Sustainable Online Reinforcement Learning for Auto-bidding

Anonymous Author(s)
Affiliation
Address
email

Abstract

Recently, auto-bidding technique has become an essential tool to increase the return on investment (ROI) for advertisers. Facing the complex and ever-changing bidding environments in the real-world advertising system (RAS), state-of-the-art auto-bidding policies usually leverage reinforcement learning (RL) algorithms to generate real-time bids on behalf of the advertisers. Due to safety concerns, it was believed that the RL process can only be carried out in a virtual advertising system (VAS) which is usually built based on the historical data generated in the RAS. In this paper, we argue that there exists significant gaps between VAS and RAS, making the RL process in the VAS suffer from the problem of *inconsistency between online and offline* (IBOO). Firstly, we formally define the IBOO and systematically analyze its causes and influences. Then, we design a safety function and prove its Lipschitz smooth property in theory, which provides theoretical foundations for solving the IBOO problem. Moreover, based on this property, we propose a sustainable online RL (SORL) framework to continuously improve the auto-bidding polices through direct interactions with the RAS, which can completely resolve the IBOO problem. Specifically, we devise a safe and efficient online exploration policy that can continuously collect data from the RAS. We also develop a variance-suppressed conservative Q-learning method to improve the auto-bidding policies with the collected data. Extensive experiments on both simulated and real-world advertising systems validate the effectiveness of our approach.

1 Introduction

In the era of Internet, online advertising business has become one of the main profit models for many companies, such as Google [7] and Alibaba [4], which, at the same time, benefits millions of advertisers who are willing to bid for impression opportunities. Contemporary online advertising systems, as auctioneers, usually have large amount of candidate advertisers contesting for numerous impression opportunities at every moment. Making auction decisions based on accurate evaluation of each impression opportunity for all advertisers within several milli-seconds is computationally infeasible. Therefore, a real-world advertising system (RAS) adopts a cascade architecture. In this architecture, the auction of each impression opportunity is completed through several stages. Without loss of generality, we here simply view the RAS as a system with two stages: stage 1 and stage 2, as shown in Fig. 1. The auction process of each impression opportunity is described as follows:

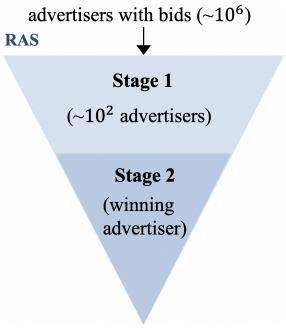


Figure 1: Two-stage cascade architecture in RAS.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

in stage 1, rough evaluations are conducted for all the candidate advertisers ($\sim 10^6$), and a group of the most promising advertisers ($\sim 10^2$) are fed to the stage 2; and in stage 2, accurate valuations and auction are carried out to determine the winning advertiser of the impression opportunity. The winner will gain the value of the impression and pay the market price. Faced with huge amount of impression opportunities at every moment, advertisers cannot bid in the granularity of individual opportunities. Recently, many auto-bidding policies have emerged to realize automatic real-time biddings for each impression opportunity on behalf of advertisers [6, 5, 3, 2], which have significantly increase their return on investment (ROI). State-of-the-art auto-bidding policies usually leverage reinforcement learning (RL) algorithms [6]. It was believed that the auto-bidding policies being updated cannot directly access to the RAS during the training process for safety concerns [2]. A common solution in most existing works [2, 3, 5, 6] is to learn the auto-bidding policy in a virtual advertising system (VAS) which is built based on the historical data of advertisers in the RAS. It is worth noting that the historical data is often collected merely from stage 2 of the RAS due to computational limitations.

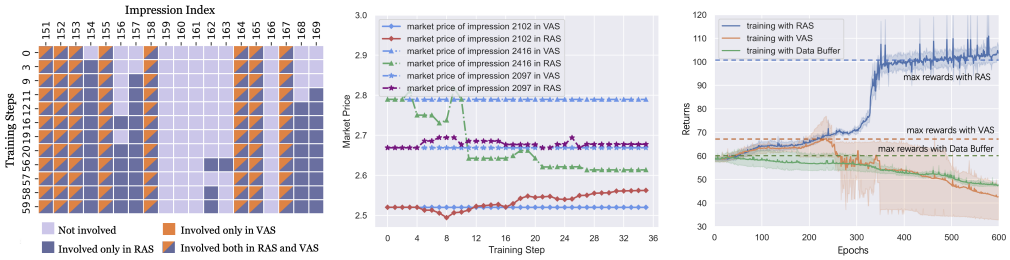
IBOO Challenges However, we argue that this common scheme suffers from the problem of *inconsistencies between online and offline* (IBOO). Here, "online" refers to the RAS, while "offline" refers to the VAS. In brief, the IBOO indicates the gaps between the RAS and VAS. Formally, the IBOO can be defined as follows.

Definition 1 (Inconsistencies Between Online and Offline)**.** *The IBOO are defined as the structural differences between the RAS and the VAS, which can degrade the real performance of auto-bidding policies trained with the VAS using RL when applied in the RAS.*

We next systematically analyze the causes and influences of the IBOO. *Causes:* There are two dominated causes of the IBOO. One is that the VAS is built based on the historical data of stage 2 that cannot accurately reflect the cascade auction architecture of the RAS. For example, an advertiser can bid for every impression opportunity in the VAS no matter how its policy changes during training. However, in the RAS, an advertiser can lose the impression opportunities that appear in the VAS if it fails to pass the stage 1, or can gain extra impression opportunities that are not in VAS if it passes the stage 1 due to policy improvements, as shown in Fig. 2(a). In addition, the bidding involving all advertisers naturally falls into the realm of multi-agent RL, where each single advertiser is affected by the policies of all other advertisers. However, the VAS does not involve the exact influences of other advertisers during RL training, which makes up the second cause. For example, in the RAS, the market prices are determined by the bids of all advertisers which can change during the training process (even in the situation where all other advertisers' policies are fixed). However, the VAS always provides constant market prices for advertisers, as shown in Fig. 2(b). *Influences of IBOO:* Essentially, the IBOO makes the VAS provides the advertiser with unreal rewards and state transition functions during RL training process. Fig. 2(c) displays the returns of the auto-bidding policies evaluated in the RAS during training with the VAS and RAS[1], respectively. It can be seen that, due to the IBOO, we can only obtain sub-optimal auto-bidding policies when training with the VAS.

Related Work To avoid the IBOO, one may consider training auto-bidding policies with traditional RL algorithms based on the data collected by a certain behavior policy directly from the RAS. However, this method will face extrapolation errors that can seriously degrade the policy performance [12]. The returns of training with a fixed data buffer shown in Fig. 2(c) testify this problem. Though we can leverage offline RL techniques [12, 14, 15] to mitigate it, the performance variance of the trained policy can be large [29] and further off policy evaluations (OPE) are required [13]. However, the OPE in auto-bidding needs to be manually designed based on the characteristics of the RAS, which can be time consuming and inaccurate. Besides, we may still have little confidence to widely apply the learned policies to advertisers for further online explorations in the RAS, even though they are proven to perform well by OPE. Note that there exist some safe online RL methods to explore in the environments with safety guarantees [21, 24, 26, 25, 23, 22]. Some popular algorithms [21, 23, 22] are built based on Constraint Markov Decision Process (CMDP) [17] where the constraint function naturally exists under each state-action pair. However, there are no such off-the-shelf constraint functions in the environment of the RAS. Other approaches are often designed based on systems with

[1] As we cannot directly train a standard RL algorithm with the RAS, we here conduct all the experiments in Fig. 2 with a simulated two-stage advertising system s-RAS and a virtual advertising system s-VAS generated from the historical data of s-RAS. Appendix A.2 for details.



(a) Impression opportunities. (b) Market prices. (c) RL rewards.

Figure 2: Structural differences and influences of IBOO. (a) shows that the impression opportunities within a certain period involving in the bidding of an advertiser remains the same in the VAS, but changes in the RAS in stage 2 during RL training process. (b) shows that the market prices remain the same in the VAS, but can rise, decrease or fluctuate around during RL training. (c) shows that auto-bidding policies can achieve higher rewards when training with the RAS than with the VAS and a fixed data buffer.

specific models and extra assumptions, which are not suitable for the auto-bidding problem. Recently, with the development of the offline RL field, many algorithms for efficient online explorations on the premise of having an offline dataset [16, 27] have emerged. However, they often focus on the efficiency of RL training process rather than the safety of it. Hence, new schemes need to be devised so as to avoid the IBOO and improve the performance of auto-bidding policies. Besides, it is worth noting that the IBOO resembles the *sim2real* problems studied in other realms, such as computer visions [9] and robotics [10]. Nonetheless, to the best of our knowledge, this is the first work that systematically analyzes and resolves the IBOO in the field of auto-bidding. Comprehensive related works are presented in Appendix B.

Main Contributions The contributions of this work are in four-fold. Firstly, we formally define the IBOO in auto-bidding and systematically analyze its causes and influences. Secondly, we design a safety function and prove its Lipschitz smooth property in theory, which provides theoretical foundations for solving the IBOO problem. Thirdly, based on this property, we propose a sustainable online RL (SORL) framework to continuously improve the auto-bidding polices through direct interactions with the RAS, which can completely resolve the IBOO problem. Specifically, we develop a safe and efficient exploration (SER) policy that can continuously collect data from the RAS. We also design a variance-suppressed conservative Q-learning (V-CQL) method to improve the auto-bidding policies based on the collected data with low performance variance. Furthermore, extensive experiments on both simulated and real-world advertising systems validate the effectiveness of our approach.

2 Problem Formulation

In this paper, we study the auto-bidding problem from the perspective of a single advertiser, which can be viewed as an episodic task of $T \in \mathbb{N}_+$ time steps. Between time step t and $t+1$, there are $N_t \in \mathbb{N}_+$ impression opportunities, each of which has a positive value $v_{j,t} \leq v_M$ and is sold to the winning advertiser at a market price $p_{j,t} \leq p_M$, where $v_M > 0$ and $p_M > 0$ are the upper bounds for values and market prices, respectively, $j \in \{1, 2, ..., N_t\}, t \in \{1, 2, ..., T\}$. Let the total budget of the advertiser be $B > 0$. The auto-bidding problem can be formulated as a CMDP $< \mathcal{S}, \mathcal{A}, R, \mathbb{P}, \gamma, C >$. The state $s_t \triangleq [b_t, T-t, B-b_t] \in \mathcal{S}$ is composed of three elements, including the budget left $s_t(1) = b_t$, the time left $s_t(2) = T - t$ and the budget consumed $s_t(3) = B - b_t$. The action $a_t \in \mathcal{A} \triangleq [A_{\text{min}}, A_{\text{max}}]$ is the bidding price at time step t, where $A_{\text{max}} > A_{\text{min}} > 0$ are the upper and lower bounds, respectively. The reward $r_t(s_t, a_t)$ and constraint function $c_t(s_t, a_t)$ are the total value of impressions won between time step t and $t+1$ and the corresponding costs, while $\mathbb{P}$ denotes the state transition rule. Note that R, C and $\mathbb{P}$ are all directly determined by the RAS. Moreover, $\gamma \in [0, 1]$ is the discounted factor. We denote the policy as $\mu : \mathcal{S} \to \mathcal{A}$, and let Π be the policy space.

The goal of the auto-bidding problem is to maximize the total value of impression opportunities earned by the advertiser under the budget constraint, which can be expressed as

$$\max_{\mu \in \Pi} \mathbb{E}_{s_{t+1} \sim \mathbb{P}, a_t \sim \mu} \left[\sum_{t=1}^{T} \gamma^t r_t(s_t, a_t) \right], \quad \text{subject to} \quad \sum_{t=1}^{T} c_t(s_t, a_t) \leq B. \tag{1}$$

One can leverage standard RL algorithms to learn a state-of-the-art auto-bidding policy [6] with the VAS, where the constraint in (1) is met by terminating the training episode early once the budget runs out. However, as stated before, this way of learning suffers from the challenge of IBOO. Hence, we first theoretically analyze the safety zones during online explorations in the RAS in Section 3 and then, in Section 4, propose the SORL framework to solve the auto-bidding problem in an online fashion, which can completely avoid the IBOO.

3 Theory: Lipschitz Safety Function

Unlike the safety functions in other realms such as robotics, it is not appropriate to construct an *immediate-evaluated* safety function to assess the safety degree of each state-action pair merely based on their values in auto-bidding. Actually, any action in any state would be safe as long as the summation of all subsequent rewards maintains at a high level. This implies that the safety degree of a state-action pair is equivalent to its Q value which is highly related to the behaviors in the following states. Hence, we here design the safety function directly as the Q function $Q^\mu(s_t, a_t)$.

Proposition 1 (Safety function)**.** *A proper safety function in auto-bidding should be a function of states, actions and policies, i.e., $\mathcal{S} \times \mathcal{A} \times \Pi \to \mathbb{R}$, which can directly be the Q function $Q^\mu(s_t, a_t)$.*

The detailed proofs can be found in Appendix D.2. Remind that the safety function can help determine which action is safer in the same state under the same policy, while comparing the value of the safety function between actions in distinct states or policies would be meaningless. We next analyze the characteristics of the safety function. According to the Bellman equation, we have:

$$Q^\mu(s_t, a_t) = r_t(s_t, a_t) + \mathbb{E}_{s_{t+1} \sim \mathbb{P}(\cdot|s_t, a_t)}[Q^\mu(s_{t+1}, \mu(s_{t+1}))]. \tag{2}$$

For theoretical analysis, we formulate the analytical expressions of the reward function R, the constraint function C and the transition rule $\mathbb{P}$. The proof can be found in Appendix D.1.

Proposition 2 (Analytical expressions of R, C and $\mathbb{P}$)**.** *Based on the characteristic of the two-stage cascaded auction in the RAS, we can formulate the reward function R as $r_t(s_t, a_t) = \sum_j \mathbb{1}\{a_t v^1_{j,t} \geq p^1_{j,t}, a_t v^2_{j,t} \geq p^2_{j,t}\} v_{j,t}$, the constraint function C as $c_t(s_t, a_t) = \sum_j \mathbb{1}\{a_t v^1_{j,t} \geq p^1_{j,t}, a_t v^2_{j,t} \geq p^2_{j,t}\} p_{j,t}$, and the state transition rule $\mathbb{P}$ as $s_{t+1} = s_t + [\triangle s_t(1), \triangle s_t(2), \triangle s_t(3)]$, where $\triangle s_t(1) = -\triangle s_t(3) = -\sum_j \mathbb{1}\{a_t v^1_{j,t} \geq p^1_{j,t}, a_t v^2_{j,t} \geq p^2_{j,t}\} p_{j,t}$ and $\triangle s_t(2) = -1$. Note that $p^1_{j,t}$ and $v^1_{j,t}$ denote the market price and rough value of impression j in stage 1, and $p^2_{j,t}$ and $v^2_{j,t}$ denote the market price and accurate value of impression j in stage 2.*

We make the following assumption on the impression opportunities, whose rationalities can be found in Appendix C.1.

Assumption 1 (Bounded Impression Distributions)**.** *Between time step t and $t+1$, we assume the numbers of winning impressions with action a_t in the first stage $n_{t,1}$ and the second stage $n_{t,2}$ can both be bounded by linear functions, i.e., $n_{t,1} \leq k_1 a_t, n_{t,2} \leq k_2 a_t$, where $k_1, k_2 > 0$ are constants.*

We claim that the reward function $r_t(s_t, a_t)$ is Lipschitz smooth with respect to actions a_t at any given state s_t. See Appendix E.1 for the proof.

Theorem 1 (Lipschitz Smooth of $r_t(s_t, a_t)$)**.** *Under Assumption 1, the reward function $r_t(s_t, a_t)$ is L_r-Lipschitz smooth with respect to actions a_t at any given state s_t, where $L_r = (k_1 + k_2) v_M$.*

We make the following mild assumptions, whose rationalities can be found in Appendix C.2.

Assumption 2 (Bounded Partial Derivations of $Q^\mu(s_t, a_t)$)**.** *We assume that the partial derivation of $Q^\mu(s_t, a_t)$ with respect to $s_t(1)$ and $s_t(3)$ is bounded, i.e., $\left|\frac{\partial Q^\mu(s_t, a_t)}{\partial s_t(1)}\right| \leq k_3$ and $\left|\frac{\partial Q^\mu(s_t, a_t)}{\partial s_t(3)}\right| \leq k_4$, where $k_3, k_4 > 0$ are constants.*

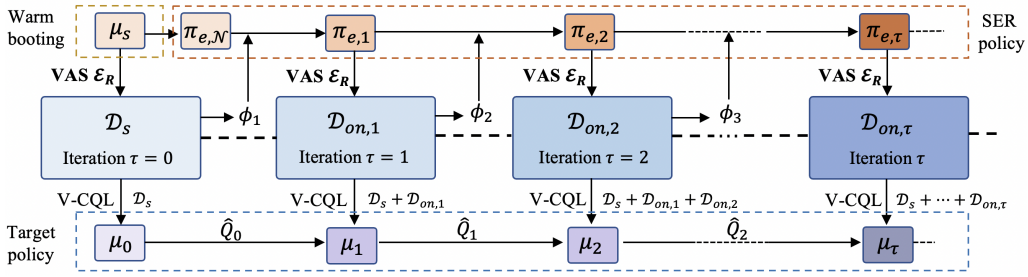


Figure 4: Sustainable online reinforcement learning (SORL) framework which learns the auto-bidding policy directly with the RAS in a safe and efficient way The SORL works in an iterative manner.

Based on Theorem 1, we can prove that the Q function $Q^\mu(s_t, a_t)$ is also Lipschitz smooth. See Appendix E.2 for proof.

Theorem 2 (Lipschitz Smooth of $Q^\mu(s_t, a_t)$)**.** *Under Assumption 1 and 2, the Q function $Q^\mu(s_t, a_t)$ is an L_Q-Lipschitz smooth function with respect to the actions a_t at any given state s_t, where $L_Q = [v_M + (k_3 + k_4)p_M](k_1 + k_2)$.*

This gives us a way to find the safety zone of actions a_t at each state s_t under policy μ with $Q^\mu(s, a)$. Suppose we have a safe policy μ_s that can direct interact with the RAS (e.g., a state-of-the-art auto-bidding policy [6] will do). Then, as shown in Fig. 3, a safety zone of actions a_t in state s_t under policy μ_s can be determined as $[\mu_s(s) - \xi, \mu_s(s) + \xi]$, where $\xi > 0$ is a small value. Specifically, if we wish the decrease of safety degrees (i.e., the Q value) within the safety zone to be no more than a threshold $\epsilon_s > 0$, then we can design ξ as

$$\xi \leq \frac{\epsilon_s}{[v_M + (k_3 + k_4)p_M](k_1 + k_2)} = \frac{\epsilon_s}{L_Q}. \tag{3}$$

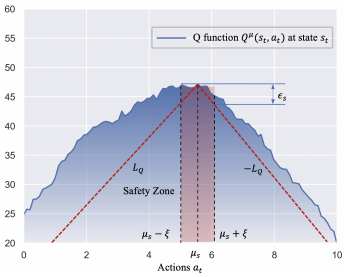


Figure 3: The safety function $Q^\mu(s, a)$ is L_Q-Lipschitz smooth, and the safety zone can be determined as $[\mu_s(s) - \xi, \mu_s(s) + \xi]$, where $\xi \leq \frac{\epsilon_s}{L_Q}$.

In addition, we remind that, theoretically, the safety zone of a state is effective only when the subsequent actions strictly follow the safe policy μ_s until the end of the episode. Nonetheless, we can relax this constraint in practice for the sake of exploration efficiency. For example, we can generate actions in several (even all) states by not following the safe policy μ_s, but by sampling within the safety zones. Experiment results in Section 5.1 demonstrate the feasibility of such operations. See Appendix F.2.1 for further illustrations on practical implementations. We highlight that the online explorations in the RAS would be possible if we always stay within the safety zones.

4 Algorithm: SORL Framework

We propose a sustainable online RL (SORL) framework for auto-bidding, where the policy is learned directly with the RAS. In the SORL framework, we design a safe and efficient exploration (SER) policy based on the Lipschitz property of the safety function to continuously collect data from the RAS; and we also develop a variance-suppressed conservative Q-learning (V-CQL) method to learn the target policies with the collected data. Fig. 4 shows the whole structure of the SORL framework.

Warm Booting To start with, we need a safe policy μ_s to boot the explorations in the RAS. Previous works on the auto-bidding problem provide us with a number of such safe policies, many of which have already been actually applied to the RAS. Here, we exploit the state-of-art auto-bidding policy [6], an RL method trained with the VAS, to boot the online explorations. Denote the data collected by μ_s as $\mathcal{D}_s = \{(s_k, a_k, r_k, s'_k)\}_k$.

4.1 Variance-suppressed Conservative Q-learning

Training the target policy with a fixed dataset $\mathcal{D}_s$ naturally falls into the realm of offline RL, where many advanced algorithms have been proposed, such as CQL [14], BCQ [12], etc. As shown later in Fig. 6, the performance of offline RL methods initialized with different parameters or hyper-parameters can have high variance in practice. Thus, extra parameter selection work needs to be done [29]. However, existing OPE methods for auto-bidding problem are not very accurate, and target policies with poor performance can affect the efficiency of subsequent online explorations (see Appendix F.2.2 for explanations). Hence, to decrease the performance variance of target policies, we develop the V-CQL method which is a variant of the CQL algorithm. Specifically, we implement the regularizer $\mathcal{R}(\mu)$ in the CQL with a novel expression:

$$\mathcal{R}(\mu) = \beta \mathbb{E}_{s_k \sim \mathcal{D}_s} \left[D_{\text{KL}} \left(\underbrace{\frac{\exp(\widehat{Q}_{\text{new}}(s_k, \cdot))}{\sum_a \exp(\widehat{Q}_{\text{new}}(s_k, a))}}_{\text{distribution of } \widehat{Q}_{\text{new}}}, \underbrace{\frac{\exp(\widehat{Q}_{\text{old}}(s_k, \cdot))}{\sum_a \exp(\widehat{Q}_{\text{old}}(s_k, a))}}_{\text{distribution of } \widehat{Q}_{\text{old}}} \right) \right], \tag{4}$$

where $\widehat{Q}_{\text{new}}$ is the Q function for the target policy[2], $\widehat{Q}_{\text{old}}$ represents the old Q function (here means the Q function of μ_s), $D_{\text{KL}}(\cdot, \cdot)$ represents the KL-divergence, and $\beta > 0$ is a constant controlling the weight of $\mathcal{R}(\mu)$. To interpret, the regularizer $\mathcal{R}(\mu)$ tries to limit the distance between the distribution of target policy's Q function and the distribution of the old Q function. This can limit the derivation target policy from old policy μ_s. Due to the Lipschitz smooth property of the safety function, the performance of target policies can have smaller variance. See Appendix F.1 for detailed illustrations. We denote the obtained target policy using the V-CQL based on $\mathcal{D}_s$ as μ_0.

4.2 Safe and Efficient Online Exploration Policy

Despite we control the variance of target policy's performance by the V-CQL, μ_0 is still not a good choice for further online explorations. On the one hand, μ_0 is a deterministic policy that is often lack of exploratory characteristics. On the other hand, its derivation degree from the safe policy μ_s cannot be regulated during explorations. Hence, we need to design a new stochastic exploration policy whose randomness and derivations from μ_s can be easily adjusted.

Vanilla Exploration Policy Due to the Lipschitz smooth property of the safety function, a vanilla online exploration policy can directly be a Gaussian stochastic policy $\pi_{e,\mathcal{N}}$ with mean of safety action $\mu_s(s_t)$ and variance of σ^2, i.e.,

$$\pi_{e,\mathcal{N}}(a_t|s_t) = \mathcal{N}(\mu_s(s_t), \sigma^2). \tag{5}$$

Based on the three-sigma rule, we can let $\sigma = \frac{1}{3}\xi \leq \frac{\epsilon_s}{3L_Q}$ to guarantee the safety of explorations.

SER Exploration Policy However, $\pi_{e,\mathcal{N}}$ resembles the policy μ_s to a large extent, which makes the explorations conservative. To lift up the efficiency of explorations, we shift the distribution of $\pi_{e,\mathcal{N}}$ towards the state-actions pairs that are potentially promising, while constraining its derivations within a threshold $\epsilon_e > 0$. This can be formulated as a functional optimization problem:

$$\max_{\pi_e, \forall s_t} \quad \mathbb{E}_{a_t \sim \pi_e(\cdot|s_t)} \phi(s_t, a_t) \qquad \text{s.t.} \quad D_{\text{KL}}(\pi_e, \pi_{e,\mathcal{N}}) \leq \epsilon_e, \tag{6}$$

where the optimization variable π_e is the SER policy, $\phi(s_t, a_t)$ is the *potential function* evaluating the potential degrees of state-action pairs. Note that a vanilla choice for $\phi(s_t, a_t)$ can directly be the Q function $\widehat{Q}_{\text{new}}$ trained on $\mathcal{D}_s$ with the V-CQL. Further designs of $\phi(s_t, a_t)$ are shown in Appendix F.2.2. Based on Euler equations, we can derive the form of π_e from (7) as

$$\pi_e = \frac{\pi_{e,\mathcal{N}}}{C(s_t)} \exp\left\{ \frac{1}{\lambda} \phi(s_t, a_t) \right\} = \frac{1}{C(s_t)\sqrt{2\pi\sigma^2}} \exp\left\{ \underbrace{-\frac{(a_t - \mu_s(s_t))^2}{2\sigma^2}}_{\text{safety}} + \underbrace{\frac{1}{\lambda}\phi(s_t, a_t)}_{\text{efficiency}} \right\}, \tag{7}$$

[2]Note that we use $\widehat{Q}$ to denote the Q function obtained from RL training, and Q to denote the real Q function.

where $C(s_t) = \int_{a_t} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\{-\frac{(a_t-\mu_s(s_t))^2}{2\sigma^2} + \frac{1}{\lambda}\phi(s_t, a_t)\}\mathrm{d}a_t$ acts as the normalization factor. The complete deductions of π_e are given in Appendix F.2.3. We can interpret π_e as a deviation from $\pi_{e,\mathcal{N}}$ towards the potential function $\exp\{\frac{1}{\lambda}\phi(s_t, a_t)\}$, where $\pi_{e,\mathcal{N}}$ guarantees the safety and $\exp\{\frac{1}{\lambda}\phi(s_t, a_t)\}$ ensures the efficiency. Note that σ and λ in (7) are both hyper-parameters that can control the randomness and derivations from μ_s of the SER policy, respectively. As shown in Fig. 5, the smaller the value of λ, the larger the derivations of π_e from μ_s; besides, the bigger the value of σ, the greater the randomness of π_e. Hence, one can control the randomness and derivations from the safe policy μ_s of exploration policy easily by adjusting the value of σ and λ. We note that for safety concerns, we should keep $\sigma \leq \frac{\epsilon_s}{3L_Q}$. A way to implement the SER policy π_e in practice is described in Appendix F.2.4.

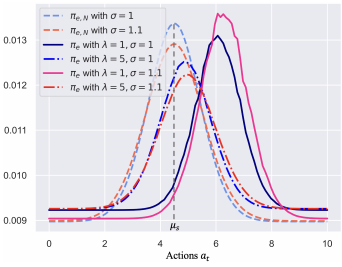


Figure 5: Exploration policies π_e under different σ and λ, as well as $\pi_{e,\mathcal{N}}$ with different σ.

4.3 Iteratively Updated Structure

The SORL framework is designed to work in an iterative manner, as shown in Fig. 4. Specifically, at each iteration $\tau \in \mathbb{N}$, the SER policy collects data from the RAS, and both the target policy and the SER policy are updated based on the collected data for next iteration. Denote the SER policy and the target policy in the τ-th iteration as $\pi_{e,\tau}$ and μ_τ, and the data collected in the τ-th iteration as $\mathcal{D}_{on,\tau}$. A straight design for $\pi_{e,\tau+1}$ can be the policy in (7) where $\pi_{e,\mathcal{N}}$ is substitute by $\pi_{e,\tau}$, i.e.,

$$\pi_{e,\tau+1} \leftarrow \frac{\pi_{e,\tau}}{C_{\tau+1}(s_t)} \exp\left\{\frac{1}{\lambda_{\tau+1}}\phi_{\tau+1}(s_t, a_t)\right\} = \frac{\pi_{e,\mathcal{N}}}{\prod_{i=1}^{\tau+1} C_i(s_t)} \exp\left\{\sum_{i=1}^{\tau+1} \frac{1}{\lambda_i}\phi_i(s_t, a_t)\right\}, \quad (8)$$

where $C_{\tau+1}(s_t)$ acts as the normalization factor, λ_τ is the hyper-parameter used in iteration τ, and $\pi_{e,0} = \pi_{e,\mathcal{N}}$. However, (8) requires us to store all the previous potential functions to calculate the current SER policy, which is costly. Hence, we omit the history potential functions $\{\phi_i\}_{i=1}^{\tau}$ and gradually decrease λ_τ to zero, i.e., $\lambda_1 > \lambda_2 > \cdots \lambda_\tau > 0$. Thus, the SER policy is modified to $\pi_{e,\tau+1} = \frac{1}{C'_{\tau+1}(s_t)}\pi_{e,\mathcal{N}} \exp\{\phi_{\tau+1}(s_t, a_t)/\lambda_{\tau+1}\}$. We apply the V-CQL to iteratively improve the target policy. Specifically, at iteration τ, we substitute $\widehat{Q}_{\text{old}}$ in (4) with $\widehat{Q}_{\tau-1}$ which is the Q function learned in iteration $\tau - 1$, and train a new Q function $\widehat{Q}_{\text{new}} \leftarrow \widehat{Q}_\tau$ for the next iteration. The final target policy would be $Q^* = \lim_{\tau\to\infty} \widehat{Q}_\tau$. A summary of the SORL framework is present in Appendix F.3.

5 Experiments

We conduct experiments on both the simulated advertising system and the RAS. The following three questions are mainly studied in the experiments: (1) Does the V-CQL help to reduce the performance variance of the trained policy? (2) Are the proposed vanilla exploration policy and SER policy safe when exploring in the RAS (with a safety threshold of $\epsilon_s = 5\%$)? (3) Can the proposed SORL framework continuously improve the target policy?

Experiment Setup *RAS:* We conduct our experiments on one of the world's largest E-commerce platforms, TaoBao. See Appendix G.1 for details. *Evaluation Metrics:* We utilize four metrics to evaluate the performance of our approach. The first metric the total **consumed budget (ConBdg)** of the advertiser. Note that the ConBdg will not exceed the total budget due to the mechanisms in the advertising system. Hence, the larger the ConBdg, the more revenue the advertiser can obtain. The second metric is the **ROI** which is defined as the ratio between the total revenue and the ConBdg of the advertiser. The third metric is the **BuyCnt** which is the total number of the purchased impressions. The fourth metric is the **cost per acquisition (CPA)** which is defined as the average cost for each successfully converted impression. Clearly, large values of BuyCnt and ROI with a

small CPA indicates a good performance of the auto-bidding policy. *Baselines:* **Unified Solution to Constrained Bidding (USCB)** [6] is the state-of-the-art auto-bidding policy which outperforms previous algorithms in terms of both effectiveness and stability. Specifically, USCB is a RL method trained in the VAS. **Conservative Q-learning (CQL)** [14] is a popular offline RL method that can mitigate extrapolation errors by learning conservative Q functions. The initial safety policy μ_s for booting the explorations in the RAS is trained by the USCB.

5.1 Main Results

To Answer Question (1): We leverage the V-CQL and CQL algorithm to train a set of the target policies based on the collected data with 100 different initial parameters. To ensure fairness, we also apply the USCB to train a set of target policies in the RAS with exactly the same 100 initial parameters. We carry out the OPE for these three sets of target policies using the VAS where the initial safety policy μ_s is trained, as shown in Fig. 6. Note that the $\mathbf{R}/\mathbf{R}^*$ is a main metric used in the OPE for auto-bidding (see Appendix G.2). We can see that the maximum $\mathbf{R}/\mathbf{R}^*$ of the V-CQL is larger than that of the USCB as well as that of the μ_s, which indicates that the V-CQL is capable of training a better policy. Although the maximum $\mathbf{R}/\mathbf{R}^*$ of the V-CQL and the CQL are about in the same level, the variance of $\mathbf{R}/\mathbf{R}^*$ of the V-CQL is smaller than that of the CQL. This indicates that the V-CQL can reduce the performance variance compared with CQL when initializing with different parameters. For real-world experiments, we apply the three sets of policies trained by the V-CQL, CQL and USCB to $1,600$ advertisers for seven days in the RAS, and the average values of the metrics are shown in Table 1. We can see that the V-CQL outperforms USCB and CQL methods in almost all metrics.

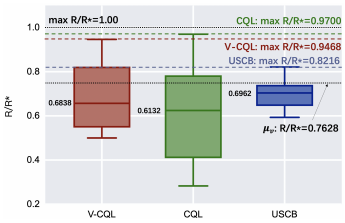


Figure 6: OPE for V-CQL, CQL and USCB methods.

Table 1: The performance of the V-CQL, CQL and USCB algorithms in the RAS.

Methods	USCB: newly trained target policies				USCB: initial safety policy			
	ROI	BuyCnt	CPA	ConBdg	ROI	BuyCnt	CPA	ConBdg
USCB	3.90	40,926	20.71	847,403.12	3.82	35,627	21.58	**768,832.64**
CQL	3.87	40,462	21.42	845,621.15	3.71	34,103	22.83	761,361.19
V-CQL	**3.95**	**42,236**	**20.29**	**856,913.14**	**3.97**	**37,090**	**20.61**	764,467.39

To Answer Question (2): As the performance variance of the V-CQL is still larger than that of the USCB, we need to fully examine the safety of the SER policy using different $\phi(s,a)$ corresponding to target policies of different performance levels. In the simulated advertising system, we train seven different versions of target policies with V-CQL method which have various performance levels. We construct seven SER policies with the potential functions $\{\phi_i(s,a)\}_{i=1}^7$ being the Q functions of the target policies. We also construct a vanilla exploration policy $\pi_{e,\mathcal{N}}$ for comparison. The hyper-parameters for all exploration policies are $\sigma = 1, \lambda = 0.1$. We apply the $\pi_{e,\mathcal{N}}$ and seven SER policies to the simulated advertising system, and the ROIs of explorations policies are shown in Fig. 7. We can see that the ROI of the SER policy rises with the performance level of the target policy trained by the V-CQL. The worst ROI of the SER policy drops about 3.52% compared with the initial safety policy, which is smaller than the safety threshold $\epsilon_s = 5\%$, as well as the decline rate in ROI of the corresponding target policy 6.76%. This indicates the safety of the SER policy. Moreover, we can control the safety degree of the SER policy by regulating the hyper-parameters σ and λ, and the results are shown in Appendix G.3.2. In addition, we can see that the ROI of SER policy is a higher than the vanilla policy when the performance of target policies is better than the initial safety policy (this is why we propose the V-CQL method to reduce the performance variance of CQL).

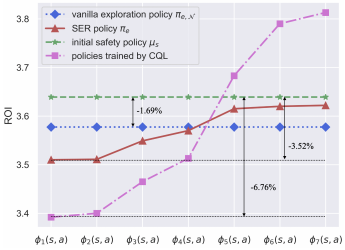


Figure 7: The ROI of two vanilla exploration policies and SER policies when using different versions of policies trained by CQL.

For real-world experiments, we apply π_e and $\pi_{e,\mathcal{N}}$ to $1,300$ advertisers in the RAS and compare the performance with the initial safety policy μ_s, as shown in Table. 2. We can see that the varieties of the ConBdg and BuyCnt of π_e are all with in the safety threshold 5%, and are better than those of $\pi_{e,\mathcal{N}}$, which indicates the safety of π_e.

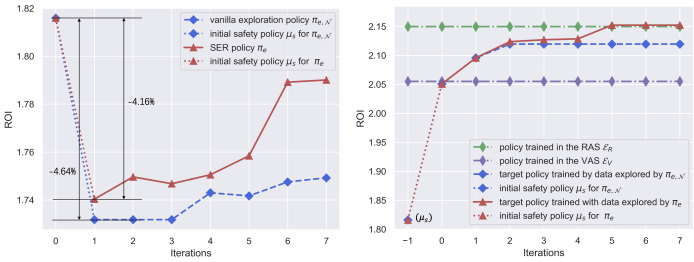
Table 2: The performance variety of π_e and $\pi_{e,\mathcal{N}}$ compared to μ_s in the RAS.

Methods	BuyCnt	ConBdg
vanilla $\pi_{e,\mathcal{N}}$	-3.32%	1.11%
SER policy π_e	**-1.83%**	**0.67%**

Table 3: The results of SORL framework in the real-world advertising system.

Iterations	A/B Tests with previous target policy				A/B Tests with USCB (state-of-the-art)			
	ROI	BuyCnt	CPA	ConBdg	ROI	BuyCnt	CPA	ConBdg
0-th: μ_0	**1.28%**	**3.20%**	**-2.01%**	**1.12%**	**1.28%**	**3.20%**	**-2.01%**	**1.12%**
1-th: μ_1	**1.96%**	**0.65%**	**-1.27%**	-0.62%	**2.88%**	**3.41%**	**-0.93%**	**2.45%**
2-th: μ_2	**0.26%**	**0.47%**	**-0.13%**	**0.33%**	**1.60%**	**3.57%**	**-0.98%**	**2.55%**
3-th: μ_3	**3.20%**	**0.95%**	**-1.01%**	**0.06%**	**2.48%**	**3.75%**	**-3.91%**	-0.15%

To Answer Question (3): We first apply the SORL algorithm in the simulated advertising system, and the results during iterations are shown in Fig. 8. Specifically, from Fig. 8(a), we can see that the decline rate in ROI of both SER policies and the vanilla exploration policies are smaller than $\epsilon_s = 5\%$, which implies the safety of the proposed exploration policies. Moreover, the ROI of the SER policy $\pi_{e,\tau}$ rises with iterations and is alway higher than that of $\pi_{e,\mathcal{N}}$, which indicates that the SER policy is safer. From Fig. 8(b), we can see that the ROI of the target policy μ_τ trained with data collected by $\pi_{e,\tau}$ rises with the number of iterations and converges to the optimal ROI (i.e., the ROI of the policy trained directly in the simulated advertising system) at the 5-th iteration. In addition, the ROI of target policy trained with data collected by $\pi_{e,\tau}$ is higher than the ROI of the target policy trained with data collected by $\pi_{e,\mathcal{N}}$, as well as the ROI of the policy trained with the VAS. This indicates the efficiency of the SER policy. For real-world experiments, we utilize $10,000$ advertisers to collect data from the RAS with π_e and compare the target policies in 4 iterations on $1,500$ advertisers using A/B tests, and the results are shown in Table. 3. We can see that the performances of target policies are getting better with iterations and exceeds the state-of-the-art auto-bidding policy USCB, which validates the effectiveness of the SORL framework.



(a) ROI of $\pi_{e,\tau}$ and $\pi_{e,\mathcal{N}}$. (b) ROI of target policy μ_τ.

Figure 8: The change of ROI of both the SER policy $\pi_{e,\tau}$ and target policy μ_τ with iterations τ when applying the SORL framework.

6 Conclusions

In this paper, we define the IBOO in auto-bidding problems and systematically analyze its manifestations, causes and influences. Then, we design a safety function to identify the safety zones in auto-bidding and prove its Lipschitz smooth property, which provides theoretical guarantees for solving the IBOO problem. Based on this property, we propose a SORL framework to directly learn auto-bidding policies with the RAS. The SORL works in an iterative way, where the SER policy is designed to safely and efficiently collect data from the RAS, and the V-CQL is designed to continuously improve the target auto-bidding policies. Finally, extensive experiments on both the simulated advertising system and the RAS validates the effectiveness of the proposed SORL framework. A promising future work can be extending the SORL framework to all online advertisers.

References

[1] X. Yang, Y. Li, H. Wang, D. Wu, Q. Tan, J. Xu, and K. Gai, Bid optimization by multivariable control in display advertising, in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining,* Jul. 2019.

[2] D. Wu, X. Chen, X. Yang, et al. Budget constrained bidding by model-free reinforcement learning in display advertising, in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management,* Oct. 2018.

[3] X. Hao, Z. Peng, Y. Ma, et al., Dynamic knapsack optimization towards efficient multi-channel sequential advertising, in *International Conference on Machine Learning,* 2020.

[4] Alibaba. 2022. alimama, https://www.alimama.com/index.htm.

[5] H. Cai, K. Ren, W. Zhang, et al., Real-time bidding by reinforcement learning in display advertising, in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining,* Jan. 2017.

[6] Y. He, X. Chen, D. Wu, et al., A unified solution to constrained bidding in online display advertising, in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, Aug. 2021.

[7] Google. 2022. google ads, https://ads.google.com/.

[8] Z. Wang, L. Zhao, B. Jiang B, et al. Cold: Towards the next generation of pre-ranking system, *arXiv preprint arXiv:2007.16122,* Aug. 2020.

[9] D., Carl, and A Zisserman, Sim2real transfer learning for 3d human pose estimation: motion to the rescue, *Advances in Neural Information Processing Systems* 32, 2019.

[10] S. Höfer, K. Bekris, A. Handa, et al. Sim2Real in robotics and automation: Applications and challenges, *IEEE transactions on automation science and engineering* vol. 18, no. 2, pp. 398–400, Apr. 2021.

[11] X. Liu, C. Yu, Z Zhang, et al., Neural auction: End-to-end learning of auction mechanisms for e-commerce advertising, *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining,* Aug. 2021.

[12] S. Fujimoto, D. Meger, and D. Precup, Off-policy deep reinforcement learning without exploration, *International Conference on Machine Learning,* 2019.

[13] C. Voloshin, HM. Le, N. Jiang, and Y. Yue, Empirical study of off-policy policy evaluation for reinforcement learning, *arXiv preprint arXiv:1911.06854,* 2019.

[14] A. Kumar, A. Zhou, G. Tucker, and S. Levine, Conservative q-learning for offline reinforcement learning, *Advances in Neural Information Processing Systems,* 2020.

[15] A. Kumar, J. Fu, M. Soh, G. Tucker, and S. Levine, Stabilizing off-policy q-learning via bootstrapping error reduction, *Advances in Neural Information Processing Systems,* 2019.

[16] A. Nair, A. Gupta, M. Dalal, and S. Levine, Awac: Accelerating online reinforcement learning with offline datasets, *arXiv preprint arXiv:2006.09359*, 2020.

[17] E. Altman, Constrained Markov decision processes: stochastic modeling, *Routledge,* 1999.

[18] TP. Lillicrap, JJ. Hunt, A. Pritzel, et al., Continuous control with deep reinforcement learning, *arXiv preprint arXiv:1509.02971,* 2015.

[19] V. Mnih, K. Kavukcuoglu, D. Silver, et al. Human-level control through deep reinforcement learning, *nature,* vol. 518, no. 7540, pp. 529–533, 2015.

[20] J. Shi, Y. Yu, Q. Da, et al. Virtual-taobao: Virtualizing real-world online retail environment for reinforcement learning, *Proceedings of the AAAI Conference on Artificial Intelligence.,* 2019.

[21] H. Bharadhwaj, A. Kumar, N. Rhinehart, et al. Conservative safety critics for exploration, *arXiv preprint arXiv:2010.14497,* 2020.

[22] DRR. Scobee, SS. Sastry, Maximum likelihood constraint inference for inverse reinforcement learning, *arXiv preprint arXiv:1909.05477,* 2019.

[23] U. Anwar, S. Malik, A. Aghasi, et al. Inverse Constrained Reinforcement Learning, *arXiv preprint arXiv:2011.09999,* 2020.

[24] B. Thananjeyan, A. Balakrishna, U. Rosolia, et al., Safety augmented value estimation from demonstrations (saved): Safe deep model-based rl for sparse cost robotic tasks, *IEEE Robotics and Automation Letters,* vol. 5, no. 2, pp. 3612–3619, 2020.

[25] F. Berkenkamp, M. Turchetta, A. Schoellig, et al., Safe model-based reinforcement learning with stability guarantees, *Advances in neural information processing systems,* 2017.

[26] T. Koller, F. Berkenkamp, M. Turchetta, et al. Learning-based model predictive control for safe exploration, *IEEE conference on decision and control (CDC),* 2018.

[27] S. Lee, Y. Seo, K. Lee, et al. Offline-to-Online Reinforcement Learning via Balanced Replay and Pessimistic Q-Ensemble, *Conference on Robot Learning,* 2022.

[28] D. Wu, X. Chen, X. Yang, et al. Budget constrained bidding by model-free reinforcement learning in display advertising, *Proceedings of the 27th ACM International Conference on Information and Knowledge Management,* 2018.

[29] TL. Paine, C. Paduraru, A. Michi, et al., Hyperparameter selection for offline reinforcement learning, *arXiv preprint arXiv:2007.09055,* 2020.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 6
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We believe there is no negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 3 and Appendix C
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix D and Appendix E
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A] Our work uses own data, and the data used in the simulation experiments and real-world experiments is released in supplementarymaterial.
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]