Appendix

A. Background of Online Advertising

Online advertising is a marketing strategy involving the use of advertising *platform* as a medium to obtain website traffics and targets, and deliver marketing messages of *advertisers* to the suitable *customers*.

Platform. Advertising platform plays an important role in connecting consumers and advertisers. For consumers, it provides multiple advertising channels, e.g., channels on news media, social media, E-commerce websites and apps to explore. For advertisers, it provides automated bidding strategies to compete for consumers in all channels under the setting of real-time bidding (RTB), in which advertisers bid for ad exposures and the exposures opportunities go to the highest bidder with a cost which equals to the second-highest bid in the auction.

Consumers. Consumers explore multiple channels during the several visits to the platform within a couple of days. A consumer's final purchase of an item is usually a gradually changing process, which often includes the phases of Awareness, Interest, Desire, and Action (AIDA) (Roberge, 2015). The consumer's decision to purchase a product (conversion) is usually and has to be driven by multiple touchpoints (exposures) with ads. Each advertising exposure during the sequentially multiple interactions could influence the consumers mind (preferences and interests) and therefore contribute to the final conversion.

Advertisers. The goal of advertisers is to cultivate the consumer's awareness, interest and finally driving purchase. As different ad strategies can affect consumers' AIDA, an advertiser should develop a competitive strategy to win the ad exposures in RTB setting. When the ad is displayed to a consumer, in Cost Per Click (CPC) setting, the advertisers should pay commission to the platform after the consumer clicking the ad. When the consumer purchases the advertised item, the advertiser will get the corresponding revenue.

The objective of an advertiser is usually to optimize the accumulated revenue within a time period under a budget constraint. A strategy that maximizes short-term revenue of each ad exposure on different channels independently is obviously unreasonable, since the final purchase is a result of long-term ad-consumer sequential interactions and the consumer's visits between different channels are interdependent. Therefore, the advertiser must develop a strategy to overcome following two key challenges: (1) Find the optimal interaction sequence including interaction times, channels selection and order of channels for a targeted consumer; (2) Choose targeted consumers and allocate predefined limited budget to them in multiple interaction sequences.

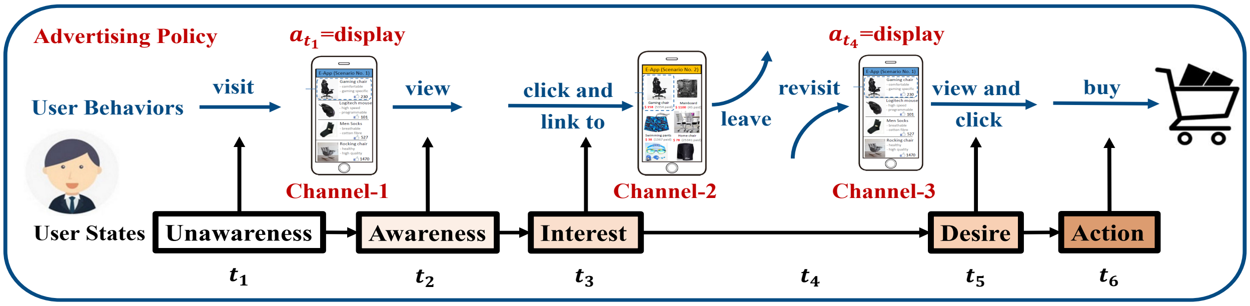


Figure 9. An illustration of the sequential multiple interactions (across different channels) between a user and an ad. Each ad exposure has long-term influence on the user's final purchase decision.

An example of a user's shopping journey is shown in Figure 9. At time t_1, a user visits the news media channel and triggers an advertising exposure opportunity. Then, the advertising agent executes a display action and leaves an exposure on the user. After that, the user becomes aware of and interested in the commodity, so he clicks the hyperlink. Quickly, the user is induced into the landing (detail) page of the commodity in the shopping app. After fully understanding the product information, the user leaves the shopping app. After a period of time, the user comes back to the shopping app at time t_4 and triggers an exposure opportunity of banner advertising. The advertising agent executes a display action as well. Consequently, the users desire is stimulated. At time t_6, the user makes a purchase. In this example, the ad exposure at time t_1 influences the users mind and contributes to the ad exposure at time t_4 and the delayed purchase, which means the ad exposure on one channel would influence the users preferences and interests, and therefore contributes to the final conversion. Thus, the goal of advertising should maximize the total cumulative revenue over a period of time instead of simply maximizing the immediate revenue.

B. Proof and Analysis

B.1. Knapsack Problem in Online Advertising Settings

Theorem 2. *The greedy solution to the proposed dynamic knapsack problem of online advertising is λ approximately optimal where $\lambda > 99.9\%$*

Proof. In the proposed online advertising problem, each user is with value V_G (i.e. the profit of advertiser when the user purchase the commodity) and weight V_C (i.e. the total budget consumption for the target user in the real-time bidding to reach the final purchase). As the item (i.e. user) is non-splittable, the proposed dynamic knapsack problem is essentially a 0-1 knapsack problem which aims to maximize the total value of the knapsack given a fixed capacity B. For each item, we can calculate the Cost-Performance Ratio (CPR) as V_G/V_C. Sort all items in descending order of CPR, i.e. $(V_{G_1}, V_{C_1}), (V_{G_2}, V_{C_2}), \ldots, (V_{G_n}, V_{C_n})$ where CPR$_i \geq$ CPR$_j, \forall i \leq j \leq n$. For $V_C > 0$, $V_G > 0$ and $B > 0$, we first define that this 0-1 knapsack problem has optimal solution $K^*(V_C, V_G, B)$ and greedy solution $K(V_C, V_G, B)$ where K^* and K represent the total value of the knapsack.

Assume B_{end} is the remaining budget after greedy algorithm, the following inequality holds:

$$\frac{B - B_{end}}{B} K^*(V_C, V_G, B) \leq K(V_C, V_G, B) \leq K^*(V_C, V_G, B) \tag{13}$$

This is because:

1) If the knapsack can hold all the items after the greedy algorithm, that is, the optimal solution is equal to the greedy solution. As $B_{end} \geq 0$, we have $\frac{B-B_{end}}{B} K^*(V_C, V_G, B) \leq K^*(V_C, V_G, B) = K(V_C, V_G, B)$

2) If the knapsack cannot hold all the items after the greedy algorithm, as $\frac{V_{G_1}}{V_{C_1}} \geq \frac{V_{G_2}}{V_{C_2}} \geq ... \geq \frac{V_{G_l}}{V_{C_l}}$, we have $V_{G_l} \sum_{j=1}^{l-1} V_{C_j} \leq V_{C_l} \sum_{j=1}^{l-1} V_{G_j} \Leftrightarrow V_{G_l}(B - B_{end}) \leq V_{C_l} K(V_C, V_G, B) \Leftrightarrow \frac{V_{G_l}}{V_{C_l}} \leq \frac{K(V_C, V_G, B)}{B - B_{end}} \Leftrightarrow K(V_C, V_G, B) \geq K^*(V_C, V_G, B) - \frac{B_{end} K(V_C, V_G, B)}{B - B_{end}}$ where l is the index of last item picked by greedy algorithm. This derivation can be simplified to $K(V_C, V_G, B) \geq \frac{B-B_{end}}{B} K^*(V_C, V_G, B)$.

In online advertising settings, the budget spent on a single user is much smaller than the advertiser's total budget. We conduct statistics on one of the world's largest E-commerce platforms to prove it. On Feb 3rd of 2020, a total of 1136149 ads result in 983414548 user-ad sequences (a user sequence consists of multiple interactions of the same user with the same ad), with an average of 865 user sequences per ad. Interactions with users of each ad forms a knapsack problem, where each user sequence is an item in the knapsack. The average maximum budget consumed by each user sequence accounts for 0.07068% of the total budget capacity of the advertisers. We also list details of 5 ads with largest budget consumption in Table 4, where the maximum budget consumed by each user sequence is much smaller than 1/1000 (smaller than 3/10000 specifically) of the total budget of each ad.

As proposed in Dantzig (1957), $\forall i \in 1, 2, \ldots, n, V_{C_i} \leq (1 - \lambda)B, 0 \leq \lambda \leq 1$, the greedy algorithm achieves an approximation guarantee of λ. We can conclude from above statistics that $\max_i \frac{V_{C_i}}{B} \leq \frac{1}{1000}$, which means λ is much greater than $1 - \frac{1}{1000}$.

The thesis above can be further proved:

1) If the knapsack can hold all the items after the greedy algorithm, that is, the greedy solution is obviously equal to the optimal solution, which is also the λ approximately optimal solution.

Ad	#Users Sequences	Budget	Avg Cost	(Avg Cost)/Budget	Max Cost	(Max Cost)/Budget
Ad 1	2460976	119352.51	0.048498039	0.0000406343%	20.04	0.0167905979%
Ad 2	2674738	114388.54	0.04276626	0.000037388%	26.22	0.0229218766%
Ad 3	2848816	90113.08	0.031631766	0.0000351023%	15.29	0.0169675701%
Ad 4	2107497	82951.82	0.03936035	0.0000474497%	5.6	0.0067509067%
Ad 5	1087011	77140.49	0.070965694	0.0000919954%	19.32	0.0250452130%

Table 4. Detailed Comparison between an ad's total budget and cost on a user sequence.

2) If the knapsack cannot hold all the items after the greedy algorithm, we have $V_{C_l} > B_{end}$, that is, $B_{end} < V_{C_l} \leq (1-\lambda)B$. According to Formula 13, we have

$$\begin{aligned} K(V_C, V_G, B) &\geq \frac{B - B_{end}}{B} K^*(V_C, V_G, B) \\ &> \frac{B - (1-\lambda)B}{B} K^*(V_C, V_G, B) \\ &= \lambda K^*(V_C, V_G, B) \end{aligned} \tag{14}$$

Therefore, in theory, the greedy solution in our online advertising settings is λ approximately optimal and the λ is much greater than 99.9% in our case.

B.2. Regretless Optimal Bidding Strategy b_t^*

Theorem 3. *During the online bidding phase, the bidding agent can always set the bid price as:*

$$\begin{aligned} \mathbf{b}_t^* = \Bigg[&\left(\frac{Q_G(s, \widehat{a}_t = 1)}{\text{CPR}_{\text{thr}}^*} - Q_C^{\text{next}}(s, \widehat{a}_t = 1) \right) \\ &- \left(\frac{Q_G(s, \widehat{a}_t = 0)}{\text{CPR}_{\text{thr}}^*} - Q_C^{\text{next}}(s, \widehat{a}_t = 0) \right) \Bigg] \end{aligned} \tag{15}$$

where $Q_C^{next}(s, \widehat{a}_t) = \mathbb{E}[\sum_{k=t+1}^{T_j} c_k | s, \widehat{a}_t, \pi_j]$. $\boldsymbol{b}_t^*$ is a regretless optimal bidding strategy without any loss of accuracy.

Proof. Since $\textbf{bid}_t^{\textbf{2nd}}$ is unknown until the current auction is finished, we prove the regretless of $\mathbf{b}_t^*$ from the following two cases:

1) If $\mathbf{b}_t^* > \textbf{bid}_t^{\textbf{2nd}}$: $\mathbf{b}_t^* > \textbf{bid}_t^{\textbf{2nd}} \Leftrightarrow Q(s, \widehat{a}_t = 1) > Q(s, \widehat{a}_t = 0)$, which means the agent should take action $\widehat{a}_t = 1$ in this case. Exactly, $\mathbf{b}_t^*$ is greater than the second highest price $\textbf{bid}_t^{\textbf{2nd}}$ based on the condition for entering the current branch. Thus, the agent will always win the auction and the executed action is indeed $\widehat{a}_t = 1$.

2) If $\mathbf{b}_t^* \leq \textbf{bid}_t^{\textbf{2nd}}$: $\mathbf{b}_t^* \leq \textbf{bid}_t^{\textbf{2nd}} \Leftrightarrow Q(s, \widehat{a}_t = 1) \leq Q(s, \widehat{a}_t = 0)$, which means the agent should take action $\widehat{a}_t = 0$ in this case. Exactly, $\mathbf{b}_t^*$ is less than the second highest price $\textbf{bid}_t^{\textbf{2nd}}$ according to the condition. Thus, the agent will always lose the auction and the executed action is indeed $\widehat{a}_t = 0$.

Thus, we complete the proof.

B.3. Convergence Analysis of *MSBCB*

The overall framework of *MSBCB* can be described as follows:

(1) Let the budget constraint of an advertiser be B. Given a CPR$_{\text{thr}}$, we can use reinforcement learning algorithms to ensure that each user i is optimized according to $\pi_i^* := \text{argmax}_{\pi_i} [V_G(i|\pi_i) - \text{CPR}_{\text{thr}} * V_C(i|\pi_i)]$ and converges to the optimal policy π_i^* under the current CPR$_{\text{thr}}$. Further, picking all users whose CPR$_i \geq$ CPR$_{\text{thr}}$ will result in a total cost of B' (i.e., the advertiser spends a budget B').

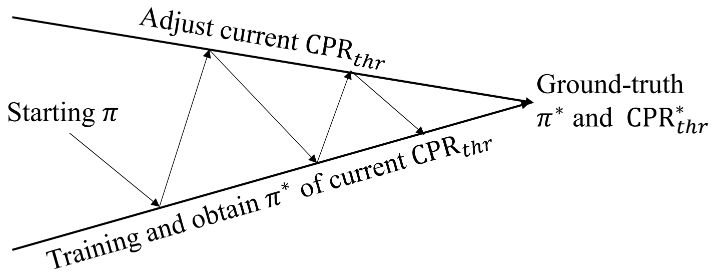

Adjust current CPR_{thr}
Starting π
Ground-truth
π^* and CPR^*_{thr}
Training and obtain π^* of current CPR_{thr}

Figure 10. Convergence demonstration of *MSBCB*

(2) As the current estimated threshold CPR_{thr} might have some bias from the optimal $\text{CPR}^*_{\text{thr}}$, B' may not equal to the budget B. Thus, we design a PID controller to dynamically adjust the estimated $\text{CPR}^*_{\text{thr}}$ so as to minimize the gap between the budget constraint B and the actual feedback of the daily cost B'.

As described in Figure 10, *MSBCB* repeats the above two steps iteratively. Given an updated CPR_{thr}, each π will be optimized by the lower-level reinforcement learning algorithms and π will move towards the optimal π^*. As a result, users whose optimized $\text{CPR}_i \geq \text{CPR}_{\text{thr}}$ will be selected and we get the daily cost B'. Then, the current CPR_{thr} will be updated so that the gap between the cost B' and the budget B will be further minimized. Thus, CPR_{thr} will move towards the optimal $\text{CPR}^*_{\text{thr}}$ gradually. As long as the learning rates of π and CPR_{thr} are small enough, the overall iterations will finally converge. In this paper, we also validate the convergence of our *MSBCB* in the experiments. As shown in Section 4.2 of the paper, our method converges quickly and finally reaches an approximation ratio of 98.53%.

C. Deployment

Here we give the online deployment details of our *MSBCB*.

C.1. Myopic to Non-Myopic Advertising System Upgrade Solution

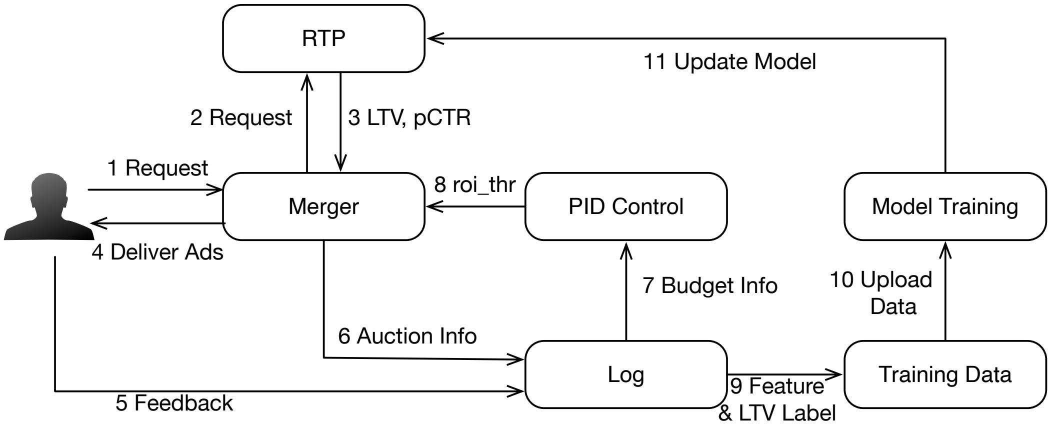


Figure 11. Online System

A myopic advertising system includes several key components as Figure 11 shows: (1) Log module collects auction information and user feedback. (2) Training data are constructed based on log followed by model training with offline evaluation. (3) Real-time prediction (RTP) module provides service for myopic value prediction of user-ad pairs. RTP periodically pulls newly trained models. (4) Merger module receives the user visit, requests RTP for myopic value with which ad bid adjustment ratios and ranking scores are calculated (In advertising, ranking score is $ecpm = pCTR * bid$ where $pCTR$ is predicted Click Through Rate and bid is the bidding price). Finally, top-scored ads are delivered to the user. Above myopic advertising system can upgrade to a non-myopic system by considering the following key changes.

(1) Log module needs to keep long-term auction information and users' feedback, and these data are used to construct features and long-term labels for training. Besides, logged data have to track each advertised item's budget and current cost data which are fed to a PID control module to compute CPR_{thr} for users selection in Merger. (2) Model training can use Monte Carlo (MC) or Temporal Difference (TD) methods. For MC, the long-term labels are cumulative rewards of a sequence and the training becomes a supervised regression problem. For TD, one-step or multi-step rewards are used to compute a bootstrapped long-term value using a separate network for training. (3) RTP module should periodically pull both myopic and non-myopic newly trained models and provide corresponding value prediction service. (4) Merger maintains an $< \text{item}, \text{CPR}_{\text{thr}} >$ table which is updated periodically from PID module. When a user visit comes, Merger requests RTP for both $pCTR$ and long-term values (long-term GMV i.e. V_G and $cost$ i.e. V_C in our paper), and with CPR_{thr} decides the selection of current user and bid adjustment.

C.2. Long-Term Value Prediction Model

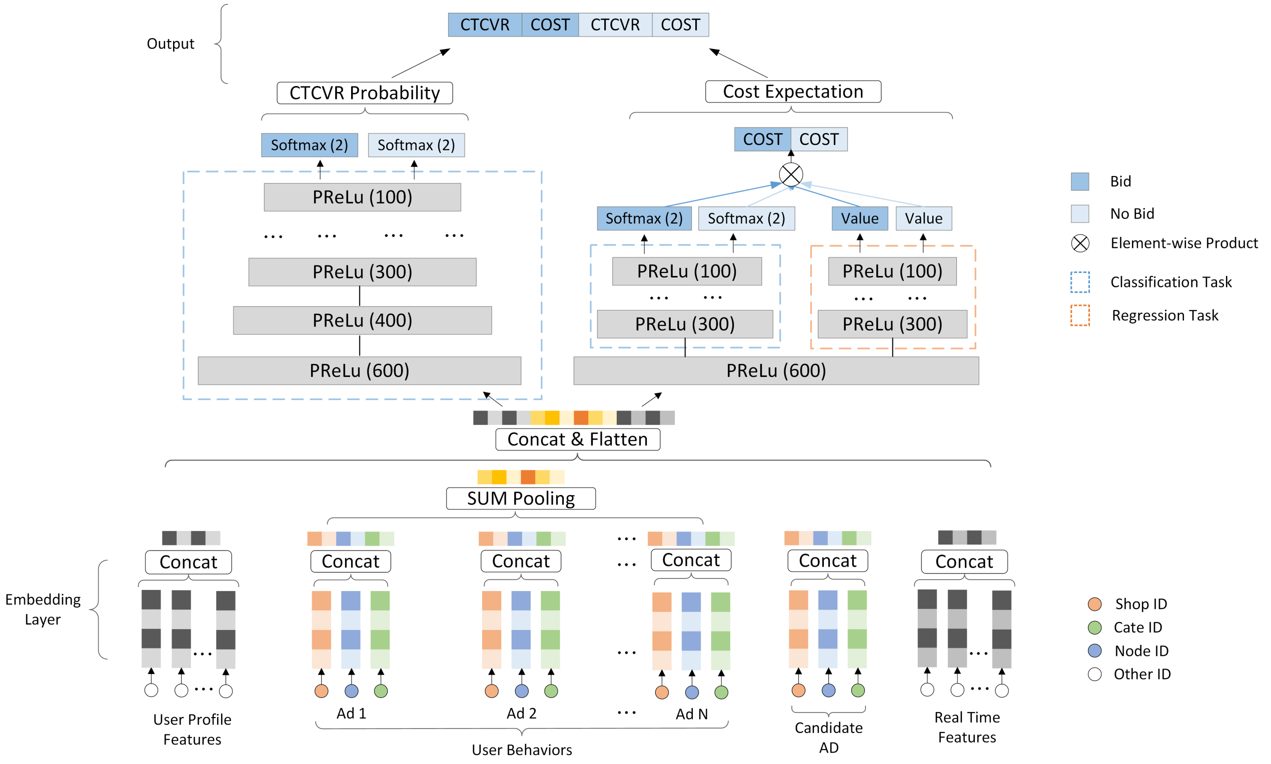


Figure 12. Long-Term Value Prediction Model

C.2.1. Features and Labels

Features for long-term value prediction should contain sufficient user's static profile and historical behavior information. Most myopic advertising systems already have a sound feature system which can summarize user-oriented, ad-oriented and user-ad interactive history very well. Besides, due to the large amount of data collected by the online advertising system, these features are able to generalize across large number users where each user-ad pair's interaction is considered as a separate MDP, thus, help the prediction model learning. To be specific, the state s_t at step t includes: 1) user profile features; 2) user behavior features; 3) real-time user behavior features; 4) context features; 5) user-ad interaction histories; 6) user feedback before current step t and so on. Features are constructed based on past 7-14 days data before user visit time t. For the MC training method, labels are constructed using the following 7 days data after user visit time t. For the TD method, labels are the instant rewards at time t and the long-term labels are constructed using a bootstrap method.

C.2.2. Model Architecture

The long-term value model architecture is shown in Figure 12, where the model takes the features as input and output long-term value of GMV (i.e. V_G in our formulation) and $cost$ (i.e. V_C in our formulation) for both action=1 (display the ad) and action = 0 (do not display the ad).

We use one model to output multiple long-term values (GMV and $cost$ for action=1 and action=0). Multiple prediction tasks share the same bottom layers because we consider the underlying knowledge of the user's sequence behaviors such as opening the app, jumping across channels, turning off the phone and revisiting the app should be learned together and shared. The shared layer converts input features to embeddings and embeddings in the same group are concatenated. The user-behavior group embeddings are then pooled with sum operation. User-profile embeddings, user-behavior embeddings, candidate ad embeddings, and real-time features are finally concatenated and flattened as the output of the bottom layers.

Following the shared bottom layers, the network is split into two forward-pass branches where one is for long-term GMV prediction and one for long-term cost prediction. We find this two-branch design can reduce the influences among different tasks and stabilize the learning. For the long-term GMV prediction, since each user usually buys a commodity only once, we only have to predict $P(buy > 0|feature)$ denoted as $CTCVR$. In the online inference phase, the long-term GMV is computed with $GMV = P(buy > 0|feature) * item\_price$ where $item\_price$ is the price of the commodity. For the long-term cost prediction, in CPC (Cost-Per-Click) advertising, a user usually clicks several times before buys and the cost per click along with each click varies, thus, the long-term $cost$ prediction cannot be decomposed as the long-term GMV prediction and the only way is to regress the long-term cost value. However, as most sequences' costs are zero, the direct regression learning process will be very noisy. Therefore, we design an additional hidden layer to compute $P(cost > 0|feature), P(cost = 0|feature)$ and $E(cost|cost > 0, feature)$. Then, the predicted long-term cost is computed as $pcost = P(cost > 0|feature) * E(cost|cost > 0, feature) + P(cost = 0|feature) * 0 = P(cost > 0|feature) * E(cost|cost > 0, feature)$ where $P(cost > 0|feature)$ and $P(cost = 0|feature)$ are learned using logistic regression loss and $pcost$ is learned using mean-square error loss $(pcost - cost)^2$. We find the above designs help improve the model's prediction performance in practice. For $CTCVR$ and $P(cost > 0|feature)$, $P(cost = 0|feature)$, we use GAUC (Zhou et al., 2018) as metric, and for $pcost$ regression, we use mean-square error and reverse order metrics.

D. Empirical Evaluation: Supplementary of Offline Experiments

D.1. Experiments Settings.

Considering the potential losses of assets and money, it's usually forbidden to do a lot of trial and error and thoroughly comparisons between available baselines in a live advertising system. Thus we implement a fairly general simulation environment so that we could make extensive analyses of our approach. All experiments are conducted on an Intel(R) Xeon(R) E5-2682 v4 processor based Red Had Enterprise Linux Server, which consists of two processors (each with 16 cores), running at 2.50GHz (16 cores in total) with 32KB of L1, 256 KB of L2, 40MB of unified L3 cache, and 128 GB of memory and 2 Tesla M40 GPUs.

D.2. Simulation Environment.

Here, we give the detail of the simulation environment. Similar to (Ie et al., 2019), the simulation environment includes the following 5 modules:

- *Advertisements and Users Interests Model:* We assume a set of ads $\mathcal{D}$ representing the content available for advertising. We also assume a set of commodity categories $\mathcal{T}$ that capture fundamental characteristics of users interest to the ad; we assume categories are indexed $1, 2, ...|\mathcal{T}|$. Each commodity $d \in \mathcal{D}$ has an associated user interest vector $\mathbf{d} \in [0, 1]^{|\mathcal{T}|}$, where d_j is the degree to which d reflects user interest j. Each ad $d \in \mathcal{D}$ also have an inherent quality $Q_d \in [0, 1]$, representing the category-independent attractiveness to the average user.
- *Consumer Interest and Satisfaction Model:* Each user i has various degrees of interests in categories, ranging from 0 (completely uninterested) to 1 (fully interested), with each user i associated with an interest vector $\mathbf{u} \in [0, 1]^{|\mathcal{T}|}$. Consumer i's interest in advertisement d is given by the dot product $I(u, d) = \mathbf{ud}$. We assume some prior distribution P_u over user interest vectors, but user i's interest vector is dynamic, i.e., influenced by their advertisement consumption (see below). Besides, a user's satisfaction $S(u, d)$ with a consumed (viewed) advertisement d is a function $f(I(u, d), Q_d)$ of user i's interest and ad d's quality. Here, we assume a simple convex combination

$S(u, d) = (1 - \alpha)I(u, d) + \alpha Q_d$. Satisfaction influences user dynamics as we discuss below.

- *Consumer Choice Model:* The user's Click-Through Rate (CTR) and Conversion Rate (cvr) are represented by $I(u, d)$ and $S(u, d)$ respectively. Each user has the probability of clicking and buying an advertising commodity according the CTR and CVR.
- *Consumer Dynamics:* We assume that a user's interest evolves as a function of the ads consumed (viewed). When user i consumes ad d, her interest in category $T(d)$ is nudged stochastically, biased slightly towards increasing her interest, but allows some chance of decreasing her interest. In this paper, we set $\mathbf{u} \leftarrow \gamma\mathbf{u} + \beta * S(u, d) * \mathbf{d}$, where γ is the interest decay rate and $\beta \in [-1, 1]$ is a user independent parameter.
- *Consumer Visiting Model and Advertising System Dynamics:* The users' request sequence are generated from a stable distribution P_{req}. For each user's request, all advertisements $d \in \mathcal{D}$ give a bid and competes with other bidders in real-time. The winner has the privilege to display its ad to the user, which could further influence the user's interest and behavior.

D.3. Codes.

The codes to reproduce our offline experiments are provided here.

D.4. Cost Comparison.

The consumption of budget during the training process is shown in Figure 13. As we can see, the costs of all approaches converge to about 12000, which is exactly equal to the budget we set in experiments. Specific costs of each approach can be found in Table 1 of paper.

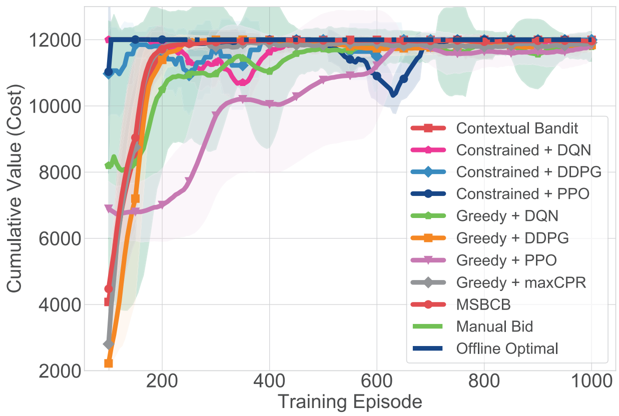
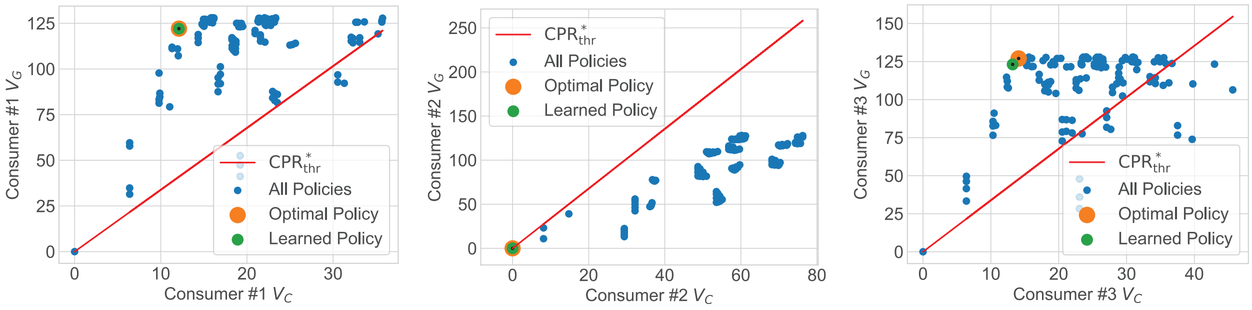


Figure 13. The learning curves of costs of our *MSBCB* and the other baseline approaches.

D.5. Convergence Analyses

D.5.1. CONVERGENCE OF EACH π_i^* GIVEN ANY CPR$_{\text{THR}}$.

As shown in Figure 14, given a CPR$_{\text{thr}}$, the learned advertising policy π of our *MSBCB* converges to the optimal π_j^*. In Figure 14, the x-axis denotes the cumulative cost, the y-axis denotes the cumulative value and the dots in blue represent the cumulative values and costs of all possible policies for each user. The red line represents $y =$ CPR$_{\text{thr}} * x$, whose slope is CPR$_{\text{thr}}$. The orange point represents the optimal policy π_i^* computed by enumerating all possible policies (blue points) and finding the one which maximize $V_G(i|\pi_i) -$ CPR$_{\text{thr}} * V_C(i|\pi_i)$ according to **Theorem 1**. The green point denotes the learned policy of *MSBCB*. *In theory, the point of the optimal policy is the one whose CPR $>$ CPR$^*_{thr}$ and vertical distance is the farthest from the red line.* A proof is provided in the **Theorem 4** in the later part. We present 3 convergence examples of different types in Figure 14. In Figure 14 (a) and (b), the learned π by the RL algorithm is exactly the same with the optimal π^*. In Figure 14 (b), the optimal policy is do not advertise to this user. In Figure 14 (c), the learned π is approximately optimal. Detail convergence statistics on the proportion of users whose policies converged to the optimal ones among all users are shown in Table 5. For each user, we denote the vertical distance of the learned policy to the



(a) user #1's learned policy. (b) user #2's learned policy. (c) user #3's learned policy.

Figure 14. Three examples of the convergence of each π_i^* given a fixed CPR_{thr}.

CPR^*_{thr} line as $\text{dis}^*_{learned}$ and the vertical distance of the optimal policy π^* to the CPR^*_{thr} line as $\text{dis}^*_{optimal}$. We denote $\text{R}^*_{opt} = \text{dis}^*_{learned}/\text{dis}^*_{optimal}$ as the approximation ratio. According to **Theorem 4**, if the R^*_{opt} is 100%, then the learned strategy is exactly the optimal strategy. Otherwise, we denote that the learned strategy is the R^*_{opt}-approximation strategy. As shown in Table 5, there are 74.9% policies achieve more than 90%-approximation ratios and 53.3% policies achieve exactly the optimal.

Table 5. Optimal types of each π_i^* of 10000 users

R^*_{opt}	100%	[90%, 100%)	[0%, 90%)
Percentage	53.3%	21.6%	25.3%

Theorem 4. *The point of the optimal policy is the one whose CPR $>$ CPR^*_{thr} and vertical distance to CPR^*_{thr} line (red line) is the farthest among all policy dots in Figure 14.*

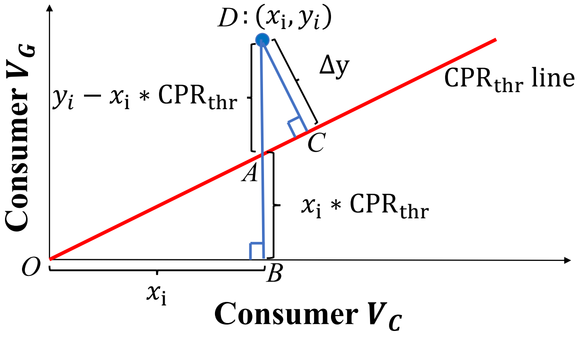

$D: (x_i, y_i)$
Δy
CPR_{thr} line
$y_i - x_i * \text{CPR}_{\text{thr}}$
C
A
$x_i * \text{CPR}_{\text{thr}}$
Consumer V_G
O
B
x_i
Consumer V_C

Figure 15. Proof of Optimal Policy Dot

Proof. Here we give a simple proof of **Theorem 4**. As we can see in Figure 15, the blue dot $D : (x_i, y_i)$ is an arbitrary policy i in Figure 14. Suppose the vertical distance of D to CPR^*_{thr} line (red line) is Δy_i (segment DC in the figure). We then draw a vertical line of x-axis from dot D to dot B. We can then calculate the length of segments: $OB = x_i$, $BA = x_i * \text{CPR}_{thr}$, $DA = y_i - x_i * \text{CPR}_{thr}$. It's evident that $\triangle OAB \sim \triangle DAC$, which means $\frac{DC}{OB} = \frac{DA}{OA} = \frac{DA}{\sqrt{(OB)^2+(AB)^2}}$. We can derive that

$$\frac{\Delta y_i}{x_i} = \frac{y_i - x_i * \text{CPR}_{thr}}{\sqrt{x_i^2 + (x_i * \text{CPR}_{thr})^2)}} \tag{16}$$

As $x_i > 0$, we can further derive that

$$\Delta y_i = \frac{y_i - x_i * \text{CPR}_{thr}}{\sqrt{1 + \text{CPR}_{thr}^2}} \tag{17}$$

Suppose the dot of a policy is (x^*, y^*), which has farthest vertical distance Δy^* from the CPR_{thr}^* line, that is, for a dot of arbitrary policy i, we have $\Delta y_i \leq \Delta y^*$. According to Equation 17, we have

$$\frac{y_i - x_i * \text{CPR}_{thr}}{\sqrt{1 + \text{CPR}_{thr}^2}} \leq \frac{y^* - x^* * \text{CPR}_{thr}}{\sqrt{1 + \text{CPR}_{thr}^2}} \tag{18}$$

Then we get $y_i - x_i * \text{CPR}_{thr} \leq y^* - x^* * \text{CPR}_{thr}$, which means (x^*, y^*) is the dot of the optimal policy. Thus, we complete the proof.

D.5.2. Convergence of $\text{CPR}_{\text{THR}}^*$.

In Figure 16, we plot the learning curves of the CPR_{thr} of our *MSBCB* as well as 3 RL approaches. The dotted blue line denotes the optimal $\text{CPR}_{\text{thr}}^*$ computed by the *MSBCB (enum)* of Table 1 of paper. Figure 16 shows that the learned CPR_{thr} of our *MSBCB* could gradually converge to the optimal $\text{CPR}_{\text{thr}}^*$ approximately, which is much better than the other 3 RL approaches.

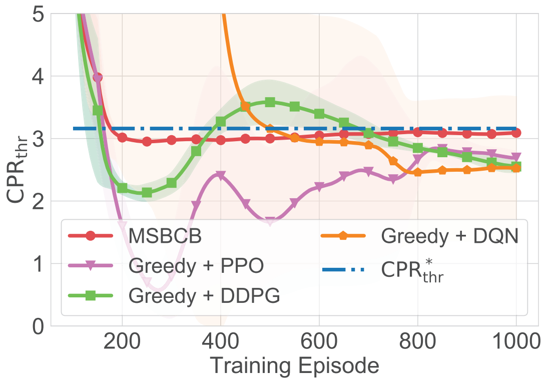


Figure 16. The convergence of $\text{CPR}_{\text{thr}}^*$

D.6. Gap to Market Second Price.

Figure 17 shows average gaps between the bid of the agent of different approaches and the second price in the auction. Results indicate that the bid prices given by the *MSBCB* agent are closer to the second price in the auction, which can reduce the risk of economic loss when the market price fluctuates.

D.7. Effectiveness of Action Space Reduction.

Here we give a more detailed comparison of *MSBCB* and RL baselines to demonstrate the effectiveness of action space reduction. As shown in Figure 18 and Table 6, *MSBCB* (with action space reduction) can reach exactly the same cumulative value much more quickly than the other 3 RL baselines. *MSBCB* can reach a cumulative value of 85000 in only 104 epochs, which proves that action space reduction can effectively improve the sample utilization to converge to higher performance with faster speed.

E. Empirical Evaluation: Supplementary of Online A/B Testing

In online A/B Testing, we conduct further analyses to verify the effectiveness of our *MSBCB* and find out whether our approach could benefit most advertisers.

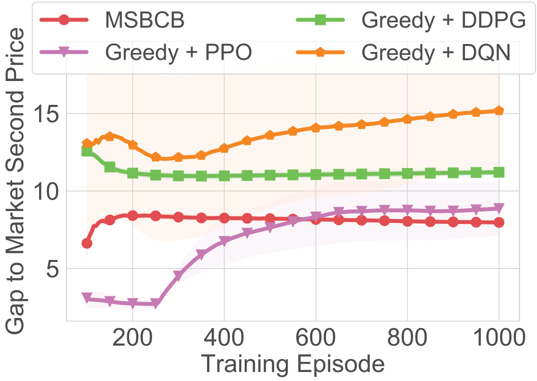


Figure 17. The average gaps of the bids to the second prices in the auction by different methods.

Table 6. The training epochs and the number of samples needed by different approaches when achieving the same revenue level.

Cumulative Value	60000		65000		70000		75000		80000		85000	
Method	#Epoch	#Samples	#Epoch	#Samples	#Epoch	#Samples	#Epoch	#Samples	#Epoch	#Samples	#Epoch	#Samples
Greedy + PPO	251	1280000	299	1530880	776	3973120	817	4183040	-	-	-	-
Greedy + DDPG	68	343040	76	389120	92	471040	154	788480	853	4362240	-	-
Greedy + DQN	90	455680	109	558080	153	783360	373	1909760	754	3855360	-	-
MSBCB	**22**	**112640**	**33**	**163840**	**48**	**245760**	**61**	**312320**	**71**	**363520**	**104**	**532480**

Firstly, we analyze the performance of our *MSBCB* for each advertiser. To guarantee the statistical significance, only the advertisers with more than 100 conversions in a week are included. The detail results of top-10 advertisers with the largest costs are shown in Table 7. In Table 7, under the same budget constraint, our *MSBCB* can increase the Revenues and ROIs of most advertisers compared with the myopic *Contextual Bandit* approach. Although the ROI of advertiser 7 drops slightly, our *MSBCB* contributes to much more PVs (Page Views).

Besides, in Figure 19, we give the detail proportions of advertisers whose ROIs are improved. Among all advertisers, 85.1% advertisers obtain positive ROI improvements while the rest of 14.9% advertisers are in the so-called quantity and quality exchange situations: their PV increments are larger than the ROI drops. We say that its also acceptable for some advertisers because the PV increments might lead to secondary exposures to an advertiser and thus lower the ROI within the current time period. But the increase in PV may leave deeper impressions to the users and contribute to the long-term future revenues. In addition, Figure 19 demonstrates that our *MSBCB* can be well applied to the multi-agent setting (which involves multiple advertisers) in the real-world auction environment, which could increase the overall revenue for most advertisers.

In order to highlight the advantage of our method in long-term revenue optimization, we compared the average number of

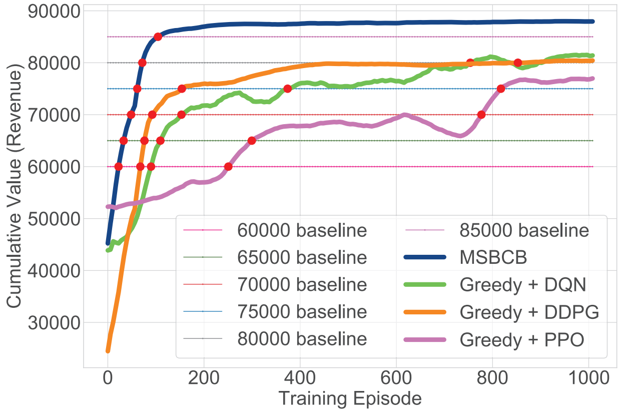


Figure 18. The comparison of the number of training episodes needed by different approaches when achieving the same revenue level.

	Revenue	Cost	CVR	PV	ROI
Advertiser 1	5.1%	-6.3%	17.2%	9.6%	12.2%
Advertiser 2	7.5%	2.1%	5.2%	12.2%	5.3%
Advertiser 3	48.6%	10.9%	27.6%	28.9%	33.9%
Advertiser 4	3.1%	2.8%	1.1%	9.6%	0.3%
Advertiser 5	12.7%	1.7%	12.9%	17.8%	10.8%
Advertiser 6	10.8%	2.2%	4.4%	13.8%	8.4%
Advertiser 7	1.9%	3.8%	4.6%	31.5%	-1.8%
Advertiser 8	5.6%	-4.8%	2.9%	10.7%	11.1%
Advertiser 9	6.7%	-2.4%	6.3%	21.0%	9.4%
Advertiser 10	5.8%	-0.8%	2.5%	8.0%	6.7%

Table 7. The improvements in Revenue, CVR, PV and ROI of our *MSBCB* compared with the myopic *Contextual Bandit* method.

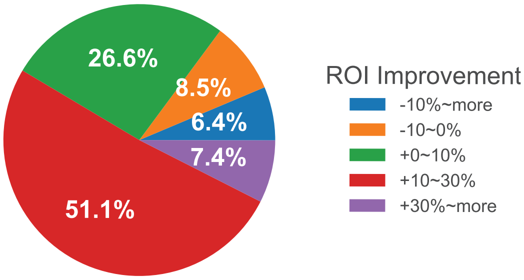


Figure 19. The distribution of ROI improvements for all advertisers of our *MSBCB* compared with the myopic *Contextual Bandit* method.

times (we call the sequence length) that a user contact with an advertisement under different approaches. Figure 20 shows the extent of *MSBCB*'s improvement relative to *Contextual Bandit* in the proportion of the user sequence length. The results show that our *MSBCB* can increase the proportion of the sequences with larger sequence length. Especially, the ratio of sequence length of 7 is increased by nearly 30%. It shows that our method can promote longer user behavior sequences, and longer user behavior sequence means more opportunities to affect the user's mentality towards an advertisement, thereby improving the long-term revenue for an advertisement.

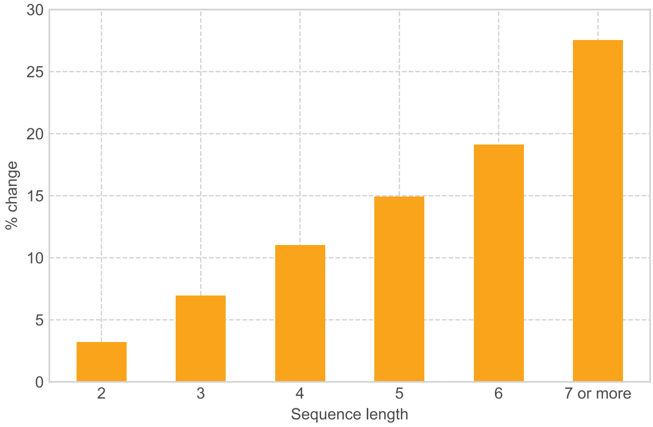


Figure 20. The proportion improvements in the sequence length of our *MSBCB* compared with the myopic *Contextual Bandit* method.

Further, we also analyze the ROI performances of the compared 3 algorithms (i.e., *CEM*, *Contextual Bandit* and our *MSBCB*) in different channels. Figure 21 shows the budget allocation distributions of all approaches among 6 channels and the corresponding ROIs. The left axis represents the ROI, and the ROI performances of each algorithm among different channels are given by the corresponding bar charts. The right axis represents the increments or decrements of the actual costs of *MSBCB* and *Contextual Bandit* relative to *CEM*, which are indicated by the line charts. In Figure 21, we observe the

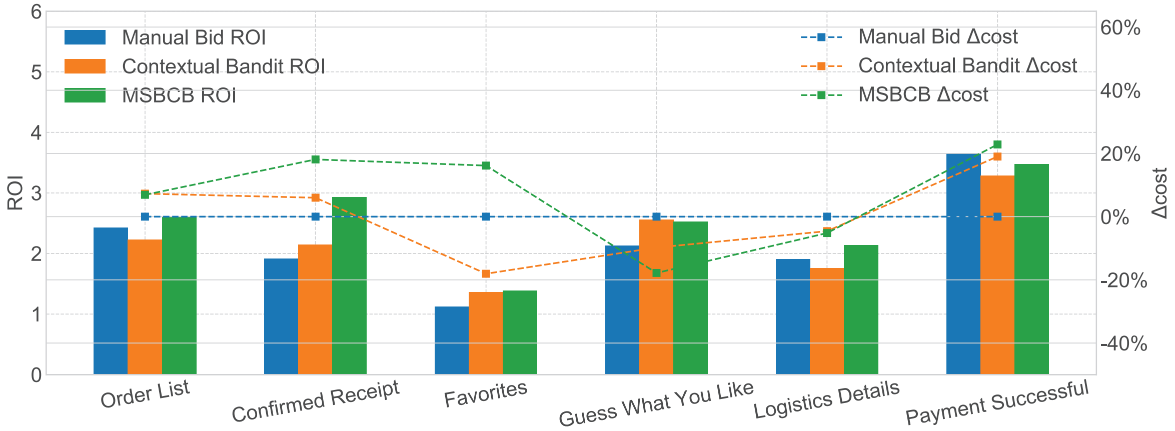


Figure 21. ROI and budget allocation among different channels.

following two phenomena:

1) *MSBCB* and *Contextual Bandit* both spend more budgets on channels with higher ROIs, especially on the *Payment Successful* channel, where the average ROI is much higher.

2) Compared with *Contextual Bandit*, *MSBCB* allocates more budget from the *Guess What You Like* channel to other channels, especially the *Favorites* channel, *Confirmed Receipt* channel and the *Payment Successful* channel.

These phenomena show that our *MSBCB* can reasonably allocate budgets among different channels and spend more budgets in channels with higher ROIs. In addition, compared with the myopic method *Contextual Bandit*, our long-term *MSBCB* is more optimistic about channels during and after purchasing, which shows that our *MSBCB* prefers a longer interaction sequence to optimize cumulative long-term values.

References

Dantzig, G. B. Discrete-variable extremum problems. *Operations research*, 5(2):266–288, 1957.

Ie, E., Jain, V., Wang, J., Navrekar, S., Agarwal, R., Wu, R., Cheng, H.-T., Lustman, M., Gatto, V., Covington, P., et al. Reinforcement learning for slate-based recommender systems: A tractable decomposition and practical methodology. *arXiv preprint arXiv:1905.12767*, 2019.

Roberge, M. *The Sales Acceleration Formula: Using Data, Technology, and Inbound Selling to go from* 0*to100 Million*. John Wiley & Sons, 2015.

Zhou, G., Zhu, X., Song, C., Fan, Y., Zhu, H., Ma, X., Yan, Y., Jin, J., Li, H., and Gai, K. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1059–1068. ACM, 2018.