Model-Based Reinforcement Learning for Auto-bidding in Display Advertising

Shuang Chen*
Ant Group
Shanghai, China
shuangchen.cs@alibaba-inc.com

Qisen Xu*
Ant Group
Shanghai, China
qisen.xqs@antgroup.com

Liang Zhang
Ant Group
Shanghai, China
zhuyue.zl@antgroup.com

Yongbo Jin
Ant Group
Shanghai, China
yongbo.jyb@antgroup.com

Wenhao Li†
The Chinese University of Hong Kong, Shenzhen
Shenzhen, China
liwenhao@cuhk.edu.cn

Linjian Mo†
Ant Group
Shanghai, China
linyi01@antgroup.com

ABSTRACT

Real-time bidding (RTB) achieves outstanding success in online display advertising, which has become one of the most influential businesses. Given historical ad impressions under the second price auction mechanism, the advertiser's optimal bidding strategy is determined by the core parameter corresponding to the optimal solution of a constrained optimization problem. However, the sequentially arrived impressions in online display advertising make it highly non-trivial to obtain the optimal core parameter in advance without knowing the complete impression set. For this reason, recent methods have generally transformed the core parameter determination problem into a sequential parameter adjustment problem and solved it using reinforcement learning (RL). This paper proposes a simple and effective **M**odel-**B**ased **A**utomatic **B**idding algorithm, **MBAB**, which explicitly models the uncertainty of the dynamic auction environment and then uses the dynamic programming algorithm to obtain the current optimal adjustment of the core parameter. MBAB can avoid burdensome simulated environment construction and is more suitable for production deployment without the thorny sim-to-real issue than model-free methods. Furthermore, MBAB uses the optimal bidding formula to carry out coarse-grained modeling of the online market environment to alleviate the scalability problem caused by fine-grained environment modeling of previous model-based methods. In order to accurately describe the impression distribution and non-stationarity of the online market environment, we introduce the probabilistic modeling method and propose a novel monotonicity constraint to regulate the model output. Numerical experiments show that the proposed MBAB substantially outperforms existing baselines on various constrained RTB tasks in the production environment.

KEYWORDS

Real-Time Bidding; Constrained Bidding; Display Advertising; Model-Based Reinforcement Learning

[*] Equal contribution
[†] Corresponding authors

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom. © 2023 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

ACM Reference Format:
Shuang Chen, Qisen Xu, Liang Zhang, Yongbo Jin, Wenhao Li, and Linjian Mo. 2023. Model-Based Reinforcement Learning for Auto-bidding in Display Advertising. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 9 pages.

1 INTRODUCTION

Real-time bidding (RTB) achieves outstanding success in online display advertising, which has become one of the most influential businesses, with $56.7 billion revenue and holds a 30% share of total internet advertising revenue for FY 2021 in US alone [13]. In RTB, advertisers need to adopt a bidding strategy to deliver their bids to compete for each ad impression opportunity, which is usually associated with the expectation of the desired ad clicks and conversions. The actual bidding cost of advertisers depends on the auction mechanism adopted by the advertising platforms. In this paper, we focus on the second-price auction mechanism, i.e., each ad impression opportunity is assigned to the highest bidder, which only needs to pay the second highest bid [8].

In order to satisfy the different demands of advertisers, such as maximizing conversions while keeping the average cost per impression/conversion within specific budgets, advertising platforms usually provide advertisers with customized bidding strategies. This bidding strategy generally corresponds to an online optimization problem with constraints, or more specifically, a feasible solution to the online knapsack problem [16, 30]. Under the second price auction mechanism, given the value v for each ad impression opportunity, an advertiser's optimal bidding strategy corresponds to the hyperparameter w value. However, in RTB of online display advertising, the optimal value for w is highly non-trivial to obtain because the bidding strategy optimization algorithm needs to consider a large number of ad impressions and the specific needs of different advertisers at the same time, coupled with the non-stationary nature of the bidding environment [27].

In other words, since the impressions arrive sequentially in a day, it is of great challenge to calculate the optimal value for hyperparameter w in advance without the complete impression set. Therefore, recent works consider the online data stream properties of the impressions described above and transform the determination of the optimal value of w into a sequential parameter tuning problem, or a sequential decision problem, in the framework of constrained bidding. Thanks to the excellent performance of deep reinforcement learning (RL) algorithms in dealing with sequential decision problems, the above methods can be divided into two main categories, model-free reinforcement learning (MFRL) methods [10, 27] and model-based reinforcement learning (MBRL) methods [4].

Directly deploying the bidding agent to interact with the environment in an online task will likely lead to economic losses and customer churn. Therefore, RL methods can only train to bid agents "offline" [1] based on historical bidding data. The existing MFRL and MBRL methods mainly take two approaches to distill knowledge from historical data, respectively. The former simulates online RTB scenarios by directly replaying historical data, supplemented by random disturbances and advertising platform filtering rules to build a simulation environment. The latter converts the constrained RTB problem into a Markov Decision Process (MDP), preprocesses the historical data, and converts it into transition data to support the training of transition dynamic model based on supervised learning.

However, the construction of the simulator involves the calibration of a large number of parameters to fit the realistic environment for model-free methods. In addition, existing model-free methods mainly adopt the independent learning scheme [23], and the agent does not perceive the existence of other agents in the environment when optimizing the policy but regards these agents as part of the environment. A non-stationary environment due to simultaneous policy updates of all agents will make the migration of the agents trained in the simulator to the production environment face the thorny sim-to-real [32] issue. Although model-based methods can naturally avoid the above issues, the algorithm needs to consider hundreds of millions of impressions to provide advertisers with the optimal bidding strategy in online RTB markets, making it difficult to accurately model the state transition and extend the existing model-based methods to practical tasks. The engineering overhead associated with simulator construction, policy transfer, and generalization from simulated to actual environments, or the difficulty of modeling high-dimensional stochastic RTB environments, leave much room for improvement in existing RL-based automated bidding methods for RTB tasks in online display advertising.

In this paper, we focus on retaining the advantages of the model-based method while enhancing its practicability by re-modeling the RTB at a coarse-grained level under the MBRL context inspired from the *optimal bidding model* [31]. Unlike existing model-based methods, the MDP we constructed does not estimate the market price of each impression and compute the auction-level metrics but instead models the uncertainty of budget consumption and win value, which significantly reduces the learning complexity of the transition model. In addition, to better capture the highly non-stationary bidding market due to dynamic bidding behavior and independent learning modeling, we explicitly consider the uncertainty of state transitions, which are naturally incorporated into the modeling of coarse-grained MDPs.

Based on the coarse-grained bidding MDP, we propose a simple and effective **M**odel-**B**ased **A**utomatic **B**idding framework, **MBAB** (Figure 1), which is divided into two stages, the learning of the non-stationary transition model and the model predictive control based on the dynamic programming of the learned model to obtain the optimal bidding strategy under constraints. In order to accurately describe the impression distribution and non-stationarity of the online market environment, we introduce the probabilistic modeling method and propose a novel monotonicity constraint to regulate the model output. Numerical experiments show that the proposed MBAB substantially outperforms existing baselines on various constrained RTB tasks in the production environment.

Our main contributions are as follows: (1) we model the auto-bidding problem as a coarse-grained MDP based on the optimal bidding model, which significantly reduces the learning complexity of the model-based method; (2) we propose a simple and effective model-based RL framework, MBAB, which can avoid burdensome simulated environment construction and is more suitable for production deployment without the thorny sim-to-real issue; (3) we introduce the probabilistic modeling method and propose a novel monotonicity constraint to accurately describe the impression distribution and non-stationarity of the online market environment in the transition model learning.

2 PRELIMINARIES AND RELATED WORK

2.1 Constrained Bidding

In the RTB system [29], advertisers bid for ad impression opportunities with impression value estimation, e.g., predicted click-through rate (PCTR) [21] and predicted conversion rate (PCVR) [6]. Their optimization goal is to maximize the total value of winning impressions under long-term constraints. Advertisers in the real world consider various long-term constraints, such as daily budget, cost per action (CPA), and return on investment (ROI). Under the second price auction mechanism, Zhang et al. [31] proves the optimal bidding formula for bidding with the budget constraint, and He et al. [11] further derives a unified optimal bidding formula for the constrained bidding problem by the primal-dual method [2]. According to different constraints and optimization goals, these optimal bid formulas take different bid formats, whose core parameters can be solved numerically by historical data [31]. However, obtaining optimal parameters for future bidding is not trivial. As advertisers do not know the complete impression set in advance, and the auction competitors are dynamic [5], the optimal parameters calculated from historical data may not be optimal. As a result, we should dynamically adjust the parameters in the optimal bidding formula under the actual market environment. Automated bidding (auto-bidding) is a machine-based bidding strategy designed to calculate the real-time price the advertiser would like to pay for an ad opportunity. Based on this technology, a straightforward bidding strategy is to dynamically tune core parameters of the optimal bidding formulas to satisfy bidding constraints [28]. In this paper, we also take advantage of the optimal bidding formula. An uncertainty-aware model-based reinforcement learning framework is proposed for auto-bidding to adjust the core parameters dynamically.

[1] This paper does not consider offline RL [14, 20] as a technical route, and to the best of our knowledge, there is no work that implements offline RL into online real-time bidding scenarios. However, offline RL is theoretically well suited for solving constrained bidding problems in online display advertising, and we will study it in depth in our future work.

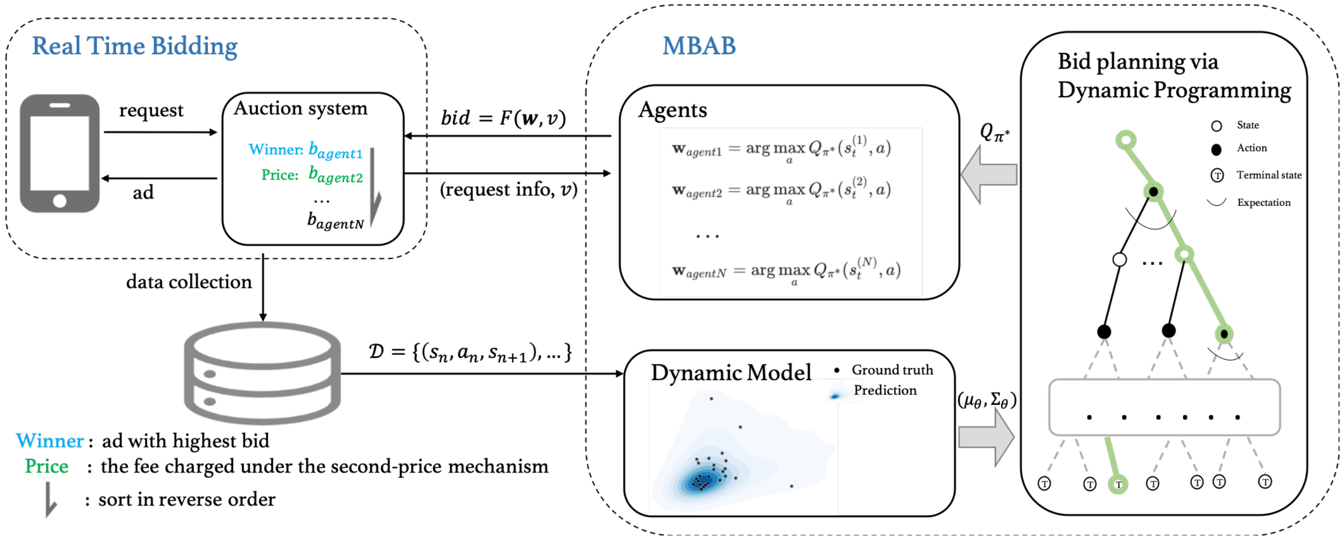

Real Time Bidding
request
ad
Auction system
Winner: b_{agent1}
Price: b_{agent2}
…
b_{agentN}
$bid = F(\mathbf{w}, v)$
(request info, v)
data collection
$\mathcal{D} = \{(s_n, a_n, s_{n+1}), \dots\}$
Winner : ad with highest bid
Price : the fee charged under the second-price mechanism
: sort in reverse order
MBAB
Agents
$\mathbf{w}_{agent1} = \arg\max_a Q_{\pi^*}(s_t^{(1)}, a)$
$\mathbf{w}_{agent2} = \arg\max_a Q_{\pi^*}(s_t^{(2)}, a)$
…
$\mathbf{w}_{agentN} = \arg\max_a Q_{\pi^*}(s_t^{(N)}, a)$
Q_{π^*}
Dynamic Model
Ground truth
Prediction
$(\mu_\theta, \Sigma_\theta)$
Bid planning via Dynamic Programming
State
Action
Terminal state
Expectation
T

Figure 1: Model-based reinforcement learning for Auto-Bidding (MBAB). The dynamic environment model $\hat{\mathcal{P}}_\theta$ is trained using the dataset $\mathcal{D}$ collected from the logs in the online RTB system. With the parameterized distribution over the state transition predicted by $\hat{\mathcal{P}}_\theta$, MBAB conducts the bid planning via the dynamic programming and computes the optimal action-value function Q_{π^*} under optimal bidding policy π^*. At each time step, the parameters w of agents are updated by maximizing $Q_{\pi^*}(s_t, \mathbf{w})$. In a real auction environment, bidding agents receive the request info and estimate the winning value v , and then compute the real-time bids by optimal bidding formula $F(\mathbf{w}, v)$.

2.2 Reinforcement Learning

Reinforcement Learning (RL) is an area of machine learning used to solve the MDP. The goal of an RL agent is to maximize the expected reward when sequentially interacting with the MDP environment. An MDP can be described by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ where $\mathcal{S}$ and $\mathcal{A}$ indicate the state and action space respectively. The immediate reward $r(s_t, a_t, s_{t+1}) \in \mathcal{R}$ received from the environment at new state $s_{t+1} \in \mathcal{S}$ after an action $a_t \in \mathcal{A}$ at timestep t is taken under the current state $s_t \in \mathcal{S}$ is determined by the reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$. State transition probability function $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ measures the uncertainty of state transitions. Generally, most RL algorithms can be categorized into model-free RL (MFRL) [10] or model-based RL (MBRL) [17] framework. Specifically, model-free methods learn a deterministic policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ or a stochastic one $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ to maximize the expected cumulative reward only through trial and error, but the policy in model-based methods is learned by predicting the dynamics of the environment (reward and/or the transition dynamic) additionally. This paper focuses on the MBRL to learn the optimal bidding strategy.

2.3 RL-Based Auto-Bidding

Recently, some works [1, 4, 26] have been formulating the bidding process as an MDP and solving it by RL [22]. Wu et al. [26] employs a model-free RL algorithm DQN [19] to regulate the optimal bidding parameter sequentially instead of directly producing bids. To alleviate the sparse reward problem during training, Wu et al. [26] designs a deep neural network to predict the total reward in a period. As an improvement, USCB [10] derives a unified optimal bidding formula and uses another model-free RL algorithm DDPG [15] to learn the optimal parameter selection policy in continuous action space. Apart from the single-agent modeling, some researchers tried to solve the auction problem through multi-agent reinforcement learning (MARL) [9] from the advertising platform perspective. Wu et al. [25] proposes a MARL approach to derive cooperative policies for the impression allocation problem. Furthermore, Wen et al. [24] focuses on the equilibrium state of competition and cooperation, where a temperature-regularized credit assignment is leveraged to make a competition and cooperation trade-off among agents. To the best of our knowledge, there are few auto-bidding methods based on MBRL. Cai et al. [4] uses a model-based RL agent to generate the bidding policy by modeling the state transition in an auction process. However, the method in Cai et al. [4] requires estimating the market price of each impression and computing the auction-level value function by dynamic programming, which is computationally intensive. We use the optimal bidding model to construct a coarse-grained environment model to reduce the complexity. Then an uncertainty-aware MBRL framework is proposed for auto-bidding to adjust the core parameters adaptively.

3 PROBLEM FORMULATION

In this paper, we focus on model-based RL methods to alleviate two thorny issues, i.e., (1) the engineering overhead associated with simulator construction and (2) policy transfer from simulated to real environments, that cause the performance limitations of the currently popular model-free methods in handling non-stationary markets in production environments. This paper did not first propose model-based auto-bidding methods. The fine-grained modeling of advertisers' bidding behavior by existing model-based frameworks can produce better bidding strategies. However, the resulting learning complexity makes them difficult to apply to production environments other than artificial datasets, further contributing to the current status quo of model-free methods becoming mainstream. In order to retain the advantages of the model-based method while enhancing its practicability, we re-model the sequential decision problem of real-time automatic bidding at a coarse-grained level under the MBRL context, starting from the *optimal bidding model* [31], while ensuring the superiority of the strategy. In addition, to better capture the highly non-stationary bidding market due to dynamic bidding behavior and independent learning modeling, we explicitly consider the uncertainty of state transitions, which are naturally incorporated into the modeling of coarse-grained MDPs. Below, we will describe the optimal bidding model under the second-price auction mechanism and the corresponding coarse-grained Markov decision process for auto-bidding.

3.1 Optimal Bidding Model

During a period, the ad impression opportunities sequentially arrive, and the bidding agent competes with other agents to win the ad impression. After the auction, the agent with the highest bid can show ads to an audience and enjoy the impression value. Without loss of generality, we consider clicks as the optimization target of advertisers, while the budget is the long-term constraint:

$$\max \sum_{i=1\ldots N} x_i \cdot v_i, s.t. \sum_{i=1\ldots N} x_i \cdot c_i \leq B, \tag{1}$$

where N is the total number of impressions and x_i is a binary indicator of whether the bidding agent wins impression i. v_i, c_i represent the winning value and the cost the advertiser needs to pay, respectively. Budget B is the maximum spendable amount set by the advertiser. Under the second price auction mechanism, Zhang et al. [31] proved the optimal bidding formula is as follows:

$$b_i = F(w, v_i) = w \cdot v_i, \tag{2}$$

where w is a scaling parameter, the optimal parameter w^* can be calculated through the historical data. Based on this idea, He et al. [11] extends the optimal bidding formula to a more general form with multiple constraints and corresponding parameters:

$$b_i = F(\mathbf{w}, v_i) = w_0 \cdot v_i - \sum_{j=1}^{M} w_j (q_{ij}(1 - \mathbb{I}_{CR_j} - k_j \cdot p_{ij}), \tag{3}$$

where M is the number of constraints, p_{ij} and q_{ij} can be any performance indicators, $\mathbf{w} = [w_j]_{j=0}^{M} \in \mathbb{R}^{M+1}$ is the core constrain-related parameter vector, $\mathbb{I}_{CR_j}$ is an indicator function of whether constraint j is cost-related and k_j is the upper bound of constraint j, which the advertiser provides in advance. Likewise, the optimal parameters $\mathbf{w}^*$ can be calculated from the historical data. Nevertheless, in practical applications, $\mathbf{w}^*$ is hard to obtain as the entire impression set can not be collected until the end of the day. Thus, the bidding agent must adopt an approximate real-time strategy to adjust the core parameters $\mathbf{w}$ under the current state.

3.2 Coarse-grained Bidding MDP

Inspired by the optimal bidding model, we formulate the online adjustment of the core parameters $\mathbf{w}$ as a *coarse-grained* Markov decision process under the independent learning scheme based on state-of-the-art model-free methods [10, 26].

In an episode, the bidding agent starts with an initial bid parameter vector $\mathbf{w}_0$ and will sequentially modify it T times (typically 1 hour between decisions) until the end of a day, which means that the episode length is equal to 24. At timestep t, the agent observes the current state $s_t \in \mathcal{S}$ and then takes action $a_t \in \mathcal{A}$, which is used to generate a new parameter vector for calculating the bidding in next hour. After applying the new parameter vector to bid, the agent will transition to a new state s_{t+1} with the probability $p(s_{t+1}|s_t, a_t)$ and receive a reward r_t, which is denoted by the reward function $r(s_t, a_t, s_{t+1})$. We now describe the key components of the proposed coarse-grained MDP as follows:

- $\mathcal{S}$: From the perspective of advertisers, the state should reflect the current agent's advertising status, which includes its *budget consumption*, the *accumulated winning value*, and the current *timestep*.
- $\mathcal{A}$: We directly output the next bid parameter vector and are different from outputting the adjustment rate to the parameters, which is adopted by previous model-free methods. The action space $\mathcal{A} = \times_{j=0}^{M} \mathcal{A}_j$ contains $K = \prod_{j=0}^{M} K_j$ discrete actions, where $|\mathcal{A}_j| = K_j$. Then the update of the parameters at timestep t takes the form $\mathbf{w}_t = a_t, a_t \in \mathcal{A}$.
- $\mathcal{R}$: Since the optimization target of advertisers is to maximize the total winning value $\sum_{i=1}^{N} x_i \cdot v_i$ under long-term $M + 1$ constraints, the reward should both take account of the winning value and the satisfaction of constraints. The format of the reward function is discussed in Section 5 since the design of the reward needs to be oriented towards the specific task in the production environment. Furthermore, we assume that the reward function is known to the bidding agent and does not need to be estimated by model-based learning.
- $\mathcal{P}$: Since the market competition fluctuates with traffic, the winning rate of an advertiser with the same bid will also fluctuate. Unlike previous model-free methods, we explicitly model the non-stationarity of state transitions and denote the transition dynamic $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ as a probability distribution over $\mathcal{S}$. As a model-based RL approach, learning $\mathcal{P}$ is the primary task when modeling the environment, and we also introduce a probabilistic modeling method in Section 4 to learn this probability distribution effectively.
- γ : In online display advertising, the goal is to maximize the total reward regardless of considering the reward decay over time. In other words, bidding agents do not need to weigh short- and long-term benefits so we set discount factor $\gamma = 1$.

The solution of the above coarse-grained MDP we aim to obtain in the paper is just a deterministic policy $\pi^* : \mathcal{S} \rightarrow \mathcal{A}$, which defines the bidding strategy in RTB for online display advertising, and is learned by maximizing the expected cumulative reward.

Unlike existing MBRL methods, the MDP we constructed does not estimate the market price of each impression and compute the auction-level metrics, but instead models the uncertainty of budget consumption and win value, which greatly reduces the learning complexity. This enables the design of model-based methods based on this coarse-grained MDP to bypass the two thorny issues that limit the performance of model-free methods in production environments with non-stationary markets and ensure practicality. Next, we will build a simple and effective model-based automatic bidding framework based on the above coarse-grained MDP.

4 MODEL-BASED AUTO-BIDDING

In this section, we propose a model-based RL framework for auto-bidding, MBAB (Figure 1), which is divided into two stages: learning the non-stationary environment model via a parameterized probabilistic model and the model predictive control based on the dynamic programming of the learned environment model to obtain the optimal bidding strategy under constraints. The explicit uncertainty modeling of the dynamic auction environment makes MBAB avoid burdensome simulated environment construction and is more suitable for production deployment without the thorny sim-to-real issue than model-free methods.

Specifically, according to the coarse-grained MDP, we consider learning the state transition $\hat{\mathcal{P}}_\theta : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ which is parameterized by θ. The complex changes of the bidding market in the production environment and the independent learning scheme adopted by the coarse-grained MDP make the state transition highly uncertain or non-stationary. This prompts us to model the transition function $\hat{\mathcal{P}}_\theta$ as a parameterized probabilistic distribution. Given the dataset $\mathcal{D} = \{s_t^\ell, a_t^\ell, s_{t+1}^\ell\}_{\ell=1}^L$ collected from the RTB, the non-stationary transition function $\hat{\mathcal{P}}_\theta$ then be approximately fitted via two novel loss functions based on supervised learning.

Once a dynamic environment model is learned, a straightforward utilization for the agent is to find an optimal bidding strategy by planning on it. Borrowed from many model-based RL methods [12, 18], we introduce a classical framework called model predictive control (MPC) [3] to select the bidding action at the current timestep greedily. Under the MPC framework, we use a dynamic programming method to obtain bidding planning results on all possible future state sequences based on the learned environment model. Then only the first-step plan in the sequence is chosen to limit the accumulation of errors due to the estimated environment model. Below we elaborate on the learning of the non-stationary environment model and the design of the MPC framework.

4.1 Non-stationary Environment Modeling

In general, the market price is strongly associated with the number of auction competitors. For example, an advertiser with a fixed bid wins more ad impressions when there are fewer competitors, while its win rate decreases in a more competitive market. As a result, the deterministic transition model fails to capture this aleatoric uncertainty [7], which arises from the inherent stochasticities of the auction environment. We adopt a neural network (NN) to output sufficient statistics of a parameterized distribution to model the aleatoric uncertainty in the RTB system.

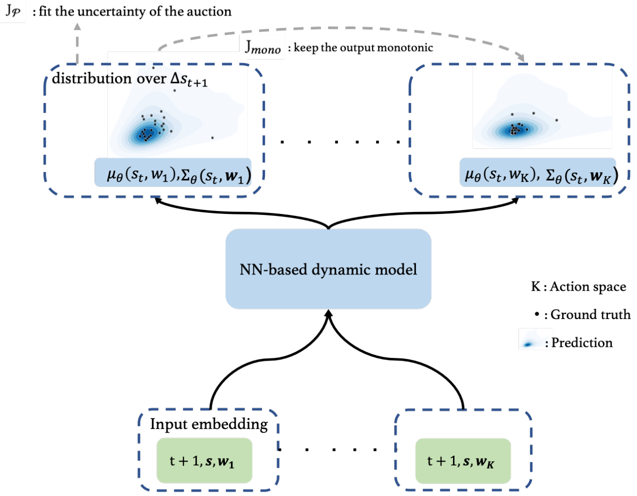

$J_{\mathcal{P}}$: fit the uncertainty of the auction
J_{mono} : keep the output monotonic
distribution over Δs_{t+1}
$\mu_\theta(s_t, w_1), \Sigma_\theta(s_t, \boldsymbol{w}_1)$
$\mu_\theta(s_t, w_K), \Sigma_\theta(s_t, \boldsymbol{w}_K)$
NN-based dynamic model
K : Action space
• : Ground truth
: Prediction
Input embedding
$t + 1, \boldsymbol{s}, \boldsymbol{w}_1$
$t + 1, \boldsymbol{s}, \boldsymbol{w}_K$

Figure 2: Our probabilistic dynamic model is shown as an distribution over Δs_{t+1}. The model's input is an embedding of the current bidding state s and bidding parameter w.

In the RTB system, the state (budget consumption and the total winning value of ad impressions) of a bidding agent is gradually accumulated, which means the NN needs to predict the distribution over the increment of the current state in a specific time interval. The next state is determined by the previous state and the predicted content. The architecture of NN based dynamic state transition model is depicted in Figure 2, whose output captures the RTB environment's aleatoric uncertainty by parameterizing a joint probability distribution function over the budget consumption and winning value in the next time interval. We assume that the output of the NN follows a log-normal distribution and consider the negative log prediction probability as one part of the loss function:

$$J_{\mathcal{P}} \quad = \quad - \sum_{\ell=1}^{|\mathcal{D}|=L} \log \hat{\mathcal{P}}_\theta(\Delta s_{t+1}^\ell \mid s_t^\ell, a_t^\ell = \mathbf{w}_t^\ell), \tag{4}$$

where s_t^ℓ and s_{t+1}^ℓ are two consecutive states. Δs_{t+1}^ℓ refers to the state shift in next time interval and the predicted next state s_{t+1}^ℓ is determined by the current state s_t^ℓ and $\Delta s_{t+1}^\ell \sim \hat{\mathcal{P}}_\theta(\Delta s_{t+1}^\ell | s_t^\ell, a_t^\ell)$, i.e., $s_{t+1}^\ell = s_t^\ell + \Delta s_{t+1}^\ell$. For distinguish, we define $s_{t+1}^\ell \sim \hat{\mathcal{P}}_\theta(s_{t+1}^\ell | s_t^\ell, a_t^\ell) = s_t^\ell + \hat{\mathcal{P}}_\theta(\Delta s_{t+1}^\ell | s_t^\ell, a_t^\ell)$ in the following. We define our predictive model to output a multivariate normal distribution with a mean vector $\mu_{\theta_\mu}(s_t^\ell, a_t^\ell)$ and a diagonal covariance matrix $\Sigma_{\theta_\Sigma}(s_t^\ell, a_t^\ell)$ parameterized by θ_μ and θ_Σ respectively. And then we have $\theta = \{\theta_\mu, \theta_\Sigma\}$. Based on this parameterized distribution, the predicted $\Delta s_{t+1}^\ell \sim \mathcal{N}(\mu_{\theta_\mu}(s_t^\ell, a_t^\ell), \Sigma_{\theta_\Sigma}(s_t^\ell, a_t^\ell))$ follows a Gaussian distribution, and then the loss function 4 becomes

$$\begin{aligned} J_{\mathcal{P}} = \sum_{\ell=1}^{|\mathcal{D}|=L} \left[\mu_{\theta_\mu} \left(s_t^\ell, a_t^\ell \right) - \Delta s_{t+1}^\ell \right]^\top \Sigma_{\theta_\Sigma}^{-1} \left(s_t^\ell, a_t^\ell \right) \cdot \\ \left[\mu_{\theta_\mu} \left(s_t^\ell, a_t^\ell \right) - \Delta s_{t+1}^\ell \right] + \log \operatorname{Det} \Sigma_{\theta_\Sigma} \left(s_t^\ell, a_t^\ell \right). \end{aligned} \tag{5}$$

Besides, considering a specific fact for the market competition: the rise of the bid value usually leads to an increase in the budget consumption and win rate. We also optimize a gradient-based loss function to guarantee the monotonicity of predictions:

$$J_{mono} = \sum_{\ell=1}^{|\mathcal{D}|=L} \max \left(0, - \frac{\partial (\mu_{\theta_\mu}(s_t^\ell, a_t^\ell) + \operatorname{Diag}(\Sigma_{\theta_\Sigma}(s_t^\ell, a_t^\ell)))}{\partial a_t^\ell} \right). \tag{6}$$

Then the final loss function takes the form of $J_{\mathcal{P}} + \beta * J_{mono}$, where β is a weight parameter. Using the prediction model loss of (5) and (6), the learning of the transition model can adopt the supervised learning scheme, which is well-studied in previous works.

4.2 Model Predictive Constrained Bidding

Once the environment model is available, we can use it to plan various bidding strategies by predicting future outcomes with a virtual bid. Then, a set of candidate bidding trajectories are obtained by sequentially predicting the bidding result under different strategies. Within these trajectories, a current optimal policy for the agent is to seek an action sequence to maximize the expected reward of the successor auctions.

$$\max_{a_{t:T}} \mathbb{E}_{s_{t'+1} \sim p(s_{t'+1} | s_{t'}, a_{t'})} \left[\sum_{t'=t}^{T} r\left(s_{t'}, a_{t'}\right) \right], \tag{7}$$

where the state will recursively evolve from one timestep to the next one through the transition model, e.g.:$s_{t'+1} \sim p\left(s_{t'+1} \mid s_{t'}, a_{t'}\right)$. For many MBRL methods [12, 18], a classical framework, model predictive control (MPC) [3], is often applied to plan an optimized sequence of actions in the model. Among the real applications, the agent will select the first action a_t of the action sequence $a_{t:T}$ and take it to interact with the environment at each timestep. In our proposed MBRL framework for auto-bidding, a similar technique is adopted to choose the current optimal bidding parameters.

Random shooting is a basic method for an MPC controller to optimize the action sequence. Generally, the controller randomly samples a number of uniform action sequences in the action space at each time step and applies them in the model via state transitions. After that, the accumulated reward of each trajectory can be computed to evaluate the action sequence. This online planning process is easy to implement while computationally intensive due to the numerous rollouts at each time step. In our work, we consider to use the dynamic programming method to estimate the value function $Q(s_t, a_t)$ in advance, which denote the expected reward-to-go with starting state s_t, taking action a_t.

$$\begin{aligned} Q^{\pi}\left(s_t, a_t\right) &= \mathbb{E}_{s_{t+1} \sim \hat{\mathcal{P}}_{\theta}, a_{t+1} \sim \pi}\left[G_t \mid s = s_t, a = a_t\right] \\ &= \mathbb{E}_{s_{t+1} \sim \hat{\mathcal{P}}_{\theta}, a_{t+1} \sim \pi}\left[r_t + \gamma G_{t+1} \mid s = s_t, a = a_t\right] \\ &= r_t + \gamma \int_{s_{t+1}} \hat{\mathcal{P}}_{\theta}\left(s_{t+1} \mid s_t, a_t\right) V\left(s_{t+1}\right) ds_{t+1}, \end{aligned} \tag{8}$$

where $G_t = \mathbb{E}[\sum_{t=1}^{T} \gamma^{t-1} r_t]$ and $V(s_t)$ is the value function, which represents the expected return with starting state s_t, following the policy π

$$V\left(s_t\right) = \max_{a_t \in \mathcal{A}} \left\{ r_t + \gamma \int_{s_{t+1}} \hat{\mathcal{P}}_{\theta}\left(s_{t+1} \mid s_t, a_t\right) V\left(s_{t+1}\right) ds_{t+1} \right\}. \tag{9}$$

Similarly, with the value function, we have the optimal policy in state s_t as:

$$\pi^*\left(s_t\right) = \underset{a_t \in \mathcal{A}}{\operatorname{argmax}} \left\{ r_t + \gamma \int_{s_{t+1}} \hat{\mathcal{P}}_{\theta}\left(s_{t+1} \mid s_t, a_t\right) V\left(s_{t+1}\right) ds_{t+1} \right\}. \tag{10}$$

To settle the integration over s_{t+1} in (9), we discretize the bidding state and compute the approximation of the optimal value function $V(s_t)$ in the discrete state space $\mathcal{S}^d$:

$$V\left(s_t\right) = \max_{a_t \in \mathcal{A}} \left\{ r_t + \gamma \sum_{d=1}^{|\mathcal{S}^d|} \hat{\mathcal{P}}_{\theta}\left(s_{t+1}^d \mid s_t, a_t\right) V\left(s_{t+1}^d\right) \right\}. \tag{11}$$

The pseudocode of the dynamic programming is described further in Algorithm 1.

Algorithm 1: Offline Training of MBAB.

```
Input  : historical data $\mathcal{D} = \{s_t^\ell, a_t^\ell, s_{t+1}^\ell\}_{\ell=1}^{L}$ collected from
         the online RTB system, episode length $T$, state
         space $\mathcal{S}^d$, action space $\mathcal{A}$;
Output : optimal action-value function $Q^*(s_t, a_t)$;
Learn the dynamic model $\hat{\mathcal{P}}_\theta$ with historical data $\mathcal{D}$;
for each $s \in \mathcal{S}$ do
    Initialize $V(s) = r(s)$;
end for
for $t \leftarrow T-1$ to $0$ do
    for each pair $(s_t, a_t) \in \mathcal{S} \times \mathcal{A}$ do
        Obtain the estimated $s_{t+1}$ from the output
          distribution of $\hat{\mathcal{P}}_\theta\left(s_{t+1} \mid s_t, a_t\right)$;
        Update $V(s_t)$ via (11);
    end for
    Evaluate actions as:
    $Q^*\left(s_t, a_t\right) = r_t + \gamma \sum_{d=1}^{|\mathcal{S}^d|} \hat{\mathcal{P}}_\theta\left(s_{t+1}^d \mid s_t, a_t\right) V\left(s_{t+1}^d\right)$.
end for
```

After updating the value function, we can use it to adjust parameters in the optimal bidding model adaptively. At each time step, the optimal action is state s_t is

$$a_t^* = \underset{a \in \mathcal{A}}{\operatorname{argmax}} Q^*\left(s_t, a\right). \tag{12}$$

The online auto-bidding algorithm is described in Algorithm 2.

Algorithm 2: MPC-like Online Deployment.

```
Input  : action value function $Q^*(s_t, a_t)$;
for $t \leftarrow 0$ to $T-1$ do
    Observe current state $s_t$;
    Select optimal action via (12);
    Update bidding parameter: $\mathbf{w}_t = a_t^*$.
end for
```

5 EXPERIMENTS

5.1 Experimental Setup

Datasets: In this paper, we evaluate the performance of MBAB on two real-world constrained bidding tasks. Our datasets are built from the online bidding log of the Alipay display advertising platform for two specific constrained bidding problems. The first bidding task is to maximize the clicks, and the constraint is the daily budget, whose optimal bidding formula takes the form of Eq. 2. The dataset for the first task comprises about 20 million impressions

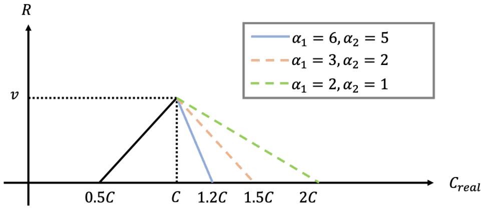


Figure 3: Examples of R function with different parameters

per day with the value estimations (i.e., $pCTR_i$) of each campaign from 12th Aug. to 13th Aug. 2022. The second bidding task is to maximize the conversions under the CPA constraint [33], and the optimal bidding formula takes the form of

$$b_i = \quad w * CPA_{target} * pCTR_i * pCVR_i, \tag{13}$$

where CPA_{target} is the average expected cost per action, and the advertisers are charged per click. The dataset for this cost-constrained task comprises about 12 million impressions per day from 10th Oct. to 11th Oct. 2022. For both datasets, we use the first day of data for training and the next day for evaluation. Each day comprises an episode, and each episode consists of 24-time steps (an hour between two consecutive time steps)

Evaluation Metrics: The two constrained bidding tasks to aim to maximize the total value (clicks or conversions) of winning impressions while satisfying the constraints. As we mentioned above, the optimal parameters w^* can be calculated from the static data from a posterior perspective, then the optimal bidding result R^* [26] can be obtained using the optimal w^*. Similarly, the total value of winning impressions under the current bidding policy is denoted by R. As a result, $R/R*$ is a simple and practical metric to evaluate the difference between the current and optimal policies. Specifically, the R in the budget-constrained task is the sum predicted CTR of winning impressions. While considering the cost satisfaction, the R in the CPA-constrained task is a trade-off between the total value v (the sum predicted $pCVR$) of winning impressions and the constrain of the average cost per action (CPA), which takes the form of $R = p * v$. The weight p is a penalty factor used to measure constraint satisfaction.

$$p = \begin{cases} max(0, \alpha_1 - \alpha_2 \frac{C_{real}}{C}), & C_{real} > C, \\ 1, & C_{real} = C, \\ max(0, \frac{2C_{real}}{C} - 1), & C_{real} < C, \end{cases} \tag{14}$$

where C_{real} and C respectively denote the real average CPA and the target average CPA, α_1 and α_2 are the hyper-parameters to control the penalty strength. Examples of R with different penalty factors are shown in Figure 3, which indicates that R decays to 0 when $C_{real} \le 0.5C$ or $C_{real} \ge \frac{\alpha_1}{\alpha_2} C$.

5.2 Compared Methods

In this paper, we compare our proposed method with three other bidding strategies, which are introduced as follows:

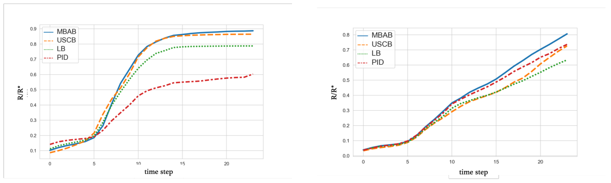
Linear Bidding (LB): a method that bids with optimal bidding model and fixed parameters, which can be set empirically, or the optimal one is calculated from the historical data.

Table 1: R/R* comparison of different methods in two datasets with $[0\%, 5\%]$ w-derivation

R/R*	*Budget constraint*	*CPA constraint*
LB	0.8657	0.7238
PID	0.6029	0.7472
USCB	0.8697	0.7294
MBAB	**0.8835**	**0.8094**

Table 2: R/R* comparison of different methods in two datasets with different levels of deviation

	Budget constraint				CPA constraint			
Deviation	*LB*	*PID*	*USCB*	*MBAB*	*LB*	*PID*	*USCB*	*MBAB*
$[0\%, 5\%]$	0.8657	0.6029	0.8697	**0.8835**	0.7238	0.7472	0.7294	**0.8094**
$[5\%, 30\%]$	0.7258	0.5791	**0.8713**	0.8700	0.6339	0.7367	0.7263	**0.8062**
$[30\%, 50\%]$	0.5181	0.5668	0.7811	**0.8418**	0.3922	0.6791	0.7390	**0.7712**
$[50\%, +\infty]$	0.3690	0.5714	0.4870	**0.6766**	0.2527	0.5372	**0.6608**	0.6360
Average	0.6197	0.5800	0.7523	**0.818**	0.5006	0.6751	0.7139	**0.7557**



(a) Performance under budget constraint with w-deviation $\in (0\%, 5\%)$

(b) Performance under CPA constraint with w-deviation $\in (0\%, 30\%)$

Figure 4: Average performance on two tasks.

PID: a method that combines the optimal bidding formula with current information about the satisfaction of constraints. For example, the PID bidding policy adjusts parameters in the cost-constrained task by computing the difference between the target CPA and the actual average CPA so far. In the budget-constrained task, the PID controller gives a smooth bidding policy by constraining the budget consumption ratio to equal the time consumption ratio in an episode. When the left budget ratio is lower than the time left ratio, the agent increases the bid, otherwise decreases the bid.

Unified Solution to Constrained Bidding (USCB): the method proposed in [10] is the state-of-the-art algorithm for bidding parameters control, a unified solution to constrained bidding using the model-free RL. Likewise, the model-free agent takes advantage of the optimal bidding formula and dynamically adjusts the parameters: $\mathbf{w}_{t+1} = (1 + a_t)\mathbf{w}_t$.

Model-based RL for Auto-Biding (MBAB): The method proposed in this paper, which also utilizes the optimal bidding formula and learns a model-based RL agent to plan a new parameter via Eq. 12 at each time step.

5.3 Evaluation Results

In this section, we conduct experiments to compare the performance of LB, PID, USCB, and MBAB. As mentioned above, the optimal parameters w^* can be calculated from the historical data and used as a prior for bidding strategies. In our experiments, we use this prior to initializing the w_0 at the beginning of each episode. Due to the non-stationary auction environment and the dynamic of auction competitors, the previous optimal w^*_{pre} may deviate from the optimal w^* of the current episode. Firstly, the performance comparison of different methods on two datasets with small w-derivation ([0%-5%]), defined as $|w_0 - w^*|/w^*$, are reported in Table 1. In the comparison on R/R^*, we find that (i) our proposed method performs the best on both budget-constrained tasks and CPA-constrained tasks, verifying the effectiveness of MBAB; (ii) Compared to the other three bidding strategies, MBAB performs particularly well on the CPA-constrained task; (iii) PID gives the worst performance on the budget-constrained task since its smooth delivery strategy is not suitable for uneven distribution of traffic quality.

Furthermore, to investigate the performance of each method with different levels of w-deviation, we divide campaigns in the dataset into four groups according to the w-deviation and evaluate different bidding strategies for each group. Table 2 provides a detailed comparison of different methods on two constrained bidding tasks with different levels of w-deviation. All methods' performance gradually decreases as the w-deviation increases, especially the LB. To explain this phenomenon, we should understand that w-deviation is an indicator of how volatile the market is, which is affected by many factors, such as budget, auction competitors, and market price distribution. Take the budget-constrained task as an example, an advertiser with a big budget tends to bid aggressively. However, the same strategy may lead to budget over when the budget decreases significantly. Due to the unawareness of the market volatility, we can see that the LB strategy using a fixed w_0 gives the worst performance. On the contrary, the PID strategy performs stably since its bidding strategy has a low correlation with w_0. Among all w-deviation settings, RL-based bidding strategies perform better than other bidding strategies, and MBAB achieves the best average performance on two tasks. It shows that MBAB is more robust when the auction environment is unstable. USCB, a model-free RL method, has stable performance when the w-deviation is not large (w-deviation $\in [0\%, 30\%]$), but R/R* drops significantly (lower than 0.5) on budget-constrained task when w-deviation exceeds 50%. A possible reason for this phenomenon is that the learning of USCB is influenced by previously identified episodes, lacking the capture of the uncertainty of the auction environment. To better understand the performances of all bidding strategies, we also present the average R/R* increase over time steps on two constrained bidding tasks in Figure 4.

By calculating and normalizing the average bidder per ad impression at each timestep, we evaluate the fierce market competition on two constrained bidding tasks in Figure 5(a). For budget-constrained tasks, competition in the morning will be more fierce, which means that the optimization space of bidding strategies is also larger. Compared to the LB bidding strategy, Figure 5(b) and Figure 5(c) show the average R/R* improvement of two RL-based methods over timesteps. We observe that (i) For both MBAB and USCB, the main improvement on the budget-constrained task is in the morning (about 5:00 am to 11:00 am); (ii) For the CPA-constrained task, MBAB performs better than USCB at each time step and enjoys positive improvements throughout the day, which shows that the performance of MBAB is more stable than the model-free method.

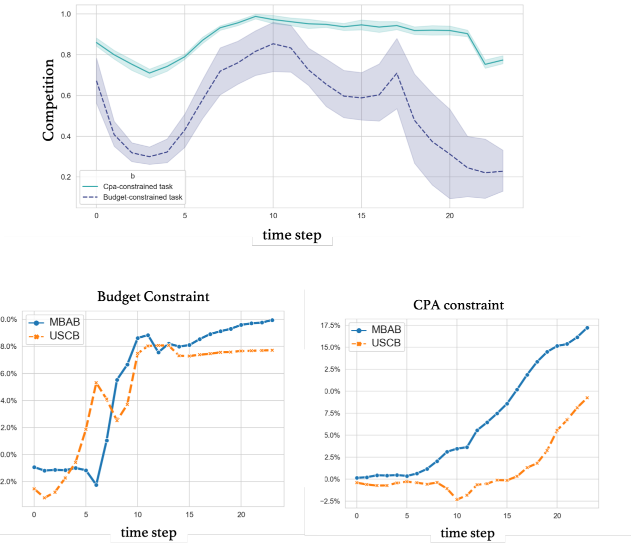


Figure 5: (Top): Normalized market competition. (Bottom): Average R/R* improvement over time steps on two constrained bidding tasks.

Table 3 shows the performance of MBAB under different scales of datasets. We observe that (i) the improvement is not significant when the training data exceeds 30% of datasets; (ii) The increment of training data does not always benefit the performance of MBAB.

Table 3: Performance under various scales of datasets.

*R/R**	*10%*	*30%*	*50%*	*60%*	*80%*
Budget constraint	0.5066	0.8514	0.8641	**0.8835**	0.8782
CPA constraint	0.6659	0.7877	0.7582	0.7758	**0.8076**

6 CLOSING REMARKS

This paper proposes a MBRL algorithm MBAB for constrained auto-bidding in display advertising. The bidding strategy takes advantage of the optimal bidding formula and sequentially adjusts its core parameters. According to the MDP formulation, a parameterized distribution is learned to capture the uncertainty of the auction environment. After that, an action-value function is used to represent how good it is for an agent to choose a particular bidding parameter in a state, which is equal to the expected total reward from now on. The optimal action-value function is then updated using dynamic programming, and the optimal policy is derived by taking the action that maximizes the expected total reward at each time step. Experimental results on two real-world constrained bidding tasks demonstrated the superiority of MBAB over several baselines in the industry and the state-of-the-art method.

ACKNOWLEDGMENTS

This work was supported in part by Postdoctoral Science Foundation of China (2022M723039) and a grant from Shenzhen Institute of Artificial Intelligence and Robotics for Society.

REFERENCES

[1] Kareem Amin, Michael Kearns, Peter B. Key, and Anton Schwaighofer. 2012. Budget Optimization for Sponsored Search: Censored Learning in MDPs. ArXiv abs/1210.4847 (2012).

[2] Achim Bachem and Walter Kern. 1992. Linear programming duality. In Linear Programming Duality. Springer, 89–111.

[3] Marko Bacic. 2003. Model predictive control.

[4] 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. Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (2017).

[5] Ye Chen, Pavel Berkhin, Bo Anderson, and Nikhil R. Devanur. 2011. Real-time bidding algorithms for performance-based display ad allocation. In KDD.

[6] Kuang chih Lee, Burkay Orten, Ali Dasdan, and Wentong Li. 2012. Estimating conversion rate in display advertising from past erformance data. In KDD.

[7] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. 2018. Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models. In NeurIPS.

[8] 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.

[9] Sven Gronauer and Klaus Diepold. 2022. Multi-agent deep reinforcement learning: a survey. Artificial Intelligence Review 55, 2 (2022), 895–943.

[10] 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 & Data Mining. 2993–3001.

[11] 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. Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (2021).

[12] Lukas Hewing, Kim Peter Wabersich, Marcel Menner, and Melanie Nicole Zeilinger. 2020. Learning-Based Model Predictive Control: Toward Safe Learning in Control. Annual Review of Control, Robotics, and Autonomous Systems (2020).

[13] IAB. 2022. Internet Advertising Revenue Report: Full Year 2021 Webinar. Technical Report. https://www.iab.com/wp-content/uploads/2022/04/IAB_Internet_Advertising_Revenue_Report_Full_Year_2021.pdf

[14] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. 2020. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. arXiv preprint arXiv:2005.01643 (2020).

[15] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Manfred Otto Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2016. Continuous control with deep reinforcement learning. CoRR abs/1509.02971 (2016).

[16] Chi-Chun Lin, Kun-Ta Chuang, Wush Chi-Hsuan Wu, and Ming-Syan Chen. 2016. Combining powers of two predictors in optimizing real-time bidding strategy under constrained budget. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. 2143–2148.

[17] Fan Luo, Tian Xu, Hang Lai, Xiong-Hui Chen, Weinan Zhang, and Yang Yu. 2022. A Survey on Model-based Reinforcement Learning. ArXiv abs/2206.09328 (2022).

[18] Rowan McAllister and Carl Edward Rasmussen. 2016. Improving PILCO with Bayesian Neural Network Dynamics Models.

[19] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charlie Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. Nature 518 (2015), 529–533.

[20] Rafael Figueiredo Prudencio, Marcos ROA Maximo, and Esther Luna Colombini. 2022. A Survey on Offline Reinforcement Learning: Taxonomy, Review, and Open Problems. arXiv preprint arXiv:2203.01387 (2022).

[21] Matthew Richardson, Ewa Dominowska, and Robert J. Ragno. 2007. Predicting clicks: estimating the click-through rate for new ads. In WWW '07.

[22] Richard S. Sutton and Andrew G. Barto. 2005. Reinforcement Learning: An Introduction. IEEE Transactions on Neural Networks 16 (2005), 285–286.

[23] Ming Tan. 1993. Multi-agent reinforcement learning: Independent vs. cooperative agents. In Proceedings of the tenth international conference on machine learning. 330–337.

[24] Chao Wen, Miao Xu, Zhilin Zhang, Zhenzhe Zheng, Yuhui Wang, Xiangyu Liu, Yu Rong, Dong Xie, Xiaoyang Tan, Chuan Yu, Jian Xu, Fan Wu, Guihai Chen, and Xiaoqiang Zhu. 2022. A Cooperative-Competitive Multi-Agent Framework for Auto-bidding in Online Advertising. Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (2022).

[25] Di Wu, Cheng Chen, Xun Yang, Xiujun Chen, Qing Tan, Jian Xu, and Kun Gai. 2018. A Multi-Agent Reinforcement Learning Method for Impression Allocation in Online Display Advertising. ArXiv abs/1809.03152 (2018).

[26] Di Wu, Xiujun Chen, Xun Yang, Hao Wang, Qing Tan, Xiaoxun Zhang, and Kun Gai. 2018. Budget Constrained Bidding by Model-free Reinforcement Learning in Display Advertising. Proceedings of the 27th ACM International Conference on Information and Knowledge Management (2018).

[27] 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. 1443–1451.

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

[29] Shuai Yuan, Jun Wang, and Xiaoxue Zhao. 2013. Real-time bidding for online advertising: measurement and analysis. ArXiv abs/1306.6542 (2013).

[30] 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. 1077–1086.

[31] Weinan Zhang, Shuai Yuan, and Jun Wang. 2014. Optimal real-time bidding for display advertising. Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (2014).

[32] Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. 2020. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In 2020 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, 737–744.

[33] Han Zhu, Junqi Jin, Chang Tan, Fei Pan, Yifan Zeng, Han Li, and Kun Gai. 2017. Optimized Cost per Click in Taobao Display Advertising. Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2017).