Dynamic Coupon Targeting Using Batch Deep Reinforcement Learning: An Application to Livestream Shopping

Xiao Liu

Abstract

We present an empirical framework for creating dynamic coupon targeting strategies for high-dimensional and high-frequency settings, and we test its performance using a large-scale field experiment. The framework captures consumers' intertemporal tradeoffs associated with dynamic pricing and does not rely on functional form assumptions about consumers' decision-making processes. The model is estimated using batch deep reinforcement learning (BDRL), which relies on Q-learning, a model-free solution that can mitigate model bias. It leverages deep neural networks to represent the high-dimensional state space and alleviate the curse of dimensionality. The empirical application is in a multi-billion-dollar livestream shopping context. Our BDRL solution increases the platform's revenue by twice as much as static targeting policies and by 20% more than the model-based solution. The comparative advantage of BDRL comes from more effective and automatic targeting of consumers based on both heterogeneity and dynamics, using exceptionally rich, nuanced differences among consumers and across time. We find that price skimming, reducing discounts for attractive hosts, and increasing the coupon discount level at a faster rate for low spenders are effective strategies based on dynamics, consumer heterogeneity, and the two combined, respectively.

Keywords: dynamic pricing, coupon, targeting, deep reinforcement learning, reference price, livestream shopping, e-commerce

1 Introduction

This paper presents a new solution to the dynamic coupon targeting problem based on batch deep reinforcement learning, and we conduct a large-scale field experiment to demonstrate its comparative advantages against state-of-the-art benchmarks.

In the e-commerce era, dynamic and personalized pricing strategies can improve market efficiency, allowing firms to serve more customers and generate higher profits (Furman et al. 2019). To ensure fairness and protect customer trust, companies often do not implement personalized pricing directly but instead use personalized discounts or coupons (UK Competition and Markets Authority 2018). Companies send coupons very frequently, and coupon redemptions account for tens of billions of dollars in annual marketing spending. For example, Whole Foods sends at least one coupon to each Amazon Prime member every week.[1] The aggregate value of digital coupon redemption is forecasted to surge from $47 billion in 2017 to $91 billion by 2022.[2] The high frequency of coupon delivery makes the coupon allocation problem a dynamic targeting problem, and fine-grained personalized targeting is more attractive than ever as modern marketers increasingly have access to detailed data on individual consumers and their shopping patterns.

However, having detailed consumer data is only half of the equation, as one must also have a feasible method of processing that data to generate effective targeting strategies. In this paper, we aim to create dynamic coupon targeting strategies for high-dimensional and high-frequency settings. We address three research questions: (1) How can we develop a theoretical framework that incorporates the intertemporal tradeoffs in dynamic coupon targeting to design a policy that maximizes revenue? (2) How can we evaluate the performance of dynamic coupon targeting policies? (3) What are the gains, if any, from using a dynamic targeting framework relative to a static benchmark, and what explains the differences in performance?

We face three technical challenges in answering these questions. First, we need to determine the optimal dynamic pricing strategy (e.g., price penetration, price skimming, or cyclical) when the type(s) and magnitude of demand dynamics are unknown and when firms' repeated coupon interactions with consumers may also shape consumers' behaviors over time. Second, we need to predict consumer responses to coupons without using a parametric formulation that can introduce model bias. Third, we need to overcome the curse of dimensionality problem because firms tend to collect high-dimensional data on consumer and contextual features for targeting.

[1] Statista. (2017, October). Retrieved from https://bit.ly/3CXIunS
[2] Juniper Research (November 2017) shorturl.at/ehCEN]

We overcome these three challenges by developing a framework to design and evaluate dynamic personalized pricing strategies. Our solution is a model-free, Batch Deep Reinforcement Learning (BDRL) algorithm. It is based on deep reinforcement learning techniques similar to those that were used to create artificial intelligence agents such as AlphaGo and the DQN algorithm, which defeated the best professional human players in the games of Go and Atari 2600, respectively (Mnih et al. 2015).

First, we formulate the dynamic coupon targeting problem as a Markov Decision Process (MDP) and build on the dynamic pricing literature (see Seetharaman 2009 for a review) to incorporate consumer intertemporal tradeoffs. We empirically test three dynamic pricing theories—the reference price effect, loyalty/inertia, and variety-seeking—and we identify the optimal coupon allocation sequence for each. Second, we use a model-free reinforcement learning solution, Q-learning, to solve the MDP. Our solution mitigates model bias because it does not rely on any functional form assumptions to model consumer purchase behaviors (the reward function) and the state transition process. Third, we use Deep Neural Networks (DNNs) to approximate the action-specific value functions, thereby alleviating the curse of dimensionality problem.

We apply the BDRL framework, to a novel, multi-billion-dollar livestream shopping context. Livestream shopping is a new format of e-commerce that allows brands and social media influencers to reach consumers directly through live interactive video sessions, in which hosts showcase and sell just about every product under the sun, from fresh fruits to fine jewelry. During a livestream session, consumers can ask hosts questions and make purchases in real time. High-frequency coupons are widely used by livestream shopping platforms to incentivize consumer engagement. Our data come from Taobao Live, the largest livestream shopping platform in the world. Because Taobao Live is a relatively new product (launched in 2016), the platform was still in an experimental mode at the time of our study, and it was testing the performance of coupon strategies using randomized trials. However, the platform was seeking an algorithmic solution for more efficient coupon allocation. Specifically, the platform was looking for an algorithm to decide who should receive coupons of which value (i.e., targeting based on consumer heterogeneity) and how the value should vary as each consumer gains purchase experience (i.e., targeting based on dynamics). Due to the firm's concern regarding potential negative effects of experimental policies, we use batch reinforcement learning in training and the Doubly Robust off-policy policy evaluation method to assess the performance of the proposed policy offline, against a comprehensive set of benchmarks, including both static targeting and model-based dynamic targeting policies. After demonstrating the effectiveness of BDRL offline, we run a

large-scale field experiment to provide out-of-sample validation.

The results show that our solution increases the platform's gross merchandise value by 63%. The BDRL algorithm is approximately twice as effective as static targeting strategies, suggesting the importance of demand dynamics. BDRL is also 1.5 times more effective than a dynamic structural model, indicating possible model bias and a strong economic incentive for the implementation of model-free solutions. The reason behind the superiority of the seemingly black-box BDRL method is rather intuitive and explainable. BDRL can more effectively and automatically identify when to target consumers (dynamics) and who to target (heterogeneity) based on exceptionally rich, nuanced differences among consumers and across time. Regarding dynamics, static policies suffer from myopia—they ignore the long-term negative consequences of the reference price effect. By contrast, BDRL recommends small-discount coupons at the beginning, and the recommended discount level increases gradually to avoid the negative reference price effect in the long run. The advantage regarding heterogeneity is best demonstrated with an example: BDRL suggests smaller-discount coupons when the consumer is visiting the channel of a more attractive host. This granular consumer behavioral factor would be easy to overlook with non-scalable human approaches. Lastly, BDRL organically combines dynamics and heterogeneity. For instance, the strength of the reference price effect may vary across individuals and also across time within each individual. BDRL can detect these nuances and recommend different pricing trends for different consumers, specifically, a faster increase in the coupon discount level for low spenders than for high spenders.

The paper makes three contributions. First, we make a methodological contribution by developing a new framework with which managers and researchers can design and evaluate high-frequency and high-dimensionality dynamic targeting strategies for coupons and pricing. Second, from a managerial perspective, we use a large-scale field experiment to demonstrate the real-world effectiveness of a robust machine-learning application in an area of practical importance (livestream shopping) and theoretical importance (optimal coupon allocation strategies). Third, substantively, we contribute to the dynamic pricing literature by conducting empirical tests of the three dynamic pricing theories, and we contribute to the personalized pricing literature by comprehensively comparing static targeting, structural-model-based targeting, and model-free dynamic targeting strategies. Our empirical finding that a flexible functional form does better out-of-sample/off-policy is an important data point in the debate about the raison d'être for structural models.

The rest of the paper is organized as follows. §2 reviews the literature. §3 introduces the business context

and data. §4 and §5 describe our model and the benchmarks. §6 presents the results, and §7 concludes.

2 Literature

2.1 Dynamic Pricing

Because coupons provide price discounts, a dynamic coupon targeting strategy is essentially a personalized dynamic pricing strategy (Van Heerde and Neslin 2017). Therefore, to design dynamic coupons, we draw on the dynamic pricing literature, which has documented three dynamic effects of price promotions on demand (Seetharaman 2009): (1) reference price, (2) loyalty/inertia, and (3) variety-seeking.

Reference price refers to the idea that consumers may use historical prices to construct a reference price (Winer 1986). If the firm provides a high discount now, then consumers might adopt this low price as the reference price and become disinclined to purchase in the future, as the price almost certainly will be higher than the reference price. In the future, the firm would have to provide an even higher discount to entice consumers to make purchases. The pricing implication of the reference price effect is that a firm should start with a low discount and gradually increase the discount amount over time.

Inertia or loyalty, also known as state dependence (Jeuland 1979; Dubé et al. 2010), refers to the idea that consumers might become increasingly loyal as they gain purchase experience because of switching costs or learning effects. If so, then a low discount (relative to a high discount) will increase the paid amount per transaction and short-term profits, but it also will attract and lock in a relatively small customer base, thereby securing fewer repeated purchases in the future. The recommended strategy based on inertia is penetration pricing, namely, providing an initial high discount and decreasing the discount over time.

Variety-seeking, also known as negative state dependence (Kahn et al. 1986), refers to the idea that consumers become satiated with the product after a purchase and prefer to switch products in subsequent purchases. It implies that a higher discount will increase short-term purchases but decrease long-term purchase intentions for the same product. Therefore, the variety-seeking framework recommends that firms provide a consistent, low discount (Seetharaman and Che 2009).

We add to this literature by providing empirical tests of the three intertemporal tradeoffs and generalizing to individual-specific (i.e., *personalized*) dynamic pricing strategies.

2.2 Personalized Pricing

Our paper also relates to the empirical literature on personalized pricing and first-degree price discrimination (Rossi et al. 1996; Dubé and Misra 2019), which we extend by considering pricing dynamics.

2.3 Model-Based and Model-Free Reinforcement Learning

One popular approach to solving the dynamic coupon targeting problem is to formulate a parametric model of the consumer's responses to coupons, i.e., a structural model in which the parameters are assumed to be policy-invariant.[3] This approach is advantageous in that it provides a means of predicting how customers will behave in scenarios (i.e., states and/or actions) that do not arise in the historical data. For example, even if the platform sent high-discount coupons only to new consumers, the model could predict how loyal consumers would respond to the same coupons. This approach may also better account for changes in customer behavior that result from changes in the targeting policy. For instance, if consumers made myopic decisions in the historical data but become forward-looking under the new policy, the structural model could be used to estimate consumers' shifted responses.

These benefits come with some costs. First, structural models are sensitive to the specification of the consumer utility function, which is challenging when the consumer decision-making process is complex and difficult to measure and the environment is high-dimensional and information-rich. Model misspecification can result in model bias and suboptimal policy designs. We discuss the rationale for and evidence of model bias in §3.4.3. Moreover, the computation difficulty associated with structural model approaches could render them infeasible for deployment in high-frequency business production systems. By contrast, our reinforcement learning solution does not require the estimation of structural models and thus avoids the potential issues relating to mis-specifying utility functions, model bias, and the curse of dimensionality.

Our paper contributes to the literature that uses reinforcement learning to solve sequential policy design problems in marketing (Hauser et al. 2009, Misra et al. 2019). Instead of a stateless multi-armed bandits problem, we consider full reinforcement learning where the action can change future state transitions.

[3] For example, Bertsekas 2019 outlines a method that involves estimating two structural models (one for reward and another for state transition) solving the exact dynamic programming problem using the Bellman Equation, and then using the solution to inform the new coupon policy.

2.4 Livestream and Video Marketing

Our dynamic coupon targeting problem comes from a new business context: livestream shopping. Livestreaming and video marketing is an emerging area. Existing papers emphasize the importance of leveraging unstructured data in livestreams or videos to enhance the accuracy of predictions (Zhang et al. 2019b). Similarly, we leverage summary statistics generated from image and audio features in livestreams to represent the information-rich environment experienced by consumers.

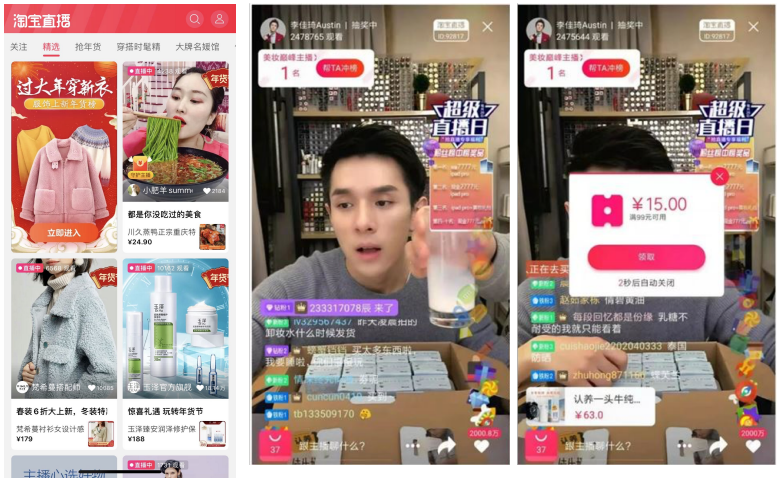
3 Data Patterns and Stylized Facts

3.1 Setting

Our empirical setting is livestream shopping. In the U.S., livestream shopping is still a relatively new concept, but it has seen rapid adoption by industry giants in e-commerce, social networking, and traditional brick-and-mortar retail. Amazon launched Amazon Live in 2019, followed by competing livestream shopping initiatives from Wayfair, Facebook, YouTube, and Walmart in 2020 and 2021. In China, where livestream shopping is more established, the industry was forecasted to generate over 500 billion RMB (71 billion USD) in annual sales transactions in 2021.[4] Major Chinese livestream shopping platforms include Taobao Live, Kuaishou, and TikTok.

Figure 1: The User Interface of Livestream Shopping

(a) Search Page (b) Channel Page (c) Coupon Page



Source: http://www.chuangyejia.com/article-12355989.html

Our dataset comes from Taobao Live, the largest livestream shopping platform in the world. The platform was launched in 2016 and has attracted over two million hosts and 30 million daily active users. Fig.1

[4]Hallanan L. (2019, March 15). Retrieved from https://bit.ly/2UwavBC

displays the user interface of Taobao Live. The typical consumer performs the following sequence of actions.

1. Land on the search page (panel (a) of Fig.1). When a consumer arrives on the livestream shopping platform, the landing page resembles a newsfeed. The consumer can browse the thumbnails of many livestream channels, each accompanied by a topic, seller name, average price, number of viewers, and number of likes for the livestream session. The thumbnails are ordered by a ranking algorithm similar to Google's PageRank; the algorithm considers factors such as whether a channel is currently live or not, the popularity of the host, and the preference match between the consumer and the channel.
2. Navigate to a channel page (panel (b) of Fig.1). If interested, the consumer will click the thumbnail of a channel to go to the channel page, where the consumer can watch the livestream content and interact with the host. Interactions may take multiple forms: add items to the cart, purchase, chat, patronize, report the host for bad behavior, share the livestream on social media, and like/upvote the livestream session.
3. Receive a coupon (panel (c) of Fig.1). Consumers are highly engaged on the Taobao Live platform. An average Taobao Live user watches two livestream videos and spends 20 minutes on the platform per week. High engagement is attributable partly to the high frequency of coupon distribution. Panel (c) of Fig.1 presents an example scenario: Shortly after the consumer navigates to the channel page, a coupon for ¥15 off ¥99 pops up on the screen. (In §3.3, we discuss all available coupons.) The consumer can click the coupon to claim it; the coupon will be applied automatically if the consumer makes a purchase before leaving the channel. Because Taobao Live is a relatively new product, at the time of our study the platform was still in the experimentation stage and used simple randomization to allocate coupons to customers.
4. Exit the channel. The consumer can exit the channel by clicking the "x" button in the top right corner of the screen, thereby returning to the search page. The coupon received within the channel is invalid after exit.

3.2 Summary Statistics

We collect data on 1,020,898 Taobao Live consumers during three months of 2019. These consumers received 25,886,094 coupons and completed 1,539,424 transactions while watching 200,568 distinct livestreams, created by 11,926 hosts who sell products in 5 categories: men's apparel, women's apparel, children's apparel, cosmetics, and jewelry. All consumers in the dataset were "active," which the platform defines as receiving at least 10 coupons from Taobao Live during the sample period. We focus exclusively on active users because the platform wants to create strategies to optimize the revenue from these consumers.

3.2.1 Raw Variables

Our data is at the coupon reception incidence level. Table 1 defines the variables in each observation.

Table 1: Data Overview

Category	Variable	Definition
User ID	User ID	The user's unique identifier
Time ID	Time ID	The coupon reception incidence, that is, the number of times that the consumer has received a coupon during the sample period
Targeting variables	Consumer	A vector of static and dynamic consumer characteristics, to be introduced in §4.2.1
	Product	A vector of static and dynamic characteristics for each product promoted in the livestream videos watched by the consumer, to be introduced in §4.2.1
	Host	A vector of static and dynamic characteristics for each host of the livestream videos watched by the consumer, to be introduced in §4.2.1
	Livestream channel	A vector of static and dynamic video characteristics for each livestream channel watched by the consumer, to be introduced in §4.2.1
Coupon information	Coupon type	A vector of indicator variables that specify the type of coupon received in the incidence. The coupon types are defined by the discount and threshold ratios introduced in §3.3
Revenue	Revenue	The revenue generated from the coupon reception incidence. If the consumer does not purchase anything, the revenue is 0. If the consumer makes a purchase, the revenue equals the payment amount (after applying the coupon)

3.3 Coupon Effect

The platform can send various kinds of coupons to consumers. Each coupon is characterized by two attributes, the discount value and threshold value. For example, a "¥15 off ¥99" coupon[5] has a discount value of ¥15 and a threshold value of ¥99. Because the value of the coupon is relative to the price of the product, we operationalize the coupons by calculating two metrics: the discount ratio (ratio of the discount value to the threshold value) and the threshold ratio (ratio of the threshold value to the average price in the livestream[6]). For example, if the average price is ¥120, then a "¥15 off ¥99" coupon has a discount ratio of 15.15% (=15/99) and a threshold ratio of 82.5% (=99/120). Note that consumers prefer higher discount ratios and lower threshold ratios.

Both discount ratio and threshold ratio are continuous variables, but we discretize both variables to five levels: LL (extra-low), L (low), M (middle), H (high), and HH (extra-high), based on the quintiles in their distributions. This approach reduces dimensionality and makes the solution more tractable. The quintiles are displayed in Table 2. In the running example of a "¥15 off ¥99" coupon, the 15.15% discount ratio falls in the LL discount level, while the 82.5% threshold ratio falls in the HH threshold level. In sum, we consider the platform's choice set to include 25 types of coupons (5 discount levels * 5 threshold levels).

[5]The displayed coupons include only these two values with no other variation in message content.

[6]The "average price" includes all products displayed in the same livestream video. On Taobao Live, a consumer receives only one coupon per livestream video, but the coupon can be applied to any of the products featured in the video. We calculate the threshold ratio with the average price because we need to assign a threshold value to every coupon reception incidence, even when we do not know which product the coupon was applied to, for instance, when a consumer did not purchase any products. Usually, all products within the same livestream have similar prices, so the threshold ratio based on the average price represents the consumer's overall impression of the difficulty of meeting the coupon's threshold.

Table 2: Coupon Type Discretization by Discount Ratio and Threshold Ratio

	Ratio Level				
	LL	L	M	H	HH
Discount Ratio	0%-15%	16%-25%	26%-45%	46%-70%	71%-100%
Threshold Ratio	0%-20%	21%-35%	36%-60%	61%-70%	70%-Inf

Note: The support of the discount ratio is from 0 to 100%, and that of the threshold ratio is from 0 to infinity. (When the threshold ratio is greater than 100%, the consumer has to purchase multiple items to qualify for the discount.)

Figure 2: Redemption Rate by Coupon Type

(a) By Discount Level (b) By Threshold Level

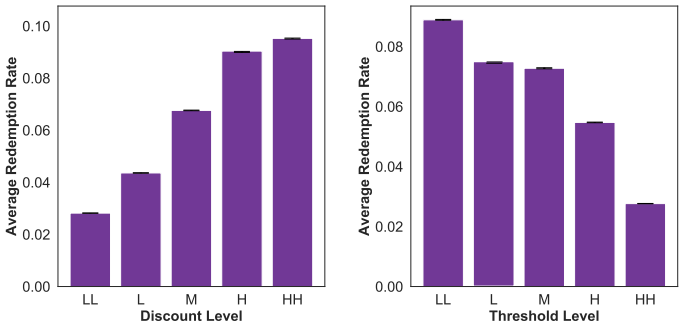


Fig.2 displays the coupon redemption rate for each discount level and threshold level. As the discount level increases (from LL to HH), the redemption rate increases (Fig.2a) because a higher discount (i.e., lower price) gives a stronger incentive to purchase. As the threshold level increases, the redemption rate decreases (Fig.2b) because a higher threshold makes it harder for a transaction to qualify for the coupon, making the coupon less attractive to consumers.

3.4 Model-Free Evidence

Compared to one-size-fits-all coupon strategies, the benefits of moving to dynamic targeting depend on two conditions: consumer-level heterogeneity in the effectiveness of coupons, and intertemporal tradeoffs in the impacts of the coupon strategy between the current period and future periods. §3.4.1 and §3.4.2 provide evidence of both conditions in the data. Moreover, to motivate our choice of a model-free reinforcement learning approach, we provide descriptive evidence of the possibility of model bias in §3.4.3.

3.4.1 Heterogeneity in Sensitivity to Discounts

Consumers have different price sensitivities, so the optimal coupon discount level likely varies across consumers. In Fig.3, we graph the effectiveness of two coupons, one with a low discount level ("L," dark purple color) and the other with a high discount level ("H," light pink color), for two segments of consumers, high spenders (the top 10% in the spending amount) and low spenders (the bottom 10% in the spending amount).

A coupon's effectiveness is measured by the revenue it generates (i.e., the transaction amount). For high spenders, the low discount level coupon generates more revenue than the high discount level coupon (¥36.4 vs. ¥31.1, p_value < 0.001); for low spenders, the pattern reverses: the high discount level coupon generates more revenue (¥3.8 vs. ¥1.7, p_value < 0.001). The opposite patterns might be driven by different price sensitivities.[7] It is possible that high spenders are less price-sensitive than low spenders, so the existence of a discount may nudge the high spender toward a purchase, but the magnitude of the discount does not matter much. If so, for high spenders, low discounts may achieve the optimal balance between enticing purchases and maintaining high profit margins. For low spenders, however, their heightened price sensitivity may require deeper discounts (rather than the mere existence of a discount) to entice purchases. As these summary graphs illustrate, the consumers in our dataset are heterogeneous in their sensitivities to different coupons, fulfilling the first criterion for a setting that would benefit from a dynamic targeting strategy.

Figure 3: Heterogeneity

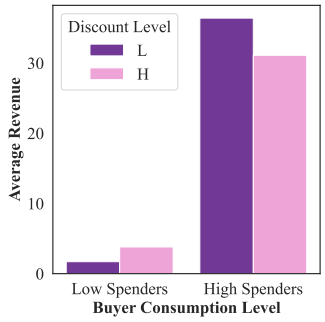


Figure 4: Reference Price

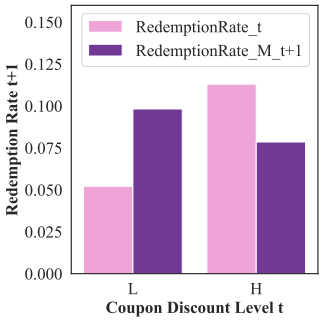
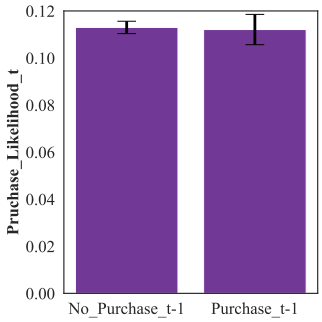


Figure 5: State Dependence



3.4.2 Intertemporal Tradeoffs

This section tests whether consumers in our setting exhibit the three types of demand dynamics documented in the literature (§2.1). We find that consumers in our setting are affected by reference prices, but their purchase decisions do not appear to be state dependent or forward-looking.

3.4.2.1 Reference Price

Reference price theory states that consumers often adopt past prices as reference points for evaluating future prices. Prices higher than the reference points are thought of as losses, and prices lower than the reference points are thought of as gains. Thus, reference price theory predicts that increasing discounts over time would lead to higher sales in later periods, while decreasing discounts over time would lead to lower sales.

[7] We interpret the findings in Fig.3 as the causal impact of the coupon type on revenue by consumption level. Although high spenders and low spenders may choose different livestream channels or products, within each channel, the coupons are randomly allocated (according to the platform's current strategy). Thus, the interpretation of the plot is not subject to selection bias; that is, consumers do not self-select into different treatments (coupon types).

This is precisely the pattern that we see in the data. Fig.4 plots the redemption rate for two coupons, one with a low discount level ("L"), and the other with a high discount level ("H"),[8] over two consecutive coupon reception incidences: period t in light pink and period $t+1$ in dark purple. Period t is the current period, that is, the period in which the coupon with either a high or low discount level was allocated; period $t+1$ is the next period, in which a middle discount level ("M") was allocated. Fig.4 shows that the redemption rate in the current period is higher with a high discount than with a low discount (11.30% vs. 5.21%, p_value < 0.001), but in the next period, the redemption rate is higher among consumers who received the low discount, not the high discount, in period t (7.85% vs. 9.82%, p_value < 0.001). The reversal is consistent with the theory that consumers adopt the coupon level in the current period as the reference price for the next period, so the middle discount in the next period is more attractive than the reference price established by the low-discount coupon but less attractive than the reference price established by the high-discount coupon.

3.4.2.2 State Dependence (Loyalty/Inertia and Variety-Seeking)

Another type of demand dynamics is the state dependence effect. A positive state dependence effect captures customer loyalty/inertia, while a negative state dependence effect captures variety-seeking. In our setting, state dependence would occur if a consumer who made a purchase (vs. who did not make a purchase) during incidence t-1 is either more or less likely to make a purchase during incidence t (Dubé et al. 2010).[9] Following the literature, we evaluate state dependence by comparing the purchase likelihoods between consumers who did and did not make a purchase in the previous period after receiving a coupon of a given discount level. In appendix B, we provide additional evidence that helps disentangle structural state dependence from the possible confounds of unobserved heterogeneity.

We do not find evidence of state dependence in our setting. Fig.5 includes the subset of consumers who received high-discount coupons in two consecutive periods. Some consumers made a purchase in incidence t-1 while others did not, but the purchase decision at t-1 did not affect the purchase likelihood at t (11.30% vs. 11.21%, p_value = 0.58). In other words, consumers who purchased in the previous period did not appear to have been locked in or disincentivized from making purchases in subsequent periods, as would be predicted by loyalty/inertia and variety-seeking, respectively. We speculate that this setting might lack of state dependence because the most popular categories in livestream shopping are apparel and cosmetics,

[8] The same effect holds when we use any other pair of the five discount levels, LL, L, M, H, and HH.

[9] We test state dependence at the level of the platform (Taobao Live), not the brands, because the decision maker in our problem is the platform, not brand manufacturers or hosts. Even if consumers develop loyalty to specific hosts (we show evidence of this in Appendix A), this behavior does not affect the dynamic pricing decision for the platform.

which are often associated with low customer loyalty.[10]

3.4.2.3 Forward-Looking

The three demand dynamics tested in the previous sections assume that consumers are backward-looking, i.e., a consumer's reaction to a coupon in the present is affected by coupons received in the past. Another form of demand dynamics involves forward-looking consumers, whose objective is to maximize not current utility but rather total discounted utility from now on. Forward-looking consumers form expectations about the platform's future targeting policies when making current decisions. For example, if a consumer foresees that future coupons will be more generous, she might forego purchases now and instead engage in strategic waiting. Or, if the consumer anticipates that future coupons will be less generous, she might stockpile now.

We formally test whether consumers are forward-looking in the livestream shopping setting in Appendix D. In short, we find no evidence of forward-looking behaviors, and we build the model framework (§4) under the assumption that consumers are backward-looking, only. We offer two explanations for the absence of forward-looking behaviors. First, the popular product categories on Taobao Live are seasonal and non-durable, so they are incompatible with either strategic waiting or stockpiling. Second, the platform allocated coupons randomly during the sample period. If the platform instead used strategies such as price penetration or price skimming, consumers might have learned to behave strategically and become forward-looking in the long run. We present an extension of the model framework that allows forward-looking behaviors in Appendix D.2, and in §D.3, we discuss potential pricing strategies that platforms could adopt if facing forward-looking consumers.

3.4.3 Model Bias

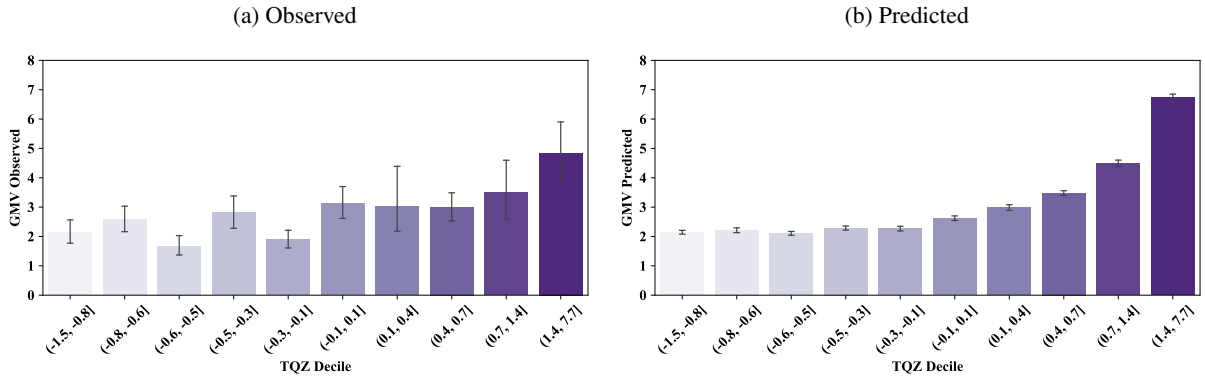
Modeling consumer purchase behaviors in a complex shopping environment is challenging and may be subject to model bias.[11] Model bias could emerge from two sources: distributional mismatch (i.e., covariate shift) and wrong functional forms. In distribution mismatch, the model is unbiased in the training data, which was collected under the current policy, but it becomes biased under the new policy because the two policies have different state-action distributions. Distribution mismatch is inevitable because the state-action distributions under the two policies are always different. The second source of model bias due to

[10] Evidence of low customer loyal in apparel and cosmetics can be found here (https://bit.ly/2UwavBC) and here (https://bit.ly/3zHzvFi). One possible explanation for the absence of variety-seeking is that only the frequent buyers become satiated enough to seek variety. We test this hypothesis by dividing consumers into two groups: infrequent buyers and frequent buyers. Results in appendix C indicate no variety-seeking even for the frequent buyers.

[11] See Appendix E for an example.

wrong functional forms could also exist in our data. Fig.6 shows the relationship between the consumer activity level (TQZ)[12] and gross merchandise value (GMV). Panel (a) in Fig.6 shows that, in the observed data, the relationship between TQZ (discretized into deciles) and GMV is non-monotonic, with many ups and downs. In panel (b), however, a widely used machine learning model, Gradient Boosting Decision Tree (GBDT, Friedman 2001), predicts a monotonic relationship. A comparison of the observed and predicted values suggests that commonly used models often are mis-specified, thereby creating bias and leading to suboptimal policy decisions.

Figure 6: Relationship Between TQZ and GMV: Comparison of Observed and Predicted Values



The above model-free evidence indicates that to effectively implement dynamic coupon targeting in our setting, we need a model that can incorporate rich consumer heterogeneity and intertemporal tradeoffs and can avoid model bias. Rich consumer heterogeneity, intertemporal tradeoffs, and difficult-to-parameterize consumer preferences are common not only in livestream shopping settings but also for any marketer in the digital age. Before we introduce the model framework, in the next section, we discuss the generalizability of the findings from our setting to e-commerce.

3.5 Generalizability of the Setting

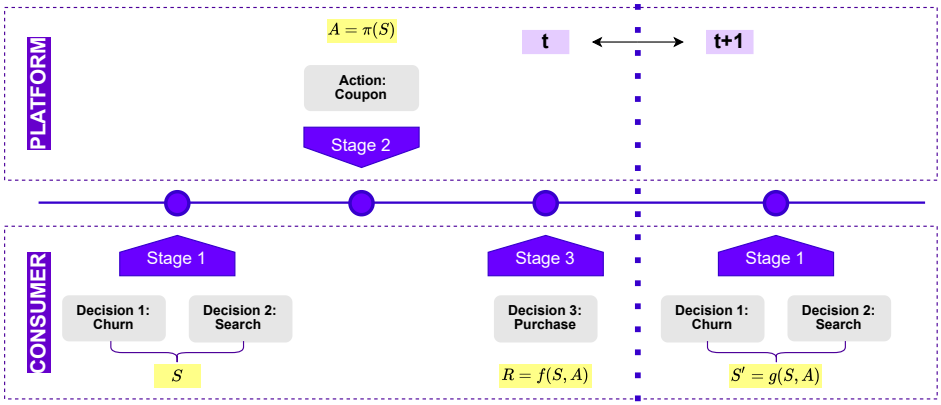
In Fig.7, we present a stylized framework that can be used to extend our model of the dynamic coupon targeting problem to a more generalized e-commerce setting.[13] Our BDRL approach analyzes the livestream shopping decision process in terms of the consumer's decisions about churn, search, and purchase, which is a framework that applies to most e-commerce settings. In our stylized sequence of events, each period t comprises three stages. In Stage 1, the consumer makes a churn decision: whether to leave the e-commerce

[12] TQZ is a score that Taobao uses to indicate the consumer's activity level (higher TQZ is more active), calculated from the consumer's recent search and purchase histories, payment amounts, and review posting activity.

[13] The mathematical notations in Fig.7 are explained in §5.3.

platform permanently. If not, then the consumer visits the platform and starts the search process by choosing and visiting a product or seller's webpage (in our context, the consumer clicks on a thumbnail to access a livestream channel). The churn and search decisions jointly determine the shopping context, which is defined by the features of the consumer, seller (i.e., host), product, and webpage (in our context, the livestream channel). In Stage 2, the platform decides which coupon to send to the consumer or, more generally, determines the price for the consumer. In Stage 3, the consumer makes the purchase decision based on the shopping context and the coupon/price information. This concludes period t. When the next period, $t+1$, arrives, the consumer will return to the churn decision and repeat the process. In sum, the consumer's decision process can be characterized by the churn, search, and purchase decisions.

Figure 7: Structural Model Sequence of Events



The stylized consumer decision process applies to many e-commerce contexts including livestream shopping, traditional online shopping, customer relationship management (CRM), and mobile apps for consumer services such as Uber. In the next section, we introduce a general model framework to solve the dynamic coupon targeting problem.

4 Model

4.1 Problem Definition

We formally define the dynamic targeting problem as follows. Each consumer i ($i = 1, 2, \ldots, I$) on the e-commerce platform[14] could visit many sellers' product pages and receive a total of T_i coupons within a window of time. Consumers can receive only one coupon each time she visits a seller's page, so we treat every consumer visit to a seller's product page as a coupon reception incidence.[15] The platform chooses the

[14] In the general framework, we use a single term, "the e-commerce platform," to represent all the entities that face the dynamic targeting problem. Examples include livestream shopping platforms, generic e-commerce platforms, and firms using CRM apps.

[15] If a consumer visits more product pages, she receives more coupons. There are no browsing or purchase occasions without a coupon.

coupon; in each coupon reception incidence t, the platform sends a coupon $A_{it} \in \mathbb{A}$ to consumer i where $\bar{A}$ is the number of available coupon types and $\mathbb{A}$ is the set of all coupons.[16] The platform's decision of which coupon to send during each consumer visit is based on context information including characteristics of the consumer, seller, webpage (i.e., virtual store), and product. These context characteristics are denoted as the state variables, $\mathbf{S}_{it} \in \mathbb{S}$. The platform aims to create a targeting policy $\pi(A_{it}|\mathbf{S}_{it}) : \mathbb{S} \times \mathbb{A} \to [0,1]$, which is a mapping from the state and action space to probabilities (i.e., $\pi(A_{it}|\mathbf{S}_{it})$ is the probability of choosing action A_{it} in state $\mathbf{S}_{it}$, $0 \leq \pi(A_{it}|\mathbf{S}_{it}) \leq 1$)[17] for the purpose of maximizing its reward, $R_{it} \in \mathbb{R}$. The reward of the platform is defined as the GMV, which is the total revenue generated by consumers' purchases.[18] Intuitively, revenue is equal to the after-coupon price if the consumer stays on the platform and makes a purchase, and 0 otherwise. Thus, the revenue function is:

$$R_{it}(\mathbf{S}_{it}, A_{it}) = \begin{cases} price(\mathbf{S}_{it}) * (1 - discount_{A_{it}}) * \mathbb{1}\{Buy_{it}(\mathbf{S}_{it}, A_{it})\} & \text{if } \mathbf{S}_{it} \neq Churned \\ 0 & \text{othewise} \end{cases} \tag{1}$$

where *Churned* is the absorbing state, when a consumer permanently leaves the platform[19]; $\mathbb{1}\{Buy_{it}(\mathbf{S}_{it}, A_{it})\}$ is an indicator function that takes the value of 1 when consumer i makes a purchase and 0 otherwise; and $discount_{A_{it}}$ is the discount ratio associated with action A_{it}. The discount ratio has two competing effects on revenue: a higher discount ratio lowers the consumer payment amount but increases the purchase likelihood.

[16] For example, in our field setting, the set of coupons $\mathbb{A}$ that is available to the Taobao platform consists of coupons that reflect any combination of discount values and threshold values that the platform would like to implement.

[17] We use the specification of stochastic policies, where each action has a probability between 0 and 1, because Taobao Live used stochastic allocation policies to generate our training data. The notation for stochastic policies is more general and all-inclusive than that for deterministic policies, where the probability of each action is either 0 or 1.

[18] Managers at Taobao and other e-commerce companies told us that the GMV usually is the objective function of choice for pricing decisions on e-commerce platforms. Our BDRL method can easily be adapted to maximize other reward objectives such as Taobao Live's gross or net profits.

[19] Taobao Live defines a "churned consumer" as one who has not used Taobao Live for X consecutive days, where X is a number between 1 and 365. We cannot disclose the exact value X because it is a trade secret. This type of churn definition is an industry common practice and consistent with the "silent" churn definition in Ascarza et al. (2018). We acknowledge that our definition of churn is only an approximation because, in our non-contractual setting, there is no mechanism that would prevent a consumer from leaving and then returning to the platform. However, customer attrition from the platform is a real phenomenon, and Taobao management recognizes that if a customer remains inactive for a long time, then she is statistically unlikely to return to the platform. We use a simplified assumption in which consumers choose between staying and permanently leaving because it allows us to formulate the problem as an episodic MDP (with a terminal state instead of a continuing one), which "is mathematically easier because each action affects only the finite number of rewards subsequently received during the episode" (Sutton and Barto 2018). Of course, the policy learned under this assumption may be more myopic than the optimal one. If some consumers can come back to the platform after a "hibernation" period (X or more days), then our policy overlooks the impact of a before-the-hibernation coupon (action) on the after-the-hibernation revenue (reward). Underestimating the future impact may motivate the platform to be aggressive and provide coupons that are more generous than necessary, hence reducing the platform's total revenue. Future research could explore more sophisticated estimates of churn to relax our assumption.

The optimal discount ratio balances the two competing effects to achieve the highest revenue. The platform's objective function is to maximize the expected, total discounted GMV across consumers and across time by creating the optimal targeting policy π^*:

$$\pi^* = argmax_\pi E_\pi \left[\sum_{i=1}^{I} \sum_{t=0}^{T_i} \delta^t R_{it}\left(\mathbf{S}_{it}, A_{it}\right) \right] = argmax_\pi E \left[\sum_{i=1}^{I} \sum_{t=0}^{T_i} \delta^t \sum_{A_{it} \in \mathbb{A}} \pi\left(A_{it}|\mathbf{S}_{it}\right) R_{it}\left(\mathbf{S}_{it}, A_{it}\right) \right]$$

where δ is the discount factor,[20] and the expectation is taken over the initial state probabilities $p(\mathbf{S}_{i0})$, the state transition probabilities $p(\mathbf{S}_{it+1}|\mathbf{S}_{it}, A_{it})$, and the action distribution $\pi(A_{it}|\mathbf{S}_{it})$.

To solve the optimization problem, we collect historical data in batch mode, meaning that all the data become available to the econometrician at the same time; the econometrician cannot collect new data on the go. The data are in a panel format where, for each consumer, we observe the {state, action, reward} tuple repeating multiple times. Formally, the data are denoted as $\mathscr{H} = \{\mathbf{S}_{i0}, A_{i0}, R_{i0}, \mathbf{S}_{i1}, A_{i1}, R_{i1}, \ldots, \mathbf{S}_{iT_i}, A_{iT_i}, R_{iT_i}\}_{i=1}^{I}$.

We note several assumptions behind our definition of the dynamic targeting problem.

- The decision maker (of coupon allocation) is the platform, not the sellers. Although each seller can determine the product price on her own webpage,[21] the seller has no control over the platform's decisions about coupon allocation. This assumption might seem restrictive, but it applies to most e-commerce platforms and offline retailers that sell many products, where the coupon allocation decision is made by a centralized system.
- The action set is fixed, and there is no capacity constraint for products or coupons. As described briefly in §3.3 and in more detail in §6.1.2, we discretize the action space to $\bar{A}$ coupon types. When a consumer arrives, the platform can send any of the $\bar{A}$ coupons in the action space to the consumer. Moreover, the platform can send unlimited coupons of every type. We solve only the problem of coupon allocation as the coupon supply process is outside the scope of the current study. We assume that the coupon supply process

[20] The discount factor applies to future incidences, which may not occur at fixed intervals of calendar time. That said, at the time of each coupon offer decision, although the platform does not know the precise time intervals between the consumer's future visits, the platform knows the expected intervals. As such, the discount factor in this setting can be thought of as the discount rate based on the expected time intervals between each consumer's visits. Our discount rate specification is also consistent with Sutton and Barto (2018): "The MDP framework is abstract and flexible and can be applied to many different problems in many different ways. For example, the time steps need not refer to fixed intervals of real time; they can refer to arbitrary successive stages of decision making and acting." Appendix O provides more discussion of the implications of not adjusting the discount rate by the time intervals between consumer visits.

[21] The livestream shopping market is characterized by monopolistic competition; each seller can determine the product price on her own webpage, and the products are differentiated from one another. There are millions of sellers in the market, and each seller independently sets prices to maximize her own profit. The platform, however, makes the coupon allocation decision using a centralized system. The coupons affect the end price that consumers pay but not the list price, which is controlled by sellers.

is exogenously given and, thus, is not affected by the coupon targeting policy.

- We do not model the consumer's choice of when to use the coupon. Coupons expire quickly and can be used to purchase only the products promoted on the corresponding webpage; consumers cannot accumulate many coupons over time and compare them across webpages before making a purchase decision.
- The consumer's search behavior (decision of which webpage to visit and in which sequence) is exogenous, not affected by coupons.[22]
- There are no fairness concerns. Consumers are familiar with the platform's targeting practices. Although different consumers receive different coupons, they do not feel they have been treated unfairly. The ubiquity of personalized coupons from different brands justifies this assumption.[23]
- Consumers are not forward-looking. As discussed in §3.4.2, in each period, the consumer's objective is to maximize current utility, not future utility. Consumers cannot anticipate the platform's future strategy. In Appendix D, we discuss how to extend the current model to the forward-looking scenario.

4.2 Batch Deep Reinforcement Learning (BDRL) Framework

As illustrated in §3.4.2, the coupon targeting policy likely affects both immediate revenue and future revenue because consumers may use historical coupon levels as reference points for evaluating future coupons. To solve for an optimal coupon targeting policy that has the flexibility to account for such intertemporal tradeoffs, we employ a deep reinforcement learning framework (Fig.8). The deep reinforcement learning problem is defined as a Markov decision process (MDP): $(\mathbb{S}, \mathbb{A}, p, R, \delta)$, where

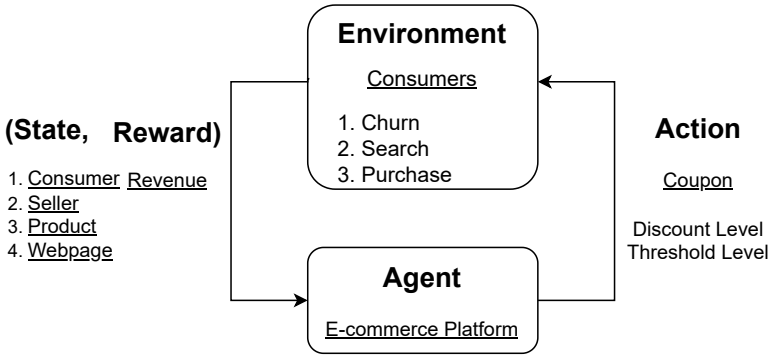
- $\mathbb{S}$ and $\mathbb{A}$ denote the state and action spaces, respectively.
- At time step t, the agent of the reinforcement learning system (the e-commerce platform) takes action $A_t \in \mathbb{A}$ in state $\mathbf{S}_t \in \mathbb{S}$, receives the reward R_t, and observes the next state $\mathbf{S}_{t+1}$, according to the transition process $p(R_t, \mathbf{S_{t+1}} | \mathbf{S_t}, A_t)$. The transition process captures the environment that the agent faces.
- The agent's goal is to maximize the total discounted reward, which is also called the return: $Return = \sum_{t=\tau+1}^{\infty} \delta^t R_t(\mathbf{S}_t, A_t, \mathbf{S}_{t+1})$ where the discount factor is δ.

[22] The assumption of exogenous search behavior is informed by empirical evidence, but it also can be relaxed. On the one hand, we tested whether coupons affect a consumer's subsequent webpage choices by comparing the distributions of seller characteristics (e.g., seller attractiveness) in a store visit, conditional on the type of coupon received in the previous incidence. The data pattern fails to reject the null hypothesis (see appendix A). On the other hand, we could relax the assumption of exogenous search behavior by allowing the seller-related and webpage-related state variables to be dynamic instead of stationary. See more details in §4.2.1.

[23] When our data collection occurred, Taobao had been operating for 20 years, and consumers were used to the personalized promotion activities. Even though livestreaming creates an interactive environment in which consumers and hosts can chat with each other, we found no evidence that consumers publicly share individual coupon information in the chat.

In our setting, the reward R_t is the GMV from the consumer, which is determined by how the coupons affect the consumer's ultimate purchase decision. Within the reinforcement learning framework, the "environment" is the process that generates the reward and state changes based on the agent's action and current state values. As discussed in §3.5, the environment is the combination of three consumer decisions: 1) churn, that is, whether to leave the platform and never come back (which affects state transitions), 2) search, that is, which webpage to visit (which affects state transitions), and 3) purchase, that is, whether to purchase after receiving the coupon (which affects the reward and state transitions).

Figure 8: Reinforcement Learning Framework



We provide the details for the primitives of the MDP in the following subsections.

4.2.1 States

The states contain the contextual information collected by the platform for targeting purposes. In the shopping context, states can be categorized into four groups of characteristics: consumer, seller, product, and webpage. Furthermore, we divide states within each group into static states (Table 3) and dynamic states (Table 4) because only some state variables are affected by actions. The distribution of a dynamic state changes after an action is performed, while the distribution of a static state variable is independent of actions.

4.2.1.1 Static States[24]

Table 3: Static State Variables

Group	State Variables
Consumer	Demographics (e.g., gender), behavioral variables (e.g., purchasing power), product category preference, the consumer's preference for specific sellers
Seller	Demographics, the seller's popularity (e.g., monthly revenue, quality rating)
Product	Product category, price, market share, rating, etc.
Webpage	Engagement scores, e.g., the average number of consumers who visit the webpage and who add a product to the cart; unstructured data such as the webpage aesthetics features

[24] Static features are defined as those not affected by actions. Static features may come from a stationary distribution instead of being a fixed number.

4.2.1.2 Dynamic States

Our BRDL approach uses dynamic states based on the recency, frequency, and monetary value (RFM) framework (Fader et al. 2005), which is commonly used by industry practitioners to quantify customer transaction histories. Recency refers to how much time has passed since the customer has performed the activity of interest (e.g., making a purchase or visiting a webpage); frequency refers to how many times the customer has performed the activity within a set period, and monetary value refers to the dollar value associated with the activity.[25] Dynamic states are categorized into five groups: consumer-related, seller-related, product-related, webpage-related, and the terminal state.

The dynamic states can incorporate the intertemporal tradeoffs discussed in §2.1 and §3.4.2. Because our model-free evidence shows that state dependence (inertia and variety-seeking) is not present in our setting, we focus on the reference price effect alone.[26] We capture the reference price effect using the monetary value associated with the coupon (monetary_coupon): specifically the average, minimum, and maximum of the discount ratio and threshold ratio of the coupons received by the consumer since the beginning of the sample period.[27]

Table 4: Dynamic State Variables

Group	State Variables
Consumer	Related to coupon reception behaviors: the number of days since the consumer last received a coupon (recency_coupon), the number of coupons the consumer has received since the beginning of the sample period (frequency_coupon), and the average, minimum, and maximum of the discount ratio and threshold ratio of the coupons received since the beginning of the sample period (monetary_coupon)
Seller	The total number of sellers visited by the consumer since the beginning of the sample period (frequency_seller)
Product	The number of periods (coupon reception incidences) since the consumer last purchased a product (recency_product), the number of products purchased since the beginning of the sample period (frequency_product), the average, maximum, and minimum prices of the products purchased (monetary_product), and the cumulative spending since the beginning of the sample period (monetary_product)
Webpage	The number of periods since the consumer last visited a webpage (recency_webpage) and the number of webpages visited since the beginning of the sample period (frequency_webpage)
Terminal	Once the consumer stops coming back to the platform, the terminal state occurs. This state captures consumer churn.

[25] The RFM variables could lead to a potential endogeneity problem if firms previously used these variables to determine how to target consumers. Although our input data does not suffer from this problem because the platform allocated the coupons randomly (i.e., without targeting), future researchers need to exercise caution in scenarios with the endogeneity issue and consider solutions such as the instrumental variable approach (Gönül et al. 2000) or the latent trait approach (Rhee and Russell 2009).

[26] We also consider a specification with state dependence. The results remain qualitatively unchanged (see Appendix N).

[27] Identifying references price effects is typically challenging in scanner data applications, where one does not know whether a customer observed the price (Rajendran and Tellis 1994). In livestream shopping, however, the identification of reference price effects is less ambiguous for two reasons. First, the livestream shopping app records the livestream channel visited by each consumer, the product the consumer clicked, and the targeted coupon received, so the econometrician knows whether the consumer observed the price. Second, our training data was generated using a random targeting policy, so no targeting endogeneity exists.

Table 5: The Transition Processes of Dynamic State Variables (RFM)

Variable	Transition
Recency_product	$r\_product_{t+1} = \begin{cases} 0 & \text{if } purchase_t = 1 \\ r\_product_1 + 1 & \text{otherwise} \end{cases}$
Frequency_product	$f\_product_{t+1} = \begin{cases} f\_producttt_t + 1 & \text{if } purchase_t = 1 \\ f\_producttt_t & \text{otherwise} \end{cases}$
Monetary_product	$avg\_product_{t+1} = \begin{cases} \frac{avg\_product_t + price_t}{f\_product_t + 1} & \text{if } purchase_t = 1 \\ avg\_product_t & \text{otherwise} \end{cases}$
	$max\_product_{t+1} = \begin{cases} price_t & \text{if } price_t > max\_product_t \\ max\_product_t & \text{otherwise} \end{cases}$
	$min\_product_{t+1} = \begin{cases} price_t & \text{if } price_t < min\_product_t \\ min\_product_t & \text{otherwise} \end{cases}$

Note: This table presents only the transition process for product-related dynamic states. The corresponding transition process for consumer-related, seller-related, and webpage-related dynamic states are derived similarly.

4.2.2 State Transition

We discuss the state transition process for static state variables and dynamic state variables separately.

Static states have two types of transition processes: fixed and stationary. The values of some static states (e.g., the consumer's gender) are fixed throughout the sample period. The values of other static variables change from time to time, but their distributions are stationary, not affected by the platform's actions. For example, the webpage that a consumer visits, and the associated seller and product characteristics, change in each consumer visit. However, based on the data patterns in Appendix A, we assume that the coupon a consumer receives in the current period does not affect her webpage choice in the next period. In other words, the consumer's webpage choice is independent of the coupon targeting policy. Under this assumption, we allow the state transition process for the seller, product, and webpage characteristics to be stationary.

The transitions of dynamic states are all stochastic. Table 5 illustrates the product dynamic states as examples. For instance, recency_product measures the number of periods since a consumer purchased a product last time. If a consumer purchases a product in incidence t, then recency resets to 0; otherwise, recency increases by 1. This transition is stochastic from one period (which consists of three stages, Fig. 7) to the next because whether the consumer purchases or not is stochastic. Once the terminal state, Churned (i.e., the consumer permanently leaves the platform), is reached, there are no subsequent state transitions.

4.2.3 Action

As explained in §3.3, we discretize the platform's coupon choice action space using two metrics: the discount ratio and the threshold ratio. After discretization, the total number of actions is $\bar{A}$.

4.2.4 Reward

The reward function R, as defined in Equation 1, is a mapping from the state and action space to a real-valued number, $\mathbb{S} \times \mathbb{A} \rightarrow \mathbb{R}$.

4.3 Policy Learning

Algorithm 1 Q-Learning (Watkins 1989)

1. Algorithm parameters: learning rate $\alpha \in (0,1]$, small $\varepsilon > 0$
2. Initialize $Q(\mathbf{S},A)$, for all $\mathbf{S} \in \mathbb{S}, A \in \mathbb{A}(s)$,arbitrarily except $Q(terminal,\cdot) = 0$
3. Loop for each ε:
4. Initialize $\mathbf{S}$
5. Loop for each step of the episode:
6. Choose A from $\mathbf{S}$ using policy derived from Q (e.g., ε-greedy)
7. Take action A, observe, R, $\mathbf{S}'$
8. $Q^{new}(\mathbf{S},A) \leftarrow (1-\alpha)\underbrace{Q(\mathbf{S},A)}_{oldvalue} + \alpha * \underbrace{(R + \delta \max_{A} Q(\mathbf{S}',A))}_{update}$
9. $\mathbf{S} \leftarrow \mathbf{S}'$
10. until $\mathbf{S}$ is terminal

To solve the MDP problem described in §4.2, we use the Batch Constrained Q-Learning algorithm (BCQ, Fujimoto et al. 2019). To understand how this algorithm works, we first introduce the concept of Q-learning (Algorithm 1). "Q" stands for the "quality" of an action taken in a given state; "Q-learning" involves a function that calculates the return (total discounted rewards) used to provide the reinforcement feedback in the learning process.[28] Before learning begins, Q is initialized to a possibly arbitrary fixed value (line 2 of algorithm 1). Then, at each time t (line 5), the agent (the platform, in our case) selects an action A (line 6), observes a reward R and a new state $\mathbf{S}'$ (which may depend on both the previous state and the selected action, line 7). Given the new reward R and new state $\mathbf{S}'$, Q is updated to Q^{new} using a weighted average of the Q value from the previous iteration and the new information ($R + \delta \max_A Q(\mathbf{S}',A)$); the weight α is the learning rate (a hyperparameter), and δ is the discount factor (line 8).

Q-learning is model-free because it does not rely on any functional form assumptions about the reward function or the state transition function. Rather, Q-learning uses samples collected from the environment (in the historical data, in our case) to directly observe the reward and the next state. The sampled values (i.e., R and $\mathbf{S}'$ in step 8 of Algorithm 1) [29] enable the Q-function to update even though we know only the reward value, not the functional form of the reward as a function of the state and action.

[28] The Q-function is often defined as a table, called the Q-table, which stores one state-action pair and the associated Q-value in each row.

[29] Although both Q-learning and the Conditional Choice Probability (CCP) estimator (Hotz and Miller 1993) use sample observations (the so-called cell estimators) in the estimation procedure, they are fundamentally different algorithms. We compare the two algorithms in Appendix F.

The original Q-learning algorithm is appropriate when the state and action spaces are small. For enormous state and action spaces (e.g., with continuous state values), however, we cannot expect to find an optimal value function within the limit of infinite time and data. Instead, the value function needs to be a parameterized function, e.g., DNNs with many hidden layers (Mnih et al. 2015) or the random forest (RF) algorithm (Kim et al. 2021). We choose DNNs, a parametric model with a large number of parameters, as the functional approximator.[30]

Another important feature of our solution is that we use batch reinforcement learning, which means that our dataset is fixed; no further interactions between our policy and the environment can occur, and our problem does not involve the exploration-exploitation tradeoff. We choose to use batch reinforcement learning because we face a high-stakes problem involving millions of consumers and billions of dollars in revenue. A suboptimal proposed policy could cause a significant profit loss for the platform and could hurt customer experience and satisfaction. Batch reinforcement learning enables us to ensure the safety of our proposed policy before launching a field experiment. (Note that in practice, one would need to train the model with the newest available data to ensure that the model does not become stale.)

Although batch analysis using fixed historical data sounds like the standard practice in econometric modeling, batch reinforcement learning algorithms have been shown to be susceptible to extrapolation error (Fujimoto et al. 2019), induced by generalization from the neural network function approximator. When selecting action A, such that the pair $(\mathbf{S},A)$ is distant from the data contained in the batch, the estimate $Q(\mathbf{S},A)$ may be arbitrarily poor, introducing extrapolation error. We address this extrapolation error problem by using a Batch Constrained Q-learning algorithm (Fujimoto et al. 2019), which favors a state-action visitation similar to some subset of the provided batch. Specifically, the algorithm will adjust a threshold value τ for a state-action pair and will allow only the actions whose relative probability is above the threshold τ (line 5). The probability of actions in each state is calculated by training a generative model G_ω (lines 2 and 7). We use the GBDT model as G_ω.

[30] Other function approximators such as Chebyshev polynomials were previously used in the dynamic programming literature (Rust 1996). Chebyshev polynomials are preferred for smooth value functions (Cai and Judd 2010), but DNNs can approximate non-smooth functions effectively (Imaizumi and Fukumizu 2019). Recent theoretical work (Fan et al. 2020) has proved the accuracy of the deep learning approximation in reinforcement learning settings.

Algorithm 2 Batch Constrained Q-Learning

1. Input: Batch data $\mathscr{B}$, horizon T, target network update rate Υ, mini-batch size N, threshold τ.
2. Initialize Q-networks Q_θ, generative model G_ω (trained in a standard supervised learning fashion, with cross-entropy loss), and target networks $Q_{\theta'}$,with $\theta' \leftarrow \theta$.
3. for $t = 1$ to T, do:
4. Sample mini-batch M of N transitions $(\mathbf{S}, A, R, \mathbf{S}')$ from $\mathscr{B}$
5. $A' = \arg\max_{A' | \frac{G_\omega(A'|\mathbf{S}')}{\max_{\hat{A}} G_\omega(\hat{A}|\mathbf{S}')} > \tau} Q_\theta(\mathbf{S}', A')$
6. $\theta \leftarrow \arg\min_\theta \sum_{(\mathbf{S},A,R,\mathbf{S}') \in M} l_\kappa (R + \gamma Q_{\theta'}(\mathbf{S}', A') - Q_{\theta'}(\mathbf{S}, A))$
7. $\omega \leftarrow \arg\min_\omega - \sum_{(\mathbf{S},A) \in M} \log G_\omega(A|\mathbf{S})$
8. If $t \bmod \Upsilon = 0$: $\theta' \leftarrow \theta$
9. end for

4.4 Evaluation

4.4.1 In-sample Policy Evaluation

After the model learns a new policy, the next question is whether the policy is effective. Because we have only historical data, we need to solve an in-sample off-policy policy evaluation problem, where the policy to be evaluated is different from the policy used to generate the historical batch data. Formally, given a new policy, π^e, we want to calculate the value function associated with this policy, V^{π_e}:

$$V^{\pi_e} = E_{P,\pi_e} \left[\sum_{t=0}^{T} \delta^t R_t \right] \tag{2}$$

Note that Equation 2 also represents the net present value of a customer or, loosely speaking, the customer lifetime value (CLV, Fader et al. 2005) (associated with coupon redemption) during time frame T. For ease of interpretation, we report the policy performance in terms of the CLV in later sections.

We estimate the value of the new policy using the Doubly Robust Estimator (Dudík et al. 2014). Most studies in the reinforcement learning literature use one of two classes of methods for policy evaluation. One class is the Inverse Propensity Weighting (IPW) method, also known as the Importance Sampling (IPS) method. The idea behind IPS is that, when making counterfactual predictions for a new policy that generates a different action distribution than the distribution observed in the data, we can use propensity scores to reweigh the original observations such that we can use the reweighed observations as a control group for estimating the value of the new policy. In other words, IPS uses importance weighting to correct for the incorrect proportions of actions in the historical data. The second class of approaches for policy evaluation is called the direct method, which uses the historical data to learn the functional mapping between the state-action space and the reward. Then, the estimated reward function can be used to calculate the value function. The Doubly Robust Estimator combines the two methods, so it is unbiased if either the propensity score estimator is correct or the direct method estimator is correct (i.e., the two methods do not have to be correct at the same time).

Mathematically, the Doubly Robust Estimator is

$$V^{\pi_e} = E_n\left[\sum_{t=0}^{T-1}\delta^t\left(\underbrace{\xi_{0:t}\left[R_t - m_t\left(\mathbf{S}_t, A_t\right)\right]}_{\text{Importance Sampling: low bias}} + \underbrace{\xi_{0:t-1}\left\{\Sigma_{A\in\mathscr{A}} m_t\left(\mathbf{S}_t, A\right)\pi_e\left(A|\mathbf{S}_t\right)\right\}}_{\text{Direct Method: low variance}}\right)\right] \tag{3}$$

where $\xi_{0:t} = \prod_{\tau=0}^{t}\frac{\pi_e(A_\tau|\mathbf{S}_\tau)}{\pi_b(A_\tau|\mathbf{S}_\tau)}$ are the inverse propensity weights, and $\pi_b\left(A_\tau|\mathbf{S}_\tau\right)$ is the behavioral policy. In our case, the behavioral policy is known because the platform used a random allocation policy with pre-determined action probabilities. $m_t\left(\mathbf{S}_t, A_t\right)$ is the direct method estimator, and for the functional form, we chose GBDT (Chen and Guestrin 2016) in which the Q value is the dependent variable and the state action variables $(\mathbf{S}, A)$ are the independent variables. When the behavioral policy is known, both the IPS and the Doubly Robust methods produce unbiased evaluations, but the Doubly Robust Estimator has lower variance than IPS (Dudík et al. 2011). We use the Doubly Robust Estimator in our application to compare BDRL against a set of benchmarks, introduced in §5.

4.4.2 Field Experiment

We also provide out-of-sample evaluation of the new policy using a field experiment, in which we divide all the users into conditions with different coupon targeting policies: a fully random allocation policy in the control condition, the policy learned using the BDRL algorithm in one treatment condition, and a benchmark policy in another treatment condition.

5 Benchmark Policies

Now we introduce the benchmark algorithms. Table 6 presents the design matrix, which has three components: whether the objective is to design a static or dynamic targeting policy, whether the treatment is homogeneous or heterogeneous across consumers, and whether the solution is model-based or model-free. To generate the static homogeneous treatment policy benchmark, we use a linear regression without the interaction between states and actions (§5.1). We generate the static heterogeneous treatment policy benchmarks using three alternative models: GBDT, DNN, and orthogonal random forest (ORF). We generate the dynamic heterogeneous treatment policy using a structural approach (§5.3). Finally, BDRL is the model-free dynamic heterogeneous treatment approach (§4.2).

Table 6: Benchmark Models

Treatment	Model	Objective	
		Static	Dynamic
Homogeneous		Regression w/o interaction (§5.1)	
Heterogeneous	Model-based	GBDT w/ interaction (§5.2)	
		DNN w/ interaction (§5.2)	
		ORF (§5.2)	Structural (§5.3)
	Model-free		BDRL

5.1 Static Policy with Homogeneous Treatment

Our first benchmark is the static targeting policy with homogeneous treatments across consumers. We use a linear regression model without the interaction between states and actions (Equation 4). The dependent variable is the revenue from one coupon reception incidence, and the independent variables are states and actions. The model estimates the average treatment effect of each coupon (action A) and chooses the optimal coupon, that is, the coupon that generates the highest predicted reward, $\hat{R}(\mathbf{S},A)$ (Equation 5). Although states are used to estimate the model coefficients, states do not affect the choice of the optimal action because there is no interaction between states and actions in Equation 4. In fact, the optimal action is the same across all states. In other words, all consumers will receive the same coupon, which is chosen to maximize revenue.

$$R_{it} = \beta_0 + \mathbf{S}_{it}\boldsymbol{\beta}_s + \beta_A A_{it} + \varepsilon_{it}, \varepsilon_{it} \sim N\left(0,\sigma^2\right) \tag{4}$$

$$\pi(A|\mathbf{S}) = \begin{cases} 1 & A = \arg\max_{a\in\mathbb{A}} \hat{R}(\mathbf{S},A) \\ 0 & o.w. \end{cases} \tag{5}$$

5.2 Static Policy with Heterogeneous Treatments

The homogeneous treatment approach is hardly optimal because it assumes that one coupon is most effective for every consumer in every context. By contrast, static targeting policies that allow for heterogeneous treatments recognize differences among consumers, though these policies still optimize for each period separately. This type of static targeting strategy is prevalent in the industry and can be categorized into two groups: indirect or direct methods. Most indirect methods are machine learning algorithms that use the state variables and actions as inputs to predict the reward output. After the model has been trained, a post-selection exercise loops over all the actions and picks the one that is predicted to generate the highest reward. We use GBDT, which is used by Facebook (He et al. 2014), and DNN as the indirect method benchmarks. The direct methods[31] were developed more recently in the causal inference literature. In contrast to indirect

[31] Here, "direct methods" belong to a different class of treatment estimation algorithms than the policy evaluation methods in §4.4.1.

methods, direct methods calculate the heterogeneous treatment effect of each action and identify the optimal action as the one with the highest treatment effect. We choose the ORF model (Oprescu et al. 2018) as the direct method benchmark because it is less sensitive than others with respect to the estimation error of nuisance parameters. In the extant literature, it is unclear how direct methods perform compared to indirect methods, so we empirically test both.

5.3 Dynamic Policy with Heterogeneous Treatments: Structural Model

Model-based dynamic algorithms, unlike the methods in §5.2, consider the intertemporal tradeoffs created by targeting policies. We take a structural approach in which we construct a function f for the reward (equation 6) and a function g for the state transition process (equation 7).

Both the reward function and the state transition function depend on the consumer decision process. As explained in Fig.7, each period t comprises three stages. In Stage 1, the consumer makes a churn decision (i.e., whether to continue using the platform or to leave permanently); if the consumer chooses to continue using the platform, she starts the search process. The churn and search decisions jointly determine the state variables $\mathbf{S}$, which comprise the consumer, seller, product, and webpage features. In Stage 2, the platform decides which action to take based on a policy function $\pi(\mathbf{S})$. In Stage 3, the consumer makes the purchase decision based on the state and action she observes. The purchase decision results in a reward for the platform, and the reward function is specified as $R = f(\mathbf{S}, A)$. This concludes period t. In the next period $(t+1)$, the consumer again makes the churn and search decisions, which jointly determine the new state $\mathbf{S}'$. The state transition is specified by function g, so $\mathbf{S}' = g(\mathbf{S}, A)$.

The existing marketing literature has proposed many models for the search, purchase, and churn behaviors. In Appendix G, we present the sequential search model in a full structural fashion.[32] We also consider a simplified version in which both the f and g functions are formulated as GBDT models (equation 6 and 7), which are more flexible than the full structural model. Once we fit the reward function ($f(\mathbf{S}, A)$) and the state transition function ($g(\mathbf{S}, A)$), we use backward induction to calculate the Q function (i.e., the policy-specific value function, equation 8). The optimal policy is the one that maximizes the Q function (equation 9).

[32] Moreover, to account for the reference price effect, we follow the literature (Bell and Lattin 2000) and add two additional state variables in this model that represent the gain and loss effects of the reference price.

$$\hat{R}_{it}\left(\mathbf{S_{it}}, A_{it}\right) = \sum_{k=1}^{K} f_k\left(\mathbf{S_{it}}, A_{it}\right), f_k \in \mathscr{F} \tag{6}$$

$$\mathbf{S}_{it+1} = \sum_{l=1}^{l} g_l\left(\mathbf{S_{it}}, A_{it}\right), g_l \in \mathscr{G} \tag{7}$$

$$Q\left(\mathbf{S}, A\right) = \hat{E}\left[\hat{R}_A + \delta \max_{A'} Q\left(\mathbf{S}', A'\right) | \mathbf{S}\right] \tag{8}$$

$$\pi\left(A|\mathbf{S}\right) = \begin{cases} 1 & A = \arg\max_{A \in \mathbb{A}} \hat{Q}\left(\mathbf{S}, A\right) \\ 0 & o.w. \end{cases} \tag{9}$$

6 Empirical Application and Results

6.1 Empirical Application

This section discusses the empirical application of the modeling framework presented in §4. First, in §6.1.1 and §6.1.2, we introduce the primitives of the MDP (namely, the states and action) in our empirical context of livestream shopping. Then, in §6.1.3, we present the train-test split and hyperparameters.

6.1.1 States

States can be categorized into four groups: consumer, seller (host), product, and webpage characteristics. We provide the specific variables used in the livestream shopping setting and their summary statistics in Appendix H. Taobao keeps a complete history of every customer's interactions with the platform, so we can precisely calculate the state variable values from the beginning of the sample period without an initial condition problem (Heckman 1981).

6.1.2 Action

As explained in §3.3, we discretize the action space into 25 coupons, each characterized by two metrics, the discount ratio and the threshold ratio, with 5 levels each based on the quintiles in their respective distributions. Importantly, the action set does not change over time and can be set ex-ante, exogenously. Although the cutoff points depend on the data, which may vary over time, the entire support of the two dimensions can be fully covered by the five levels. For instance, for the threshold ratio, the HH level spans 70% to infinity; for the discount ratio, the HH level spans 70% to 100%. We chose the two dimensions carefully, with sufficiently broad numerical support to accommodate the anticipated range of experimentation. For example, the largest threshold ratio observed in our data is 300%, but if a coupon with a 400% threshold ratio is required in the future, it would still be categorized as the HH type.

6.1.3 Train-Test Split

We need to train a reward prediction model and then derive the corresponding policy for the three benchmark models: the linear regression model (§5.1), GBDT, and DNN (§5.2). To assess the prediction accuracy of the three models, we create the training and test sets with an 80-20 split (i.e., the training set contains 80% of the consumers, and the test set contains the other 20%). All observations from the same consumer belong to the same subset. We provide the model prediction accuracy in Appendix I and the hyperparameters for the different estimation approaches in Appendix J.

6.2 Policy Evaluation

We measure the 'in-sample" performance of the dynamic coupon targeting policy using the Doubly Robust Estimator (Equation 3) introduced in §4.4.1.

Table 7: Model Comparison Based on the Doubly Robust Estimator

		1. Static Policy with Homogeneous Treatments	2. Static Policy with Heterogeneous Treatments			3. Model-Based Dynamic Policy with Heterogeneous Treatments	4. Model-Free Dynamic Policy with Heterogeneous Treatments
		A: Regression	B: GBDT	C: DNN	D: ORF	E: Structural[33]	F: Proposed BDRL
CLV (Return)	Mean	6.53	7.52	6.95	7.49	8.37	9.53
	Std	2.00	2.29	2.27	2.29	2.77	3.23
Gain	Mean	11%	28%	18%	28%	43%	62%
T-test[34]		222.99	524.25	346.60	515.47	714.27	945.57
P_value		<0.001	<0.001	<0.001	<0.001	<0.001	<0.001

Note: All the gains are compared to the mean CLV of ¥5.87. The total sample size is 1,020,898 consumers.

Table 7 compares the CLV estimates from our model-free dynamic targeting policy (BDRL) with those from the benchmark policies. The mean CLV observed in the data is ¥5.87;[35] the value is relatively small because only 6% of the incidences result in a purchase (i.e., 94% of the incidences have a reward of 0). The static policy with homogeneous treatments, estimated using linear regression, increases the CLV by 12% compared to the original coupon allocation rule, which was purely random without any optimization. The three static policies with heterogeneous treatments, estimated using GBDT, DNN, and ORF, all increase

[33] The result is for the full structural model. The result from the simplified version is qualitatively similar and available upon request.

[34] The t-tests compare the mean CLV of different policies. Appendix P reports distributional differences measured by the Kolmogorov-Smirnov test.

[35] To protect the privacy of the platform, the amounts have been changed by an affine transformation.

the CLV by around 20%; the improvement over the static policy with homogeneous treatments is evidence of the importance of personalization. Interestingly, our results corroborate existing findings that there is no strict hierarchy in the relative effectiveness of indirect methods (GBDT and DNN) and direct methods (ORF). The dynamic policy with heterogeneous treatments, estimated using a structural model, increases the CLV by 43%, indicating that demand dynamics play an important role in coupon effectiveness, so it is important to account for how coupon targeting strategies affect consumers over time. Finally, the dynamic targeting policy generated using BDRL increases the CLV by 63%, which translates to an increase of $13.1 million (91.8 million RMB) from the consumers in our sample and $18 billion (126 billion RMB) from the entire Taobao Live customer base during the three-month sample period.[36]

6.3 Field Experiment

Table 8: Field Experiment Results

		Random Allocation	Model-Based Dynamic Policy With Heterogeneous Treatments (Structural)	Model-Free Dynamic Policy With Heterogeneous Treatments (BDRL)
CLV	Mean	6.98	9.70	11.16
(Return)	Std	2.43	3.15	3.86
Gain	Mean	~	+39%	+60%
T-test		~	690.80	925.95
P_value		~	<0.001	<0.001

We further test the performance of the new policy using an "out-of-sample" field experiment on the Taobao Live platform. The consumers who participated in the field experiment are the same as those in our batch data (1,020,898 consumers). To minimize the risk of losing revenue, we test only three coupon allocation strategies: (1) random allocation, (2) model-based dynamic targeting based on the structural model, and (3) model-free dynamic targeting based on BDRL.[37] Consumers in our sample were assigned to conditions (1), (2), or (3) with probabilities of 80%, 10%, and 10%, respectively.[38] The experiment lasted two weeks in January 2020. Table 8 reports the results. The return value (total discount rewards during the experiment

[36] Caveat: our data includes only active users, defined as at least 10 coupon reception incidences in the three-month sample period. If the proposed policy is more effective for active users than for the average customer, then the projected increase in the GMV would be an upper bound. The projection to the entire Taobao Live customer base is based on the reported annual sales of $28 billion (200 billion RMB) in 2019. Pak J. (2020, April 6). Retrieved from: https://https://bit.ly/3ANCY5i

[37] We assume that BCQ finished learning the optimal coupon allocation policy after training on the historical batch data. It is possible, however, that the learning process is incomplete, and as new data come in, parameters in the model will keep updating. Future research could explore another experimental condition with an online reinforcement learning algorithm (e.g., DQN) where the initial values of parameters come from the converged parameter values in the batch mode (e.g., BCQ) and continue updating in each iteration, as new data arrive.

[38] The number of consumers in each condition is 816,718, 102,090, and 102,090, respectively.

period, or CLV) is rescaled for privacy protection purposes. As shown, the model-based dynamic targeting policy (structural model) was 39% more effective than the random allocation policy, confirming the importance of incorporating demand dynamics and using optimization to determine the best coupons. The model-free dynamic targeting policy (BDRL) performed even better, with a 60% increase in the CLV. In Fig.9, we plot the histogram of the CLV gain (the percentage increase in the CLV from random allocation to the BDRL policy). The distribution has a log-normal shape with a mean of 60%. For some consumers, the CLV gain is as large as 800%.

The model-free approach outperforms the model-based approach by avoiding model bias. We note that we were able to utilize the model-free approach in this case because two conditions were met: data from the firm's previous randomized coupon experiments provided sufficient stochasticity such that we do not need to rely on functional form assumptions, and the empirical evidence suggests an absence of structural shifts in consumer behaviors (e.g., forward-looking) under alternative policies. As such, whether our model-free BDRL approach or a structural model approach would provide better results depends on the specifics of the setting. However, when large amounts of randomized historical data are available and structural shifts in consumer preferences are unlikely, our BDRL approach provides substantial advantages.

Tables 8 and 7 show that the findings from the field experiment are consistent with the policy evaluation results. For cross-reference, in Fig.10, we plot the predicted CLV gain of each consumer using the Doubly Robust Estimator against the actual CLV gain in the field experiment. The predictions visually align with the realized values.

Figure 9: Histogram of the CLV Gain

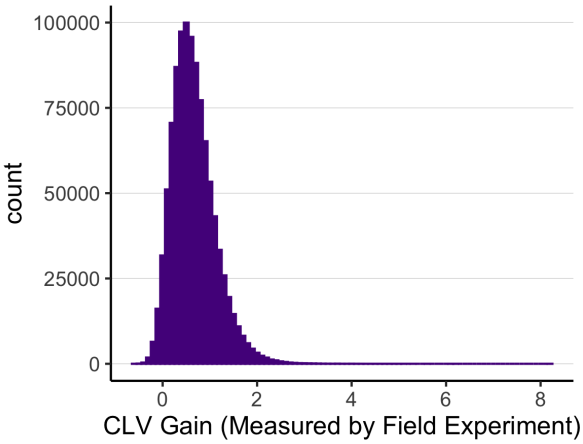
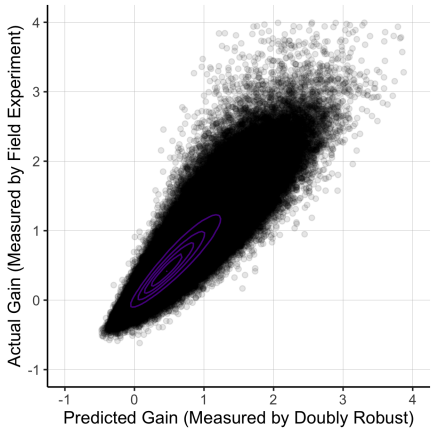


Figure 10: Actual vs. Predicted CLV Gain (%)



The dynamic targeting policy generated using our BDRL approach performs the best out of the three strategies tested in the field experiment, but we warn readers not to over-generalize this finding because of multiple

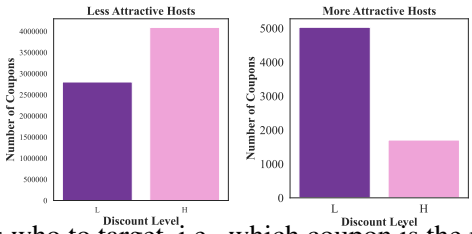
caveats. First, our policy change did not "systematically alter the structure" (Lucas 1976), so structural models might not have clear advantages in our setting. It is possible that our model's superiority comes from the similarity between the environment in the field experiment and the environment in which we collected the estimation data. If the estimation data had been collected under an alternative pricing policy, then the model-free BDRL approach might not have done better. Moreover, the experiment was a very short-run intervention during which consumers might not have had time to adjust their preferences or behaviors. Another caveat is that we tested BDRL against only two alternatives (the structural model policy and random allocation). We encourage future research to consider other policies such as static targeting (with either indirect or direct methods), which is widely used in the e-commerce and advertising industry, or a homogeneous dynamic policy that includes only intertemporal price discrimination, such as markdown pricing or markup pricing (Su 2007). Future research may also consider alternative structural model specifications (see one example that does not model search behaviors in Appendix L).

6.4 Targeting Rules

This section provides substantive insights into the targeting rules created by BDRL. The next three subsections describe how our targeting rules can determine who to target (heterogeneity), when to target (dynamics), and the interaction between who and when to target.

6.4.1 Who To Target (Heterogeneity)

Figure 11: Targeting Rule Under BDRL: Host Attractiveness



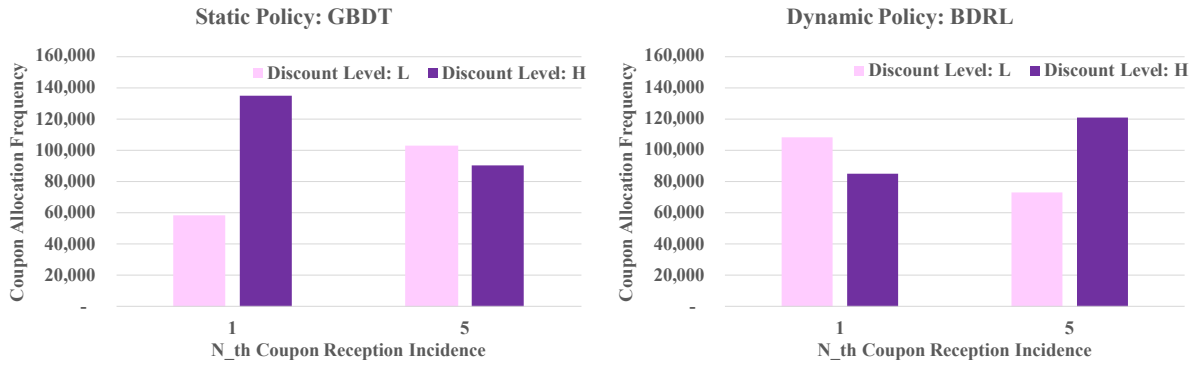
Our targeting policy recommends who to target, i.e., which coupon is the most effective for each consumer, given the consumer's context. One example of a context variable that is unique to livestream shopping is the host's attractiveness. Fig.11 displays the coupon frequency by the discount level and host attractiveness. The left panel is for less-attractive hosts (below the median of attractiveness), and the right panel is for more-attractive hosts. Our targeting policy recommends sending more high-discount (low-discount) coupons to consumers who are watching livestream content created by less-attractive hosts (more-attractive hosts), perhaps reflecting a compensatory effect in which consumers derive more utility from attractive hosts and

hence demand less monetary reward.

6.4.2 When To Target (Dynamics)

Our targeting policy also determines when to target, which can help us understand why the BDRL algorithm performs better than the static benchmarks. We plot the action distributions under the two policies in Fig.12: the static policy (GBDT) in the left plot and the dynamic policy (BDRL) in the right plot. The horizontal axis is the N_th coupon reception incidence. We consider a new customer (incidence N = 1) versus a loyal customer (N = 5). For new customers, GBDT recommends more high-discount coupons, while BDRL recommends more low-discount coupons. More generally, BDRL recommends giving increasing discounts over time, consistent with the reference price effect: if the platform gives high-discount coupons to new consumers, they may use the highly discounted prices as references in the future, making them less likely to respond to future incentives. The dynamic policy appears to recognize the long-term negative consequence of providing high-discount coupons to new consumers and thus adopts a more conservative approach of gradually increasing the discount level as consumers become more loyal.[39]

Figure 12: A Comparison of Static and Dynamic Targeting Policies



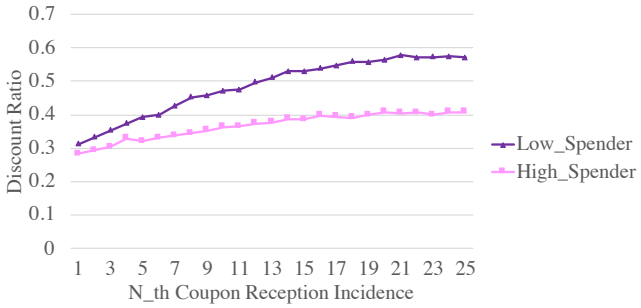
6.4.3 When To Target Who

Finally, our targeting policy recommends combining cross-sectional and intertemporal price discrimination, i.e., modifying the coupon allocation strategy based on the consumer's experience on the platform. Fig.13 plots the average discount ratio by the coupon reception incidence (from the 1st to the 25th incidences) for two segments of consumers: high spenders (square symbol) and low spenders (triangle symbol). The policy

[39] One concern about a policy that increases the discount level with loyalty is that the discount level has an upper bound (HH). It is unclear how to solve the dynamic targeting problem once the upper bound is reached. It is possible that there is a steady state at the upper bound such that the platform can achieve stable revenue with HH coupons. During our short sample period of three months, only 1.18% of the consumers reached the steady state, defined as receiving HH for at least two consecutive periods). We do not have sufficient data to study consumer behaviors after the upper bound is reached, so we leave it for future research.

recommends giving low spenders (relative to high spenders) slightly higher discounts initially and increasing the discount at a faster rate as the consumers gain experience. This coupon policy may be responding to higher price-sensitivity among low spenders than high spenders, providing larger and larger discounts to these customers to keep them continuously engaged.

Figure 13: Targeting Rule: When To Target Who



7 Conclusion and Future Directions

We design a dynamic theoretical framework, batch deep reinforcement learning (BDRL), which incorporates the intertemporal tradeoffs in dynamic pricing and coupon targeting to create a policy that maximizes a platform's revenue. We empirically evaluate the performance of the dynamic coupon targeting policy relative to diverse benchmarks: based on static and dynamic objectives, using model-based and model-free estimation, and recommending heterogeneous and homogenous treatments across consumers. Using a large-scale field experiment, we demonstrate that our BDRL approach can increase the livestream shopping platform's GMV by over 60%—far superior to the approximately 30% increase from static targeting policies and the 40% increase from a dynamic structural model-based policy. Our BDRL framework is suitable for the high-frequency and high-dimensional pricing problems that are common in e-commerce settings.

While our findings illustrate that BDRL can be effective at solving dynamic pricing and coupon targeting problems, we acknowledge several limitations that open the door for future research.

First, we assume that consumers are not forward-looking because we did not observe evidence of forward-looking behaviors in our historical data. However, such behaviors could emerge in other settings. What would be the optimal pricing strategy for forward-looking consumers? Future research can test a number of predictions from the theoretical literature on dynamic pricing, as we discuss in Appendix D.3.

Another limitation of our framework is the lack of unobserved heterogeneity, i.e., consumers' heterogeneous preferences and sensitivities to coupons beyond the observed features. Our data covers millions of consumers but for only a limited time (three months), so the identification relies heavily on pooling across

customers and the assumption of no unobserved heterogeneity.[40] Future research could explore two solutions: formulate the problem as a partially observable MDP (Hauser et al. 2009), or use multi-agent reinforcement learning and treat each consumer segment as a separate agent.

One other limitation in our study is sample selection because we examine only "active" users. Although we conducted a sensitivity test in Appendix M, it would be interesting to explore whether our conclusions generalize to less-active or inactive users, where the zero-inflated nature of the purchase frequency distribution could make the estimation challenging and require new methodological solutions.

Future research could also explore demand dynamics on a longer horizon. It is possible that the loyalty effect or variety-seeking effect would dominate the reference price effect in the long run.

Finally, due to data limitations, we could not directly incorporate unstructured data, such as image, audio, and video features, as state variables. Instead, we used summary statistics that were pre-defined by the company. However, livestream shopping involves an information-rich environment, so we hope that future research will test the informational values of both structured and unstructured data.

We see BDRL as a powerful framework for solving many marketing problems associated with sequential decision making, such as ad targeting, chatbot conversation design, recommender systems, and dynamic targeted pricing in both online and offline settings. We encourage future marketers to experiment with and apply BDRL.

References

Eva Ascarza, Oded Netzer, and Bruce GS Hardie. Some customers would rather leave without saying goodbye. *Marketing Science*, 37(1):54–77, 2018.

David R Bell and James M Lattin. Looking for loss aversion in scanner panel data: The confounding effect of price response heterogeneity. *Marketing Science*, 19(2):185–200, 2000.

Dimitri Bertsekas. *Reinforcement learning and optimal control*. Athena Scientific, 2019.

Yongyang Cai and Kenneth L Judd. Stable and efficient computational methods for dynamic programming. *Journal of the European Economic Association*, 8(2-3):626–634, 2010.

Jean-Pierre Dubé and Sanjog Misra. Personalized pricing and customer welfare. *Available at SSRN 2992257*, 2019.

[40] Similarly, Dubé and Misra (2019) assumed that heterogeneity in customers' price sensitivities can be characterized by an observed, high-dimensional vector containing a sparse subset of observable customer characteristics.

Jean-Pierre Dubé, Günter J Hitsch, and Peter E Rossi. State dependence and alternative explanations for consumer inertia. *The RAND Journal of Economics*, 41(3):417–445, 2010.

Miroslav Dudík, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 1097–1104. Omnipress, 2011.

Miroslav Dudík, Dumitru Erhan, John Langford, Lihong Li, et al. Doubly robust policy evaluation and optimization. *Statistical Science*, 29(4):485–511, 2014.

Peter S Fader, Bruce GS Hardie, and Ka Lok Lee. Rfm and clv: Using iso-value curves for customer base analysis. *Journal of Marketing Research*, 42(4):415–430, 2005.

Jianqing Fan, Zhaoran Wang, Yuchen Xie, and Zhuoran Yang. A theoretical analysis of deep q-learning. In *Learning for Dynamics and Control*, pages 486–489. PMLR, 2020.

Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.

Scott Fujimoto, Edoardo Conti, Mohammad Ghavamzadeh, and Joelle Pineau. Benchmarking batch deep reinforcement learning algorithms. *arXiv preprint arXiv:1910.01708*, 2019.

Jason Furman, Diane Coyle, Amelia Fletcher, Derek McAules, and Philip Marsden. Unlocking digital competition: Report of the digital competition expert panel. *Report prepared for the Government of the United Kingdom, March*, 2019.

Füsun F Gönül, Byung-Do Kim, and Mengze Shi. Mailing smarter to catalog customers. *Journal of Interactive Marketing*, 14(2):2–16, 2000.

John R Hauser, Glen L Urban, Guilherme Liberali, and Michael Braun. Website morphing. *Marketing Science*, 28(2):202–223, 2009.

Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, et al. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, pages 1–9, 2014.

James Heckman. J. 1981, the incidental parameters problem and the problem of initial conditions in estimating a discrete time-discrete data stochastic process, 1981.

V Joseph Hotz and Robert A Miller. Conditional choice probabilities and the estimation of dynamic models. *The Review of Economic Studies*, 60(3):497–529, 1993.

Masaaki Imaizumi and Kenji Fukumizu. Deep neural networks learn non-smooth functions effectively. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 869–878, 2019.

Abel P Jeuland. Brand choice inertia as one aspect of the notion of brand loyalty. *Management Science*, 25 (7):671–682, 1979.

Barbara E Kahn, Manohar U Kalwani, and Donald G Morrison. Measuring variety-seeking and reinforcement behaviors using panel data. *Journal of Marketing Research*, 23(2):89–100, 1986.

Minkyung Kim, K. Sudhir, and Kosuke Uetake. A structural model of a multitasking salesforce: Multidimensional incentives and plan design. *Working paper*, 2021.

Robert E. Lucas. Econometric policy evaluation: A critique. *Carnegie-Rochester Conference Series on Public Policy*, 1:19–46, 1976. ISSN 0167-2231. doi: https://doi.org/10.1016/S0167-2231(76)80003-6. URL https://www.sciencedirect.com/science/article/pii/S0167223176800036.

John Joseph McCall. Economics of information and job search. *The Quarterly Journal of Economics*, pages 113–126, 1970.

Kanishka Misra, Eric M Schwartz, and Jacob Abernethy. Dynamic online pricing with incomplete information using multiarmed bandit experiments. *Marketing Science*, 38(2):226–252, 2019.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

Miruna Oprescu, Vasilis Syrgkanis, and Zhiwei Steven Wu. Orthogonal random forest for causal inference. *arXiv preprint arXiv:1806.03467*, 2018.

Kharan N Rajendran and Gerard J Tellis. Contextual and temporal components of reference price. *Journal of marketing*, 58(1):22–34, 1994.

Eddie Rhee and Gary J Russell. Forecasting household response in database marketing: A latent trait approach. In *Advances in business and management forecasting*. Emerald, 2009.

Peter E Rossi, Robert E McCulloch, and Greg M Allenby. The value of purchase history data in target marketing. *Marketing Science*, 15(4):321–340, 1996.

John Rust. Numerical dynamic programming in economics. *Handbook of Computational Economics*, 1: 619–729, 1996.

PB Seetharaman and Hai Che. Price competition in markets with consumer variety seeking. *Marketing Science*, 28(3):516–525, 2009.

PB Seethu Seetharaman. 17 dynamic pricing. *Handbook of Pricing Research in Marketing*, page 384, 2009.

Stephan Seiler. The impact of search costs on consumer behavior: A dynamic approach. *Quantitative Marketing and Economics*, 11(2):155–203, 2013.

Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An introduction*. MIT Press, 2018.

UK Competition and Markets Authority. Pricing algorithms: Economic working paper on the use of algorithms to facilitate collusion and personalised pricing. 2018.

Glen L Urban, Guilherme Liberali, Erin MacDonald, Robert Bordley, and John R Hauser. Morphing banner advertising. *Marketing Science*, 33(1):27–46, 2013.

Harald J Van Heerde and Scott A Neslin. Sales promotion models. In *Handbook of Marketing Decision Models*, pages 13–77. Springer, 2017.

Christopher John Cornish Hellaby Watkins. Learning from delayed rewards. 1989.

Hong Wen, Jing Zhang, Quan Lin, Keping Yang, and Pipei Huang. Multi-level deep cascade trees for conversion rate prediction in recommendation system. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 338–345, 2019.

Russell S Winer. A reference price model of brand choice for frequently purchased products. *Journal of Consumer Research*, 13(2):250–256, 1986.

Qiang Zhang, Wenbo Wang, and Yuxin Chen. In-consumption social listening with moment-to-moment unstructured data: the case of movie appreciation and live comments. *Marketing Science*, 2019b.

Online Appendix

A Exogenous Search

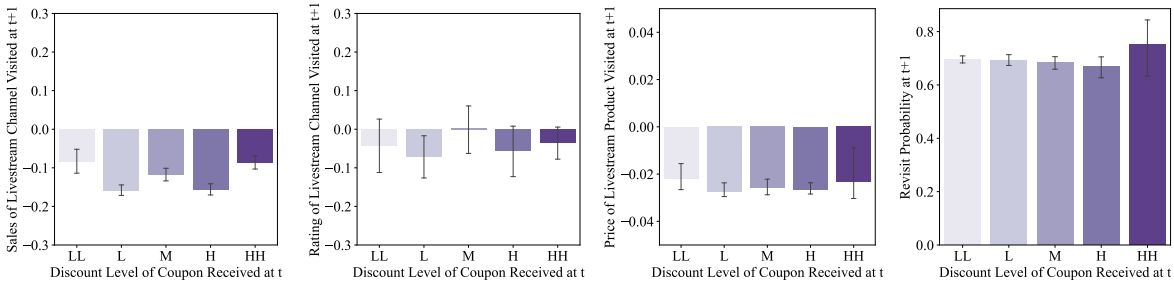
Our current setting assumes that consumers' search behaviors are independent of the platform's coupon allocations. In this section, we provide empirical evidence of this assumption. Fig.14 plots the relationship between the discount level of the coupon received by a consumer in purchase incidence t and her subsequent search behavior in incidence $t+1$.[41] Specifically, we consider four search behaviors that could plausibly be affected by the previous coupon allocation: the seller's sales, the seller's rating, the average price of the products in the channel, and the probability that the consumer has visited the channel before (i.e., the revisit probability). For each behavior, we conduct an analysis of variance (ANOVA).

In panel (a), we find no significant difference in the sales of the seller visited in incidence t+1 based on the discount level of the coupon received in incidence t (F-stat = 1.491, p_value = 0.202). In other words, a consumer who receives a big discount now is not more or less likely to seek out a channel with high or low sales in the next search. We find analogous results in the remaining panels. In panel (b), we find no significant difference in the average rating of the seller visited in incidence t+1 based on the discount level of the coupon received in incidence t (F-stat = 1.643, p_value = 0.160). In panel (c), we find no significant difference in the average price of the product considered in incidence t+1 based on the discount level of the coupon received in incidence t (F-stat = 2.016, p_value = 0.089). In other words, there is no evidence that consumers switch to highly (or poorly) rated sellers or to low-priced (or high-priced) products after receiving a deep discount. Finally, we test whether consumers are more likely to revisit a particular host after receiving a high-discount coupon from this host. In panel (d), we find no significant difference in the likelihood that the consumer is revisiting a channel in incidence t+1 based on the discount level of the coupon received in incidence t (F-stat = 1.369, p_value = 0.242). The revisit probability is always around 70%.

[41] We fix the coupon's threshold level at M to ensure a fair comparison. This rule applies to all the subplots in Fig.14.

Figure 14: Evidence of Consumer Exogenous Search Behavior

(a) Seller Sales_t+1 vs. Coupon Discount Level_t (b) Seller Rating_t+1 vs. Coupon Discount Level_t (c) Product Price_t+1 vs. Coupon Discount Level_t (d) Revisit_t+1 vs. Coupon Discount Level_t



We acknowledge that consumers' search behaviors could be influenced by coupon allocations in other marketing contexts, in which case the researchers could re-categorize the state variables associated with the host, product, and livestream to be dynamic instead of static.

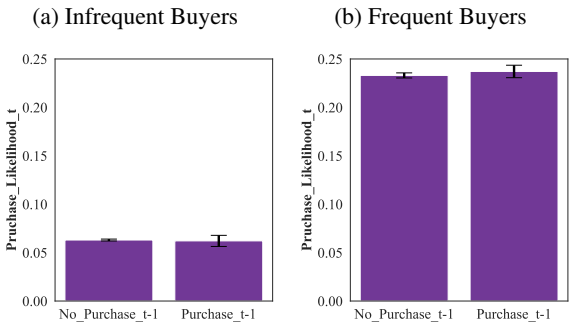
B Disentangling Unobserved Heterogeneity and State Dependence

We address a concern about the test for state dependence: if the conditional purchase probability is higher than the marginal purchase probability, it may reflect not structural state dependence or inertia but rather the confounds of unobserved heterogeneity. That is, consumers may differ along a serially correlated, unobserved propensity to make purchase decisions (Heckman 2007). We use the method proposed in Dubé et al. (2010) to tease apart state dependence from unobserved heterogeneity. Specifically, we rely on spells during which the consumer first made a purchase initiated by a discount (defined as a purchase for which the consumer redeemed a coupon with the highest discount level, HH) and then continued to purchase after prices returned to "typical" levels (a coupon with discount level L or lower). We compare this repeat-purchase rate with the marginal purchase probability, and we find no significant difference (10.86% vs. 10.79%, p_value = 0.65). We conclude that state dependence is absent even after controlling for unobserved heterogeneity.

C Additional Evidence of the Absence of Variety-Seeking

Although we did not find evidence of variety-seeking in our initial analysis, it is possible that variety-seeking behavior occurs only among heavy repurchasers. To test this hypothesis, we recreate the variety-seeking plot with two groups of consumers: infrequent buyers (< 5 purchases in the sample period) and frequent buyers (at least 5 purchases in the sample period). Fig.15 shows that the purchase likelihood is not negatively influenced by prior purchase experience for either group of consumers, indicating an absence of variety-seeking behavior.

Figure 15: Coupon Redemption Rate by Prior Purchase Experience and Purchase Frequency



D Forward-Looking Behavior

D.1 Evidence of the Absence of Consumer Forward-Looking Behaviors

To assess whether consumers in our sample are forward-looking, we compare the purchase frequency under three scenarios: (1) the consumer receives coupons with a stable discount level over time, (2) the consumer receives coupons with an increasing discount level over time, and (3) the consumer receives coupons with a decreasing discount level over time. If a forward-looking consumer anticipates a deeper discount (i.e., price reduction) in the future, she may strategically wait to make a purchase, leading to a lower initial purchase frequency than a consumer who receives the same discount every time (Su 2007). On the other hand, if a forward-looking consumer anticipates a smaller discount (i.e., price surge) in the future, she may accelerate purchases now to stockpile (Neslin et al. 1985).

We formally test for strategic waiting and stockpiling behaviors in Table 9. Because the purchase frequency is heavily influenced by the discount level, as demonstrated in Fig.2a, we conduct separate analyses at three of the five discount levels (based on the first coupon received): L, M, and H. The purchase time (mean and standard deviation) for the three scenarios are displayed in Table 9. For example, when the starting discount level is L and the discounts are increasing (i.e., scenario 2), the average purchase frequency in the beginning (when consumers receive the initial, low-discount coupon) is 0.047.[42] When consumers receive stable, low-level coupons (i.e., scenario 1), the average initial purchase frequency is 0.043, which is not significantly different from the purchase frequency in scenario 2 (t-test statistic = 0.389, p-value = 0.697). Therefore, we find no evidence of strategic waiting. Similarly, when the starting discount level is L and discounts are decreasing (i.e., scenario 3), the average initial purchase frequency is 0.033, which also is not significantly different from that in scenario 1 (p_value = 0.294), suggesting an absence of stockpiling. We find the same null results with the starting discount levels of M and H.

[42] A purchase frequency of 0.043 means 4.3 purchases for every 100 incidences.

Table 9: Evidence of No Forward-Looking Behaviors

Starting Discount Level	Purchase Frequency at the Starting Discount Level	Discount Level Trend Over Time		
		(1) Stable	(2) Increasing	(3) Decreasing
L	Mean	0.043	0.047	0.033
	Standard Deviation	0.079	0.119	0.088
	T-test (vs. Stable)		0.389	-1.050
	P-value (vs. Stable)		0.697	0.294
H	Mean	0.111	0.112	0.093
	Standard Deviation	0.120	0.286	0.255
	T-test (vs. Stable)		0.033	-0.509
	P-value (vs. Stable)		0.974	0.612

The results in Table 9 suggest that consumers in our context are myopic rather than forward-looking; they do not engage in strategic waiting or stockpiling based on assumptions about future prices. The absence of forward-looking behaviors may be attributable to characteristics of Taobao Live as a livestream shopping platform, where the best-selling product categories (apparel, cosmetics, and jewelry) are highly seasonal, so consumers might not want to delay purchases (and miss the fashion trend) or stockpile products (which soon would be obsolete).

Although the consumers in our data do not seem to be forward-looking, we acknowledge the general importance of incorporating consumer strategic behaviors into the designs of dynamic coupon targeting policies. Indeed, in many marketing contexts, it is easier for consumers to learn and form expectations about price or quality. Next, we discuss how to extend our reinforcement learning model to such contexts.

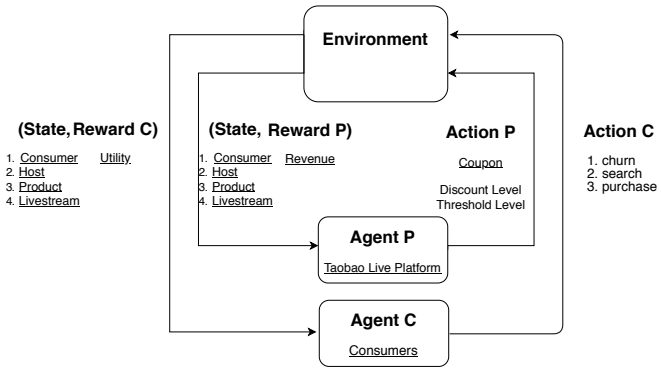
D.2 Multi-Agent Reinforcement Learning

To accommodate consumer forward-looking behaviors, future research could explore multi-agent reinforcement learning (Littman 1994), which allows consumers and the platform to be independent agents. Each agent maximizes its own total discounted rewards, and the reward function of each agent depends on the actions of all agents. In this section, we briefly define multi-agent reinforcement learning and explain how to use it in our context. We leave the solution and estimation for future research.

We use Fig.16 to illustrate how a Markov game can solve the dynamic targeting problem. There are two agents, the platform P and consumers C. At time t, when a consumer enters a livestream channel, the platform agent takes an action: it chooses a coupon for the consumer in this purchase incidence. After observing the platform's action, the consumer also takes an action: she decides whether to purchase and whether to churn after this incidence. The joint actions of the platform and the consumer determine the reward for each (revenue for the platform; utility from the purchase incidence for the consumer). Then, the environment evolves to time $t+1$, and both the platform and consumer observe the next state, which

includes features of the consumer, host, product, and livestream.

Figure 16: Multi-Agent Reinforcement Learning Framework



The Markov game illustrated in Fig.16 allows both the platform and consumers to learn from experience collected through trial-and-error strategies. Specifically, if the platform's policy is to increase the discount level (i.e., decrease prices) over time, then consumers will incorporate this policy into their purchase decisions and engage in strategic waiting because the platform's policy is part of the value function for consumers.

D.3 Optimal Pricing Strategy for Forward-looking Consumers

What would be the optimal pricing strategy for forward-looking consumers? We draw on the theoretical literature on dynamic pricing to offer some conjectures. The optimal strategy depends on the primary intertemporal tradeoff. If the reference price effect is the primary intertemporal tradeoff, then Wu et al. (2015) show that price skimming (i.e., markdown price) is the optimal pricing strategy with forward-looking consumers who form price expectations or reference prices based on the retailer's actions in the past. The presence of strategic consumers makes the firm discount less over time, leading to a more stable markdown price. If loyalty or inertia is the primary intertemporal tradeoff, then Klemperer (1987) shows that price penetration is the equilibrium optimal strategy under the assumption that consumers are forward-looking and have rational expectations for future prices. The reason is that firms will raise their prices in the second period to take advantage of the fact that first-period customers are now locked in. Firms may charge higher prices in the first period with forward-looking consumers than with myopic consumers because the demand of forward-looking consumers is less elastic; they recognize that a firm with a lower first-period price will gain a greater market share and will charge a higher price in the second period, so forward-looking consumers are wary of becoming attached to the supplier. Finally, Seetharaman and Che (2009) show that if variety-seeking is the primary intertemporal tradeoff, then collusive, enduring high prices are optimal even for forward-looking consumers. In fact, first-period prices increase even more for rational, forward-

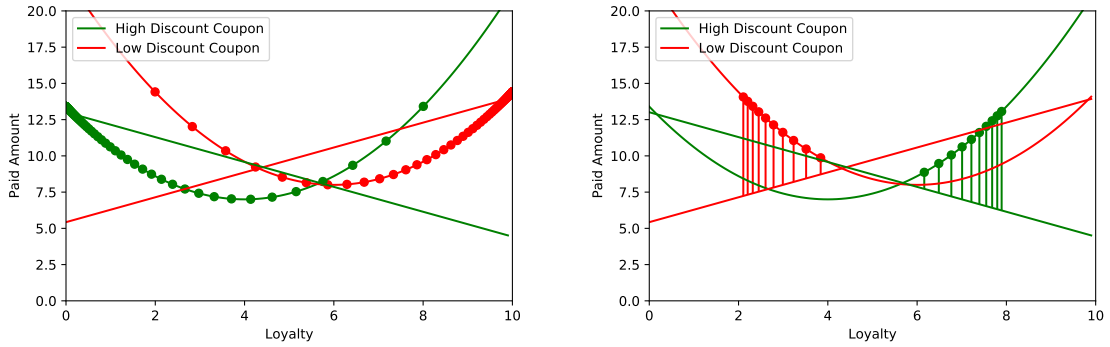
looking consumers than for myopic consumers because forward-looking consumers recognize that they will be partially locked in to an untried supplier in the second period, so they must predict second-period prices when making their first-period purchase decisions, and the prediction makes the first-period price cut less attractive.

E An Example of Model Bias

This section uses an illustrative example (Fig.17) to explain the intuition behind model bias, and we present empirical evidence of potential model bias.

Figure 17: Illustration of Model Bias

(a) Observed Policy (b) New Policy



Imagine there are two types of coupons, one with a high discount level, color-coded in green, and the other with a low discount level, color-coded in red. Fig.17 exhibits the relationship between loyalty and the paid amount (revenue) for purchases made with the two types of coupons. We define "loyalty" as the consumer's number of past purchases (e.g., a loyalty of 2 means that the consumer has bought twice before). It is possible that, under a certain behavioral mechanism, we could observe a non-linear, quadratic relationship between loyalty and the paid amount (shown as the green curved line and the red curved line). Specifically, when loyalty is low, revenue decreases with purchase experience; when loyalty is high, revenue increases with purchase experience. Also, it is possible that the high-discount coupon generates higher revenue from high-loyalty customers, whereas the low-discount coupon generates higher revenue from low-loyalty customers. Thus, in Fig.17, the green curved line is above the red curved line when loyalty is higher than 6.

We further assume that in the observed data (panel (a)), the firm's strategy rewards new consumers by giving them more high-discount (green) coupons, whereas loyal consumers receive more low-discount (red) coupons. In the plot, each dot is one observation; the green dots (high discounts) are concentrated in the low-loyalty region, and the red dots (low discounts) are concentrated in the high-loyalty region. Thus,

although the true relationship between loyalty and revenue in the data generating process is quadratic, an econometrician might instead assume that a linear relationship (the straight lines) is the best fit. The linear relationships are estimated using ordinary least squares (OLS) and are unbiased in the current data, under the current policy.

Next, we consider a new policy (panel (b)) in which the firm reverses its previous strategy and now gives more high-discount coupons to loyal customers instead of to new customers (the green dots cluster in the high loyalty region in panel (b)). Now, the revenue predicted by the linear model is negatively biased because the green straight line lies below the green curved line. The biased predictions lead to suboptimal strategy decisions. For example, in the high-loyalty region (loyalty > 6), the linear model predicts that the low-discount coupon is the best strategy (the straight red line is above the straight green line), while the quadratic model predicts the opposite (the curved green line is above the curved red line).

The example demonstrates two sources of model bias. The first source of bias is distribution mismatch or covariate shift. The reward predictor (straight lines) was trained on the past data and can perform well (i.e., be unbiased) on past data but not necessarily on data generated using a new policy. The reward predictor is formed without knowledge of the new policy and, hence, when approximating the reward, the predictor might focus too much on areas that are irrelevant for the new policy and not enough on areas that are important for the new policy (Dudík et al. 2014). In our setting, although the platform randomly distributed coupons, the probabilities associated with different coupon types are not uniform. As shown in Table 2, there are more coupons with the lowest discount level (0-15%) than with the highest discount level (85%-100%) because the platform cannot afford to offer large markdowns as often as small markdowns. So, a model built on our observed data could suffer from the first source of model bias. The second source of model bias is wrong functional forms. Consumer behaviors in the information-rich environment are challenging to model. The straight line here is an extreme example, but even flexible non-parametric models might not accurately capture the intricacies in the true data-generating process. In fact, many prior studies on consumer click-through rates and purchase likelihoods present predictions with only single-digit precision (Wen et al. 2019).

F CCP

Although both Q-learning and the Conditional Choice Probability (CCP) estimator (Hotz and Miller 1993) use sample observations (the so-called cell estimators) in the estimation procedure, they are fundamentally

different algorithms. We compare the properties of the two algorithms on three dimensions: (1) reinforcement learning primitives, (2) how the sample observations are used, and (3) the researcher's role.

First, CCP and Q-learning belong to two different subcategories of reinforcement learning and, therefore, have different primitives. CCP is an estimation algorithm for the exact dynamic programming problem, which assumes that the agent knows the environment, that is, the state transition process and the reward function. Therefore, the agent can solve the dynamic programming program and derive the optimal policy. The data generating process assumes that the agent takes actions according to the optimal policy in each step. By contrast, Q-learning is a solution to the model-free reinforcement learning problem, which assumes that the agent does not know how the environment operates, so the agent learns about the environment while gaining experience. In each step, the agent takes actions according to the policy the agent has learned, which might not yet be optimal. Moreover, CCP assumes that the agent's reward is defined by a utility function that is not observed directly in the data, while the reward for Q-learning is observed in the data.

Second, both CCP and Q-learning use sample observations, but in different fashions. In CCP, sample observations are used to calculate the conditional choice probabilities (hence, the name of the method): the action probabilities in each state and the state transition probabilities for each state-action pair. Then, the CCPs are plugged into the likelihood function in the maximum likelihood estimation routine. On the contrary, in Q-learning, sample observations are used to get the values of the reward and next state, which are plugged into the value iteration update function to iteratively calculate the Q-function (step 9 in Algorithm 1).

Third, the researcher plays different roles when applying CCP and Q-learning. In CCP, the researcher is the econometrician, whose objective is to infer or estimate the parameters in the agent's utility function. When using Q-learning, however, the researcher is the agent herself. She directly observes the reward and creates the optimal policy by learning the value function iteratively. Because Q-learning belongs to the family of model-free reinforcement learning, no model or model parameter is involved. In the tabular case, when the state space is small, no parameter needs to be estimated at all. When the state space is large, the researcher needs to estimate the parameters in the functional approximator, such as DNNs.

To sum up, although both CCP and Q-learning use cell estimators, they differ drastically on the three dimensions mentioned above. For a more detailed comparison of exact dynamic programming and model-free reinforcement learning, see Sutton and Barto (2018).

G Full Structural Model

This section presents a full structural model for consumers' sequential search, purchase, and churn behaviors. An alternative model without search is introduced in §L.

G.1 Model

Assume that all livestream channels are displayed in order on the list/search page, which shows some information about each livestream channel, including the thumbnail, topic, seller's name, average price, number of viewers, and number of likes. When a consumer clicks a channel's link, she enters the channel and can obtain more information such as the livestream content, detailed product descriptions, and a coupon. On a single day, consumers can visit multiple channels that belong to multiple product categories, and they can make multiple purchases. We treat channel visits in different categories as separate, independent search sessions. Formally, a search session is confined to one product category and, at most, one purchase, but it can include multiple clicks (channel visits) in the same category. A search session can end in one of two ways: the consumer makes a purchase, or the consumer switches from one product category to another (which both terminates the first search session and starts a new one). After examining the data, we found that, within one session, consumers rarely revisited a channel that they searched earlier in the session; all purchases were made in the last channel visited. Therefore, in each step of the decision process, the consumer's decisions are (1) whether to stop searching (churn), (2) if not churning, then which channel to search (search), and (3) whether to buy a product from the searched channel (purchase). Our setting can be characterized by a sequential search model without recall (McCall 1970).

G.1.1 Timing

A search session g can consist of multiple coupon reception incidences t. During a search session g, when a consumer visits channel j and receives a coupon, this constitutes a coupon reception incidence t. This timing definition is consistent with that in §4.

G.1.2 Search and Purchase Model

Let w_{ijg} be consumer i's valuation of the list-page information for channel $j \in \{1,\ldots,J\}$ in search session g, and v_{ijg} is the valuation after clicking and watching the livestream channel. Then, w_{ijg} is the expected utility before search, and v_{ijg} is the expected utility from search. w_{ijg} depends on $\mathbf{x}_{ijg}$, a vector of state variables related to the channel (seller) and product (e.g., average price, number of viewers, number of likes). We

assume v_{ijg} follows a normal distribution with a mean of zero and standard deviation σ_j.

Let c_{ijg} be the search cost, which includes the time and mental cost of processing information during search. The search cost depends on $\mathbf{z}_{ijg}$, a vector of consumer- and seller/product-related state variables, such as consumer demographics and the ranking position of the channel. If a consumer chooses not to make a purchase on the platform, she can take the outside option $j = 0$ with a known expected utility u_{it0} (i.e., it does not require search). Then, after search, the realized utility of channel j becomes

$$\begin{aligned} u_{ijg} &= w_{ijg} + v_{ijg}, v_{ijg} \sim N\left(0, \sigma_j^2\right) \\ w_{ijg} &= \mathbf{x}_{ijg}^{'} \boldsymbol{\beta}_i \\ c_{ijg} &= \mathbf{z}_{\mathbf{ijg}}^{'} \boldsymbol{\gamma}_i \end{aligned} \tag{10}$$

The variables ($\mathbf{x}_{ijg}$) that affect the search utility u_{ijs} include those observed after clicking because we assume that $\mathbf{x}_{ijg}$ can be decomposed into two subsets, $\mathbf{x}_{ijg}^s$ and $\mathbf{x}_{ijg}^p$, where $\mathbf{x}_{ijg}^s$ are the variables observed in the search stage (before clicking; the superscript s denotes the search stage) and $\mathbf{x}_{ijg}^p$ are the variables observed in the purchase stage (after clicking; the superscript p denotes the purchase stage). Because $\mathbf{x}_{ijg}^p$ is not observed in the search stage, we can assume that consumers form an expectation about $\mathbf{x}_{ijg}^p$, denoted $\tilde{\mathbf{x}}_{ijg}^p$; thus, in the search stage, $u_{ijg} = \mathbf{x}_{ijg}^s \boldsymbol{\beta}^s + \tilde{\mathbf{x}}_{ijg}^p \boldsymbol{\beta}^p + v_{ijg}$. If we assume that consumers have rational expectations (i.e., the expected variable values deviate from the observed variable values only by a random noise term, that is, $\tilde{\mathbf{x}}_{ijg}^p = \mathbf{x}_{ijg}^p + \boldsymbol{\iota}_{ijg}$ and $\boldsymbol{\iota}_{ijg} \sim N(\mathbf{0}, \Sigma_\iota)$), then the utility function becomes $u_{ijg} = \mathbf{x}_{ijg}^s \boldsymbol{\beta}_i^s + \mathbf{x}_{ijg}^p \boldsymbol{\beta}_i^p + \iota_{ijg}^{'} \boldsymbol{\beta}^p + v_{ijg}$. Let $\varepsilon_{ijg} = \iota_{ijg}^{'} \boldsymbol{\beta}_i^p + v_{ijg}$, then $u_{ijg} = \mathbf{x}_{ijg}^s \boldsymbol{\beta}_i^s + \mathbf{x}_{ijg}^p \boldsymbol{\beta}_i^p + \varepsilon_{ijg} = \mathbf{x}_{ijg}^{\prime} \boldsymbol{\beta}_i + \varepsilon_{ijg}$, $\varepsilon_{ijg} \sim N\left(0, \beta^p \Sigma_\iota \boldsymbol{\beta}^p + \sigma_j^2\right)$. The utility function implies that the variables in equation 10 can include all the covariates, regardless of whether they are observable before clicking. Importantly, $\mathbf{x}_{ijg}$ includes $\mathbf{A}_{it}$, the coupon the consumer receives.[43] For ease of explanation, we can assume that $\mathbf{x}_{ijg} = \begin{pmatrix} \mathbf{S}_{it} \\ \mathbf{A}_{it} \end{pmatrix}$, where $\mathbf{S}_{it}$ is the vector of the state variables and $\mathbf{A}_{it}$ is the vector of action dummy variables. Note that the state variables $\mathbf{S}_{it}$ include those that can incorporate consumer intertemporal tradeoffs (Table 18), such as the reference price effect and state dependence, so the structural model is consistent with the model free evidence documented in §3.4.

We allow for consumer heterogeneity using a random coefficient approach where both the individual utility parameters $\boldsymbol{\beta}_i$ and cost parameters $\boldsymbol{\gamma}_i$ follow a normal distribution; that is, $\boldsymbol{\beta}_i \sim N\left(\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta\right)$ and $\boldsymbol{\gamma}_i \sim N\left(\boldsymbol{\mu}_\gamma, \boldsymbol{\Sigma}_\gamma\right)$, where $\boldsymbol{\mu}_\beta$ and $\boldsymbol{\mu}_\gamma$ capture the mean effect across consumers, and $\boldsymbol{\Sigma}_\beta$ and $\boldsymbol{\Sigma}_\gamma$ (assumed to be

[43] This assumption contradicts the exogenous search evidence in Appendix A. Future research could separate the utility function into the search utility and purchase utility (Seiler 2013).

diagonal) capture the variance.

The consumer's optimal search strategy can be summarized by the following rules:

1. Selection Rule: If a channel is to be searched, it should be the channel with the highest reservation utility R_{ij}. The reservation utility is defined as the utility that makes the consumer indifferent between choosing the last-searched option (with utility R_{ij}) and continuing with the next search. The reservation utility solves the equation $\int_{R_{ijg}}^{\infty}(u_{ijg}-R_{ijg})\,dF(u_{ijg})=c_{ijg}$, where the left-hand side is the marginal utility of searching channel j, and the right-hand side is the marginal cost of search.

2. Stopping Rule: Terminate search whenever the current realized utility exceeds the reservation utility of every unsearched channel.

3. Purchase Rule: Once the consumer stops searching, compare the last-searched channel with the outside option. If the realized utility of the last-searched channel is higher than that of the outside option, then purchase. Otherwise, take the outside option.

G.1.3 Churn Model

The churn model is specified as a binary logit model. Consumer i's utility of churn from the platform in incidence t is ϕ_{ijg1}, and the utility of staying is ϕ_{ijg0}

$$\begin{aligned}\phi_{ijg1} &= \mathbf{S}_{ijg}\rho+\varepsilon_{ijg1}, \varepsilon_{ijg1}\sim Gumbel(0,1)\\ \phi_{ijg0} &= \varepsilon_{ijg0}, \varepsilon_{ijg0}\sim Gumbel(0,1)\end{aligned}$$

G.2 Estimation

The likelihood function for the structural parameters $\boldsymbol{\theta}=\left(\boldsymbol{\mu}_{\beta},\boldsymbol{\Sigma}_{\beta},\boldsymbol{\mu}_{\gamma},\boldsymbol{\Sigma}_{\gamma},\left\{\sigma_j\right\}_{j=1}^{J},\rho\right)$ can be specified based on the search rules. The selection rule implies that, in the j_{th} search, the reservation utility for the channel is higher than that of all the to-be-searched and unsearched channels. Thus, the corresponding likelihood function is $L_{ijg}^{Selection}=Pr\left(R_{ijg}\geq R_{ikg},\forall k\in\{j+1,\ldots,J\}\right)$. The stopping rule implies that 1) the reservation utility for the channel in the j_{th} search must be higher than the realized utilities of all previously searched channels; otherwise, the consumer would have stopped earlier, and 2) the realized utility of the channel in the j_{th} search is higher than the reservation utility of all the unsearched channels. Thus, the corresponding likelihood function is $L_{ijg}^{Stop}=Pr\left(R_{ijg}\geq u_{ikg},\forall k\in\{1,\ldots,j-1\}\right)\cdot Pr\left(u_{ijg}\geq R_{ilg},\forall l\in\{j+1,\ldots,J\}\right)$. The purchase rule implies that the utility of choosing the last-searched channel is higher than that of the outside option, so the corresponding likelihood function is $L_{ijg}^{Purchase}=Pr\left(u_{ijg}\geq u_{i0g}\right)I(Buy_{ijg}=1)+$

$Pr(u_{ijg} < u_{i0g})\, I(Buy_{ijg} = 0)$. The likelihood of churn (leave permanently) is $L_{ijg}^{Churn} = Pr(\phi_{ijg1} \geq \phi_{ijg0})\, I(Churn_{ijg} = 1) + Pr(\phi_{ijg1} < \phi_{ijg0})\, I(Churn_{ijg} = 0)$. Therefore, the full likelihood is $L = \prod_i \prod_g \prod_j L_{ijg}^{Selection} L_{ijg}^{Stop} L_{ijg}^{Purchase} \left(1 - L_{ijg}^{Churn}\right)$. We use the simulated maximum likelihood approach (SMM) for estimation.

G.3 Identification

Consumers may be enticed to search more under two conditions: an increase in the variance of the utility from search (σ_j) and a reduction in the search cost ($\boldsymbol{\mu}_\gamma$). Thus, σ_j and $\boldsymbol{\mu}_\gamma$ are not identified separately, so we normalize all the $\left\{\sigma_j\right\}_{j=1}^{J}$ to 1. The mean utility parameters $\boldsymbol{\mu}_\beta$ are identified by the association between the observed state variables and the frequencies of click and purchase. The heterogeneity utility parameters $\boldsymbol{\Sigma}_\beta$ are identified by the distribution of deviations of the observed clicks and purchases from the predicted values based on the mean utility parameters. The mean cost parameters $\boldsymbol{\mu}_\gamma$ are identified by the inter-consumer observations of continuing or stopping search given their current consideration sets (e.g., corresponding livestream channel positions in the list). The heterogeneity cost parameters $\boldsymbol{\Sigma}_\gamma$ are identified by the deviations from the mean predicted probabilities of stopping. The churn parameters are identified by the association between the state variables and the frequency of churn.

G.4 Result

Table 10 reports the estimated structural parameters of the search model. We include only a subset of the parameters because we use over 300 state variables in the model. The full list of estimates is available from the author upon request.

G.5 Simplification

In the structural sequential search model, the stopping rule and search rule jointly create a mapping between the current state-action pair and the next state because the rules describe whether the consumer is going to churn and, if not, which channel the consumer is going to visit. If we assume that $\mathbf{S}_{it} = \{\mathbf{x}_{it} \backslash \mathbf{A}_{it}\} \cup \mathbf{z}_{it}$, then the stopping rule and search rule can be represented by $\mathbf{S}_{it+1} = g(\mathbf{S}_{it}, \mathbf{A}_{it})$. The purchase rule describes consumer payment, so it can be represented by a mapping between the current state-action pair and the platform's reward; that is, $R_{it} = f(\mathbf{S}_{it}, \mathbf{A}_{it})$. The full structural model characterizes both mappings using the sequential search structure. A simplified model can replace the structural model with flexible functional forms such as GBDT.

Table 10: Model Estimates

	Model With Search				Model Without Search			
	Mean Parameters		Heterogeneity		Mean Parameters		Heterogeneity	
	Est.	SE	Est.	SE	Est.	SE	Est.	SE
Average price	-0.265	0.027	0.051	0.006	-0.195	0.076	0.043	0.005
Number of viewers (log)	0.592	0.054	0.280	0.072	0.444	0.032	0.234	0.064
Number of likes (log)	0.328	0.033	0.264	0.089	0.360	0.034	0.340	0.077
Coupon Threshold LL Discount LL	0.716	0.210	0.426	0.130	0.568	0.255	0.459	0.101
Coupon Threshold LL Discount LL	1.337	0.406	0.677	0.171	1.198	0.434	0.538	0.212
Coupon Threshold LL Discount M	1.994	0.535	1.107	0.278	1.719	0.573	1.349	0.319
Coupon Threshold LL Discount H	2.594	0.813	1.449	0.428	2.325	0.829	1.785	0.439
Coupon Threshold LL Discount HH	3.285	1.067	1.803	0.503	2.918	1.053	1.943	0.492
Coupon Threshold L Discount LL	0.559	0.170	0.334	0.090	0.488	0.139	0.423	0.074
Coupon Threshold L Discount L	1.306	0.349	0.706	0.189	1.152	0.359	0.681	0.191
Coupon Threshold L Discount M	1.833	0.550	0.965	0.302	1.623	0.522	0.819	0.235
Coupon Threshold L Discount H	2.480	0.740	1.333	0.415	2.205	0.746	1.109	0.356
Coupon Threshold L Discount HH	3.162	1.048	1.848	0.583	2.815	1.036	2.187	0.596
Coupon Threshold M Discount LL	0.525	0.131	0.271	0.074	0.423	0.155	0.320	0.087
Coupon Threshold M Discount L	1.108	0.312	0.633	0.209	0.978	0.306	0.819	0.253
Coupon Threshold M Discount M	1.740	0.475	0.888	0.267	1.559	0.498	0.814	0.260
Coupon Threshold M Discount H	2.494	0.626	1.390	0.363	2.178	0.582	1.344	0.390
Coupon Threshold M Discount HH	3.124	0.824	1.872	0.564	2.775	0.842	1.776	0.476
Coupon Threshold H Discount LL	0.389	0.117	0.200	0.056	0.288	0.155	0.248	0.057
Coupon Threshold H Discount L	1.122	0.313	0.633	0.177	0.966	0.286	0.778	0.210
Coupon Threshold H Discount M	1.756	0.458	1.020	0.264	1.543	0.416	0.747	0.199
Coupon Threshold H Discount H	2.406	0.769	1.206	0.347	2.132	0.768	0.896	0.421
Coupon Threshold H Discount HH	2.977	0.883	1.748	0.570	2.661	0.930	1.948	0.484
Coupon Threshold HH Discount LL	0.257	0.076	0.148	0.043	0.183	0.070	0.192	0.051
Coupon Threshold HH Discount L	0.987	0.294	0.545	0.163	0.824	0.339	0.404	0.198
Coupon Threshold HH Discount M	1.537	0.442	0.899	0.232	1.323	0.453	0.640	0.285
Coupon Threshold HH Discount H	2.191	0.657	1.189	0.386	1.884	0.691	1.046	0.407
Coupon Threshold HH Discount HH	2.987	0.771	1.599	0.402	2.656	0.799	1.879	0.391
Constant	-1.298	0.008	0.405	0.036				
Position	0.152	0.007	0.059	0.018				

Note: The coefficients of the Average price, Number of reviewers (log), Number of likes (log), and Coupon Threshold XX Discount XX are related to the search utility variables $\mathbf{x}_{ijg} = (\mathbf{S}_{it}, \mathbf{A}_{it})'$ where $\mathbf{S}_{it}$ is the vector of the state variables and $\mathbf{A}_{it}$ is the vector of the action dummy variables. The coefficient of the Position variable corresponds to the search cost variable $\mathbf{z}_{\mathbf{ijg}}$ in equation (10).

G.6 Optimization

After estimating the structural parameters, we create an optimization routine to find the optimal policy. The objective of the optimization is to maximize the total discounted revenue over a T period horizon $\max_{A_{it} \in \mathbb{A}} E\left\{\sum_{i=1}^{I} \sum_{t=0}^{T_i} \delta^t R_{it}\left(\mathbf{S}_{it}, A_{it}\right)\right\}$. The finite horizon allows for the dynamic programming problem to be solved via backward induction.

H State Variables and Summary Statistics

Table 11: Static State Variables

Group	State Variables	#
Consumer	Demographics such as age, gender, income, education, and occupation; behavioral variables such as TQZ,[44] purchasing power, product category preference, and host preference.	175
Host	Demographic features of the host as well as the popularity of the host as a seller. Specifically, we include the product categories sold by the host (e.g., jewelry, women's apparel), monthly revenue, a rating of the quality of the host's content (evaluated by experts), and the number of subscribers to the host's channel, etc. The platform pre-defines some host-level summary statistics such as attractiveness, which the platform evaluates from unstructured data, including the host's profile images and voice.	66
Product	Average values for all the products promoted in a livestream,[45] including the product category,[46] price, market share, repurchase rate, review rating, and shipping cost, etc.	36
Livestream (video)	Time of day of the live video (morning, afternoon, or evening); the average number of consumers who watch the livestream, add a product to the cart, write a comment, share the video, like the video, and reward the host; and the total payment amount generated during the livestream, etc. We considered using unstructured data such as textual features (e.g., the video script) and audio features (e.g., the music background), but the data were not available, so we leave it to future research.	27

Table 12: Dynamic State Variables

Group	State Variables	#
Consumer	Related to consumers' coupon reception behaviors: the number of days since the consumer last received a coupon (recency_coupon), the number of coupons received since the beginning of the sample period (frequency_coupon), and the average, minimum, and maximum of the discount ratio and threshold ratio of the coupons received since the beginning of the sample period (monetary_coupon). We also track these variables separately by product category (5 categories: men's/women's/children's apparel, cosmetics, and jewelry).	40
Host	The total number of sellers visited by the consumer since the beginning of the sample period (frequency_seller). Every time a consumer visits a host, she receives a coupon, so the host dynamic state is equivalent to the consumer dynamic state, frequency_coupon. To avoid redundancy, we do not add additional host dynamic state variables.	0
Product	The number of periods (coupon reception incidences) since a consumer last purchased a product (recency_product), the number of products purchased (frequency_product), the average, maximum, and minimum price of the products purchased (monetary_product), and the cumulative spending since the beginning of the sample period (monetary_product). We track these variables separately for each category.	30
Livestream	The number of periods since a consumer last visited a livestream channel (recency_webpage) and the total number of channels visited (frequency_webpage). Every time a consumer visits a channel, she receives a coupon, so the livestream dynamic states are equivalent to the consumer dynamic states, recency_coupon, and frequency_coupon. To avoid redundancy, we do not add additional livestream dynamic state variables.	0
Terminal	The terminal state occurs when the consumer stops returning to the platform. This state captures consumer churn.	1

There are many state variables, so we provide summary statistics for a subset in Table 13, and the complete table is available upon request.

[44] TQZ is an activity score, with higher values indicating higher activity. A consumer can increase her TQZ by searching and buying more products, staying for a longer duration, and writing more reviews.

[45] One livestream session can sell many products, but most of the products usually are in the same category. We use the average product characteristics within a livestream session as state variables.

[46] There are 5 categories: men's apparel, women's apparel, children's apparel, cosmetics, and jewelry.

Table 13: Summary Statistics of State Variables

Category	Static / Dynamic	State Variable	N	Mean	S.D.	Min	25%	50%	75%	Max
Consumer	Static	TQZ	25,886,094	950.58	1406.85	297	664	841	1104	4483
	Dynamic	Frequency coupon	25,886,094	24.35	62.93	0	8	14	26	833
Host	Static	Attractiveness	25,886,094	703.20	900.09	3	532	739	864	999
Product	Static	Price	25,886,094	6067.55	19375.87	12	113	193	320	11298752
Livestream	Static	# Times add to cart	25,886,094	2158.89	6578.47	7	260	810	2397	75451
	Dynamic	Frequency product	25,886,094	1.53	6.04	0	0	0	2	190

I Benchmark Prediction Performance

We evaluate the accuracies of the linear regression, GBDT, and DNN algorithms by splitting the data into training and test sets. Table 14 reports the result. GBDT is the most accurate of the three.

Table 14: Comparison of the Predictive Performance of Three Benchmark Models

		Training Set	Test Set
Number of Consumers		816,718	204,180
Number of Observations		20,688,420	5,197,674
Benchmark Model Mean	Linear Regression	0.93	0.92
	GBDT	0.84	0.89
Squared Error	DNN	0.86	0.91

J Hyperparameters

Table 15: Hyperparameter Choice

Hyperparameter	Definition	Alternatives	Optimal choice
G_ω	Action probability model, Number of trees	5, 10, 50, 100, 500	10
	Action probability model, Maximum depth	1, 2, 5, 10, 20	2
τ	Action probability threshold	0.001, 0.01, 0.05, 0.1, 0.5	0.05
Γ	Target network update rate (number of training iterations)	500, 1k, 8k, 20k	8k
M	Q-network number of nodes in the hidden layers	(16, 16) (32, 32) (64, 64) (100, 50)	(32, 32)
α	Adam optimizer learning rate	0.00001, 0.001, 0.01, 0.1	0.001
ε	Adam optimizer epsilon	0.000001, 0.2	0.2
E	Number of epochs	20, 50	20
N	Minibatch size	256, 512	256

The BCQ algorithm (§4.3) relies on a list of hyperparameters. Table 15 lists the hyperparameters considered, their definitions, value alternatives, and the final optimal values chosen. For instance, for the action probability model G_ω (GBDT), we use 10 trees, each with a maximum depth of 2. The threshold value τ is 0.05. We set the Q-value function approximator as a three-layer, fully-connected neural network. The three layers, in order, have 32, 32, and 1 hidden nodes, and the activation functions are ReLu, ReLu, and linear. The model is trained using the stochastic gradient descent algorithm (step 6 in Algorithm 2) with the Adam optimizer and an epsilon value of 0.2. The mini-batch size is 256, the learning rate is 0.001, the discount factor δ is 0.99, the total number of steps in a trajectory is 25, and the number of epochs is 20. We choose these values by doing a grid search and identifying the hyperparameter combinations with the best

convergence properties.

Following the guideline in Henderson et al. (2018), we use the average return as the evaluation metric to compare different hyperparameters. The two figures below show the average return at each time step for alternative values of two hyperparameters. As shown, our chosen hyperparameter values achieve the highest average return the fastest.

Figure 18: Average Return for Alternative Γ Values

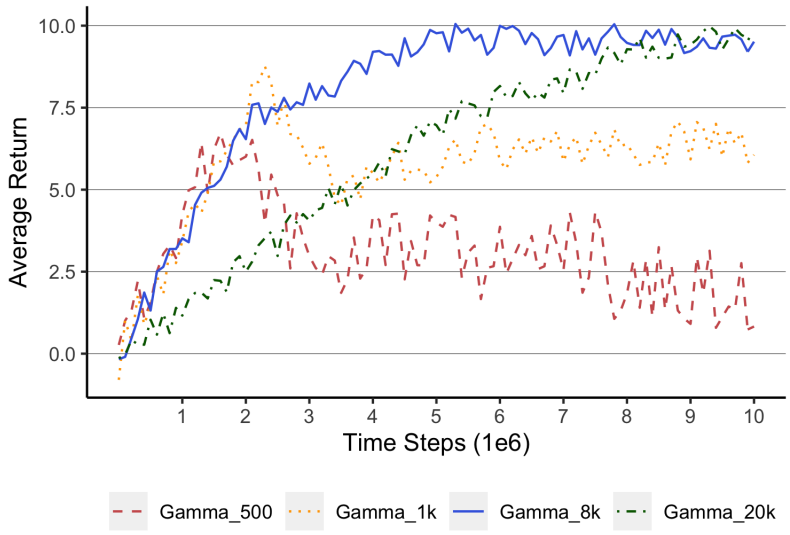
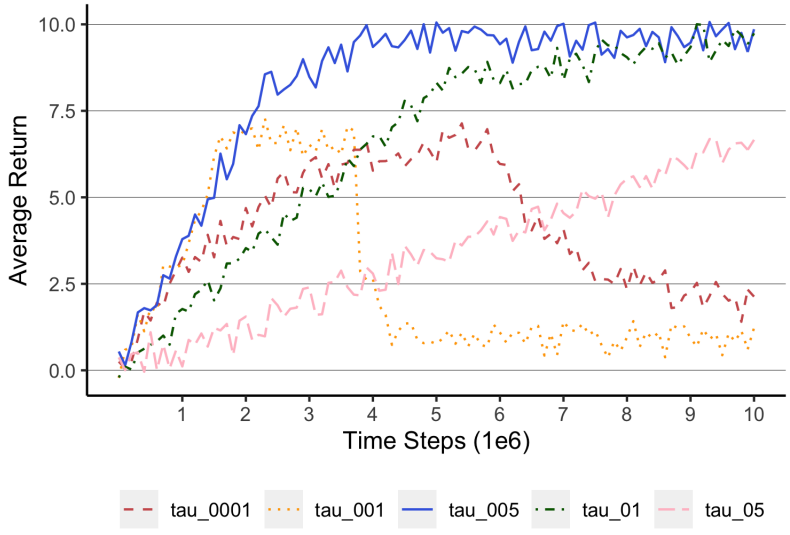


Figure 19: Average Return for Alternative τ Values



We use the XGBoost package (Chen and Guestrin 2016) to estimate GBDT. The number of trees is 200, the

learning rate is 0.1, the maximum depth of the individual regression tree is 3, and the fraction of features to consider when looking for the best split is 0.1.

For the neural network model, we use a three-layer, fully-connected neural network with network sizes of 24, 256, and 32 in each layer and ReLu as the activation function. We use the dropout method (dropout rate = 0.2) for regularization.

K Policy Evaluation by Training versus Test Sets

The CLV and gain metrics are slightly lower in the test set than in the training set, but the relative advantage of the proposed BDRL algorithm over the benchmarks is unchanged.

Table 16: Model Comparison Based on the Doubly Robust Estimator

Training Set		1. Static Homogeneous	2. Static Heterogeneous			3. Model-Based Dynamic Heterogeneous	4. Model-Free Dynamic Heterogeneous
		A: Regression	B: GBDT	C: DNN	D: ORF	E: Structural	F: Proposed BDRL
CLV	Mean	6.63	7.64	7.02	7.56	8.41	9.63
(Return)	Std	2.04	2.37	2.30	2.36	2.83	3.27
Gain	Mean	13%	31%	20%	29%	44%	65%
T-test		234.89	499.69	331.93	478.99	646.06	866.16
P_value		< .001	< .001	< .001	< .001	< .001	< .001
Test Set		1. Static Homogeneous	2. Static Heterogeneous			3. Model-Based Dynamic Heterogeneous	4. Model-Free Dynamic Heterogeneous
		A: Regression	B: GBDT	C: DNN	D: ORF	E: Structural	F: Proposed BDRL
CLV	Mean	6.33	7.29	6.88	7.32	8.30	9.33
(Return)	Std	1.88	2.15	2.22	2.27	2.62	3.21
Gain	Mean	6%	23%	16%	23%	39%	57%
T-test		59.36	197.35	134.37	195.05	310.09	392.80
P_value		< .001	< .001	< .001	< .001	< .001	< .001

Note: All the gains are compared to the mean observed CLV: ¥5.85 in the training set and ¥5.95 in the test set. The training set has 816,718 consumers, and the test set has 204,180 consumers.

L Alternative Structural Model Without Search

An alternative structural model can consider only the purchase and churn decisions without the search process (which channel to visit and the sequence of channel visits). For the purchase decision, we consider a random utility-based discrete choice model with random coefficients. The outside option is not purchasing on the livestream platform, and the mean utility is normalized to 0.

$$u_{it1} = \mathbf{x}_{it}^{'}\boldsymbol{\beta}_i + v_{it1}, v_{it1} \sim \mathit{Gumbel}\,(0,1)$$

$$u_{it0} = v_{it0}, v_{it0} \sim \mathit{Gumbel}\,(0,1)$$

$$\boldsymbol{\beta}_i \sim N\left(\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta\right)$$

The likelihood of purchase is $L_{it}^{Purchase} = Pr(u_{it1} \geq u_{it0}) I(Buy_{it} = 1) + Pr(u_{it1} < u_{it0}) I(Buy_{it} = 0)$. The churn decision is specified as a discrete choice model.

$$\phi_{it1} = \mathbf{x}_{it}^{'} \rho + \varepsilon_{it1}, \varepsilon_{it1} \sim Gumbel(0,1)$$

$$\phi_{it0} = \varepsilon_{it0}, \varepsilon_{it0} \sim Gumbel(0,1)$$

The likelihood of churn (leave permanently) is $L_{it}^{Churn} = Pr(\phi_{it1} \geq \phi_{it0}) I(S_{it} = Churn) + Pr(\phi_{it1} < \phi_{it0}) I(S_{it} \neq Churn)$. Therefore, the full likelihood is $L = \prod_i \prod_t L_{ijg}^{Purchase} \left(1 - L_{ijg}^{Churn}\right)$. We use the SMM for estimation.

Table 10 reports a subset of the estimated coefficients. The results in Table 17 indicate that consumers in the model without search (vs. in the model with search) are less price-sensitive and less responsive to coupons, a result consistent with prior research (Moraga-González et al. 2015).

Table 17: Model Comparison Based on the Doubly Robust Estimator

		Structural with Search	Structural without Search
CLV	Mean	8.39	8.15
(Return)	Std	2.79	2.52
Gain	Mean	43%	39%
T-test		716.62	657.17
P_value		<0.001	<0.001

Note: All the gains are compared to the mean CLV observed in the data, which is ¥5.87.

M Sensitivity Test for Active Users

To test whether the dynamic pricing recommendations are sensitive to the sample selection rule (users who received at least 10 coupons during the sample period), we split our data into highly-active users (who received more than 25 coupons) and relatively-inactive users (who received 10 to 24 coupons). Below, we provide model-free evidence of the intertemporal tradeoffs within each segment. Fig.20 and Fig.21 show similar patterns among the highly-active users and relatively-inactive users. Moreover, we find no behavioral research that suggests that the reference price effect applies only to active users but not inactive ones. Based on the additional data evidence and the absence of relevant research findings, we believe that many of our targeting policy recommendations, learned from active users, will generalize to inactive users.

Figure 20: Current and Future Redemption Rates by Coupon Discount Level and User Activity Level

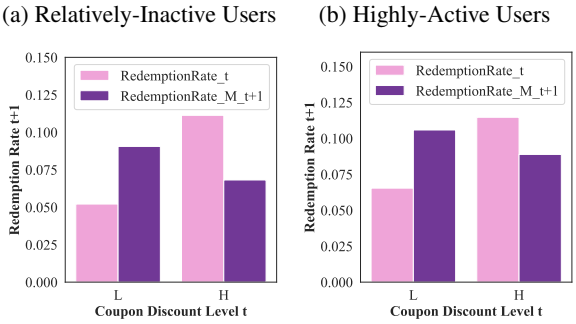
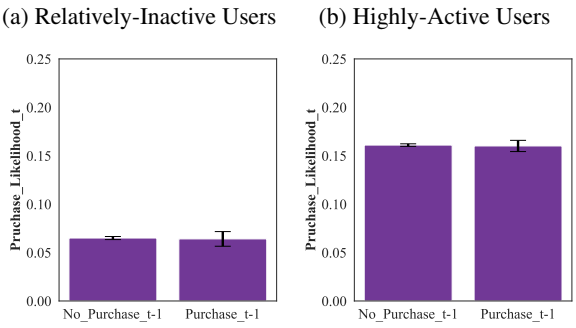


Figure 21: Coupon Redemption Rates by Prior Purchase Experience and User Activity Level



N Policy Evaluation with State Dependence

We consider an alternative specification that can incorporate all three intertemporal tradeoffs discussed in §2.1 and §3.4.2, including state dependence. As shown in Table 18, the monetary value associated with the coupon (monetary_coupon), such as the average/minimum/maximum of the discount ratio/threshold ratio of the coupons received by the consumer, can be considered the reference price, and the product purchase frequency (frequency_product) captures state dependence. A positive state dependence effect indicates inertia, whereas a negative effect indicates variety-seeking.

Table 18: Intertemporal Tradeoffs and Dynamic State Variables

Intertemporal Tradeoffs	Dynamic State Variable
Reference Price	monetary_coupon
Loyalty/Inertia	frequency_product
Variety-Seeking	frequency_product

Table 19 compares the CLV estimates from our model-free dynamic targeting policy (BDRL) and all the benchmark policies. The results are directionally consistent with those in Table 7.

Table 19: Model Comparison Based on the Doubly Robust Estimator

		1. Static Homogeneous	2. Static Heterogeneous			3. Model-Based Dynamic Heterogeneous	4. Model-Free Dynamic Heterogeneous
		A: Regression	B: GBDT	C: DNN	D: ORF	E: Structural[47]	F: Proposed BDRL
CLV (Return)	Mean	6.57	7.57	6.99	7.51	8.39	9.57
	Std	2.01	2.33	2.28	2.34	2.79	3.26
Gain	Mean	+12%	29%	19%	28%	43%	63%
T-test		237.35	536.02	357.17	515.93	716.62	950.56
P_value		<0.001	<0.001	<0.001	<0.001	<0.001	<0.001

Note: All the gains are compared to the mean CLV of ¥5.87. The total sample size is 1,020,898 consumers. For separate analyses of the training and test sets, see Appendix K.

O Undiscounted Reward

In our model, the discount factor is not adjusted by the time intervals between consumer visits, which should be the case based on the economic rationale. We choose to use the same discount rate for two reasons. First, the variance in the intervals between coupon reception incidences is relatively small, with an average inter-incidence time of 3.6 days and a standard deviation of 1.3 days. Second, prior literature that applied reinforcement learning to consumer clickstream data made the same assumption (Urban et al. 2013, Shani et al. 2005, Zheng et al. 2018, and Zou et al. 2019). However, we acknowledge that this assumption is not ideal, so we tested an alternative of undiscounted rewards to alleviate this concern.

Table 20 shows the results with undiscounted rewards. Several interesting results emerge. First, the undiscounted CLVs are larger than the discounted counterparts in Table 19, an expected result. Second, with respect to cross-sectional price discrimination, Figure 22 shows that the discount factor has minimal impact on the action distributions for less-attractive and attractive hosts. Third, the discount factor has a sizable impact on inter-temporal price discrimination. As Figure 23 shows, in the undiscounted case relative to the discounted case (Figure 12), BDRL recommends sending fewer high-discount coupon earlier (in the first incidence) and more high-discount coupon later (in the fifth incidence). This may happen because, when future rewards are not discounted, the earlier high-discount coupons have a more pronounced reference price effect, leading the algorithm to recommend fewer high-discount coupons earlier. The discount factor has no effect on the action distribution under the GBDT policy because GBDT is a static policy, not affected by the discount factor. Fourth, the discount ratio time trend in Figure 24 is steeper in the undiscounted case than in

[47] The result is for the full structural model. The result from the simplified version is qualitatively similar and available upon request.

the discounted case (Figure 13). This result is also consistent with the reference price effect; the algorithm reduces the discount ratio earlier and increases the discount ratio later to avoid the negative reference price effect. Overall, the results in the undiscounted case are qualitatively similar to those reported in the main text.

Table 20: Model Comparison with Undiscounted Rewards Based on the Doubly Robust Estimator

		1. Static Homogeneous	2. Static Heterogeneous			3. Model-Based Dynamic Heterogeneous	4. Model-Free Dynamic Heterogeneous
		A: Regression	B: GBDT	C: DNN	D: ORF	E: Structural[48]	F: Proposed BDRL
CLV	Mean	8.01	9.22	8.53	9.13	10.21	11.66
(Return)	Std	2.48	2.85	2.78	2.86	3.42	3.99
Gain	Mean	11%	28%	19%	27%	42%	62%
T-test		261.88	582.36	393.52	555.66	761.54	1,001.05
P_value		<0.001	<0.001	<0.001	<0.001	<0.001	<0.001

Figure 22: Targeting Rule Under BDRL (Undiscounted Case): Host Attractiveness

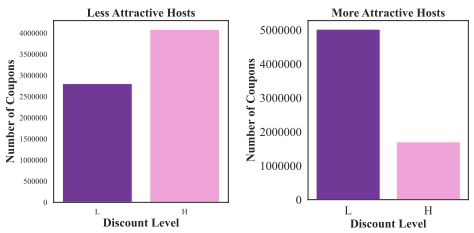
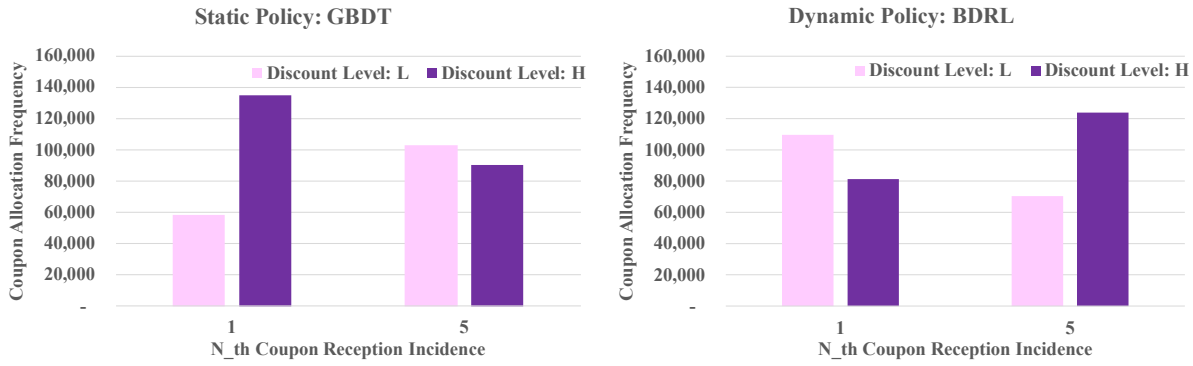


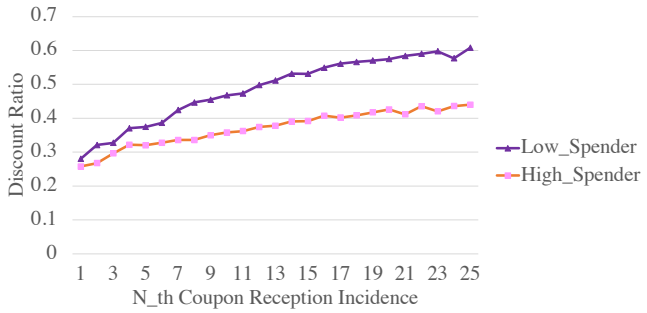
Figure 23: Comparison Between the Static and Dynamic Targeting Policies (Undiscounted Case)



[48]The result is for the full structural model. The result from the simplified version is qualitatively similar and available upon request.

O.0.1 Cross-sectional and Intertemporal Price Discrimination

Figure 24: Targeting Rule (Undiscounted Case): Intertemporal Price Discrimination



P Model Comparison Using Distributional Differences

We use the Kolmogorov-Smirnov test to measure distributional differences. As shown below, both the Doubly Robust Estimator and field experiment results are consistent with the t-test results reported previously.

Table 21: Model Comparison Based on the Doubly Robust Estimator (KS test)

		1. Static Homogeneous	2. Static Heterogeneous			3. Model-Based Dynamic Heterogeneous	4. Model-Free Dynamic Heterogeneous
		A: Regression	B: GBDT	C: DNN	D: ORF	E: Structural[49]	F: Proposed BDRL
CLV	Mean	6.57	7.57	6.99	7.51	8.39	9.57
(Return)	Std	2.01	2.33	2.28	2.34	2.79	3.26
Gain	Mean	+12%	29%	19%	28%	43%	63%
KS-test		0.14	0.31	0.21	0.30	0.42	0.54
P_value		<0.001	<0.001	<0.001	<0.001	<0.001	<0.001

Table 22: Field Experiment Results (KS test)

		Random Allocation	Model-Based Dynamic Heterogeneous (Structural)	Model-Free Dynamic Heterogeneous (BDRL)
CLV	Mean	6.98	9.70	11.16
(Return)	Std	2.43	3.15	3.86
Gain	Mean	~	+39%	+60%
KS-test		~	0.38	0.52
P_value		~	<0.001	<0.001

[49]The result is for the full structural model. The result from the simplified version is qualitatively similar and available upon request.

Q State Transition Implementation in Different Algorithms

The state transition process described in §4.2.2 is implemented differently in the three algorithms (Q-learning, BCQ, and the structural models).

In Q-learning, state transitions are observed by the agent. As shown in Algorithm 1, once the agent takes action A, the environment returns R and S' to the agent (line 7). In our livestream setting, if the firm deploys Q-learning online (in real time), then after the firm takes action A, the consumer will make the purchase, search, and churn decisions, and these consumer decisions will result in R and S'. So, the firm can observe the next state when updating the Q function. There is no need to use historical data to estimate the state transition matrix.

In BCQ, state transitions are observed from the batch data. As shown in Algorithm 2, a sample of M transitions (S,A,R,S') is drawn from the batch data in step 4. An action is chosen in step 5, and the Q function updates in step 6 (specifically, the θ parameters in the neural network approximation of the Q function). Also, the state transition S' is observed from the sampled transitions in step 6. Again, there is no need to use historical data to estimate the state transition matrix.

In the structural model, state transitions are predicted using the structural parameters. Specifically, in the value function iteration step (equation 8), the expectation is taken over the state transition probabilities, and we rely on the estimated state transition to calculate the expectation. As explained in §4.2.2, state transitions can be fixed, stationary, or stochastic. For the static state variables that follow a stationary distribution, in each coupon reception incidence t, we randomly draw a value from the stationary distribution of the state variables. The multivariate (joint) state variable distribution is estimated prior to training the targeting policy. The transitions of the dynamic (stochastic) state variables are governed by the structural model. For instance, the transitions of product-related dynamic variables can be calculated based on Table 5 once Purchase is predicted using the structural model.

References

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD*, pages 785–794, 2016.

James J Heckman. 3. heterogeneity and state dependence. In *Studies in labor markets*, pages 91–140. University of Chicago Press, 2007.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep

reinforcement learning that matters. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Paul Klemperer. The competitiveness of markets with switching costs. *The RAND Journal of Economics*, pages 138–150, 1987.

Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine Learning Proceedings 1994*, pages 157–163. Elsevier, 1994.

José L Moraga-González, Zsolt Sándor, and Matthijs R Wildenbeest. Consumer search and prices in the automobile market. 2015.

Scott A Neslin, Caroline Henderson, and John Quelch. Consumer promotions and the acceleration of product purchases. *Marketing Science*, 4(2):147–165, 1985.

PB Seetharaman and Hai Che. Price competition in markets with consumer variety seeking. *Marketing Science*, 28(3):516–525, 2009.

Guy Shani, David Heckerman, Ronen I Brafman, and Craig Boutilier. An mdp-based recommender system. *Journal of Machine Learning Research*, 6(9), 2005.

Xuanming Su. Intertemporal pricing with strategic customer behavior. *Management Science*, 53(5):726–741, 2007.

Shining Wu, Qian Liu, and Rachel Q Zhang. The reference effects on a retailer's dynamic pricing and inventory strategies with strategic consumers. *Operations Research*, 63(6):1320–1335, 2015.

Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. Drn: A deep reinforcement learning framework for news recommendation. In *Proceedings of the 2018 World Wide Web Conference*, pages 167–176, 2018.

Lixin Zou, Long Xia, Zhuoye Ding, Jiaxing Song, Weidong Liu, and Dawei Yin. Reinforcement learning to optimize long-term user engagement in recommender systems. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2810–2818, 2019.