An Auto-Coupon Framework for Advertiser Retention in Online Advertising

Yidan Xing*
School of Computer Science,
Shanghai Jiao Tong University
Shanghai, China
katexing@sjtu.edu.cn

Haoqi Zhang*
School of Computer Science,
Shanghai Jiao Tong University
Shanghai, China
zhanghaoqi39@sjtu.edu.cn

Yangsu Liu
Alibaba Group
Beijing, China
liuyangsu.lys@alibaba-inc.com

Dagui Chen
Alibaba Group
Beijing, China
dagui.cdg@alibaba-inc.com

Zhenzhe Zheng†
School of Computer Science,
Shanghai Jiao Tong University
Shanghai, China
zhengzhenzhe@sjtu.edu.cn

Fan Wu
School of Computer Science,
Shanghai Jiao Tong University
Shanghai, China
fwu@cs.sjtu.edu.cn

Abstract

The online advertising platform serves as the intermediary between advertisers and users to facilitate the product trading process and generate profits. While the vitality of both the user and advertiser community is crucial to the long-term development of platforms, compared with the enduring efforts to improve user experience, little attention has been given to methods for attracting or retaining advertisers. To address this oversight, in this work, we investigate the first intelligent solution to the advertiser retention problem in online advertising. Motivated by advertiser surveys and data analytics, we first define a set of short-term ad performance metrics as the key retention metrics we need to optimize for advertisers. In consideration of the desirable properties of a retention framework, we propose an Auto-Coupon module to distribute impression-level virtual coupons to selected advertisers during the bidding process. We decompose the resulted retention optimization problem into two stages, and adopt feedback control as well as reinforcement learning methods to realize personalized coupon distribution. The offline experiments demonstrate the effectiveness of our methods in boosting the ad performance of churning advertisers, and the large-scale online A/B test further justifies the improvement in advertisers' willingness to stay in the platform.

CCS Concepts

• Applied computing → Electronic commerce; **• Information systems → Online advertising**.

Keywords

Online Advertising; Advertiser Retention; Coupon

ACM Reference Format:
Yidan Xing, Haoqi Zhang, Yangsu Liu, Dagui Chen, Zhenzhe Zheng, and Fan Wu. 2025. An Auto-Coupon Framework for Advertiser Retention in Online Advertising. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3711896.3737191

1 Introduction

The rise of the Internet and mobile apps is driving rapid growth in online advertising. To facilitate its long-term development, it is crucial for an advertising platform to attract more active users and advertisers. The typical strategy of platforms in the last decade is to focus on the user-side development [3, 24, 30] and promotes the advertiser-side development indirectly. However, for the time being, most of the potential users have been involved, and platforms can hardly realize great boosts on user side. For example, the total number of active users on TaoBao has approached 900 million [11], which covers more than 70% of the mobile Internet users in China. In contrast, there still exist great potentials in constructing a larger and more active advertiser community. Facing the situations that more than half of the new advertisers may leave within ten days, the need to hold back the large groups of churning advertisers now becomes an urgent task for the platforms. As advertiser retention benefits the long-term platform development, the platform would like to undertake some monetary cost incurred by retention, as long as the short-term expenditure is bounded. Following this idea, the advertiser retention problem could be defined as *spending a limited budget to improve the total number of active advertisers.*

Despite the clarity of this objective, its relevant evaluation metrics—such as counting the number of active advertisers within a given period—offer limited insight into the underlying causes of advertiser churn or how to address them. To identify the contributing metrics through which we could effectively retain the advertisers, we first analyze the data collected from an advertiser churn survey regarding their intention in advertising. Most of the advertisers decide to leave the ad platform because of the dissatisfaction with

*Equal contribution. Work done during their internship at Alibaba Group.
†Zhenzhe Zheng is the corresponding author.

KDD '25, Toronto, ON, Canada

ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3711896.3737191

their return on investment (ROI) rate and the amount of obtained advertising opportunities, such as page view (PV) and clicks, and a collection of ad performance data of churning and non-churning advertisers in our platform aligns with the above claim. These analysis motivates us to help the advertisers improve the specific short-term performance metrics as the key optimization metrics to promote their willingness to stay in the platform.

Unlike classical optimization problems with well-defined interfaces, provided the key retention metrics, we need to first define the deployment framework to enable the optimization of these metrics. For example, directly allocating more display opportunities and providing coupons in the final bills are two distinct approaches to improve the ad performances of retention advertisers. Although many potential retention approaches might be proposed within the intricate advertising system, a feasible approach should simultaneously satisfy the implementation simplicity, optimization efficiency and other necessary economical properties inherited from ad auction design. Based on these considerations of potential schemes, we propose a plug-in module, called *Auto-Coupon*, as a universal solution for advertiser retention. The Auto-Coupon module distributes virtual coupons to advertisers on the impression level, *i.e.*, lifting their bids on some impressions for free, thereby improving their chances of winning auctions and gaining more impressions at a lower actual price. The Auto-Coupon module satisfies the desirable properties and allows precise control over retention budget.

With Auto-Coupon and impression-level coupons defined as decision variables, we formulate the problem of advertiser retention as a large-scale online optimization problem. This problem is challenging due to the heterogeneity of advertisers, the dynamic nature of online auctions and the real-time implementation requirement. To derive a feasible solution, we decompose this large-scale online optimization problem into two stages: the budget allocation stage to divide the total platform budget between advertisers, and the personalized coupon distribution problem to optimize the individual ad performance metrics under the allocated budget. We derive the theoretically optimal budget allocation method, and present practical allocation methods for online implementation. For the subsequent personalized coupon distribution, we figure out a near-optimal distribution formula with adjustable parameters, and design corresponding feedback control and reinforcement learning methods to control the online coupon distribution process. To validate the performances of our solution, we conduct offline experiments on a large-scale industrial dataset, and deploy the Auto-Coupon module on one representing channel of TaoBao for online A/B test. The experimental results demonstrate that the proposed methods could effectively improve the chosen advertisers' ad performance metrics by 14.0% and 16.5%, and further improve the number of active advertisers by 3.2%.

We summarize the contributions of this work as follows:

- We consider the advertiser retention problem to increase the number of active advertisers through improving their ad performances under platform budget constraint, which is underexplored in both academia and industry.
- We propose the design of Auto-Coupon module in online advertising system as a universal framework for advertiser retention. To optimize the retention effects, we decompose the auto-coupon optimization problem into two stages, platform budget allocation and personalized coupon distribution. We derive the near-optimal formula in personalized coupon distribution, and design the auto-coupon agents based on feedback control and reinforcement learning methods to adjust coupon distribution online.
- We conduct both online and offline experiments to validate the effectiveness of our two-stage auto-coupon optimization scheme. Besides the improvement in ROI and PV, our online experimental results further report the improvement in advertisers' willingness to investment, which demonstrates its effects to retain the churning advertisers.

2 Motivation

Before presenting our design of Auto-Coupon, we first discuss the underlying design choices of advertiser retention framework.

2.1 Metrics for Advertiser Retention

As we could not directly optimize the number of active advertisers in retention problem, to characterize appropriate intermediate metrics, we should first analyse why advertisers may decide to leave the platform. Our analysis combines the survey results from previous work [13] and advertiser performances data collected from TaoBao. The authors in [13] revealed that advertiser satisfaction mainly depends on their daily ad performances, including ROI, conversion rate (CVR), click through rate (CTR), page views (PV), and other complex factors. Among them, ROI plays the most crucial role since it directly reflects the cost-efficiency of advertising investment, while the other metrics are mutually related and jointly contribute to the ROI. As ROI has reflected the overall quality of the obtained ad display opportunities, to reach larger advertising impacts, the advertisers should also increase the amount of ad opportunities. To avoid being distracted by various intertwined metrics, we choose to adopt the amount of page views, the foremost step in interaction with users, as the representative metric. That is, ROI and PV, viewed as average quality and total quantity of ad traffic, are the key metrics the platform could improve to promote advertiser retention.

We further validate our selected metrics on the advertiser performances data through comparing the daily ad performances between churning advertisers and non-churning advertisers. Our data includes 11,776 churning advertisers and 21,089 non-churning advertisers with comparable daily cost. Prominently, ad performance of churning advertisers is much worse than that of non-churning advertisers, with 45.55% worse in ROI and 33.76% worse in PV. From all the data and analysis above, it is reasonable to transform the objective of improving long-term intractable retention metrics to improving the two short-term advertising performance metrics, *i.e.*, ROI and PV. Specifically, we aim to directly help advertisers optimize the ROI, and improve the PV simultaneously.

2.2 Properties of Retention Framework

To design a feasible retention framework, we also need to consider its additional impacts on the current online ad system. Combining relevant aspects, its desired properties could be listed as follows.

(1) *Optimization efficiency*: allow to boost the ad performances for certain group of selected advertisers in different aspects;

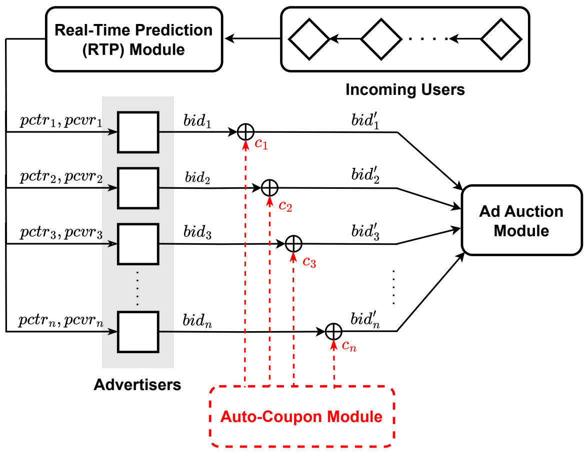

Figure 1: Advertising System with Auto-Coupon Module

(2) *Bounded budget*: the short-term revenue loss due to advertiser retention is well controlled;
(3) *Implementation simplicity*: introduce minor modifications and low additional time delay to the online advertising system;
(4) *Necessary economical properties*: preserve incentive compatibility (IC) and individual rationality (IR) [22] of ad auctions, when the advertiser retention affects the final allocation outcomes.

Although the listed properties are natural and reasonable, reaching a feasible solution to satisfy them simultaneously is non-trivial. Before our design of Auto-Coupon, we have proposed two competitive baseline schemes that are not finally adopted because of their relative defectives or violation of the above properties. The first baseline scheme is to directly adding coupons into the total bills of advertisers, which is motivated by the conventional retention approaches in the e-commerce platform. However, in online advertising, the churning advertisers may not be able to utilize the coupon well in the campaigns, leading to waste of coupon budget and inefficiency in retention effects. The second baseline scheme is to provide extra coupons to users when they purchase products from the retention advertisers, with the resulted revenue loss afforded by the platform. Although coupons in user side could produce stimulus in conversion, its retention effects cannot help with the upstream process of online advertising. For instance, advertisers with low CTR or poor abilities in getting display opportunities can hardly gain better ad performances. In contrast, our design of Auto-Coupon avoids these problems and meets the desired properties, which would be demonstrated in the next section.

3 System Model

3.1 Online Advertising with Auto-Coupon

The typical working procedure of an online ad system could be described as follows: when each user comes, the platform holds an auction for advertisers to compete the ad display opportunities. The advertisers estimate the value of this opportunity, and make bidding based on her valuation on the impression, which may depend on the predicted CTR (pCTR) and predicted CVR (pCVR). The ad auction module collects the submitted bids, and decides the winning ads and payments based on auction rules.

The Auto-Coupon module is designed to work between advertisers and the auction module, as illustrated in Figure 1. It distributes virtual coupons to chosen advertisers, namely increase their bids and reduce the payments if they win the auctions. Under this design, the interactions between Auto-Coupon module and the original ad system only involve addition and subtraction on the original bids and payments, which is simple and friendly to the system deployment. As the virtual coupons could influence auction outcomes in the impression level, the platform has the flexibility to design the detailed coupon distribution algorithms for various objectives.

3.2 Mechanism Properties and Global Optimization Problem

Now, we formalize the advertising process and relevant properties of Auto-Coupon module. There are N advertisers, denoted as $i \in \{1, \ldots, N\}$, with $n \leq N$ advertisers need retention (assumed as the first n advertisers). The advertisers compete for M impressions $j \in \{1, \ldots, M\}$, where each impression has K displaying slots.

The platform adopts the well-known generalized second-price (GSP) auctions with rank scores as the ad auction mechanism [7, 20], whose working procedure is as follows: We denote the bid of advertiser i on impression j as $b_{i,j}$, and the ad quality as $q_{i,j}$, where $q_{i,j}$ might be estimated by the platform based on the $\text{pctr}_{i,j}$ and $\text{pcvr}_{i,j}$ of advertiser i on impression j. For each impression j, the GSP auction ranks the advertisers based on the rank score $b_{i,j} \times q_{i,j}$. The k^{th} highest rank score for impression j, the corresponding advertiser and the ad quality are denoted as s^j_k, a^j_k and q^j_k, respectively. The GSP auction then allocates the k^{th} slot to advertiser a^j_k and charges payment $p^j_k = s^j_{k+1}/q^j_k$ for $k \in \{1, \ldots, K\}$ following the cost-per-mille (CPM) pricing, where payment is charged once the ad get displayed. While our main results are presented under the CPM context, they could be generalized to the cost-per-click (CPC) pricing [35] setting up to the change of one pCTR constant.

With Auto-Coupon module enabled, when virtual coupon $c_{i,j}$ is distributed to advertiser i on impression j, her rank score will become $(b_{i,j} + c_{i,j}) \times q_{i,j}$, and if she wins the k^{th} slot, her payment will be $(s^j_{k+1}/q_{i,j} - c_{i,j})^+$, where $(x)^+ = \max(x, 0)$. In other words, the virtual coupons *help the chosen retention advertisers to raise their bids and get better auction outcomes with discounts.* The platform may distribute $c_{i,j} > 0$ only when $b_{i,j} \times q_{i,j} < s^j_K$, *i.e.*, when the original bids of the advertiser is unable to win any display opportunity of impression j, and otherwise set $c_{i,j} = 0$.

In this work, we adopt the *value-maximizing* bidder model following [8, 9], *i.e.*, the bidder aims to maximize her value of allocation as long as the assigned payment does not exceed the value, regardless of the amount of payment[1]. As the advertisers typically do not have detailed valuations on getting different slots of an impression in the bidding process, we assume an advertiser i to have a same value for getting any slot on impression j, but strictly prefers to win a higher slot when payment does not exceed her value.

[1]We also discuss the properties of our design under the alternative value-maximizing bidder model that assumes additional payment preference [33] in Appendix A. Besides, we may use the term advertiser with bidder interchangeably in this work.

THEOREM 3.1. *The GSP auction with Auto-Coupon module satisfies IC and IR properties for value-maximizing bidders, when:*

(1) *the virtual coupons $c_{i,j}$ distributed to advertiser i are irrelevant to her own bids $b_{i,j}$ for any impression j; or*
(2) *the virtual coupons $c_{i,j}$ are distributed according to our design in Section 4.*

PROOF. Since the economical property of our Auto-Coupon module might depend on the coupon distribution strategy in Section 4, we delay our proof of part (2) to Appendix A, and only discuss part (1) here. Since advertisers have no control over virtual coupon $c_{i,j}$ for each single auction and are charged according to the GSP rule, any misreporting behavior to win a better allocation would lead the payment to exceed the advertiser's true value, thus preserving the IC property. As the virtual coupon $c_{i,j}$ is simultaneously added to the bid and deducted from resulted payment, the payment would not exceed the original bid and IR properties get preserved. □

As a result, the modification in Auto-Coupon module still provide the economical properties for value-maximizing advertisers, such that preserving the normal market competition order. Furthermore, we could also demonstrate that the platform revenue loss is bounded by the supplied virtual coupon, and we are thus able to control the revenue loss under the platform budget.

THEOREM 3.2. *For $q_{i,j} = 1$ with cost per mille (CPM) pricing or $q_{i,j} = pctr_{i,j}$ with cost per click (CPC) pricing, the platform revenue loss produced by Auto-Coupon module is bounded by $\sum_{j=1}^{M}\sum_{k=1}^{K} c_k^j q_k^j$.*

PROOF. By our assumption on q and payment rule, the expected platform revenue for each impression j is $\mathbf{E}_j = \sum_{k=1}^{K} p_k^j \cdot q_k^j = \sum_{k=1}^{K} s_{k+1}^j$ without Auto-Coupon module. When Auto-Coupon module is enabled, the rank score of some advertisers are affected by its virtual coupon, which might change the rank order of other advertisers. We denote the new k^{th} highest rank score for impression j, its ad quality and corresponding virtual coupon by $\tilde{s}_k^j$, $\tilde{q}_k^j$ and $\tilde{c}_k^j$, respectively. The expected platform revenue for impression j becomes $\tilde{\mathbf{E}}_j = \sum_{k=1}^{K}(\tilde{s}_{k+1}^j/\tilde{q}_k^j - \tilde{c}_k^j)^+ \cdot \tilde{q}_k^j$. Therefore, we may bound the total expected revenue loss of Auto-Coupon module as

$$\begin{aligned}\sum_{j=1}^{M}\mathbf{E}_j - \tilde{\mathbf{E}}_j &= \sum_{j=1}^{M}\sum_{k=1}^{K}\left(s_{k+1}^j - (\tilde{s}_{k+1}^j/\tilde{q}_k^j - \tilde{c}_k^j)^+ \cdot \tilde{q}_k^j\right)\\ &\le \sum_{j=1}^{M}\sum_{k=1}^{K}\left(s_{k+1}^j - (\tilde{s}_{k+1}^j/\tilde{q}_k^j - \tilde{c}_k^j) \cdot \tilde{q}_k^j\right) \\ &= \sum_{j=1}^{M}\sum_{k=1}^{K}(s_{k+1}^j - \tilde{s}_{k+1}^j + \tilde{c}_k^j \cdot \tilde{q}_k^j).\end{aligned} \tag{1}$$

Since virtual coupon is constrained to be non-negative, rank score of every advertiser should be increased or unchanged. As s_k^j are the k^{th} largest rank score for impression j, it should be non-decreasing with respect to each advertiser's rank score. Therefore, $s_{k+1}^j \le \tilde{s}_{k+1}^j$ and we then have $\sum_{j=1}^{M}\mathbf{E}_j - \tilde{\mathbf{E}}_j \le \sum_{j=1}^{M}\sum_{k=1}^{K}\tilde{c}_k^j\tilde{q}_k^j$ [2]. □

[2]We use $\sum_{j=1}^{M}\sum_{k=1}^{K} q_k^j c_k^j$ in the statement of Theorem 3.2 since the main body of our paper assumes the application of Auto-Coupon module.

Based on the optimization interface of Auto-Coupon, we now formulate the global auto-coupon retention optimization problem. As discussed in Section 2, we adopt ROI and PV as the short-term performance metrics in optimization. Considering that value-maximizing bidders who are indifferent on detailed payment but pursuing higher advertising revenue have become the mainstream of online advertising, we define ROI as $\mathrm{ROI}_i = \left(\sum_{j=1}^{M}\mathrm{pctr}_{i,j}\cdot\mathrm{pcvr}_{i,j}\right)\cdot \mathrm{rev}_i/B_i$ in our context, where rev_i is advertiser i's revenue income for one conversion, and B_i is advertiser i's campaign budget. Since the increase of ROI is proportionally equivalent to increasing the conversions, we will represent our problem in terms of conversions for ease of notations. We assume each advertiser has a *retention threshold* for her unsatisfied ad performance metric, such that she is fully satisfied when her ad performance is above the threshold; and when the ad performance is below the threshold, her willingness to stay increases in proportion to the increase in this metric. We use thr_i to denote the retention threshold for improving the conversion to retain advertiser i, which could be estimated through existing machine learning techniques [13] or advertiser survey.

To avoid disturbing normal market situations, virtual coupons are designed to help the retention advertisers win the display opportunities in a minimum way, *i.e.*, win the K^{th} slot of the impressions that they cannot win with their bare bids. We denote this impression set as $j \in [m_i]$ for advertiser i. Thus, the virtual coupon needed by advertiser i to win impression j becomes $d_{i,j} := s_K^j/q_{i,j} - b_{i,j}$, and the increase in conversions for helping advertiser i win the impression j could be denoted as $u_{i,j} := \mathrm{pctr}_{i,j}\cdot\mathrm{pcvr}_{i,j}$. Aiming to maximize the willingness to stay for advertisers, the *global auto-coupon optimization problem* can be formulated as

$$\begin{aligned}\max_{\pi}\quad & \frac{1}{n}\sum_{i=1}^{n}\min\left(\frac{\sum_{j=1}^{m_i}(u_{i,j} - \beta_i\cdot b_{i,j})\cdot x_{i,j}}{\mathrm{thr}_i}, 1\right)\\ \text{s.t.}\quad & \sum_{i=1}^{n}\sum_{j=1}^{m_i} x_{i,j}\cdot d_{i,j} \le B,\\ & x_{i,j}\in\{0,1\}\quad \forall i\in[n], j\in[m_i],\end{aligned} \tag{2}$$

where π is the virtual coupon design strategy of the platform, $x_{i,j}$ are the binary variables indicating whether to provide virtual coupon for advertiser i on her losing impression j, and B is the platform budget. The term $-\beta_i \cdot b_{i,j}$ represents advertiser's loss in originally allocated impressions because of spending more advertiser budget in auctions with coupons, where β_i is a scaled measure for bidder i's ROI in original biddings. As we will see later, under our design, $b_{i,j}$ exactly represents the consumption of advertiser budget when the coupon is effectively distributed, and we could still provide IC properties for value-maximizing bidders despite correlation between $b_{i,j}$ and $c_{i,j}$. To control revenue loss, we assign $q_{i,j} = 1$ in our CPM context and thus abbreviate those terms in budget constraint. In practice, the platform may still adopt personalized $q_{i,j}$ and correspondingly adapt our design with the bound in Theorem 3.2 changed up to a constant depending on detailed $q_{i,j}$.

3.3 Two-Stage Auto-Coupon Optimization

While Problem 2 maximizes the average willingness to stay from a global view, it is an online problem with huge amounts of variables

$x_{i,j}$ and complex estimation of key constants (*e.g.*, $\text{pctr}_{i,j}$, $\text{pcvr}_{i,j}$ and thr_i), leading it to be almost intractable for a real-time system. To overcome these difficulties, we propose to divide this problem into two stages: platform budget allocation and personalized coupon distribution. The platform would first divide the total coupon budget B to each advertiser using strategy π_B, and then enable personalized auto-coupon agents to optimize the virtual coupon distribution using strategy π_i for each advertiser i in a distributed manner.

Since the budget allocation should be finished before the start of ad auctions and coupon distribution, the platform needs an estimation on the resulted retention effects to achieve a well budget allocation. Using $U_i^{\pi_i}(B_i)$ to denote the improvement of advertising metric for advertiser i, *i.e.*, $\sum_{j=1}^{m_i}(u_{i,j} - \beta_i \cdot b_{i,j}) \cdot x_{i,j}$, when her allocated platform budget is B_i, we could formulate the platform budget allocation problem as

$$\begin{aligned} \max_{\pi_B} \quad & \frac{1}{n}\sum_{i=1}^{n} \min\left(\frac{U_i^{\pi_i}(B_i)}{\text{thr}_i}, 1\right) \\ \text{s.t.} \quad & \sum_{i=1}^{n} B_i \le B,\ B_i \in \mathbb{N}. \end{aligned} \tag{3}$$

As the online platforms usually have some smallest units for dividing budget, and each smallest unit should be able to provide the advertiser with non-negligible improvements, *w.l.o.g.*, we assume $B_i \in \mathbb{N}$ and $d_{i,j} \ll 1$. Given B_i allocated to each advertiser, the *personalized auto-coupon distribution problem* for advertiser i could then be formulated as:

$$\begin{aligned} \max_{\pi_i} \quad & \sum_{j=1}^{m_i}(u_{i,j} - \beta_i \cdot b_{i,j}) \cdot x_{i,j} \\ \text{s.t.} \quad & \sum_{j=1}^{m_i} x_{i,j} \cdot d_{i,j} \le B_i, \\ & x_{i,j} \in \{0,1\} \quad \forall j \in [m_i]. \end{aligned} \tag{4}$$

4 Methodologies

4.1 Platform Budget Allocation

The major difficulty in the budget allocation problem comes from the unknown form of budget-utility function $U_i^{\pi_i}(B_i)$. Fortunately, as $U_i^{\pi_i}(B_i)$ is determined by the subsequent personalized distribution strategy π_i, we may utilize the so-called *diminishing return submodular (DR-submodular)* property of the resulted $U_i^{\pi_i}(B_i)$ in budget allocation. In detail, $U_i^{\pi_i}$ is DR-submodular in B_i if

$$U_i^{\pi_i}(B) - U_i^{\pi_i}(B-1) \ge U_i^{\pi_i}(B+1) - U_i^{\pi_i}(B), \forall B \in \mathbb{Z}^+.$$

Intuitively, satisfying DR-submodular means the marginal return of per-unit investment decreases with the total investment, and we could show a near-optimal coupon distribution strategy π_i^* (Section 4.2) gives rises to a DR-submodular budget-utility function.

The DR-submodularity of $U_i^{\pi_i^*}$ enables us to design a *greedy algorithm* with optimality guarantee. In each iteration, we simply allocate one unit of budget to the advertiser with highest marginal retention value (increment in satisfaction for per-unit budget)

$$v_i(B_i) = \min\left(\frac{U_i^{\pi_i^*}(B_i+1)}{\text{thr}_i}, 1\right) - \min\left(\frac{U_i^{\pi_i^*}(B_i)}{\text{thr}_i}, 1\right),$$

and update the budgets until all the platform budget has been spent.

Algorithm 1: Greedy Platform Budget Allocation

Input: Total coupon budget B;
The budget-utility functions $U_i^{\pi_i^*}$;
Output: Platform budget allocation results B_i.
1 Initialize $sum = 0$, $B_i = 0$, $\forall i$;
2 Initialize a max heap, where node i has value
$v_i(B_i) = \min(\frac{U_i^{\pi_i^*}(B_i+1)}{\text{thr}_i}, 1) - \min(\frac{U_i^{\pi_i^*}(B_i)}{\text{thr}_i}, 1)$;
3 **while** $sum < B$ **do**
4 Select node i with the largest v_i from the heap;
5 $B_i = B_i + 1$, $sum = sum + 1$;
6 Update $v_i(B_i)$ for node i with B_i;

Theorem 4.1. *When $U_i^{\pi_i^*}$ are DR-submodular in B_i, the greedy algorithm based on marginal retention value (Algorithm 1) is optimal to Problem (3).*

Proof. Assume the optimal solution is $\mathbf{B}^* = (B_1^*, B_2^*, ..., B_n^*)$ and the allocation produced by Algorithm 1 is $\mathbf{B}' = (B_1', B_2', ..., B_n') \neq \mathbf{B}^*$. As $U_i^{\pi_i^*}$ is non-negative and monotonically increasing, by optimality of $\mathbf{B}^*$ and design of Line 3 in Algorithm 1, we have $\sum_i B_i^* = \sum_i B_i' = B$, and thus there must exist two advertisers i and j where $B_i^* > B_i'$ and $B_j^* < B_j'$. By our greedy design (Line 4), we have $v_j(B_j' - 1) \ge v_i(B_i')$. Since $U_i^{\pi_i^*}$ is DR-submodular, $\min\left(\frac{U_i^{\pi_i^*}(B_i)}{\text{thr}_i}, 1\right)$ is also DR-submodular, thus $v_j(B_j^*) \ge v_j(B_j' - 1) \ge v_i(B_i') \ge v_i(B_i^* - 1)$ given $B_i^* > B_i', B_j' > B_j^*$ and $B_i, B_j \in \mathbb{N}$. Therefore, reassigning $B_i^* = B_i^* - 1$ and $B_j^* = B_j^* + 1$ will not degrade the optimization objective. By repeating this process, we can transform $\mathbf{B}^*$ to $\mathbf{B}'$ without degrading the performance, showing Algorithm 1 can achieve the optimal coupon allocation. □

With the greedy algorithm, we could approach the optimal budget allocation based on the estimations of thr_i and $U_i^{\pi_i^*}$. Though existing techniques allow us to obtain those estimations [13, 37], various commercial constraints make this method too ideal in practice. To optimize the retention objective, the greedy algorithm prefers to allocate coupons to advertisers with originally low budget or low ad performance, which might lead to serious fairness concern and difficulty to provide advertisers with full interpretability about this method. Due to the above commercial constraints, we mainly consider *rule-based* methods based on clear pre-defined rules.

Towards practical deployment, we propose two rule-based methods for budget allocation in our system. The first approach is to divide budget based on advertisers' historical daily advertising investment, where advertisers with higher historical investment are supplied with more retention budget. Despite the lack of utilizing more abundant features, the conciseness of this investment-based method makes it robust for engineering implementation and appropriate for business operations. The second approach is to allocate budget based on interaction tasks, where advertisers finish specific interaction tasks assigned by the platform to obtain coupon budget.

These tasks usually guide advertisers to watch introductory videos and try advanced platform tools, which incentivizes advertisers to learn more about platform services and thus increases their adherence to the platform in a long run. Both investment-based and task-based methods have been deployed online in our platform and achieve significant retention effects. As our Auto-Coupon module and the personalized coupon distribution method work for any reasonable budget allocation approach, due to the intensive human behavioral elements involved in task-based method, we will mainly focus on the investment-based method in the subsequent analysis.

4.2 Personalized Auto-Coupon Distribution

Given the budget allocation method, the personalized auto-coupon distribution problem becomes an online optimization problem with online parameters $u_{i,j}$, $d_{i,j}$ and $b_{i,j}$. Following classical approaches in online linear programming and approximation algorithms [1], we can figure out a near-optimal solution formula to this problem.

Theorem 4.2. *Distributing coupons $x_{i,j} \cdot d_{i,j}$ to advertiser i on impression j according to $x_{i,j}$ in (5) can approximate the optimal offline solution to Problem 4 without breaking budget constraint B_i.*

$$x_{i,j} = \begin{cases} 1, & \text{if } u_{i,j} - \beta_i \cdot b_{i,j} > \lambda_i \cdot d_{i,j}, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Proof. When $d_{i,j} \ll B_i$, similar to the greedy solution to 0-1 knapsack problem [15], greedily choose impressions with higher $(u_{i,j} - \beta_i \cdot b_{i,j})/d_{i,j}$ until exhausting the budget is near optimal. Therefore, setting an appropriate λ_i as the threshold for choosing impressions, *i.e.*, letting $x_{i,j} = 1$ when $(u_{i,j} - \beta_i \cdot b_{i,j}) > \lambda_i \cdot d_{i,j}$ and 0 otherwise, is equivalent to the greedy algorithm, thus approximating the optimal offline solution. □

Proposition 4.3. *Distributing virtual coupons $c_{i,j}$ according to $\min\left(d_{i,j}, (u_{i,j} - \beta_i \cdot b_{i,j}) \times \frac{1}{\lambda_i}\right)$ with respect to some threshold parameters λ_i can approximate the optimal offline solution to Problem 4 without breaking budget constraint B_i.*

To align the format of Proposition 4.3 and Theorem 4.2, we assume the auction may allocate to a retention advertiser only when $(u_{i,j} - \beta_i \cdot b_{i,j}) \times \frac{1}{\lambda_i} > d_{i,j}$, though her rank score might become the same as the original s_K^j when $(u_{i,j} - \beta_i \cdot b_{i,j}) \times \frac{1}{\lambda_i}$ exactly takes $d_{i,j}$. When $(u_{i,j} - \beta_i \cdot b_{i,j}) \times \frac{1}{\lambda_i} \leq d_{i,j}$, the distributed virtual coupon is not intended to help the advertiser win the auction and would not be actually consumed, but instead create a benchmark of boosted rank score for other advertisers in GSP auction. This design is necessary to prevent a non-retention advertiser from misreporting a higher bid to discourage coupon distribution of the retention advertiser without breaking her own IR property (see details of proof in Appendix A). Since we have $d_{i,j} := s_K^j/q_{i,j} - b_{i,j}$, another key point for IC property is to have $\beta_i \leq \lambda_i$, such that the left-hand side of coupon distribution criteria $b_{i,j} + \frac{u_{i,j} - \beta_i \cdot b_{i,j}}{\lambda_i} > s_K^j/q_{i,j}$ is non-decreasing with bid $b_{i,j}$. Intuitively speaking, this condition should hold for reasonable configurations of β_i and λ_i. This is because β_i estimates the original ROI of current advertiser without retention, while the optimal λ_i represents the estimated conversion improvement per extra unit of coupon budget, and the original bidding strategy of retention advertisers typically perform worse than a well-designed coupon distribution strategy. To formally guarantee this condition, we impose β_i as a lower bound of λ_i in our subsequent methods.

Recall that the greedy budget allocation algorithm requires $U_i^{\pi_i}$ to be DR-submodular, we now demonstrate why this property holds for π_i^* in Proposition 4.3. Since this strategy essentially consumes coupons on impressions whose $(u_{i,j} - \beta_i \cdot b_{i,j})/d_{i,j}$ exceeds λ_i, increasing the coupon budget would lead to a decreasing optimal threshold parameter to consume more coupon budget. As a result, the decrease in optimal threshold λ_i would bring more user impressions with lower $(u_{i,j} - \beta_i \cdot b_{i,j})/d_{i,j}$. As this term represents the retention value per unit of budget consumption, DR-submodularity holds for the current $U_i^{\pi_i^*}$.

Since the exact calculation of optimal λ_i requires information about all the auctions that could not be obtained beforehand, using a fixed λ_i calculated through historical data can lead to huge deviation in the volatile advertising environment. To overcome the difficulty, we design two control-based methods to timely adjust λ_i online.

4.2.1 Feedback Control for Coupon Strategy.

Feedback control is widely adopted to deal with the outside noise and fluctuation in dynamic systems [35, 38]. In the scenario of personalized coupon distribution, we define the dynamic system in feedback control to be the coupon distribution strategy and the external advertising environment. During advertising, we distribute coupons according to Proposition 4.3 until the retention budget is depleted, and thus define the system input to be λ.

To avoid the waste of coupon budget or losing valuable ad opportunities in the future, the target of the feedback control system is to stably consume the coupons across advertising process. Therefore, dividing the advertising process into T time steps, the expected coupon budget we should consume in each future time step to realize stable budget consumption (also called as reference) at the time step t could be set as

$$reference_t = \frac{\max(0, Budget - \sum_{i=1}^{t} cost_i)}{N - t}, \tag{6}$$

where $cost_i$ denotes the amount of coupon consumed at time step i. For feedback control, we adjust λ at the end of each time step t. Thus, the error between the real-time coupon consumption and the desired amount of future consumption becomes $error_t = reference_t - cost_t$, and based on which we design a proportional controller with the linear actuator to adjust λ_t as

$$\lambda_{t+1} = \lambda_t - \gamma \times error_t, \tag{7}$$

where γ is the hyper-parameter indicating control sensitivity. Though with simple design, such feedback control method has demonstrated strong robustness and effectiveness for vast applications in practice.

4.2.2 RL-based Coupon Agent.

Independent from the robust feedback control method, we propose the RL-based coupon distribution method to integrate more historical information when adjusting the online strategy. The RL-based coupon agent has the same objective as the feedback control method to realize stable consumption of budget and to optimize the overall retention performance. For convenience in notations, we set the coupon agent to adjust the

Algorithm 2: RL-based Coupon Agent Training

Input: Training data $\mathbf{C} = \{c_1, c_2, ..., c_n\}$;
The offline theoretical optimal result $\{R_1^*, R_2^*, ..., R_n^*\}$;
The duration of the ad campaigns $\{T_1, T_2, ..., T_n\}$.
Output: Coupon agent with policy μ_θ.

Initialize the actor network μ_θ with parameter θ, the critic network Q_η with parameter η, and the replay buffer;
while *Not Converge* **do**
  Randomly select c_i from $\mathbf{C}$, and initialize parameter α_1;
  Use the coupon distribution strategy with α_1 in time step 1 to gain reward r_1, and initialize $R = r_1$;
  for $t = 2$ **to** T_i **do**
    Observe real-time coupon consumption and generate s_t;
    Take action with random noise $a_t = \mu_\theta(s_t) + \epsilon$;
    Adjust $\alpha_t = \alpha_{t-1} \times (1 + a_t)$;
    Use the auto-coupon strategy with α_t in time step t and gain reward r_t;
    Set $R = R + r_t$ and initialize $V = 0$;
    for $\tau = t + 1$ **to** T_i **do**
      Use the auto-coupon strategy with α_t in time step τ and gain reward r_τ;
      Update $V = V + r_\tau$;
    Set $G = \min\{(R + V)/R^*, 1.0\}$;
    Store experience (s_t, a_t, G) in replay buffer;
  Sample M experiences from replay buffer;
  Update the critic network Q_η by minimizing
  $\mathcal{L}(\eta) = \frac{1}{M}\sum_k \left(G^k - Q_\eta\left(s^k, a^k\right)\right)^2$;
  Update the actor network μ_θ by policy gradient:
  $\nabla_\theta J \approx \frac{1}{M}\sum_k \nabla_\theta \mu\left(s^k\right) \nabla_a Q_\eta\left(s^k, a\right)\Big|_{a=\mu(s^k)}$;

parameter $\alpha = \frac{1}{\lambda}$, and the coupon distribution rule becomes comparing $(u_{i,j} - \beta \cdot b_{i,j}) \times \alpha$ and $d_{i,j}$. We divide the advertising process into T time steps similarly, and adjust α at the beginning of each time step. We formulate this α adjustment process as a Markov Decision Process (MDP), and its elements are defined as follows.

- $\mathcal{S}$: The state $s_t \in \mathcal{S}$ describes the coupon consuming status before time step t, including the spent coupon budget, the remaining time step of advertising, the current α, and the normalized coupon consumption errors (Section 4.2.1) within the time step $t - 1$.
- $\mathcal{A}$: The action a_t represents the adjustment to α from time step $t - 1$ to t, which can be defined as $\alpha_t = \alpha_{t-1} \times (1 + a_t)$. To keep stability, we constrain the range of a_t to be $[-0.2, 0.2]$.
- $\mathcal{R}$: The immediate reward r_t is defined as the accumulative result $\sum(u_i - \beta \cdot b_i) \cdot x_i$ during time step t using parameter α_t.
- $\mathcal{T}$: We leverage model-free RL methods to generate the coupon agent, thus do not need to model the environment dynamics $\mathcal{T}$.

We adopt the actor-critic based algorithm to train the coupon agent. Inspired by the RL algorithm for automated bidding [14], we refine the classic DDPG algorithm by simplifying the training process of the actor (policy) and the critic (value) network. The detailed algorithm is shown in Algorithm 2. During each training iteration, we randomly choose an ad campaign as the training sample and initialize α_1 with prior knowledge about advertising environment. At the beginning of each time step t, we take action a_t, with random noise ϵ for exploration, to adjust α_{t-1} to α_t based on the observed state s_t. Then we use the auto-coupon strategy with α_t during time step t and gain reward r_t. Different from the traditional DDPG algorithm which approximates the value of (s_t, a_t) with the immediate reward r_t and the expected future reward of s_{t+1} based on the target value network, we directly calculate the accumulate reward $R + V$ for the entire advertising process by freezing $\alpha = \alpha_t$ in the rest of the time steps, and store the normalized performance G in the replay buffer. At the end of training iteration, we sample a batch of experiences from the replay buffer and update the value network Q by minimizing its regression loss between G and $Q(s, a)$, and update the policy network μ based on policy gradient. Using the relative performance G to replace the conventional temporal-difference method can boost the policy convergence [14].

As we can observe, both of our coupon distribution methods can be implemented efficiently for online inference: at each time step, we only need to generate state (error) in time linear to the number of impressions per time step, and then update λ via coupon agent (control function) in $O(1)$ time.

5 Experiment Results

5.1 Offline Simulation

5.1.1 Dataset. The dataset is collected from TaoBao, which includes more than 1500 ad campaigns aiming to maximize total user conversions. We assume each ad campaign in the dataset represents an independent advertiser in ad auctions, and divide the dataset into the training set and the test set with proportion 0.75:0.25. Data for each ad campaign i contains a series of bid logs recording the ad auctions that the advertiser attended during the campaign. Each bid log includes the information to compute $u_{i,j}, b_{i,j}$, and $d_{i,j}$ for the corresponding user impression j. Following the idea of investment-based budget allocation method, we allocate the coupon budget for each campaign i as $B_i = B_i^{camp} \times 0.3$, where B_i^{camp} is the original budget of the advertiser in the ad campaign i, and we set the length of each time step to be 15 minutes. The ad campaign i computes bid bid_i for all impressions according to its expected cost-per-click and pCTR, and can achieve baseline conversion cvr_i, PV pv_i and ROI roi_i without retention. The penalty parameter is set as $\beta_i = \frac{cvr_i}{B_i^{camp}} \cdot 0.3$ (we also provide experimental results of fixed β in Appendix B).

5.1.2 Metrics. In offline simulations, we mainly focus on the improvement in ad performances and budget utilization of advertisers. We adopt averaged increment of ROI and PV as evaluation metrics of ad performances, which is calculated as $\Delta ROI = \frac{1}{N} \times \sum_i \frac{roi_i' - roi_i}{roi_i}$ and $\Delta PV = \frac{1}{N} \times \sum_i \frac{pv_i' - pv_i}{pv_i}$, with N denoting the number of campaigns, roi_i' and pv_i' denoting the results after providing the coupons for advertiser i. Since a well-performed coupon distribution strategy should utilize the available budget as much as possible, we additionally evaluate the consumption ratio of coupon budget and ad campaign budget, denoted as $r(B)$ and $r(B^{camp})$, respectively.

5.1.3 Comparison Methods. For baseline methods, we consider three methods anonymous to traditional retention approach: *Direct*

Table 1: Offline Comparison Experiment Result

	ΔROI	ΔPV	$r(B^{camp})$	$r(B)$
Direct Coupon	17.10%	15.06%	78.36%	100.0%
Fixed Coupon	-0.15%	10.84%	97.62%	36.89%
Fixed-Param	50.24%	13.73%	97.66%	70.38%
Feedback Control	65.26%	17.55%	99.15%	97.07%
Coupon Agent	71.49%	18.68%	99.03%	92.94%

Coupon, *Fixed Coupon*, and *Fixed-Param*. For *Direct Coupon*, it directly increases the coupon budget B to the campaign budget B^{camp} and lets the advertiser bid by herself, which is analogous to providing fixed-amount coupons in total bill. The *Fixed Coupon* strategy, also based on the Auto-Coupon framework, distributes coupons with a fixed ratio (30% in our experiments) to the original bids until all the coupon budget is consumed. Finally, *Fixed-Param* follows the coupon distribution rule in Proposition 4.3 with a pre-defined fixed parameter λ without further adjustment during advertising.

5.1.4 Experiment Setting and Implementation Details. For *Fixed-Param*, *Feedback Control* and *Coupon Agent*, the parameters are initialized with the average offline optimal value of λ^* in the training dataset. For the *Feedback Control* method, the hyper-parameter γ_i for advertiser i is defined as $\gamma_i = \frac{0.0001 \times T_i}{B_i}$. To avoid over-tuning, we constrain the range of λ adjustment to be $[-0.0001, 0.0001]$[3]. For the RL-based *Coupon Agent*, we train the coupon agent with the campaigns in the training set for 50,000 training iterations until the agent policy converges, and adopt the policy with the best performance on the training dataset. The actor network μ includes 3 hidden layers with [100, 100, 10] nodes. The critic network Q first uses a hidden layer with 100 nodes to extract the state and concatenates it with the action. Then, it uses 3 hidden layers with [100, 50, 10] nodes to predict the long-term value G. For each iteration, we generate 20 batches of experiences for policy update, where the batch size M is 64. The exploration noise ϵ is sampled from a normal distribution $\mathcal{N}(0, 0.1)$. The computed virtual coupons are normally distributed only when the remaining retention budget is sufficient. We conduct at least 5 independent runs for each result of *Coupon Agent* and present their averaged performances.

5.1.5 Experiment Results. The performances of different methods are shown in Table 1. For *Direct Coupon*, it can hardly help the advertiser to use up their own budget or achieve large performance improvement in increasing ROI, though the provided retention budget is depleted by the design of direct coupon. For *Fixed Coupon*, it can even worsen the original performance by reducing the ROI by 0.15%. The underperformance of *Direct Coupon* and *Fixed Coupon* should be resulted from their weak assistance on the advertisers' detailed bidding strategies. For advertisers adopting unreliable bidding strategies, providing coupons without well-designed impression-level guidance could lead them to waste more budget on low cost-efficiency impressions, especially for *Fixed Coupon* that indistinguishably distribute coupons in proportional to advertiser bids.

Compared with these previous method, the subsequent methods based on our derived coupon distribution formula achieve much better performances in both improving ROI and acquiring PV, which justifies the coupon distribution rule in Proposition 4.3. For *Fixed-Param*, it can increase the ROI by 50.24% and the PV by 13.73%. For *Feedback Control*, it can improve the ROI by 65.26% and the PV by 17.55%, while the *Coupon Agent* can further achieve a better performance with ΔROI to be 71.49% and ΔPV to be 18.68% with utilization of data in training set. Meanwhile, these two methods can also help advertisers to fully consume their budget and coupons, increasing the budget consumption ratio to over 99% and the coupon consumption ratio to over 92%.

[3]To boost control efficiency in the early stage of advertising when the initial value is far from optimal, we manually set $\lambda_{t+1} = \lambda_t - 0.0001$ when $cost_t = 0$.

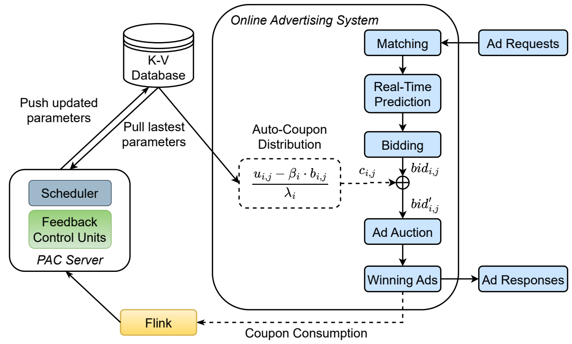

Figure 2: Auto-Coupon Module Online Deployment

5.2 Online Experiment

Due to the high cost of large-scale RL training and the comparable performances between *Feedback Control* and *Coupon Agent* method in offline experiments, we choose to implement the Auto-Coupon module with feedback control methods online. Our online experiments were conducted during the date 05-06 to 05-19, which spent over 300,000 dollars on 45,362 treatment group advertisers.

5.2.1 Deployment. The working flow of Auto-Coupon in the industrial online advertising system is illustured in Figure 2. In the online deployment, functions of Auto-Coupon module are mainly divided into two parts: an online Coupon Agent and an offline parameter adjustment control (PAC) server. The PAC server is further composed of a *Scheduler* and *Feedback Control Units*. The Scheduler triggers Feedback Control Units periodically (15 minutes in our online setting) to update parameters of the coupon distribution strategy with methods described in Section 4.2.1. In detail, Feedback Control Units retrieve the latest control parameters from Key-Value (K-V) database, update the parameters based on the feedback data in the last period, and push the updated parameters to the K-V database. The feedback data are provided by the stream-processing tool *Flink*, where Flink reads the data log from ad auction module to derive the consumption of coupon budget. Each time the advertiser submits a bid to an impression, the Coupon Agent retrieves the latest control parameters from K-V database and distributes coupons online. Our deployment leverages the existing facilities in online ad system and meets the high concurrency needs. The additional latency created in this process is less than 1 ms.

Table 2: Online Experiment Results

	ΔROI	ΔPV	$\Delta 7\text{-}rt$	$\Delta N\text{-}active$
% Improved	+16.5%	+14.0%	+1.5%	+3.2%

5.2.2 Experiment Setting. Our online experiments are conducted on a representative advertising channel designed for small or medium-sized advertisers in TaoBao, named as *JiSuTui*. The majority of advertisers in this channel are new advertisers and advertisers with low budget, who have lower loyalty to the platform and are relatively easy to churn. We randomly select advertisers in the channel and equally split them into control and treatment group. For each ad campaign of treatment group advertisers, we divide the coupon budget to be $B = B^{\text{camp}} \times 0.2$, where B^{camp} is the original budget of ad campaign. No coupon is distributed to control group advertisers.

5.2.3 Metrics. Recall that ad performance metrics are actually intermediate metrics of retention, besides ΔROI and ΔPV (Section 5.1.2), we need to further evaluate the improvement in advertisers' investment willingness as the ultimate estimation of retention effects. We define *T-days repurchase rate* ($T\text{-}rt$) as the percentage rate of advertisers who spend the same or more money in the following T days, which is measured for 7 days ($7\text{-}rt$) in this work. In addition, we define $N\text{-}active$ as the number of advertisers who make ad investment during the experiment period. We can correspondingly define $\Delta 7\text{-}rt$ and $\Delta N\text{-}active$ to be the increment of $7\text{-}rt$ and $N\text{-}active$ for treatment group compared with control group.

5.2.4 Experiment Results. The results of online experiments are presented in Table 2. As can be observed, Auto-Coupon Module can improve PV by 14.0% and ROI by 16.5% in the online system compared to the control group. In addition, we find that $7\text{-}rt$ and $N\text{-}active$ have been improved by 1.5% and 3.2%, respectively. The above results demonstrate the effectiveness of Auto-Coupon module in improving the advertisers' short-term metrics and their willingness to continue advertising in real online platform. Considering the scale of involved advertisers, the increase in $7\text{-}rt$ and $N\text{-}active$ indicates successful retention for considerable amount of advertisers. As Auto-Coupon provides additional display opportunities for advertisers to enhance their future prediction accuracy of ad metrics, we expect its benefits to persist beyond experiment period.

6 Related Work

The advertiser retention problem is closely related to advertiser modeling and advertiser services. Distinct from developed user modeling and prediction techniques [19, 28, 41], the correspondent techniques for advertisers are still under exploration. The early work [36] adopts basic machine learning models to predict advertiser churn. Leveraging the advanced deep learning techniques, some existing work have investigated the methods to model advertisers' intent and satisfaction [13], design strategy recommendation system for advertisers [12], and predict the behavior of advertisers in various scenarios [37]. Our work focus on the subsequent step of advertiser retention, and could utilize the above techniques to deal with the retention problem.

Coupons have long attracted attention on its price discrimination and advertising effects in marketing sales [4, 21]. In recent years, the coupon distribution strategy to promote user purchasing for online e-commerce platform has been widely studied [10, 16, 17, 32]. In contrast to those work, our coupons are distributed to advertisers during their bidding process for advertiser retention, which raises the bids of retention advertisers in the GSP auctions. Starting from [7, 31], a series of work has studied the auction design based on GSP auction, with maximizing revenue as the typical objective [20, 29]. The Auto-Coupon module could be viewed as a form of auction design for value-maximizing bidders with advertiser retention as the objective. While a recent line of work [18, 25, 26] also introduce coupons in ad auctions, their coupons are designed as an optional price discrimination on users to increase the platform's revenue, which is distinct with our design objective and working ways.

To solve retention optimization, we decompose it into the budget allocation and personalized auto-coupon distribution problem. Budget allocation problem has been widely considered in marketing applications [2, 6, 27, 40]. The characteristic of our budget allocation problem mainly lies in the large scale of retention advertisers and the dependence of marginal retention value on the subsequent auto-coupon distribution strategy. Both solving online optimization in the advertising environment, our methods for personalized auto-coupon distribution share similarities with the bid optimization literature [5, 34, 39, 42], where the authors of [39] firstly propose a form of optimal bidding strategy for budget-constrained bidding. Towards this problem, [35] presents a feedback control-based bidding strategy to deal with dynamic environments. Further, the authors of [14] designs a unified bidding function for various bidding options and correspondingly proposes a reinforcement learning method. A comprehensive survey for bid optimization could be found at [23]. In our context, advertisers who need retention typically do not adopt appropriate bidding strategies, and our Auto-Coupon is an independent module in addition to advertiser bidding.

7 Conclusion

In this work, we focus on the advertiser retention problem to improve the number of active advertisers within platform budget limit. The advertiser churn survey and ad performance data motivate us to improve the ad performances as the approach to promote the advertiser satisfaction. We discuss the desired properties of retention schemes, and propose the Auto-Coupon module as a universal framework for advertiser retention. We decompose the resulted large-scale online optimization problem into two stages, then propose the feasible methods of budget allocation, and finally utilize feedback control and reinforcement learning methods to realize the personalized virtual coupon distribution. Our offline and online experiments demonstrate the effectiveness of our methods in improving the ad performances of churning advertisers and promoting their willingness to stay in the platform.

Acknowledgement

This work was supported, in part by China NSF grant No. U2268204, 62322206, 62025204, 62132018, 62272307, 62372296, in part by Alibaba Group through Alibaba Innovative Research Program. The opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies or the government.

References

[1] Shipra Agrawal, Zizhuo Wang, and Yinyu Ye. 2014. A dynamic near-optimal algorithm for online linear programming. *Operations Research* 62, 4 (2014), 876–890.

[2] Noga Alon, Iftah Gamzu, and Moshe Tennenholtz. 2012. Optimizing budget allocation among channels and influencers. In *Proceedings of the 21st World Wide Web Conference 2012, WWW 2012*. ACM, 381–388.

[3] Patrik Berger and Michal Kompan. 2019. User modeling for churn prediction in E-commerce. *IEEE Intelligent Systems* 34, 2 (2019), 44–52.

[4] Helmut Bester and Emmanuel Petrakis. 1996. Coupons and oligopolistic price discrimination. *International journal of industrial organization* 14, 2 (1996), 227–242.

[5] Han Cai, Kan Ren, Weinan Zhang, Kleanthis Malialis, Jun Wang, Yong Yu, and Defeng Guo. 2017. Real-Time Bidding by Reinforcement Learning in Display Advertising. In *Proceedings of the 10th ACM International Conference on Web Search and Data Mining, WSDM 2017*. ACM, 661–670.

[6] Tianchi Cai, Jiyan Jiang, Wenpeng Zhang, Shiji Zhou, Xierui Song, Li Yu, Lihong Gu, Xiaodong Zeng, Jinjie Gu, and Guannan Zhang. 2023. Marketing Budget Allocation with Offline Constrained Deep Reinforcement Learning. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, WSDM 2023*. ACM, 186–194.

[7] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. 2007. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American Economic Review* 97, 1 (2007), 242–259.

[8] Salman Fadaei and Martin Bichler. 2016. Truthfulness and Approximation with Value-Maximizing Bidders. In *Algorithmic Game Theory - 9th International Symposium, SAGT 2016. (Lecture Notes in Computer Science, Vol. 9928)*. Springer, 235–246.

[9] Salman Fadaei and Martin Bichler. 2017. Truthfulness with value-maximizing bidders: On the limits of approximation in combinatorial markets. *European Journal of Operational Research* 260, 2 (2017), 767–777.

[10] Dmitri Goldenberg, Javier Albert, Lucas Bernardi, and Pablo Estevez. 2020. Free Lunch! Retrospective Uplift Modeling for Dynamic Promotions Recommendation within ROI Constraints. In *RecSys 2020: Proceedings of the 14th ACM Conference on Recommender Systems, RecSys 2020*, Rodrygo L. T. Santos, Leandro Balby Marinho, Elizabeth M. Daly, Li Chen, Kim Falk, Noam Koenigstein, and Edleno Silva de Moura (Eds.). ACM, 486–491.

[11] Alibaba Group. 2023. Alibaba Financial Report. https://www.alibabagroup.com/ir-financial-reports-financial-results. Accessed: 2024-05-15.

[12] Liyi Guo, Junqi Jin, Haoqi Zhang, Zhenzhe Zheng, Zhiye Yang, Zhizhuang Xing, Fei Pan, Lvyin Niu, Fan Wu, Haiyang Xu, Chuan Yu, Yuning Jiang, and Xiaoqiang Zhu. 2021. We Know What You Want: An Advertising Strategy Recommender System for Online Advertising. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2021*. ACM, 2919–2927.

[13] Liyi Guo, Rui Lu, Haoqi Zhang, Junqi Jin, Zhenzhe Zheng, Fan Wu, Jin Li, Haiyang Xu, Han Li, Wenkai Lu, Jian Xu, and Kun Gai. [n. d.]. A Deep Prediction Network for Understanding Advertiser Intent and Satisfaction. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management, CIKM 2020*. ACM, 2501–2508.

[14] Yue He, Xiujun Chen, Di Wu, Junwei Pan, Qing Tan, Chuan Yu, Jian Xu, and Xiaoqiang Zhu. 2021. A Unified Solution to Constrained Bidding in Online Display Advertising. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2021*. ACM, 2993–3001.

[15] Oscar H Ibarra and Chul E Kim. 1975. Fast approximation algorithms for the knapsack and sum of subset problems. *Journal of the ACM (JACM)* 22, 4 (1975), 463–468.

[16] Liangwei Li, Liucheng Sun, Chenwei Weng, Chengfu Huo, and Weijun Ren. 2020. Spending Money Wisely: Online Electronic Coupon Allocation based on Real-Time User Intent Detection. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management, CIKM 2020*. ACM, 2597–2604.

[17] Ying-Chun Lin, Chi-Hsuan Huang, Chu-Cheng Hsieh, Yu-Chen Shu, and Kun-Ta Chuang. 2017. Monetary Discount Strategies for Real-Time Promotion Campaign. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017*. ACM, 1123–1132.

[18] Xiaodong Liu, Zhikang Fan, Yiming Ding, Yuan Guo, Lihua Zhang, Changcheng Li, Dongying Kong, Han Li, and Weiran Shen. 2024. Optimal auction design with user coupons in advertising systems. In *Proceedings of the 32rd International Joint Conference on Artificial Intelligence, IJCAI 2024*. 2904–2912.

[19] Quan Lu, Shengjun Pan, Liang Wang, Junwei Pan, Fengdan Wan, and Hongxia Yang. 2017. A Practical Framework of Conversion Rate Prediction for Online Display Advertising. In *Proceedings of the 23th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2017*. ACM, 9:1–9:9.

[20] Brendan Lucier, Renato Paes Leme, and Éva Tardos. 2012. On revenue in the generalized second price auction. In *Proceedings of the 21st World Wide Web Conference 2012, WWW 2012*. ACM, 361–370.

[21] José Luis Moraga-González and Emmanuel Petrakis. 1999. Coupon advertising under imperfect price information. *Journal of Economics & Management Strategy* 8, 4 (1999), 523–544.

[22] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani. 2007. *Algorithmic Game Theory*. Cambridge University Press.

[23] Weitong Ou, Bo Chen, Xinyi Dai, Weinan Zhang, Weiwen Liu, Ruiming Tang, and Yong Yu. 2023. A survey on bid optimization in real-time bidding display advertising. *ACM Transactions on Knowledge Discovery from Data* 18, 3 (2023), 1–31.

[24] Gajendra Sharma and Wang Lijuan. 2015. The effects of online service quality of e-commerce Websites on user satisfaction. *The Electronic Library* 33, 3 (2015), 468–485.

[25] Weiran Shen, Pingzhong Tang, Xun Wang, Yadong Xu, and Xiwang Yang. 2020. Learning to Design Coupons in Online Advertising Markets. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems, AAMAS '20*. International Foundation for Autonomous Agents and Multiagent Systems, 1242–1250.

[26] Weiran Shen, Pingzhong Tang, Xun Wang, Yadong Xu, and Xiwang Yang. 2021. Coupon Design in Advertising Systems. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence, AAAI 2021*. AAAI Press, 5717–5725.

[27] Tasuku Soma, Naonori Kakimura, Kazuhiro Inaba, and Ken-ichi Kawarabayashi. 2014. Optimal Budget Allocation: Theoretical Guarantee and Efficient Algorithm. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014 (JMLR Workshop and Conference Proceedings, Vol. 32)*. JMLR.org, 351–359.

[28] Ning Su, Jiyin He, Yiqun Liu, Min Zhang, and Shaoping Ma. 2018. User Intent, Behaviour, and Perceived Satisfaction in Product Search. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining, WSDM 2018*. ACM, 547–555.

[29] David R. M. Thompson and Kevin Leyton-Brown. 2013. Revenue optimization in the generalized second-price auction. In *Proceedings of the 14th ACM Conference on Electronic Commerce, EC 2013*. ACM, 837–852.

[30] Ali Vanderveld, Addhyan Pandey, Angela Han, and Rajesh Parekh. 2016. An Engagement-Based Customer Lifetime Value System for E-commerce. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2016*. ACM, 293–302.

[31] Hal R Varian. 2007. Position auctions. *International Journal of Industrial Organization* 25, 6 (2007), 1163–1178.

[32] Chao Wang, Xiaowei Shi, Shuai Xu, Zhe Wang, Zhiqiang Fan, Yan Feng, An You, and Yu Chen. 2023. A Multi-stage Framework for Online Bonus Allocation Based on Constrained User Intent Detection. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023*. ACM, 5028–5038.

[33] Christopher A. Wilkens, Ruggiero Cavallo, and Rad Niazadeh. 2017. GSP: The Cinderella of Mechanism Design. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017*. ACM, 25–32.

[34] Di Wu, Xiujun Chen, Xun Yang, Hao Wang, Qing Tan, Xiaoxun Zhang, Jian Xu, and Kun Gai. 2018. Budget Constrained Bidding by Model-free Reinforcement Learning in Display Advertising. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018*. ACM, 1443–1451.

[35] Xun Yang, Yasong Li, Hao Wang, Di Wu, Qing Tan, Jian Xu, and Kun Gai. 2019. Bid Optimization by Multivariable Control in Display Advertising. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019*. ACM, 1966–1974.

[36] Sangho Yoon, Jim Koehler, and Adam Ghobarah. 2010. Prediction of advertiser churn for google adwords. (2010).

[37] Qianqian Zhang, Xinru Liao, Quan Liu, Jian Xu, and Bo Zheng. 2022. Leaving No One Behind: A Multi-Scenario Multi-Task Meta Learning Approach for Advertiser Modeling. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining, WSDM 2022*. ACM, 1368–1376.

[38] Weinan Zhang, Yifei Rong, Jun Wang, Tianchi Zhu, and Xiaofan Wang. 2016. Feedback Control of Real-Time Display Advertising. In *Proceedings of the 9th ACM International Conference on Web Search and Data Mining, WSDM 2016*. ACM, 407–416.

[39] Weinan Zhang, Shuai Yuan, and Jun Wang. 2014. Optimal real-time bidding for display advertising. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2014*. ACM, 1077–1086.

[40] Kui Zhao, Junhao Hua, Ling Yan, Qi Zhang, Huan Xu, and Cheng Yang. 2019. A Unified Framework for Marketing Budget Allocation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019*. ACM, 1820–1830.

[41] Guorui Zhou, Xiaoqiang Zhu, Chengru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep Interest Network for Click-Through Rate Prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018*. ACM, 1059–1068.

[42] Yunhong Zhou, Deeparnab Chakrabarty, and Rajan M. Lukose. 2008. Budget constrained bidding in keyword auctions and online knapsack problems. In *Proceedings of the 17th International Conference on World Wide Web, WWW 2008*. ACM, 1243–1244.

A Missing Proof of Theorem 3.1

Proof. By our design of virtual coupon distribution in Section 4.2, the coupon agent distributes $c_{i,j} = \min(s_K^j/q_{i,j} - b_{i,j}, (u_{i,j} - \beta_i \cdot b_{i,j}) \times \frac{1}{\lambda_i})$ amount of coupon to the retention advertiser i when she could not win auction j with her original bid, where s_K^j is the K^{th} highest rank score among non-retention advertisers. As a result, the rank score of retention advertiser i would be $(b_{i,j} + c_{i,j}) \cdot q_{i,j} = \min\left(s_K^j, (b_{i,j} + \frac{u_{i,j}-\beta_i \cdot b_{i,j}}{\lambda_i}) \cdot q_{i,j}\right)$ with fixed β_i and λ_i. The IR properties hold following the same argument as the first part in the main text. To verify the IC properties, we discuss the misreporting cases for the retention advertiser i and other non-retention advertisers, respectively. Note that only the retention advertiser i might get coupon from the platform, and we consider an auction to involve only one retention advertiser. We use $v_{i,j}$, $b_{i,j}$ and $p_{i,j}$ to denote advertiser i's true value, bid and payment on impression j, respectively. Here, $v_{i,j}$ represents advertiser i's valuation on impression j, which might depend on her detailed estimation on ad metrics like $\text{pctr}_{i,j}$ and $\text{pcvr}_{i,j}$. We may also use $\tilde{b}_{i,j}$ and $\tilde{p}_{i,j}$ to denote the bid and payment when the advertiser misreports her bid. For a value-maximizing bidder [8], her utility on obtaining an allocation would be $v_{i,j}$ if $p_{i,j} \le v_{i,j}$, or otherwise be $-\infty$ if $p_{i,j} > v_{i,j}$. As explained in the main text, we assume $v_{i,j}$ to be the same for getting any slot on impression j, while the advertiser strictly prefers to win a higher slot when $p_{i,j} \le v_{i,j}$.

Consider the retention advertiser i misreports her bid and fix the bids of other advertisers. Since our coupon may only help the advertiser to win the K^{th} slot, the retention advertiser has no incentive to misreport when reporting $b_{i,j} = v_{i,j}$ is sufficient to win some slot. Similarly, she has no incentive to misreport when she already wins the K^{th} slot under the help of coupon, because the only possibility to get a strictly better slot is to report $\tilde{b}_{i,j} \ge s_{K-1}^j/q_{i,j}$, which leads to $\tilde{p}_{i,j} \ge s_{K-1}^j/q_{i,j} > v_{i,j}$. Therefore, the only remaining case is that advertiser i could not win the K^{th} slot with $b_{i,j} = v_{i,j}$ even under the help of coupon, and misreport a higher $\tilde{b}_{i,j}$ to get at least the K^{th} slot. It is clear that misreporting an excessively high bid to win a slot without the help of coupon leads to $\tilde{p}_{i,j} \ge s_K^j/q_{i,j} > v_{i,j}$. If the advertiser misreport a moderately high bid $\tilde{b}_{i,j}$ to win the K^{th} slot with the help of coupon, her resulting payment would exactly be $\tilde{p}_{i,j} = \tilde{b}_{i,j}$ under our design of coupon. By our setting of $\beta_i \le \lambda_i$, we have $(b_{i,j} + \frac{u_{i,j}-\beta_i \cdot b_{i,j}}{\lambda_i}) \cdot q_{i,j}$ non-decreasing in $b_{i,j}$. Therefore, we must have $\tilde{b}_{i,j} > v_{i,j}$ to achieve $\tilde{b}_{i,j} + \frac{u_{i,j}-\beta_i \cdot \tilde{b}_{i,j}}{\lambda_i} > s_K^j/q_{i,j}$, indicating $\tilde{p}_{i,j} > v_{i,j}$ and leading to negative infinite utility for advertiser i.

Consider a non-retention advertiser i' misreports her bid and fix the bids of other advertisers. By properties of GSP auction, when advertiser i' could win some slot when truthfully reporting $b_{i',j} = v_{i',j}$, or when all winning advertisers did not leverage coupon in their bidding, misreporting bid to get a better allocation would only lead to $\tilde{p}_{i',j} > v_{i',j}$. The only remaining case is that advertiser i' could not get any allocation when reporting $b_{i',j} = v_{i',j}$, and the K^{th} slot is allocated to the retention advertiser i with coupon. Under this case, to get allocation and compete against advertiser i, advertiser i' should at least afford $\tilde{p}_{i',j} \ge \left((b_{i,j} + \frac{u_{i,j}-\beta_i \cdot b_{i,j}}{\lambda_i}) \cdot q_{i,j}\right)/q_{i',j}$, which is computed based on the boosted score of advertiser i (under coupon). However, since the retention advertiser i originally wins the K^{th} slot with the help of coupon, we have $b_{i,j} + \frac{u_{i,j}-\beta_i \cdot b_{i,j}}{\lambda_i} > s_K^j/q_{i,j}$, leading to $\tilde{p}_{i',j} > s_K^j/q_{i',j}$. Since advertiser i' did not win any slot when reporting $b_{i',j} = v_{i',j}$ under the coupon provided to retention advertiser i, we also have $v_{i',j} \le s_K^j/q_{i',j}$, leading to $\tilde{p}_{i',j} > v_{i',j}$. □

Discussions on IC properties under the alternative value-maximizing bidder model [33]. Both assuming value-maximizing bidders, the utility model [8] we adopted differs from [33] in the sense that we did not impose preference on payment when the obtained value is the same. As a result, both first-price auction and second-price auction are truthful under model [8], but only second-price auction is truthful under model [33]. Since the major objective of advertisers is to maximize their own advertising performances, adopting utility model [8] could be a reasonable choice in our context. On the other hand, we may also view our design under the alternative value-maximizing model [33]. Under [33], part (1) of Theorem 3.1 holds normally, while for part (2) of Theorem 3.1, we can only guarantee the IC and IR properties for non-retention advertisers as well as the IR property for retention advertisers. This is because our Proposition 4.3 imposes an upper bound $d_{i,j}$ for coupon consumption, which leads the payment to be exactly the bidder's original bid when coupon is distributed effectively. Nevertheless, such a design is necessary to guarantee the (approximate) optimality of coupon distribution with control parameter λ_i, which allows its implementation to be feasible for online deployment. In future work, we will explore a design of coupon distribution to further guarantee the IC properties for value-maximizing retention advertisers with payment preference [33], while still enabling effective retention and feasible implementation.

B Performances of Fixed Penalty Parameter β

When estimating β_i for different advertisers, the platform might meet the cold-start problem, *i.e.*, it might be infeasible to access the personalized data for an advertiser on cvr_i and B_i^{camp}, and the platform has to utilize the data from other advertisers for a cold-start advertiser. To validate the effectiveness of our method when handling cold-start advertisers, we additionally test the performances of applying a fixed penalty parameter β for a group of cold-start advertisers. In detail, we apply $\beta = avg(\frac{cvr_i}{B_i^{camp}}) \cdot 0.3$ to all the advertisers in the *test set*, where *avg* denotes the averaging operation across all retention advertisers in the *training set*. The corresponding results are presented in Table 3. As we can observe, using a fixed pre-defined β does not greatly degrade the retention performance, demonstrating the robustness of our algorithms.

Table 3: Offline Experiment Results Using Fixed β

	ΔROI	ΔPV	$r(B^{camp})$	$r(B)$
Fixed-Param	45.90%	12.80%	96.68%	64.94%
Feedback Control	63.54%	16.35%	97.99%	87.16%
Coupon Agent	68.09%	17.26%	97.86%	84.39%