalize learning across actions without imposing any change to the underlying reinforcement learning algorithm. Thus, there is a Q network of the focal training model and another Q_{target} network as the target model. The target model is the copy of the focal model at the beginning of the training process and gets updated every M step. Specifically, given w_a, the parameter θ estimation process is as follows.

1. Find the action a' that leads the maximal Q under state s_{t+1}: $\max_{a'} Q(s_{t+1}, a' \mid \theta)$.
2. Get the Q value of taking action a' at state s_{t+1} via the target Q network: $Q_{target}(s_{t+1}, a' \mid \theta_{target})$.
3. The TD is defined as

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

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

$$\theta = \theta - \tau \frac{\mathrm{d}TD}{\mathrm{d}\theta}. \tag{9}$$

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

Algorithm 1 (Bayesian Recurrent Q-Network Estimation)

```
1: Set the empty reply buffer $RB()$, and load the historical experiment data into the buffer.
2: Initialize RNN parameter $\theta$ and Bayesian parameter $w_a$ for each action.
3: Set $N$ as the interval for Bayesian sampling and $M$ as the interval for target model update.
4: Set $\theta_{target} = \theta$ and $w_{a,target} = w_a$.
5: while $step <= num_{steps}$ do
6:     Load a minibatch from $RB()$ with probability relative to TD error in Equation (8)
7:     Get $s_t$ based on $RNN$
8:     Get $s_{t+1}$ based on $RNN_{target}$
9:     Get TD between state $s_t$ and $s_{t+1}$ based on Equation (8)
10:    Update the RNN parameter $\theta$ based on Equation (9)
11:    Bayesian posterior update based on Equations (3) and (4)
12:    For every $N$ steps: Thompson sampling $w_a \backsim N(\mu, \Sigma)$
13:    For every $M$ steps: $\theta_{target} = \theta$ and $w_{a,target} = w_a$
14:    $step = step + 1$
15: end while
```

3.4. Guide Future Intervention Trials

Beyond learning optimal intervention policy using historical experiments, the result of the proposed model can be utilized to guide future intervention trials to further improve the learned policy, especially for (1) allocating additional samples to trial existing interventions and (2) allocating samples to evaluate new interventions. In the first scenario, the firm has extra resources to recruit additional subjects into existing interventions to refine the learned policy[5] but must decide how to allocate the limited subjects. In the second scenario, the firm aims to evaluate a new intervention but also needs to determine how to allocate subjects across different interventions.

Equations (5) and (6) define the distribution of the Q-value estimation for any state-action pair, which can be used to guide future intervention trials. It is straightforward to deal with those state-action pairs that lead to a Q value of low variance; we can exploit the state-action that leads to a Q value with a high mean, whereas the state-action with a low mean Q values will be weeded from the consideration set because of unpromising outcomes. However, for state-action pairs with high variance, we need a systematic solution to balance the promise of exploring those actions and the costs of running such expensive and uncertain trials. We adopt the Bayesian optimization to balance the need to explore uncertain actions (exploration), which might unexpectedly bring high rewards, against focusing on learned actions with known stable rewards (exploitation).

For every incoming state, we determine which action to evaluate based on the distribution of the Q values and acquisition functions (AFs).[6] Acquisition functions are heuristics for how desirable it is to evaluate an action given a state. We explore three options of acquisition functions and detail the implementation of each acquisition function in Online Appendix D.

1. Expected improvement (Močkus 1975). The 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). The acquisition function will choose the next action that has the highest probability of improvement over the current maximum reward.

3. Thompson sampling (Thompson 1933). We sample a distribution of expected cumulative reward from the posterior distribution across all the actions and choose the action that leads to the highest value in the sampled distribution.

After adopting the actions recommended by the acquisition functions, we evaluate how these acquisition functions balance exploration and exploitation in Section 4.7.

4. Empirical Analysis

4.1. Context and Data

We partner with an e-commerce platform in the United States to evaluate the performance of the proposed model. The platform is a pioneer in the industry that uses randomized field 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 the 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 test bed for this study.

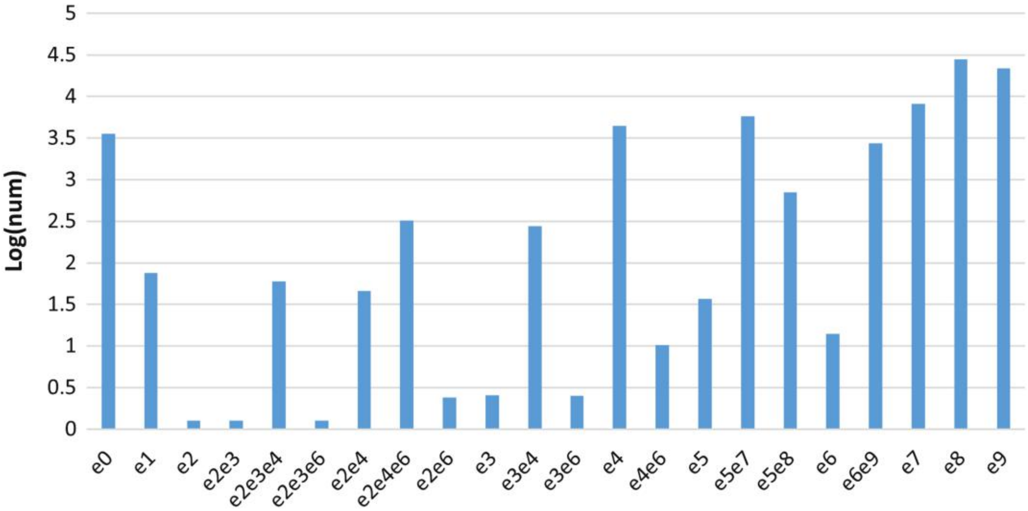
We aimed to find a period during which multiple experiments were conducted (there is no requirement for an exact alignment of start and end dates) with overlaps of treated users from different experiments to ensure a large number of distinct action combinations. Based on this criterion, we chose 10 experiments that took place from July 1 to October 31, 2017. Table 2 shows the detailed settings for each experiment, and Online Appendix E provides additional summaries. All experiments are designed and executed independently from each other. Our partner implements all experiments on a commercial randomized experiment platform to ensure that treated and control groups have similar distributions. Our data set includes 75,913 unique treated users,[7] and some are being treated in multiple experiments. We also randomly chose 74,000 users not being treated in any experiment in that period. In total, the data set includes 149,913 users across control and treated groups from 10 selected experiments. For all the selected users, we collected their clickstream data from January 1 (six months before the first experiment to construct presession vectors) to October 31 in 2017. Figure 3 shows the number of users under different action combinations in $\log_{10}()$ scale.[8]

Next, we show how to materialize state s_t and reward r_t in our empirical setting. We need to construct the time series feature as presession vectors and within-session vectors to feed two RNNs. To construct presession vectors, we temporally segment the sequence of page visits, as any two consecutive page visits that are more than 12 hours apart will be treated as two sessions. Then, we summarize the users' time spent on different web pages and the number of purchases of different types of products in that session and along with the time gap from the previous session as the presession vector, PS. To

Table 2. Historical Experiments Details

Index	Date	Descriptions	Category	No. of treated users	Page
0	7/11–8/11	Test new menu	Menu design	3,500	Project builder
1	7/18–7/26	Test new layout 1	Page design	80	Product page
2	7/27–8/02	Test new layout 2	Page design	350	Product page
3	8/11–8/17	Show rating with price	Page design	450	Home page
4	8/13–8/25	Home page redesign	Page design	5,000	Home page
5	8/16–8/25	Show an overlay to encourage the user to start the project	Page design	6,000	Project builder
6	8/16–8/25	Site coupon experiment	Discount	3,000	
7	8/25–9/20	Force template selection for product 1	Product design	12,000	Project builder
8	8/28–9/20	Add new template selection for product 2	Product design	30,000	Project builder
9	8/31–10/31	Shows carousel with new product	Page design	24,000	Product page

Figure 3. (Color online) Number of Users ($\log_{10}()$ Scale) with Different Treatment



Notes. e2 means that the users are only treated under experiment 2. e2e3 means that the users are treated under both experiments 2 and 3.

construct within-session vectors, we temporally segment a session into within-session segments based on users' web page visit(s) to any specific web page where the experiments take place. For instance, experiment 0 is conducted on the project builder page; then, a customer's visit "home page → category page → product page → project builder page → order page → home page" can be divided into two within-session segments ("home page → category page → product page → project builder page" and "order page → home page"). For each within-session segment, we construct a within-session vector *IS* similar to that of *PS*. Correspondingly, we will have $s_t \rightarrow a_{e_0} \rightarrow s_{t+1}$ based on such within-session transitions. The reward r_t associated with the transition is defined as the total money (cents) associated with the order in the second within-session segment.[9]

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 learned model in the same setting. Several studies employed simulators (e.g., Atari video games) as the test bed (Mnih et al. 2015) or directly played 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 because of unintended (negative) outcomes. On the other hand, high-fidelity simulators where these applications operate are challenging to build as well. Fortunately, precollected 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 data set 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 and (2) empirically evaluate RL models based on their ability to utilize a fixed data set 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 BRQN is built is a well-known off-policy RL algorithm (Sutton and Barto 2018); we train the model using 80% of our data set. For evaluation, we adopt fixed-*M* per-episode rejection sampling (fixed-*M* PERS) (Mandel et al. 2016) to evaluate the model performance using the holdout data set (20% of the data). Fixed-*M* PERS samples from the achieved data and more favorably selects those episodes (ordered trajectory of action, reward, and state) that follow the policy of the learned RL model. Fixed-*M* PERS is a well-established RL off-policy evaluation method with several desirable properties. (1) Episode samples accepted by fixed-*M* PERS have the same distribution as episode samples collected via the on-policy evaluation of the model. This property is known as the true sample property. (2) Based on episode samples accepted by fixed-*M* PERS while evaluating the RL model off policy, researchers can derive an unbiased estimate of the reward obtained by the RL algorithm as it runs 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 $H = 1, 4, 8$. When $H = 1$, we sample only the next observation as an episode. When $H = 4$ or 8, we aim to sample the next consecutive four or eight 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.

4.3. Model Validation

We first validate the model settings by evaluating alternative model choices. We adopt an ablation study to evaluate different alternatives and compare their performances. There are two main modules in the proposed model. The first is state representation learning. We use RNN to process historical interactions, which better accommodate POMDP. The second is Q-value estimation. We adopt Bayesian linear regression to estimate the Q-value distribution for each state-action pair. The distribution will better capture the uncertainty of the Q value rather than a singular number estimation. In the following ablation study, we will test alternatives for each module.

For the state representation learning alternative, we adopt a deep feed-forward neural network (DNN), like DQN (Mnih et al. 2015), to process only the most recent observation (no historical data) to summarize the state of the user. For Q-value estimation alternatives, (1) linear regression can be adopted to estimate the Q value for each state-action pair. (2) Following classical Bayesian Q learning (Dearden et al. 1998), the distribution of the Q value of each state-action pair is modeled as an independent normal distribution and Bayesian update; such distribution is based on the conjugate prior distribution and new observations.[10] As we have two choices for state representation learning and three choices for Q-value estimation, we train and evaluate all six combinations. Table 3 shows the average reward per episode sampled by fixed-M PERS across different model settings.

For state representation learning, we observe the consistent superiority of adopting RNNs, which confirms the need to use RNNs to process the historical data to accommodate POMDP. Otherwise, the most recent observations unveil only part of the customer state and lead to inferior decision making. For the Q-value estimation, models equipped with Bayesian linear regression improve the average reward in the range of 1.5%–7.9% over models with linear regression. Rather than estimating the singular number of Q value for each action-state pair, the proposed model draws a distribution for Q values, which directly incorporates the uncertainty into consideration, resulting in efficient exploration and exploitation. Models with Bayesian update lead to inferior performance compared with the other options. The reason is that the dimension of state-action is tremendous and leads to the curse of dimensionality concern, making the Bayesian update process inefficient. On the contrary, we set the prior distribution for each action in Bayesian linear regression rather than model each state-action pair as an independent distribution, which helps us address the curse of dimensionality concern as the number of actions is limited compared with the tremendous number of state-action pairs.[11] By incorporating the two winning strategies (RNN + Bayesian linear regression), the proposed model shows its clear advantage over all alternatives.

4.4. Comparison with Benchmarks

We further compare the performance of BRQN with state-of-the-art RL algorithms. Specifically, we explore the following benchmarks.

1. Original website intervention. For all the records in the holdout data set, we collect episodes with different lengths and calculate the average reward per episode. This original intervention policy provides a good baseline for the comparison with 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 stand-alone experiments without considering their dependency, they become a useful benchmark with which to compare model-based approaches. We sample the episodes from holdout data that match the optimal interventions derived from stand-alone experiments and obtain the average episode reward.

Table 3. Ablation Study of Model Settings

Model settings					Average reward		
DNN	RNN	Linear regression	Bayesian update	Bayesian regression	Fixed-M PERS @ 1	Fixed-M PERS @ 4	Fixed-M PERS @ 8
✓		✓			510.6	2,165.6	4,679.6
✓			✓		500.1	2,012.6	4,035.8
✓				✓	518.3	2,337.9	4,931.7
	✓	✓			527.9	2,366.9	5,521.7
	✓		✓		499.6	2,002.9	3,997.2
	✓			✓	536.5	2,429.3	5,708.6

3. Predictive model (XGboost). Following Wang et al. (2022), we adopt XGBoost (Chen and Guestrin 2016) as a benchmark model for myopic personalized targeting to predict the immediate reward given a state and action pair. We observe from the data that after the customer with state s_t is exposed to action a, this leads to immediate reward r_t. XGBoost will be trained to take s_t and a as inputs to predict r_t. Correspondingly, for an incoming customer with state s_t, the model will choose action a that will lead to the highest immediate reward prediction.

4. MABs (Katehakis and Veinott 1987). MAB is a classical RL model that balances the exploration and exploitation to optimize the immediate reward. For each state, s_t, MAB aims to learn the reward distribution (i.e., Gaussian distribution) for action a. Correspondingly, for an incoming state s_t, Boltzmann exploration (Bertsekas and Tsitsiklis 1995) is used to guide action deployment.

5. 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. Thus, DQN is equivalent to the DNN + linear regression setting in Section 4.3. To ensure learning stability and generalizability, we also adopt double-Q learning and dueling architectures in DQN.

6. DRQN (Hausknecht and Stone 2015). Similar to BRQN, DRQN also adopts two RNNs to process presession 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 reward, whereas BRQN utilizes Bayesian linear regression. Correspondingly, DRQN is equivalent to the RNN + linear regression setting in Section 4.3. To ensure learning stability and generalizability, we also adopt double-Q learning and dueling architectures in DRQN.

Table 4 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 those of the original website intervention, which indicates that it is beneficial to adopt the learned optimal intervention derived from randomized experiments. Moreover, we notice that the Q-learning family (DQN, DRQN, BRQN) that comprehensively models the interdependency of interventions from all 10 experiments achieves higher average rewards than those of the optimal intervention from one-shot experiments. Such comparisons unveil that the locally optimal interventions from one-shot experiments become suboptimal along the customer journey. Thus, it is necessary to ensemble experiments using RL models to learn the sequential dependency between interventions.

Second, it is foreseeable to see that myopic personalized targeting models, like MAB and XGboost, could achieve higher average rewards than utilizing the optimal intervention derived from experiments, indicating hat it is better to learn a personalized targeting strategy rather than nonpersonalized approaches. Interestingly, myopic personalized targeting models perform better than all Q-learning models when setting $H = 1$, but Q-learning models surpass on longer episodes $H = 4, 8$. Such a comparison shows that myopic personalized targeting models are indeed superior at learning intervention strategies to optimize the immediate reward, but Q-learning models will learn strategic policies that might sacrifice immediate reward in favor of the long-term rewards.

Third, BRQN is the best performer among the Q-learning family. As discussed in Section 4.3, such improvement can be attributed to the following. (1) Adopting RNNs to incorporate the historical data could help the model to learn state representation to better accommodate the POMDP. (2) The BRQN model draws a distribution of Q value for each state-action pair by utilizing Bayesian linear regression, which allows the model to directly incorporate the uncertainty, resulting in efficient exploration/exploitation.

Finally, by comparing the differences in average rewards across different episode lengths, we find that the improvement of the average reward is inconsistent. When $H = 1$, the improvement of BRQN is 7.6% higher than the current policy. With $H = 4$, the average reward improvement of BRQN increases to 21%. The improvement 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 BRQN's average

Table 4. Average Reward per Episode (Cents) Across Different Models

	Fixed-*M* PERS @ 1	Fixed-*M* PERS @ 4	Fixed-*M* PERS @ 8
Original intervention	498.5	2,001.3	3,989.5
Optimal intervention	502.9	2,058.0	4,250.2
XGboost	539.1	2,281.7	4,387.9
MAB	538.5	2,356.3	4,558.3
DQN	510.6	2,165.6	4,697.6
DRQN	527.9	2,366.9	5,521.7
BRQN	536.5	2,429.3	5,708.6

reward improvement surges to 43% when setting $H = 8$. These comparisons support adopting RL models, which could help firms optimize their long-term rewards.

4.5. Model Performance with Different Experimental Data

To evaluate the value of diverse exogenous interventions derived from multiple experiments, we train the BRQN model using data from all 10 experiments and contrast this with using data from only 1 experiment (the one with a maximal number of treated users) and then, from 5 experiments (the 5 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 data set. As shown in Table 5, using data from additional experiments leads to a significantly higher average reward per episode. When we train the BRQN model with data from five experiments, we improve the average reward per episode in the range of 2%–12% compared with the model that has learned from only one experiment. This improvement soared to the range of 6%–36% 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 BRQN model to explore a larger action space to learn better interventions to target different users.

Is utilizing additional experiments always helpful? To answer this question, we systematically study the sensitivity of model performance on different combinations of experimental data. We test model performance variations by training the model on (1) experiments with a different number of treated users, (2) experiments with different spans, and (3) experiments with different overlapped treated users and dates. We find that it is helpful to train the proposed model on experimental data with more treated users, longer span, bigger overlap of treated users, and experiment dates.[12] The main reason these combinations are standing out is that they are inclined to explore a larger state-action space compared with their counterparts. Thus, they have the potential to help the model to learn a better intervention policy within a larger state-action space. However, once the model is trained, we should not manually set which additional intervention to explore. It is the acquisition functions that will systematically balance such exploration versus exploitation.

4.6. Discovering Optional Sequential Interventions: An Example

Each experiment is originally designed by certain managers or teams in the company without considering other experiments. If the proposed model learned a sequential combination of experiments that is effective, it not only presents a new discovery of useful combinations but also, provides an opportunity to understand the intuition behind why these combinations work better. The intuition and understanding of underlying customer behaviors may help managers and teams better design their interventions in the future and better combine existing interventions. To provide an example in this direction, we explore our specific setting and examine all the combinations suggested by the model as the optimal sequential interventions (i.e., all episodes accepted by fixed-M PERS). It turns out that experiment 6 followed by experiment 9 is the most frequently accepted sequential combination. Specifically, experiment 6 sends out a limited-time coupon to customers offering 50% discounts on their next order equal to or higher than $50. In experiment 9, the platform shows a carousel on the product page to promote a new product, which is characterized by popular figures from a movie on the show. At first glance, we were surprised to see how this combination works out. Then, we found out that the reason is that the promoted new product in experiment 9 perfectly matched the coupon's price range requirement of $50–$60 and the time constraint (one month). Thus, the customer being treated in experiment 6 may experience "anticipatory regret" (i.e., *regret from missing out on the deal*) (Keinan and Kivetz 2008), which is an urgency driver to move customers to determine which product to redeem using the coupon (Inman and McAlister 1994), whereas the followed-up experiment 9 promotes a new and attractive product that could perfectly activate the coupon, leading to much higher order placement than other combinations. Thus, it is the

Table 5. Average Reward per Episode (Cents) of BRQN with Different Experimental Data

	Fixed-M PERS @ 1	Fixed-M PERS @ 4	Fixed-M PERS @ 8
1 Experiment	502.4	2,027.9	4,190.2
5 Experiments	515.1	2,167.5	4,711.8
10 Experiments	536.5	2,429.3	5,708.6

Notes. We train the BRQN model with data from only 1 experiment (the one with the maximal number of treated users), 5 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 lengths of one, four, and eight. The average reward of the sampled episode across different settings shows promise of fueling RL with multiple experiments.

matched incentive and temporal adjacency between experiments 6 and 9 that make them an optimal sequential intervention. Even though those experiments were originally designed independently, we can learn meaningful patterns from them ex post and make sense of them with reasonable theoretical foundations. This example shows that the proposed model could pick up insights after experiments and complement the blind spots of human intuition in the design of intervention. With more policy variations in historical experimental data, we expect more meaningful combinations to be discovered. This has the potential to facilitate the data-driven discovery of theoretical hypotheses.

4.7. Future Intervention Trial Evaluation

We next evaluate the outcomes by adopting the intervention trial recommendations from different AFs. 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 the 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 trial recommendation. Specifically, we treat 20% of the data as incoming traffic after the model is trained. For incoming traffic with state s, the acquisition function will recommend an action a. If the recommended action matches the action in the data, we use this record to update the posterior distribution of w_a via Equations (3) and (4). We simulate 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 intervention trials to further improve the policies. The detailed simulation process is shown in Algorithm 2.

Algorithm 2 (Simulate Future Intervention Trials via Holdout Data)

```
Input: holdout data set $D$, $d$ represents a record in $D$ that contains the state and action
$iteration = 1$
while $iteration \le num_{iterations}$ do
    for $d \in D$ do
        Get $s$, state from $d$
        Get $a$, action from $d$
        Get the recommended action $ra(s)$ via the chosen acquisition functions for state $s$
        if $ra(s) == a$ then
            Update BRQN model with record $d$ via Equations (3) and (4)
            $num_{update} = num_{update} + 1$
        end if
    end for
    $iteration = iteration + 1$
end while
```

4.7.1. Evaluation of the 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 learned optimal policies. Then, we summarize the mean and standard deviation (SD) of the top three Q values for all the states. In Table 6, we report the mean and SD of the Q-value estimation from the BRQN before and after the model gets updated. 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.[13] The comparisons clearly show the superiority of adopting Bayesian optimization. (1) We find that the mean values for the top three average Q values increase significantly after the model update, which shows that the recommended policy could not only exploit the learned existing policy but also, utilize additional samples to explore the state-action space to optimize the intervention to further improve the policy. (2) The SD of

Table 6. Q-Value (Mean and Standard Deviation) Comparison Before and After the Model Update in Scenario 1

		Before update		After update		
AF	Rank	Mean	Standard deviation	Mean	Standard deviation	p-value, %
Expected improvement	Top 1	2,885.36	68.62	2,951.77 (2.30%↑)	63.13 (8.01%↓)	<0.001
Expected improvement	Top 2	2,876.54	70.76	2,928.31 (1.79%↑)	66.22 (6.41%↓)	<0.001
Expected improvement	Top 3	2,853.81	79.39	2,886.16 (1.13%↑)	74.36 (6.33%↓)	<0.001
Probability of improvement	Top 1	2,885.36	68.62	2,949.13 (2.21%↑)	64.08 (6.61%↓)	<0.001
Probability of improvement	Top 2	2,876.54	70.76	2,925.66 (1.70%↑)	65.39 (7.58%↓)	<0.001
Probability of improvement	Top 3	2,853.81	79.39	2,892.84 (1.36%↑)	75.27 (5.18%↓)	<0.001
Thompson sampling	Top 1	2,885.36	68.62	2,946.96 (2.13%↑)	64.73 (5.56%↓)	<0.001
Thompson sampling	Top 2	2,876.54	70.76	2,927.75 (1.78%↑)	67.91 (4.02%↓)	<0.001
Thompson sampling	Top 3	2,853.81	79.39	2,890.56 (1.28%↑)	76.28 (3.91%↓)	<0.001
Random select	Top 1	2,885.36	68.62	2,886.62 (0.25%↑)	67.58 (1.51%↓)	3.21
Random select	Top 2	2,876.54	70.76	2,877.59 (0.21%↑)	69.86 (1.27%↓)	6.76
Random select	Top 3	2,853.81	79.39	2,854.71 (0.10%↑)	77.65 (2.19%↓)	12.58

the top three Q values decreases after the model update, which indicates that adopting the mechanism could decrease the uncertainty of the learned optimal policies. (3) All three Bayesian optimization approaches achieve the two goals, and both goals are much higher than those of the random selection.

4.7.2. Evaluation of the 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 the new intervention.[14] Thus, we first use experiments 0–8 from 80% of the original training data to train the model, and then, we update the model using the original holdout 20% of the data that contains experiments 0–9 to simulate experiment 9 as the new intervention. We compare the Q-value estimation from the BRQN 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 insignificantly different from those in scenario 1, which indicates that the best policies in the two scenarios are almost identical. As shown in Table 7, the number of trials for experiment 9 (the number of impressions associated with the treated action in experiment 9, not the number of treated users) must exceed 100,000 to ensure that the best policies derived from the two scenarios are indifferent. However, 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 intervention trials to improve the policy with greater efficiency.[15] 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 test) have been widely adopted by firms to evaluate various interventions. However, locally optimal interventions unveiled by one-shot experiments might be globally suboptimal when considering their interdependency as well as long-term reward optimization. The literature on randomized experiments lacks a holistic approach to optimize sequential interventions along the customer journey. Fortunately, the accumulation of a massive number of historical A/B tests allows us evaluate various exogenous interventions at different stages of customers' journeys and provides a new opportunity. This study proposed a Bayesian recurrent Q-network model to utilize diverse and exogenous interventions from multiple experiments to learn the sequential intervention to optimize the long-term reward. The main findings of this study are as follows.

- The empirical evaluations show that adopting the BRQN model to learn the intervention policy from historical experiments leads to a 7.6%–43% improvement in terms of reward (i.e., profits) per episode compared with the existing policy.
- Comparing DRQN with benchmark models, the superior performance of DRQN can be attributed to the model designs that better accommodate POMDP to learn customer state representation and model the distribution of reward estimation.
- Utilizing data from multiple experiments will exponentially expand the exploration in action-state space, which will, in turn, facilitate model training and empower the model with the potential to learn better sequential intervention policy.
- The model could learn meaningful sequential interventions from historical experiments. This has the potential to help managers better understand customer sequential dynamics and lay the groundwork to facilitate future intervention design.
- The model can also be used to guide the sample allocation for future interventions. The model could achieve a good balance between the exploitation of learned policy to gain stable revenue and the exploration of new/underexplored interventions to further improve the model.

This study can be applied to the e-commerce business with historical randomized experiments to learn

Table 7. Q-Value (Mean and Standard Deviation) Comparison After Model Update between Scenario 1 and Scenario 2

		Scenario 1 after update		Scenario 2 after update			
AF	Rank	Mean	Standard deviation	Mean	Standard deviation	p-value, %	No. of impression trials for new experiment
Expected improvement	Top 1	2,951.77	63.13	2,950.33	63.39	1.14	121,000 (77.83%↓)
Expected improvement	Top 2	2,928.31	66.22	2,925.25	67.31	<0.001	121,000 (77.83%↓)
Expected improvement	Top 3	2,886.16	74.36	2,883.57	75.07	0.02	121,000 (77.83%↓)
Probability of improvement	Top 1	2,949.13	64.08	2,949.75	64.58	16.76	125,000 (77.10%↓)
Probability of improvement	Top 2	2,925.66	65.39	2,922.32	66.03	0.01	125,000 (77.10%↓)
Probability of improvement	Top 3	2,892.84	75.27	2,888.46	75.15	<0.001	125,000 (77.10%↓)
Thompson sampling	Top 1	2,946.96	64.73	2,945.92	64.22	5.53	186,000 (65.93%↓)
Thompson sampling	Top 2	2,927.75	67.91	2,926.28	67.35	1.48	186,000 (65.93%↓)
Thompson sampling	Top 3	2,890.56	76.28	2,885.92	76.92	<0.001	186,000 (65.93%↓)

sequential interventions. Ideal historical randomized experiments should be those with a large number of treated users, a long span of experiment dates, and overlaps of treated users and dates between experiments. However, if the firms do not meet these prerequisites, they can still adopt the model to guide the future intervention allocation to enrich the exploration in the state-action space. However, because of the lack of exploration in the initial stage, the learned policy might not be able to significantly outperform intervention policies from other approaches. However, BRQN organically supports incremental learning that allows the model to keep improving the learned policy with additional explorations. Another limitation is that the proposed model is not able to automatically design the next intervention. The managers need to design the interventions based on domain expertise; then, the acquisition function will decide when to evaluate the intervention to balance the exploration and exploitation. We leave (1) automatic intervention design and (2) the collaboration between human experts and algorithms in the optimal experiment design as directions for future research. Finally, although the proposed approach is more complex than many benchmarks, following trends bodes well for its adoption. Data science teams at e-commerce platforms have shown a lot of interest in developing and adopting reinforcement learning methods (Feng et al. 2018). New deep learning frameworks (Paszke et al. 2019) and sequential experimentation platforms (Bakshy et al. 2018) also simplify the development and implementation of the proposed model. Nevertheless, a platform would have to consider its development and maintenance costs vis-à-vis benefits.

In summary, the findings in this study 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, it provides a holistic approach to intervention learning and optimization along the customer journey.

Acknowledgments

The authors acknowledge the support of the NVIDIA Corporation with the donation of the Graphics Processing Unit used for this research. The authors also thank the department editor, the associate editor, and the reviewers for their insightful comments and constructive suggestions throughout the review process.

Endnotes

[1] The theoretical upper bound of the unique number of action combinations given n historical experiments is 2^n. We may not be able to observe all the combinations in the empirical data because of constraints, but any additional action combination beyond n pairs of isolated actions will be helpful to facilitate RL + A/B to learn better policy.

[2] The clickstream data consist of a sequence of page visits, and page visit is the basic component in the clickstream data. Each page visit records detailed information, like time stamp, source link, length stay, user action, device, internet connection type, etc.

[3] We also explore other model structures to learn the state representation in Online Appendix A, and their performances are inferior to the proposed structure.

[4] We explore all improvements in rainbow RL (Hessel et al. 2018) that have the potential to facilitate the training of DQN in Online Appendix B, and we only adopt those that could improve BRQN performance.

[5] The historical experiments might only explore a subset of state-action space. Strategically recruiting additional subjects to trial existing interventions could enhance the coverage of state-action space to further improve the policy.

[6] The acquisition functions aim to balance the exploration and exploitation of new intervention trials rather than randomly allocate the samples to get the clean causal effect.

[7] There are 84,380 total treated user assignments from 10 experiments. As some of the users are being assigned in multiple experiments, removing the duplicates leads to 75,913 unique treated users.

[8] The theoretical maximal action combination can be $2^{10} = 1{,}024$ given 10 historical experiments, but we only observe 22 unique treated combinations plus another 1 (all 10 actions are in the control groups) in the data. Thus, the learned sequential action combination is only optimal within the explored action space and can be further improved with additional action combinations to be explored under the guidance of future intervention trials.

[9] Thus, the reward is endogenously determined by the user and is not specific to an experiment.

[10] The limitless possibility in state representation $s_{i,t}$ makes the Bayesian update intractable. Thus, we run k-means clustering on state representation $s_{i,t}$ and set the number of clusters based on the elbow method.

[11] Online Appendix G provides a detailed discussion of the curse of dimensionality concern.

[12] Refer to Online Appendix H for more details.

[13] We also tried epsilon greedy (Sutton and Barto 2018) and Boltzmann exploration (Bertsekas and Tsitsiklis 1995) to balance the exploration and exploitation for future trials in Online Appendix I.1, and we strategically select unseen samples to update the model in Online Appendix I.2. We find that utilizing Bayesian optimization is more effective in improving the learned best policy.

[14] To utilize the explored actions and reduce the cost of evaluating new intervention a', one potential solution is to find the similar intervention a we explored before and set the prior distribution for the new intervention as $w_{a'} \sim N(\mu_a, \Sigma_a)$. Thus, we can reuse the information from the explored actions and keep updating the posterior distribution of new interventions with new observations.

[15] We also explore the effect of sample size on model outcomes in scenario 2. Online Appendix I.3 provides more details.

References

Agarwal R, Schuurmans D, Norouzi M (2020) An optimistic perspective on offline reinforcement learning. *Internat. Conf. Machine Learn.* (PMLR, New York), 104–114.

Avalos E, Barrero JM, Davies E, Iacovone L, Torres J (2022) Measuring business uncertainty in developing and emerging economies (Brookings Institution), https://policycommons.net/artifacts/4141097/measuring-business-uncertainty-in-developing-and-emerging-economies/4949875/.

Azizzadenesheli K, Brunskill E, Anandkumar A (2018) Efficient exploration through Bayesian deep Q-networks. *2018 Inform. Theory Appl. Workshop ITA 2018* (IEEE, Piscataway, NJ), 1–9.

Bakshy E, Dworkin L, Karrer B, Kashin K, Letham B, Murthy A, Singh S (2018) AE: A domain-agnostic platform for adaptive experimentation. *Conf. Neural Inform. Processing Systems* (San Diego, CA), 1–8.

Bertsekas DP, Tsitsiklis JN (1995) Neuro-dynamic programming: An overview. *Proc. 1995 34th IEEE Conf. Decision Control*, vol. 1 (IEEE, Piscataway, NJ), 560–564.

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

Cassandra AR (1998) A survey of POMDP applications. *Working Notes AAAI 1998 Fall Sympos. Planning Partially Observable Markov Decision Processes*, vol. 1724 (AAAI Press, Palo Alto, CA).

Chen T, Guestrin C (2016) Xgboost: A scalable tree boosting system. *Proc. 22nd ACM SIGKDD Internat. Conf. Knowledge Discovery Data Mining* (ACM, New York), 785–794.

Dearden R, Friedman N, Russell SJ (1998) Bayesian Q-learning. Mostow J, Rich C, eds. *AAAI 98* (AAAI Press/MIT Press, Cambridge, MA), 761–768.

Feng J, Li H, Huang M, Liu S, Ou W, Wang Z, Zhu X (2018) Learning to collaborate: Multi-scenario ranking via multi-agent reinforcement learning. *Proc. 2018 World Wide Web Conf.* (International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva), 1939–1948.

Frazier PI (2018) Bayesian optimization, chapter 11. Gel E, Ntaimo L, eds. *Recent Advances in Optimization and Modeling of Contemporary Problems* (INFORMS, Catonsville, MD), 255–278.

Gallo A (2017) A refresher on A/B testing. Harvard Bus. Rev. (June 28), https://hbr.org/2017/06/a-refresher-on-ab-testing.

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

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

Hartigan J (1969) Linear Bayesian methods. *J. Roy. Statist. Soc. B* 31(3):446–454.

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

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

Hausknecht MJ, Stone P (2015) Deep recurrent Q-learning for partially observable MDPs. *AAAI 2015 Fall Symposium* (AAAI Press, Palo Alto, CA), 29–37.

Hessel M, Modayil J, Van Hasselt H, Schaul T, Ostrovski G, Dabney W, Horgan D, Piot B, Azar M, Silver D (2018) Rainbow: Combining improvements in deep reinforcement learning. *Thirty-Second AAAI Conf. Artificial Intelligence* (AAAI Press, Palo Alto, CA), 3215–3222.

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

Inman JJ, McAlister L (1994) Do coupon expiration dates affect consumer behavior? *J. Marketing Res.* 31(3):423–428.

Katehakis MN, Veinott AF Jr (1987) The multi-armed bandit problem: Decomposition and computation. *Math. Oper. Res.* 12(2): 262–268.

Keinan A, Kivetz R (2008) Remedying hyperopia: The effects of self-control regret on consumer behavior. *J. Marketing Res.* 45(6): 676–689.

Kokkodis M, Ipeirotis PG (2021) Demand-aware career path recommendations: A reinforcement learning approach. *Management Sci.* 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. *J. Fluids Engrg.* 86(1):97–106.

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

Li H, Kannan P (2014) Attributing conversions in a multichannel online marketing environment: An empirical model and a field experiment. *J. Marketing Res.* 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 Quart.* 43(3):765–786.

Mandel T, Liu YE, Brunskill E, Popović Z (2016) Offline evaluation of online reinforcement learning algorithms. *Proc. AAAI Conf. Artificial Intelligence*, vol. 30 (AAAI Press, Palo Alto, CA), 1926–1933.

Mankiw NG (2020) *Principles of Economics* (Cengage Learning, Boston).

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

Močkus J (1975) On Bayesian methods for seeking the extremum. *Optimization Techniques IFIP Tech. Conf.* (Springer, Berlin), 400–404.

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

Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, et al. (2019) Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inform. Processing Systems 32* (NIPS, San Diego, CA).

Peters M, Ketter W, Saar-Tsechansky M, Collins J (2013) A reinforcement learning approach to autonomous decision-making in smart electricity markets. *Machine Learn.* 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. *PMLR 2020* (PMLR), 2370–2380.

Schaul T, Quan J, Antonoglou I, Silver D (2015) Prioritized experience replay. *Internat. Conf. Learn. Representations 2016* (ICLR, Appleton, WI).

Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, 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 (2022) Uncovering characteristic response paths of a population. *INFORMS J. Comput.* 34(3):1661–1680.

Sutton RS, Barto AG (2018) *Reinforcement Learning: An Introduction* (MIT Press, Cambridge, MA).

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. *Proc. AAAI Conf. Artificial Intelligence*, vol. 30 (AAAI Press, Palo Alto, CA), 2094–2100.

Wang W, Li B, Luo X, Wang X (2022) Deep reinforcement learning for sequential targeting. *Management Sci.* 69(9):5439–5460.

Watkins CJ, Dayan P (1992) Q-learning. *Machine Learn.* 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 Sci.* 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. *Inform. Systems Res.* 30(3): 787–804.