big data and cognitive computing

Article

Imagine and Imitate: Cost-Effective Bidding under Partially Observable Price Landscapes

Xiaotong Luo [1], Yongjian Chen [1], Shengda Zhuo [2], Jie Lu [1], Ziyang Chen [1], Lichun Li [1], Jingyan Tian [1], Xiaotong Ye [1] and Yin Tang [1,*]

[1] School of Management, Jinan University, Guangzhou 510632, China
[2] College of Cyber Security, Jinan University, Guangzhou 510632, China
* Correspondence: ytang@jnu.edu.cn

Abstract: Real-time bidding has become a major means for online advertisement exchange. The goal of a real-time bidding strategy is to maximize the benefits for stakeholders, e.g., click-through rates or conversion rates. However, in practise, the optimal bidding strategy for real-time bidding is constrained by at least three aspects: cost-effectiveness, the dynamic nature of market prices, and the issue of missing bidding values. To address these challenges, we propose Imagine and Imitate Bidding (IIBidder), which includes Strategy Imitation and Imagination modules, to generate cost-effective bidding strategies under partially observable price landscapes. Experimental results on the iPinYou and YOYI datasets demonstrate that IIBidder reduces investment costs, optimizes bidding strategies, and improves future market price predictions.

Keywords: real-time bidding; imitation learning; bidding landscape prediction; partially observable Markov decision processes

check for updates

Citation: Luo, X.; Chen, Y.; Zhuo, S.; Lu, J.; Chen, Z.; Li, L.; Tian, J.; Ye, X.; Tang, Y. Imagine and Imitate: Cost-Effective Bidding under Partially Observable Price Landscapes. *Big Data Cogn. Comput.* **2024**, *8*, 46. https://doi.org/10.3390/bdcc8050046

Academic Editors: Dimitrios Kardaras and George Stalidis

Received: 2 April 2024
Revised: 24 April 2024
Accepted: 26 April 2024
Published: 28 April 2024

1. Introduction

Machine learning has initiated an advertising revolution from traditional "buying ad position" to "buying targeted customers" [1]. This has helped advertisers achieve precise marketing through online advertisement and made real-time bidding (RTB) the most popular way to trade online advertisements for stakeholders [2].

An advertiser's goal is to develop an optimal bidding strategy that can maximize revenue (e.g., clicks or conversions) for the advertisements placed under certain constraints such as fixed budgets and unknown market price distributions. It is particularly important to develop a bidding strategy that can balance the budget and the ad transaction price and can make decisions with censored or incomplete input. Appendix A provides domain knowledge details. Despite recent advances, we are still faced with challenges: (1) Models have to make decisions under strict distributional assumptions [3–8], which are usually not the real cases; (2) In real-world scenarios, there exist considerable amounts of censored and incomplete data, transmitted from demand-side platforms (DSPs) to which the existing models are vulnerable [9].

Most algorithms tend to make the consumption of advertising budgets as smooth as possible, with ignorance of the fluctuation of market transaction prices. The price distributions, or landscapes, are usually modeled as certain parametric distributions [3,4,10]. However, the specific assumptions, such as the Gaussian distributed assumptions, may not be the ground truth in such a changing environment. This makes the methods susceptible when bidding high-cost and high-value advertisements. Meanwhile, existing algorithms yield larger variance under partial observations than expected, bringing in higher costs, as our experiments show. On the other hand, in an RTB environment with multiple players, advertisers who have failed in bidding will not receive the transaction price just made, meaning that the information on market price is not fully accessible for the next decision process. Moreover, the model occasionally has to deal with the data with a certain percentage of missing values. To gamble with market adversaries, an agent has to make decisions with frequently incomplete information. Although some deep learning methods, such as [11], seem to work well in fitting a small number of samples containing market prices to induce their distribution, these independent and identically distributed (i.i.d.) assumption-based methods may not be well adaptable to dynamically changing RTB environments.

Based on the above analysis, this paper proposes IIBidder, based on the generative adversarial imitation learning (GAIL) [12] framework. The idea is to win the bidding by making the bidding decisions to approximate the implicit distribution of the market prices with incomplete market states. To be specific, we obtain the market price landscape by fitting the expert samples with a deep neural network. We employ proximal policy optimization (PPO) [13], the latest deep reinforcement learning algorithm, to make the strategy more stable. Meanwhile, we introduce an "Imagination" module to predict the missing value and to help "Imitation" be more robust.

Figure 1 illustrates that the proposed method has a notable improvement in terms of cost-constrained metrics. Meanwhile IIBidder reduces the variance relative to DRL methods and allows for more stable policy generation. The proposed method can help advertisers win the most valuable bidding opportunities at lower costs in the RTB business.

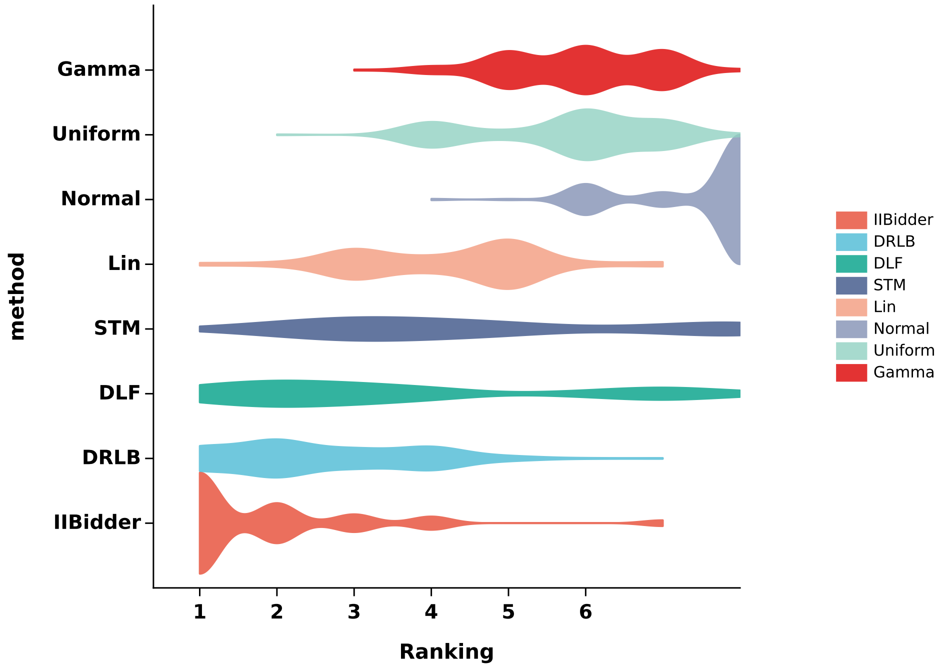


Figure 1. An overlook on the frequency of performance ranking of Imagine and Imitate Bidding (IIBidder) compared with others in a series of experimental combinations concerning budget and mask. The vertical axis lists different methods, while the horizontal axis represents ranking frequency. IIBidder ranks 1st in most of the combinations.

Overall, the method proposed in this paper contributes the following.

- Our work jointly optimizes bidding strategy and landscape prediction, leveraging market price distribution data to enhance bidding strategies.
- IIBidder introduces the *Imagination* module, enabling the model to infer hidden information and adapt to expert bidding samples, resulting in more accurate real-world bidding strategies, particularly in the presence of incomplete data.
- Experimental results demonstrate that our method outperforms current RTB methods in terms of click-through rates and winning rates considering cost constraints.

The rest of this paper is organized as follows. Section 2 describes the related work from bidding landscape prediction, bidding strategy design, and an imitation learning context. Section 3 elaborates the framework. In Section 5, experiments verify the feasibility and validity of the proposed method through baseline models. Section 6 concludes the study.

2. Related Work

2.1. Bidding Landscape Forecasting

Bid landscape prediction focuses on the problem of how to fit the distribution of transaction prices. By tracking the distribution of market prices, advertisers can predict the probability of winning auctions that may result in clicks or conversions, which helps optimize bidding strategies [14]. At the same time. Directly predicting price distribution is challenging. Existing methods model market prices as random variables and then heuristically learn single-peaked distributions from historical market price data. These distributions include normal distributions [3], log-normal distributions [4], heavy-tail distributions [5], gamma distributions [6], Gompertz distributions [7], and tanh-shaped distributions [8].

Subsequent researchers have proposed more comprehensive solutions by employing deep neural network [10] or tree models [15]. Wang et al. developed a model to better capture the multiple peaks in the market price distribution while considering the potential impact of impression-level features on the price distributions [16]. A potentially neglected problem is that only the winners of the auctions have access to the actual market prices, while the information of other competitors is truncated [9]. This right-censored issue may lead to significant biases in landscape prediction.

2.2. Bidding Strategy Design

Many suggest seeking the optimal bidding function that directly maximizes key performance indicators (KPI) of ad campaigns, e.g., total click count or revenue, based on the static distribution of input data and market competition models [17]. For instance, adhering to the assumption of a linear bidding strategy, the bidding price should be proportional to the expected click-through rate for that opportunity. Methods include segmented functions for click-through rates [18], combined predictions of click-through rates and winning bids [19], and assumptions based on gamma distributions [6].

Another way to construct a bidding function is an expected revenue function of advertisers as a nonlinear function of click rate and click cost $eCPI = CTR \times CPC$ [20]. High conversion rates are also one of the goals to seek. The lag time between clicks and conversions became an effective factor in predicting the value of ads and in assisting advertisers in pricing [10,21–23].

However, this static bidding optimization strategy may still not perform effectively in a highly dynamic RTB market due to significant discrepancies between the real data distribution during model training and the assumed data distribution [24]. A more promising branch of research is to model the bidding process as a Markov decision process (MDP), which is then solved by using deep reinforcement learning [25]. This was firstly done by Cai, who transformed the bidding strategy problem into a Markov decision process, where the reward function was quantified as the difference between the click-through rate (or conversion rate) and the bid winning price [17]. While the huge exploration space and the stochasticity of state transition in reinforcement learning still remain unsolved [26]. Efforts have been made to address these challenges. DASQN tackled synchronization and random state shifts [26]. The Soft Actor–Critic method reformulates a bidding strategy problem into hyper-parameter adjustments, avoiding the exploration space hurdles [27]. Wang proposed a course-guided Bayesian reinforcement learning (CBRL) framework adaptable to dynamic and reward-sparse RTB environments [28]. Lu addressed the oversight of sequence information and sparse state space in bidding strategy algorithms by introducing sequence information extraction and clustering-based state aggregation [29] based on the

A3C model [30]. Using multiple intelligent agent reinforcement learning with different reward functions also improved the win rate and ROI in various scenarios [31,32].

2.3. Imitation Learning and GAIL Optimization

RTB problems used to be sparsely rewarded in nature. In many cases, it requires human-designed reward functions, which may be costly. Sometimes it is often impractical to set up the reward functions manually [33] in online advertisement bidding. Imitation learning (IL), as a method to learn strategies from expert examples, is believed to be an effective solution concerning the above problems [34]. Along this line of research, behavior cloning (BC) method [35] learns example data alone; however, sometimes it ends up generating cascading errors, and has poor generalization and robustness when expert samples are insufficient [36]. On the other hand, the inverse reinforcement learning method (IRL) [37] uses expert instances to fit the unknown reward function automatically but is computationally expensive. Analogous to the idea of GAN [11], generative adversary imitation learning (GAIL) used a generator to generate action sequences and a discriminator to distinguish between the action sequences and the expert's examples [12]. GAIL automatically fits the reward function from expert samples as IRL does, and generates pseudo samples as in GAN. By combining these techniques, it avoids the errors generated by human-designed rewards and makes the model more robust. Appendix B provides more background on reinforcement learning and generative adversary imitation learning in this regard.

On the other hand, to encourage the agent to explore novel states, the intrinsic curiosity module (ICM) was proposed [38], inspired by the curiosity-driven learning mechanism in human brain research. The ICM infers the action taken by an agent based on two consecutive states. The inverse prediction error, which is the difference between the predicted and actual actions, is then used to train the Encoder. Recent advancements also include some prediction techniques in model-based agent [39]. They try to predict the next step Q values in hidden space.

3. Problem Statement

3.1. Problem Definition

The objective function is to maximize the advertiser's revenue and is subjected to conditions. The advertiser's cost spent on bidding ads should not exceed the budget, but should be larger than the bottom price. The cost of bidding should be close to but not exceeding the total budget. The symbols used in the paper is defined in Table 1. The problem can be formulated as:

$$\max \sum_{i=1}^{N} x_i v_i, \tag{1}$$

$$\begin{aligned} \text{s.t.} \qquad\qquad \sum_{i=1}^{N} x_i b_i &\leq B, \\ B - \sum_{i=1}^{N} x_i b_i &\leq \varepsilon, \\ b_i &\geq RP_i, \\ \forall i \in \{1, 2, 3, \ldots, N\}&. \end{aligned}$$

In practice, the above model carries high complexity since N is usually an unknown variable. Moreover, the advertiser's bidding strategy (bidding price b) is determined by taking into account the feature of bidding request z, the value v of the corresponding bidding request (bidding landscape), and other factors. Thus, the advertiser's bid price is mapped to the value evaluation of the ad $b = f(v, z)$, which means that the bid price could be affected by the characteristics of the ad bid and the value of the ad. We turn the problem into a conditional extreme value problem. Due to the limit of space, we put the intermediate derivation in Appendix C.

Table 1. Symbol definitions.

Symbol	Description
N	the total number of bid requests
x	whether to win the bidding request (1 if success, 0 otherwise)
v	value evaluated for bid request
b	bid price for bid request
RP	reserved price for bid request
B	total budget of an advertiser
ε	a positive number infinitely close to 0
z	feature vector denoting bid request
$b(z)$	bid price function that wins
$w(b)$	probability function of winning
$v(z)$	estimated value of advertised bid request
$p(z)$	prior distribution of the feature vector for bid request ad

The bidding process is based on the generalized second price mechanism, which means that advertisers have no access to the final knock-down price if the bidding fails. It is unlikely to accurately assess the value of the advertisement under these circumstances.

$$\lambda w(b(v(z))) = [v(z) - \lambda b(v(z))]\frac{\partial w(b(v(z)))}{\partial b(v(z))} \tag{2}$$

Inferred from Equation (2), $p(x)$ has little influence on the model. The influence of $b(v(z))$ depends on $w(b(v(z)))$ and λ, namely the bid landscape prediction problem and the bid strategy optimization problem, respectively.

3.2. Bidding Strategy as Markov Decision Process

A restricted Markov decision process $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P})$ can be transformed from the above probabilistic model.

State space $\mathcal{S}$: a state $s_t \in \mathcal{S}$ includes the time t(when an ad display opportunity arrives), the proportion of advertisers' current budget left $(LB_t = 1 - \frac{b_t}{B})$, the consumption rate of budget $BR_t = \frac{LB_t - LB_{t-1}}{LB_{t-1}}$, the characteristics of the ad itself z_t, the winning rate w_t, and the ratio of the number of winning opportunities v_t, to total number of participating bids from time $t-1$ to t. s_t is defined as below:

$$s_t = (LR_t, BR_t, z_t, w_t, v_t) \tag{3}$$

Action space $\mathcal{A}$: $a_t \in \mathcal{A}$ can be obtained by solving the regulation factor $\lambda(i)$. But under the auction mechanism of generalized second price, the bid price is the predicted value of the ad display opportunity product with the conditioning factor, (e.g., $b = v * \frac{1}{\lambda}$ [5]). Also, to simplify the scale of the problem, the action space is a discrete space, and the specific values are the adjustment parameters of the base price. The bid action a_t is considered successful when the offered price is greater than or equal to the predicted bid price b_t. Conversely, if the bid price is less than the predicted bid price, the bid is deemed unsuccessful, indicating an ineffective bidding action.

Reward function $\mathcal{R}$: To underscore the advantages of joint optimization, the rewards for this Markov chain process consist of three components, amalgamating the strengths of expert strategy, imitation strategy, and exploration mechanisms. The experiments section will further elucidate how the integration of these components contributes to the overall efficacy of the bidding strategy.

Hence, a Markov decision process-based bidding strategy model is formulated as follows.

$$\begin{aligned} \max \sum_{s \in S, a \ \in A(s)} & \pi(s) R(s,a) \\ \text{s.t.} \sum_{s \in S, a \ \in A(s)} & \pi(s) C(s,a) \leq B, \\ \sum_{a \in A(s)} & \pi(s) = 1, \forall s \in S \end{aligned} \tag{4}$$

where the objective function is to maximize the expected return. $\pi(s)$ is the bidding strategy function, denoted as the probability of executing a bid action a in state s. B is the total budget for an ad placement cycle, while $C(s,a)$ is the cost function required for bid action a in state s.

3.3. Partially Observable Scenario

A random masking process is applied to a specified percentage of the market state. This masking operation involves obscuring or hiding certain components of the state, making them unavailable for observation. The masked elements include features related to expert samples, bidding landscapes, or other relevant market dynamics. We simulate a more realistic bidding environment by masking a certain percentage of the market state, including expert samples. The motivation is to mimic the inherent uncertainty and incomplete information that advertisers often encounter in real-world bidding scenarios. The masking process randomly masks a certain percent of the state, while the Imagination module tries to recover the state by extrapolating the missing part of the state sequence with consecutive states. We apply a random masking operation to both the training and testing datasets. We construct a masking matrix $\mathbf{M}_{m \times n} = \{x_{ij} | x_{ij} \in \{0,1\}\}$ randomly with a uniform distribution:

$$x_{ij} = \begin{cases} 1 & \text{if } U(i,j) > 1 - m, \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $U(i,j)$ is a random variable uniformly distributed over the range [0, 1]. Define the masked state as $s^{\mathbf{M}} \in \mathbb{R}^{m \times n}$, which is masked using $\mathbf{M}$ from original state s element-wise.

$$s^{\mathbf{M}} = \text{Mask}(s,m) = (LR, BR, z \odot M, w \odot M, v \odot M) \tag{6}$$

4. The IIBidder Model

As shown in Figure 2, the IIBidder is composed of four parts, including *IIBidder Agent*, *Imagination module*, *Expert Sample Transitions*, and *Discriminator*.

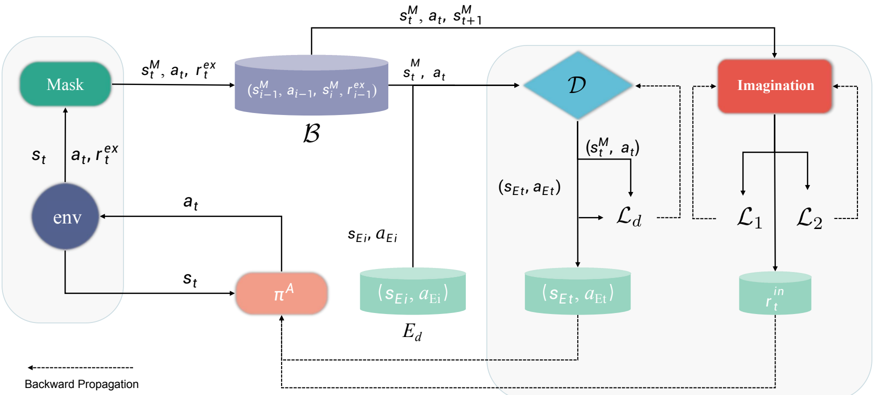


Figure 2. The Imagine and Imitate Bidding (IIBidder) framework.

4.1. IIBidder Agent

IIBidder agent executes a reinforcement learning process involving the proximal policy optimization (PPO) algorithm and the Imagination module. The reward is composed of both extrinsic rewards from the environment and intrinsic rewards generated by the Imagination module.

The extrinsic reward r^{ex} comes from the environment and is part of the transition data stored in the replay buffer. In the PPO update, the extrinsic reward is used in calculating the advantage, which measures the advantage of taking a specific action in a given state compared to the expected value.

The intrinsic reward r^{in} is generated by the Imagination module based on the prediction error of the next state and the action. This intrinsic reward promotes exploration by encouraging the agent to visit states that are not yet well understood. r^{ex}, together with r^{in}, forms the total reward used in the PPO update.

$$r(s,a) = -log(1 - (r^{ex} + r^{in})) \tag{7}$$

The generative adversarial imitation learning (GAIL) part involves training the Discriminator to distinguish between expert sample transitions and policy-generated transitions. The actor is trained to maximize the negative log probability assigned by the discriminator to the policy-generated transitions. The GAIL update includes both the expert sample transitions and policy-generated transitions in the loss calculation.

4.2. The Imagination Module

Our finding is that, although missing values are occasionally found in a single state, agents still have the chance to "guess" the missing data by observing consecutive states, which may contain enough information.

The Imagination module not only encourages an agent to expand the search space and acquire successful bidding samples that even expert strategies have not encountered to alleviate the sparse reward problem in a high-dimensional state. On top of that, it also predicts the missing value of the price landscape by learning the expert sample's distribution in consecutive states. The method also differentiates itself from the model-based Dreamer [39] and ICM [38] by predicting the missing values in a model-free environment. As shown in Figure 3, the *Imagination* module is composed of three neural networks, a forward prediction network, an inverse extrapolation network, and a state encoder.

The *Imagination* module takes (s_t, s_{t+1}, a_t) as input, where s_t denotes values taken from the state space of the Markov decision process model for real-time bidding advertising, and a_t denotes the bid price of the imitation policy. We define a neural network function $\varphi(s_t)$, to process the state s_t. The bid decision $\hat{a}_t$ is inferred by the inverse inference network $g(\varphi(s_t), \varphi(s_{t+1}))$. By sequentially comparing consecutive states, the module learns the most relevant information and infers the missing part in $\hat{s}_{t+1}$. Then, the parameters of the inverse inference network and the state encoder are, respectively, updated according to the following loss function.

$$\mathcal{L}_1 = ||a_t - \hat{a}_t||_2^2 \tag{8}$$

The forward prediction network $f(\varphi(s_t), a_t, \hat{s}_t)$ generates a next possible state $\hat{\varphi}(s_{t+1})$, and a internal reward r_t^{in} is obtained by comparing the difference between $\varphi(\hat{s}_{t+1})$ and $\varphi(s_{t+1})$, while updating the parameters of the forward prediction network from L^2-norm loss function as Formula (9).

$$\mathcal{L}_2 = ||\varphi(s_{t+1}) - \varphi(\hat{s}_{t+1})||_2^2 \tag{9}$$

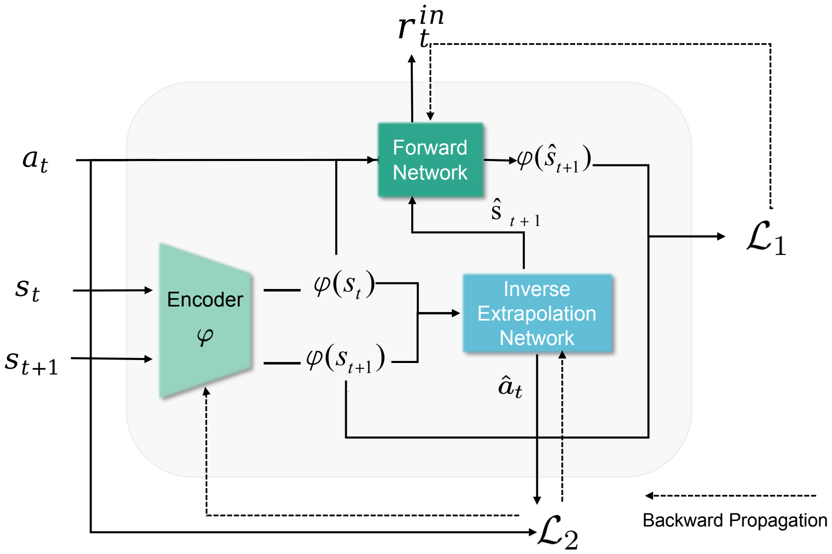


Figure 3. The Imagination module.

4.3. Expert Sample Transitions

For the bidding strategy, we can define the expert samples as samples that win the ad display opportunity transaction. Suppose the data set in the expert sample database is $E_d = \{\hat{\tau}_1, \hat{\tau}_2, \hat{\tau}_3, \ldots, \hat{\tau}_i, \ldots\}$, where $\hat{\tau}_i$ denotes a trajectory of expert experience for episode i. Let $\hat{\tau}_i = \{(s_{ij}, a_{ij})\}_{j=1}^{T}$ for each step j, where s_{ij} denotes the feature vector of bid requests z, and a_{ij} equals the bid price b when it wins the deal. Also, for the samples with failed bids, we increase their bid prices by a certain degree so that they can also be used as expert samples. With a pool of expert examples, the distribution of market prices can be fitted using deep neural networks, as probability density function $p(v|z)$. We use a deep neural network with parameters θ to approximate the conditional probability density function $p(v|z)$ during the learning process:

$$p(v|z) \approx \pi^E(z;\theta) \tag{10}$$

4.4. Discriminator

The main job of a discriminator $\mathcal{D}$ is to judge whether a bid decision is derived from an expert policy $\hat{\tau}_t$ or from an imitation policy τ_t, and to output a score indicating to what extent that the input is close to expert policy. In other words, $\mathcal{D}$ acts as the classical reward function in a reinforcement learning paradigm so that we no longer need to define the reward function manually. In general, discriminator $\mathcal{D}$ expects decisions derived from the expert policy to have higher scores than ones derived from the imitation policy, namely, $\mathcal{D}(\hat{b}_i) \geq \mathcal{D}(b_i)$. The discriminator is updated based on the following loss:

$$\mathcal{L}_d = \mathcal{D}(\hat{b}_i) - \mathcal{D}(b_i) \tag{11}$$

4.5. The IIBidder Algorithm

We integrate the GAIL with the Imagination module $\mathcal{I}$ to fit an unknown distribution of incomplete expert examples. The system consists of an expert example database E_d, an expert policy π^E, an imitation policy π^A, an Imagination module $\mathcal{I}_\eta$, and a discriminator $\mathcal{D}$. In the RTB process, training the imitation policy π^A corresponds to the problem of solving the bidding strategy, while training the expert policy π^E corresponds to the problem of fitting the market price distribution. Further, the discriminator $\mathcal{D}$ is used to update the two strategies simultaneously and make the imitation policy π^A as close as possible to the expert policy π^E.

As shown in Algorithm 1, it comprises majorly of nested loops, which are set to N epochs for the outer loop and to k steps for the inner loop. At the n-th epoch, initialize the RTB environment in a linear time and execute the inner loop, including the updating of the policy network, imagination module and discriminator network. In the inner loop, masking the state, obtaining action and reward, storing tuple in transition, and calculating score, can all be done in linear time. Therefore, the time complexity for training is $\mathcal{O}(N \cdot k)$.

Algorithm 1: The IIBidder algorithm.

```
Input Expert policy $\pi^E$, Expert samples $E_d \sim \pi^E$, Imitation policy $\pi_\theta^A$ with
  random weight $\theta$, experience buffer $\mathcal{B}$, Discriminator $\mathcal{D}$ with random weight $\alpha$,
  Imagination module $\mathcal{I}_\eta$, mask percentage $m$
Output Bidding Strategy
for each epoch do
    Initialize RTB environment $s_0 = env.init()$ ;
    for each step do
        $s_t^M = Mask(s_t, m)$;
        Get action and reward by $a_t = \pi^A(s_t^M)$;
        $s_t^M, s_{t+1}^M, r^{ex} = env.step(a_t)$;
        Store $(s_t^M, s_{t+1}^M, a_t, r^{ex})$ in $\mathcal{B}$;
        $r^{in} = \mathcal{I}_\eta(s_t^M, s_{t+1}^M, a_t)$;
        Calculate score $= \sum_t^T (r^{ex} + r^{in})$;
        Set $r(s^M, a) = -\log(1 - score)$;
    Update $\alpha_{i+1} \leftarrow \alpha_i$ with Equation (11);
    Update $\theta_{i+1} \leftarrow \theta_i$ based on $r(s^M, a)$;
    Update $\eta_{i+1} \leftarrow \eta_i$ with Equations (8) and (9);
```

5. Experiments

The experiments aim to answer the following questions:

- Under budget constraint, what is the performance of IIBidder in comparison to other classic algorithms? In terms of modeling dynamic price environments, what advantages does this algorithm have?
- In an incomplete data scenario, is IIBidder effective in addressing the challenge of missing values within the competitive bidding environment?
- Does the incorporation of the Expert sample module, Discriminator module, and Imagination module effectively stimulate favorable behavior in the agent? Under budget constraints and incomplete data landscapes, can these modules operate effectively?

5.1. Experimental Settings

5.1.1. Implementation Details

For the experimental setting, the hardware consists of a CPU (Intel Xeon Gold 6140) and a GPU (Tesla V100 with 32 GB of memory) with Ubuntu 18.04 running on it. The model is implemented using PyTorch, a well-known deep learning framework.

5.1.2. Datasets

Due to the availability and fair comparisons, we choose the popular iPinYou and YOYI datasets for the experiments.

iPinYou encompasses over 15 million impressions and user feedback data from nine distinct campaigns by various advertisers over a ten-day period in 2013. Each bid request in the log contains comprehensive information, including user details (e.g., segmentation), advertiser specifics (e.g., creative format and size), and publisher information. The data set has 26 original features that roughly fall into three categories: ad bids, ad exposure, and user feedback (clicks and conversion rate). Details are shown in Table A1. Meanwhile, the data set includes nine advertisers of various industries in Table A2. Appendix D provides the details of the datasets.

YOYI recorded multi-device advertising during 8 days in January 2016, which contains 5.64 million impressions. The first 7 days of data are used for training, while the rest are used for testing. To facilitate fair comparison, all records are used as a single campaign since the specific campaign information is unavailable.

5.1.3. Data Preprocessing

There are 10 days' records in the data set, which are partitioned into a training set and a test set by a ratio of 7:3. That is, the records from the first 7 days of the data set are used as the training set, and the ones from the last 3 days as the test set. Except for the mobile e-commerce advertiser with ID 2997, the click-through rate of real-time bidding ads is below 0.01%, mainly because advertiser's ads are usually placed on the pages of mobile APPs, resulting in high probabilities of mistaken touch by users. Meanwhile, since the test set is used to verify the effectiveness of the algorithm, all samples in the test set are samples of successful bids, which means the win rate is 100%.

Moreover, judging by a significant difference between different types of advertisers, the results indicate that advertisers in different industries will face different bidding environments. Therefore, in this research, different types of advertisers will be trained and tested separately.

5.1.4. Evaluation Metrics

Traditionally, the RTB performances were assessed using metrics like click-through rate (CTR) and winning rate (WR). However, we opt for a more sophisticated evaluation by introducing cost-effective rate (CER) and win rate cost (WRC). CER and WRC provide a more comprehensive evaluation of securing valuable clicks and successful bidding actions associated with the costs, particularly under certain constraints of budgets.

CER emphasizes the effectiveness of the bidding strategy in generating clicks relative to the cost incurred. A higher CER suggests an effective strategy in maximizing clicks relative to the overall cost. The average CER is computed as follows:

$$\overline{CER} = \frac{1}{n}\sum_{i=1}^{n}\frac{click^2(i)}{cost(i)}. \tag{12}$$

WRC is a metric that gauges the effectiveness of the bidding strategy by evaluating the win rate relative to the cost incurred. A higher WRC signifies a more cost-effective strategy. Average WRC is calculated as below, where $cost(i)$ denotes the cost per click spent by the advertiser i.

$$\overline{WRC} = \frac{1}{n}\sum_{i=1}^{n}\frac{winrate(i)}{cost(i)}, \tag{13}$$

where $n = 9$ represents the number of advertisers involved.

In the context of a simulated environment, characterized by dynamic variations in budget allocations and information masking levels, CER and WRC emerge as more suitable metrics considering the cost-effectiveness. Their capacity to handle the intricate interplay between budget constraints and data masking conditions provides a more nuanced and insightful evaluation of algorithmic performance. This adaptability ensures that the metrics accurately capture the true cost-effectiveness of bidding strategies in scenarios where missing values are a prevalent factor.

CER and WRC not only enhance the depth and relevance of our evaluation but also ensure a more precise reflection of algorithmic efficiency within the challenging landscape of our simulated environment, marked by the intricate interplay of budget considerations and varying levels of information masking.

5.1.5. Budget, Masking and Hyper-Parameter Setting

To optimize advertising revenue within specified budget constraints and considering the partial observability of the market state, we incorporate a masking mechanism.

The mask is designed to simulate real-world scenarios where advertisers have incomplete information about the pricing landscape. The masking scale, denoted by m, represents the proportion of state values that are concealed. We consider six masking scales: $m \in$ {0.0, 0.1, 0.2, 0.3, 0.4, 0.5}, where $m = 0.0$ means no masking.

To obtain the maximum advertising revenue under certain budget constraints, we adopt the budget allocation formula commonly used for reinforcement learning bidding strategies [17].

$$B_{\text{test, camp}} = c_0 \cdot B_{\text{train,total}} \cdot \frac{N_{\text{test,camp}}}{N_{\text{train,total}}}, \tag{14}$$

where $B_{(\text{test,camp})}$ denotes the budget allocated in the testing process for each advertiser. $B_{(\text{train,total})}$ is the total cost of the training set for all advertisers. $N_{(\text{test,camp})}$ is the total number of records for each advertiser, and $N_{(\text{train,total})}$ is the total number of records in the training set for all advertisers. The specific values of all of the above four variables can be derived from the totaling row of Tables A3 and A4. We set a scaling factor $c_0 \in \{1/2, 1/4, 1/8, 1/16, 1/32\}$, where a larger value corresponds to a higher allocated budget.

5.1.6. Baseline Models

The baselines are categorized into two types: parametric models and reinforcement learning models. We introduce linear, deep reinforcement learning and random models for comparison. For random policy, the study covers several common distributions, such as *Uniform* , *Gamma*, and *Normal*.

- *Uniform*: This strategy assumes that the bid price is uniformly distributed between the lowest bid $b_{\min}$ and the highest bid $b_{\max}$, namely
$$b_i \sim Uniform(b_{\min}, b_{\max}). \tag{15}$$
- *Gamma* [6]: Similarly, the strategy is made upon the Gamma distribution, characterized by a shape parameter k and a scale parameter θ. The probability density function (PDF) of the Gamma distribution is given by:
$$f(x; k, \theta) = \frac{x^{k-1} e^{-\frac{x}{\theta}}}{\theta^k \Gamma(k)}, \tag{16}$$
where $x \geq 0$, $\Gamma(k)$ is the Gamma function, and $k, \theta > 0$ are the shape and scale parameters, respectively.
- *Normal*: The logarithm assumes that market price follows the Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$.
- Lin [19]: The bid b_i for the ith ad display opportunity depends on the historical average click-through rate CTR_{avg}, the estimated click-through rate $pCTR_i$ for that opportunity, and the base price b_0.
$$b_i = b_0 \cdot \frac{pCTR_i}{CTR_{\text{avg}}}. \tag{17}$$
- GMM [16]: a Gaussian mixture model to describe and discriminate the multimodal distribution of market price by utilizing the impression-level features.
- DLF [40]: A method combining deep recurrent neural networks and survival analysis to forecast the distribution of market prices.
- DRLB [41]: A DQN-based reinforcement learning bidding strategy. It converts the problem of solving the bidding strategy into determining a factor γ according to the optimal bidding theory [41], where the bid price is the ratio of the estimated click-through rate $pCTR_i$ to γ.
$$Q(s_t, a_t) = Q(s_t, \frac{pCTR_i}{\gamma}). \tag{18}$$

5.2. Experimental Results and Discussions

5.2.1. Performance under Budget Constraints (Q1)

Cost-Effectiveness: As shown in Table 2,the IIBidder outperforms other classical algorithms on both datasets and across both metric dimensions. This indicates that, under budget constraints, IIBidder better manages the cost-effectiveness of ad bidding, achieving superior performance. The heatmap in Figure 4 gives the detail ranking w.r.t. each budget and mask combination for metric CER and WRC, respectively.

Table 2. Performance comparison of different methods on various budgets. The best one in each column is marked as **bold**.

	Budget	**1/32**		**1/16**		**1/8**		**1/4**		**1/2**	
	Metric	**CER**	**WRC**	**CER**	**WRC**	**CER**	**WRC**	**CER**	**WRC**	**CER**	**WRC**
Lin	ipinyou	0.669	5.313	1.166	5.062	1.734	4.978	2.105	5.052	2.177	5.135
	yoyi	1.420	5.105	5.826	4.817	15.900	4.743	19.999	4.740	19.999	4.740
Normal	ipinyou	0.055	2.550	0.128	2.537	0.271	2.573	0.486	2.502	0.724	2.585
	yoyi	0.180	1.429	0.231	1.433	0.425	1.432	1.705	1.433	4.751	1.430
Uniform	ipinyou	0.089	3.350	0.193	3.895	0.456	4.115	0.915	3.928	1.401	4.090
	yoyi	0.521	2.288	0.540	2.278	1.129	2.287	3.053	2.285	11.553	2.282
Gamma	ipinyou	0.112	4.162	0.208	3.975	0.471	4.143	0.859	3.887	1.379	3.997
	yoyi	0.462	2.283	0.497	2.285	1.010	2.283	2.862	2.287	11.560	2.280
GMM	ipinyou	0.173	1.718	0.405	1.725	1.141	1.677	2.334	1.687	3.718	1.687
	yoyi	0.118	1.427	0.202	1.529	0.171	1.353	0.166	14.3	0.166	1.592
DLF	ipinyou	0.555	2.816	1.015	2.828	2.306	27.725	5.211	2.615	6.272	2.611
	yoyi	0.208	1.482	0.244	1.466	0.247	1.481	0.212	1.475	0.243	1.472
DRLB	ipinyou	8.962	29.510	9.822	28.615	8.235	15.083	7.088	19.427	4.476	13.360
	yoyi	**79.398**	**73.275**	46.065	23.460	58.283	42.432	44.447	**31.852**	52.897	**43.047**
IIBidder	ipinyou	**26.249**	**52.373**	**24.055**	**45.688**	**14.176**	**54.110**	**16.567**	**22.567**	**22.132**	**71.337**
	yoyi	50.913	23.827	**71.430**	**60.452**	**75.622**	**56.535**	**55.325**	28.522	**53.773**	30.577

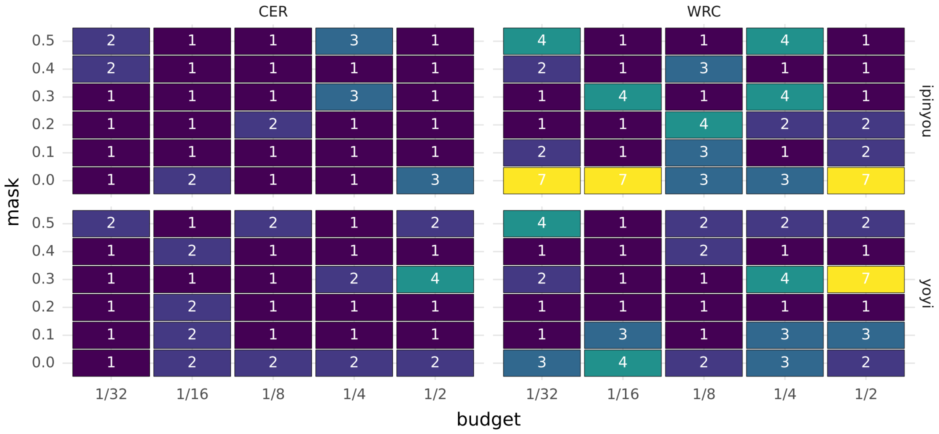


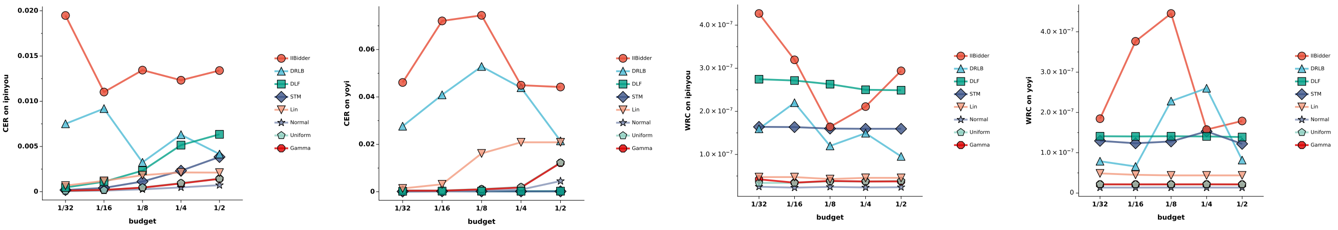
Figure 4. The rankings of our method for certain mask and budget combinations for each dataset and metric. Each cell represents an experiment of a specific mask-budget combination, with a ranking number inside. The smaller the ranking, the better. Our method ranks the first in most of experiments.

Regarding the CER metric, IIBidder outperforms all models on both IPinYou and YOYI datasets. This underscores IIBidder's ability to generate cost-effective bidding strategies, yielding higher click-through rates at lower costs. DRLB ranks second with a click-through rate of 0.03 per unit cost, while Normal models perform the poorest, with only 0.0008 click-through rate per unit cost. Results for Gamma, Uniform, Lin, GMM, and DLF are also below 0.007 click-through rate per unit cost.

Concerning the WRC metric, IIBidder achieves superior performance under most budget constraints, securing a lower unit win rate per cost. IIBidder performs well on the IPinYou dataset with an average WRC of around 4×10^{-7}. In the majority of budget conditions, it outperforms other algorithms, exhibiting high unit cost win rate values.

On the YOYI dataset, it significantly surpasses Lin, Gamma, Normal, Uniform, GMM, and DLF, and in many cases, outperforms DRLB.

The results under different budget factor settings are shown in Figure 5.



(**a**) $\overline{CER}$ on iPinYou (**b**) $\overline{CER}$ on YOYI (**c**) $\overline{WRC}$ on iPinYou (**d**) $\overline{WRC}$ on YOYI

Figure 5. The overall performance of Imagine and Imitate Bidding (IIBidder) under different budget constraints on datasets and metrics.

Dynamic modeling capabilities: Reinforcement learning models, including IIBidder and DRLB, demonstrate superior dynamic modeling capabilities compared to parametric models and sequence models on both IPinYou and YOYI datasets. The results shed light on the two modeling types in dynamic temporal scenarios: (1) Parametric models often struggle to focus on specific localities, resulting in suboptimal performance, while reinforcement learning models leverage neural networks to effectively handle fluctuating markets. (2) Sequence models, such as DLF, outperform parametric models on two metrics but fall short compared to reinforcement learning models. (3) With the support of the imitation and imagination mechanism, IIBidder stands out for its exceptional dynamic modeling capabilities, showcasing the substantial disparity in dynamic modeling prowess between parametric and reinforcement learning models.

5.2.2. Analysis on Incomplete Data Scenario (Q2)

Increasing Trend: The overall increasing trend in metrics with different levels of masking is attributed to the construction of the missing environment. Masking click information and bidding prices in the dataset cover a portion of click situations, implying that algorithms treat unknown click situations as non-clicks. Additionally, some bidding prices are masked, prompting each algorithm to adopt more aggressive bidding strategies. As a result, with increasing mask levels, the performance of CER and WRC tends to rise.

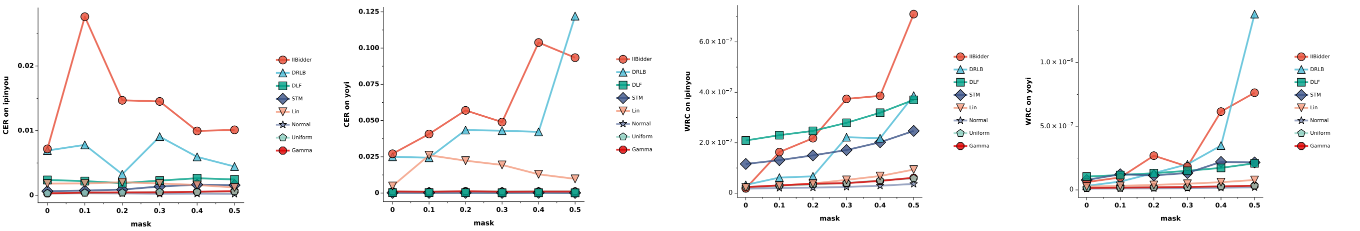
Handling Complete Landscape Capabilities: As Table 3 and the Figure 6 show, IIBidder outperforms DRLB, DLF, and parametric models in dealing with missing values. Parametric models show no significant difference in performance under different masking levels and struggle to handle situations with missing landscape information.

Table 3. Comparative performance on different masking scales. The best one in each column is marked as **bold**.

	Masked	**0%**		**10%**		**20%**		**30%**		**40%**		**50%**	
	Metric	**CER**	**WRC**	**CER**	**WRC**	**CER**	**WRC**	**CER**	**WRC**	**CER**	**WRC**	**CER**	**WRC**
Lin	ipinyou	1.875	2.260	1.812	2.992	1.707	3.994	1.531	5.178	1.381	6.854	1.117	9.370
	yoyi	13.491	2.624	16.653	3.276	14.258	4.046	12.656	4.980	10.646	6.182	8.068	7.866
Normal	ipinyou	0.360	1.724	0.380	2.054	0.350	2.316	0.330	2.502	0.300	3.014	0.277	3.686
	yoyi	1.642	1.009	1.453	1.124	1.285	1.272	1.269	1.446	1.644	1.698	1.459	2.040
Uniform	ipinyou	0.475	2.454	0.571	2.798	0.621	3.496	0.668	3.884	0.676	5.106	0.654	5.516
	yoyi	3.490	1.566	3.468	1.762	3.387	2.004	3.232	2.316	2.835	2.734	3.744	3.322

Table 3. *Cont.*

	Masked	0%		10%		20%		30%		40%		50%	
	Metric	**CER**	**WRC**	**CER**	**WRC**	**CER**	**WRC**	**CER**	**WRC**	**CER**	**WRC**	**CER**	**WRC**
Gamma	ipinyou	0.451	2.152	0.588	3.100	0.626	3.664	0.634	4.008	0.660	5.004	0.678	6.268
	yoyi	3.450	1.564	3.265	1.758	3.280	2.004	3.040	2.318	2.782	2.740	3.853	3.318
GMM	ipinyou	1.355	1.162	1.474	1.324	1.530	1.498	1.481	1.728	1.696	2.012	1.793	2.468
	yoyi	0.159	0.778	0.168	1.222	0.177	1.226	0.153	1.362	0.180	2.202	0.151	2.008
DLF	ipinyou	3.332	20.2	3.240	2.244	2.915	2.466	2.982	2.766	2.802	3.182	3.176	3.692
	yoyi	0.275	1.054	0.279	1.172	0.254	1.312	0.209	1.296	0.211	1.736	0.155	2.082
DRLB	ipinyou	6.504	3.284	6.074	6.164	11.526	16.534	8.944	24.560	8.244	33.692	5.007	42.960
	yoyi	27.604	3.300	25.999	5.546	40.039	13.634	38.998	23.592	52.123	42.806	**152.546**	**168.000**
IIBidder	ipinyou	**12.234**	**4.302**	**25.757**	**20.780**	**32.898**	**46.128**	**18.115**	**45.060**	**22.070**	**87.720**	**12.739**	**91.300**
	yoyi	**28.259**	**6.194**	**41.269**	**14.328**	**58.737**	**26.820**	**56.365**	**41.412**	**102.873**	**85.820**	80.970	65.320



(**a**) $\overline{CER}$ on iPinYou (**b**) $\overline{CER}$ on YOYI (**c**) $\overline{WRC}$ on iPinYou (**d**) $\overline{WRC}$ on YOYI

Figure 6. The overall performance of Imagine and Imitate Bidding (IIBidder) for different masking scales on different datasets and metrics.

IIBidder also demonstrates superior performance over DRLB, DLF, and GMM. As the masking level increases, its advantage becomes more evident. When the mask is set to 0, the CER of parametric models remains below 0.02, while reinforcement learning models can achieve around 0.03. However, when the mask is set to 0.4, parametric models still stay below 0.02, DRLB, GMM, and DLF below 0.06, while IIBidder exceeds 0.1. This indicates its proficiency in handling missing data. The WRC metric yields a similar conclusion.

The incorporation of expert and imitation strategies, along with an imagination mechanism, contributes to the success of IIBidder. It can reference successful cases from the expert pool during the learning process, enhancing its probability of successful bidding. The Imagination module allows IIBidder to "imagine" future states based on previous data, thereby alleviating the problem of missing data.

5.2.3. Ablation Studies on Different Module (Q3)

To assess the efficacy of the Expert sample module, Discriminator module, and Imagination module, the ablation study is to determine whether or not the incorporation of the aforementioned modules successfully induces positive behavior in the agent.

We conducted PPO, AC-GAIL (using the Actor–Critic instead of PPO), PPO-GAIL (without the *Imagination* module), and IIBidder (GAIL-PPO with *Imagination* module), across diverse budget constraints and masking scales.

Ablation Studies under Budget Constraints: With the increase in the budget, IIBidder maintains a consistently superior position in both CER and WRC metrics, showcasing a stable performance trend independent of budget variations. In terms of CER, IIBidder ranks highest. Comparatively, IIBidder demonstrates a remarkable 43% increase in CER and a substantial 68% rise in WRC when compared to PPO-GAIL. In contrast, the increase in CER is marginal at 9%, accompanied by a modest 0.6% decrease in WRC when transitioning from PPO to PPO-GAIL. Furthermore, in comparison to AC-GAIL, PPO-GAIL showcases a significant 48% improvement in CER and an impressive 77% boost in WRC. This emphasizes that the AC architecture struggles to effectively learn competitive bidding strategies under budget constraints. In summary, the architectural elements of IIBidder, such as the expert pool, imagination module, and discriminator module, collectively contribute to more accurate bidding strategies under both budget constraints and scenarios involving missing landscape data.

Ablation Studies on Incomplete Data Scenarios: With an increasing degree of masking, the advantages of the imagination module become increasingly evident. In the realm of CER, IIBidder takes the lead, followed by IIBidder, PPO-GAIL, PPO, and AC-GAIL. In comparison to PPO-GAIL, IIBidder exhibits a substantial 43% increase in CER. Similarly, PPO-GAIL outperforms PPO with a modest 9% increase in CER. This underscores the crucial role played by the imagination module in enhancing IIBidder's performance in coping with incomplete price landscape scenarios. The expert pool also contributes to the decision-making process of the agent. The imagination module's mechanism to predict unknown distributions based on existing data distribution allows IIBidder to resist interference from missing values, resulting in a more robust bidding strategy. Contrasted with AC-GAIL, PPO-GAIL shows a remarkable 62% improvement in CER, even surpassing PPO by 21%. Hence, in a straightforward AC architecture, reinforcement learning models face challenges in handling scenarios involving missing prices. Similar conclusions apply to the analysis of the WRC metric.

Moreover, our approach also alleviates the cold start problem by leveraging other advertisers' expert bidding as prior domain knowledge to initialize the bidding strategy. Advertisers can utilize industry insights, market trends, and competitor analysis to make informed decisions in the absence of historical bidding data. By incorporating domain expertise into the bidding process, advertisers can establish a solid foundation for their bidding strategy and adapt it over time as more data becomes available. We found that an advertiser can be provided some reliable expert samples from other advertisers' expert transition in the cold start.

6. Conclusions

Aiming at the bidding strategy optimization for RTB, this study proposes a framework for the joint optimization of bidding strategy and bidding landscape prediction, capable of handling incomplete price landscapes with cost-effectiveness. The experimental results on the popular data sets show that the proposed method achieves impressive results over existing approaches. This study has proved that inference on market price distribution has a positive effect on developing bidding strategies under the market of incomplete information. Further, in this kind of dynamically changing environment, it is unrealistic for one to gain the maximum advertising revenue in the long run by artificially setting up a reward function. Therefore, the Imitation and Imagination mechanism we proposed in the paper is universal and effective in pursuing long-term revenue.

In RTB, the process of solving the bidding strategy is highly time-dependent, and the neural networks used are fully connected neural networks. As such, recurrent neural networks (LSTM, GRU, etc.) might be considered in extracting the temporal features in the bidding process. Meanwhile, for advertisers without bid samples, IIBidder cannot build an expert sample pool, so pre-training methods can be considered to solve the cold start problem of bidding strategies.

Author Contributions: Conceptualization, X.L. and Y.T.; methodology, Y.C.; software, X.L. and Y.C.; formal analysis, S.Z., investigation, J.L.; investigation, Z.C.; resources, Y.T.; validation, L.L.; data curation, J.T.; writing—original draft preparation, X.L. and Y.C.; writing—review and editing, Y.T. and S.Z.; visualization, X.Y.; supervision, project administration and funding acquisition, Y.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (No.62272198), Guangdong Provincial Science and Technology Plan Project (No.2021B1111600001), and Outstanding Innovative Talents Cultivation Funded Programs for Doctoral Students of Jinan University (No.2023CXB022).

Data Availability Statement: https://github.com/JNU-Tangyin/imagineRTB.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Appendix A. Real-Time Bidding for Online Advertisement

In an online advertisement RTB eco-system, advertisers and publishers interact with each other via some intermediaries, including supply-side platforms (SSP), demand-side platforms (DSP), and ad exchange (ADX), where ADX acts as an exchanging hub between SSP and DSP. This paper focuses on DSPs (the advertisers). Real-time bidding (RTB) [2] incorporates big data technology to analyze the user's cache information, and then to explore the user's potential interest for advertising. The goal of RTB is to show the most appropriate ad only to the user of interest. A real-time bidding advertising system consists of the following roles, the interaction between which is also shown in Figure A1.

- Advertiser: An individual consumer or organization that needs to display an advertisement who has a certain budget and bids actively in the bidding process to get the opportunity to display their ads.
- Publisher: A provider with online advertising resources, usually the owner of web pages, search engines, and mobile apps. Publishers earn profits by selling their ad positions.
- Supplier Side Platform (SSP): SSP represents the interests of publishers. SSP integrates media advertising resources for centralized management. It provides publishers with functions such as bottom price setting and automatic ad placement. Since SSPs connect to a large number of partners, they need to formulate a reasonable allocation strategy for each partner to obtain the same amount of advertisers.
- Demand Side Platform (DSP): DSP represents the interests of advertisers, who register on the DSP platform and initiate the requests for ad placement. DSP recommends the most valuable ad positions for advertisers by various means such as big data analysis and user behavior analysis. At the same time, DSP provides advertisers with functions such as user response prediction, budget management and control, and automatic real-time bidding.
- Ad Exchange (ADX): ADX is the hub of the RTB system, which connects many DSPs and SSPs and handles all ad bidding requests in the system. Each bidding process is conducted under the auspices of ADX, making sure that the whole process is fair and open. Meanwhile, ADX sells ad display opportunities to the bidder who has given the highest price according to the generalized second price (GSP) mechanism [42].
- Data Management Platform (DMP): After collecting information such as users' cookies, click records, purchase records, and search contents, DMP tries to clean, transform, and refine the collected information. It adopts machine learning methods to mine user preferences and recommend accurate target customers to the advertisers.

Online exchange requires recommending ads to target users automatically and instantly, so it necessitates close cooperation from all parts of the RTB ecosystem. Figure A1 illustrates a typical bidding process.

1. When a user opens an app or web page developed by a third party (publisher), an ad request will be generated and sent to an SSP with the user's information. The request contains the user's cookie information and the ad context information (e.g., web page URL, ad position, ad position length and width, ad position reserve price, etc.).

2. Once the SSP receives the ad display request, it forwards the request to the ADX with an attachment.
3. ADX broadcasts the bid request, including the user's cookie information and contextual information of the ad, to all advertisers on the DSP platform.
4. After receiving the bid request, an advertiser makes a bid, probably with a bidding robot. The ADX platform will compare these bid prices and determine a final winner. Meanwhile, the process needs to be completed within a specified time (usually 100 ms). When the time is out, the bidding opportunity is considered abandoned.
5. The winning advertiser only needs to pay the second-highest price. If all advertisers' bid prices are lower than the reserved price defined by the publisher, the bidding process for that ad display is considered abandoned.
6. ADX notifies the winner and automatically deducts the cost from the winner's budget. Then, the ADX platform sends back the winner's ad content to the corresponding SSP.
7. The SSP shows the ad to the user and records the user's feedback, such as clicks and conversions.

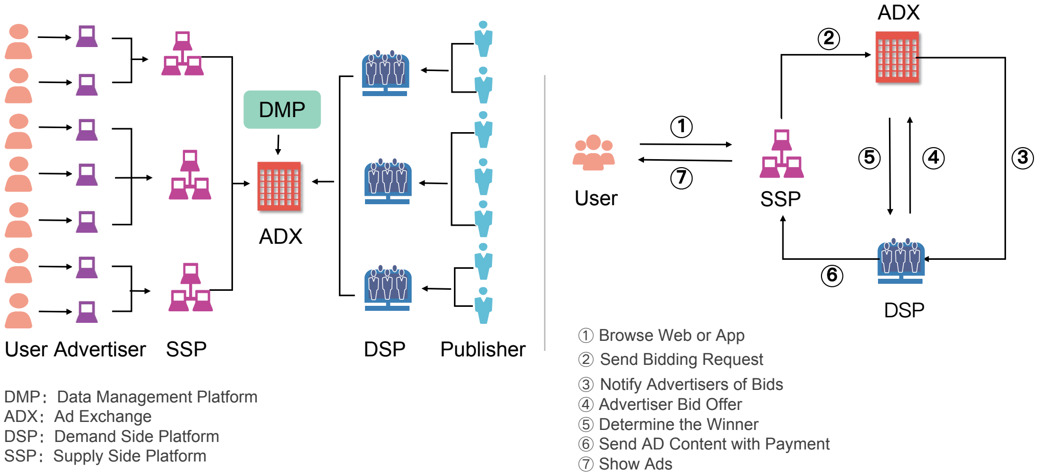


Figure A1. The real-time bidding (RTB) ecosystem and its process.

Appendix B. From Reinforcement Learning to Generative Adversarial Imitation Learning

Reinforcement learning has long been an active area of research [25], aiming at automatic decision making in many domains. A reinforcement learning architecture usually consists of two parts, an agent and an environment. It models states and decision sequences (trajectory) as Markov decision processes (MDP), where the agent continuously interacts with the environment by receiving a state s_t and a reward r_t, and by resending an action a_t back to the environment. The environment, on the other hand, affected by the action received, returns a reward r_{t+1}, together with the next state s_{t+1} if available. The purpose of the agent is to obtain the maximum cumulative reward, with a discount factor γ recursively scaling down the future reward. The interaction can be defined as a state value function:

$$V(s) = \underbrace{R_t}_{\text{instant reward}} + \underbrace{\gamma \sum_{s' \in S} P(s' \mid s) V(s')}_{\text{discounted sum of future rewards}} \tag{A1}$$

Reinforcement learning (RL) has advanced to deep reinforcement learning (DRL) with the emergence of the deep neural network in recent years. It can tackle a variety of domain problems, such as electronic games [43], mechanical control [44], recommendation systems [45], and financial investment [46]. The major idea of DRL is to replace the tabular representation of the agent with a deep neural network. With the strength of approximating significantly high dimensional functions, the deep neural network overcame the problem of high dimensional feature space and equipped the agents with the power to deal with more complicated problems such as video games and robotic control.

There are three main types of deep reinforcement learning agents, namely the value-based methods (e.g., Deep Q-network [47], or DQN for short), policy-based methods (e.g., Proximal Policy Optimization, or PPO for short [13]), Actor–Critic methods, or combination. Value-based methods are less likely to fall into local optimum, having the advantages of relatively high sampling efficiency and small variance of value function estimation. However, it cannot handle continuous action space problems. On the other hand, policy-based methods enjoy simple strategy parameterization, fast convergence, and suitability of continuous or high-dimensional action spaces. Last but not least, Actor–Critic methods combine the advantages of both methods and bring in disadvantages of both as well.

The convergence of reinforcement learning depends heavily on the design of the reward function. Unfortunately, real-world problems are often sparsely rewarded in nature, which usually require human-designed reward functions. Yet it is often impractical to set up the reward functions manually [33]. Therefore, Imitation Learning (IL), as a method to learn strategies from expert examples, is believed to be an effective solution concerning the above problems [34]. It can be summarized as follows:

$$\min_{\pi} V(\pi^{E}) - V(\pi) \tag{A2}$$

where function V is cumulative return function, π^E is expert strategy fit by expert samples, and π is the parameterized imitation strategy. In other words, the key of imitation learning is how to recover an expert strategy π^E.

Along this line of research, the behavior cloning (BC) method is first proposed [35] to replicate the expert strategies to speed up the learning process. However, Ross revealed that the BC method using example data alone may generate cascading errors, and had poor generalization and robustness when expert samples were insufficient [48]. He proposed Dagger (a Dataset Aggregation algorithm) to generate data through continuous interaction with the environment and to constantly update the strategy, which could reduce the number of unvisited states and cascading errors by using data augmentation and strategy iteration [36]. The inverse reinforcement learning method (IRL) uses expert instances to fit the unknown reward function automatically [37]. However, the IRL method consumes quite a lot of computational resources in its iterative process. Wang proposed the GAIL algorithm [12] based on GAN and IRL, which used a generator to generate action sequences and a discriminator to distinguish between the action sequences and the expert's examples.

The generative adversarial network (GAN) [11] borrows the idea of a zero-sum game from the game theory. The vanilla version of GAN consists of two neural networks, a generator and a discriminator. The generator continuously creates data for the discriminator, while the discriminator distinguishes whether the data are 'genuine' or not. Both parties play with each other and obtain results from each other so that they can continuously upgrade themselves. The objective function of GAN is as follows:

$$\min_{G}\max_{D} V(D,G) = \mathbb{E}_{x\sim p(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))], \tag{A3}$$

where both generator $\mathcal{G}$ and discriminator $\mathcal{D}$ are parameterized neural networks. The first part of the formula seeks to maximize the discriminative ability of $\mathcal{D}$, while the second part seeks to minimize the difference between the data distribution $p(z)$ generated by $\mathcal{G}$ and the real data distribution $p(x)$. Analogous to the idea of GAN, GAIL [12] automatically fits the reward function from expert samples as IRL does, and generates pseudo samples as in GAN. By combining these techniques, GAIL can avoid the errors generated by human-designed rewards and make the model more robust. We summarize the objective function of the GAIL method as:

$$\min_{\pi\in\Pi}\max_{S\times A}\mathbb{E}_{(s,a)\sim\hat{d}^{\pi}}[\log(D(s,a))]+\mathbb{E}_{(s,a)\sim d^{\pi}}[1-\log(D(s,a))], \tag{A4}$$

where d^{π} is a sample obtained from the interaction of the imitation strategy with the environment, while $\hat{d}^{\pi}$ is a sample obtained from the interaction with the environment according to the expert strategy, and discriminator $\mathcal{D}$ judges whether the action a is generated by imitation strategy π or the expert strategy π^{E}. The imitation strategy π is a deep reinforcement learning agent, while the discriminator $\mathcal{D}$ is a parametric neural network whose main function is to fit the reward function using the expert's example data.

Appendix C. Formula Derivation

We further convert the formula into a probabilistic one. The market price is represented by $v(v \geq 0)$ and $p(v|z)$ to denote the probability density function of the market price given a bid request z. If for any bid request, the price is $b(b \geq 0)$, the bid will be successful when $b \geq v$. We can, therefore, obtain the winning function of the bid request $w(b)$ and the losing function $l(b)$, respectively.

$$w(b) = w(f(v,z)) = P(v < b) = \int_0^b p(v|z)\mathrm{d}v \tag{A5}$$

$$l(b) = P(v \geq b) = 1 - w(b) = \int_b^\infty p(v|z)\mathrm{d}v \tag{A6}$$

Further, the probabilistic model on the bidding strategy is as follows:

$$\max\sum_{i=1}^{N}\int_0^t v(z_i)w(b(v(z_i),z_i),z_i)b(v(z_i),z_i)p(z_i)dz_i \tag{A7}$$

$$\text{s.t.}\quad \sum_{i=1}^{N}\int_0^t w(b(v(z_i),z_i),z_i)b(v(z_i),z_i)p(z_i)dz_i \leq \mathrm{B}$$

$$\mathrm{B}-\sum_{i=1}^{N}\int_0^t w(b(v(z_i),z_i),z_i)b(v(z_i),z_i)dz_i \leq \varepsilon$$

The probabilistic model is further simplified into a model considering a single ad bid request. The bidding strategy is only related to the ad value [18], so the strategy function $b(v(z),z)$ can be simplified to $b(v(z))$ as:

$$(w^*,b^*) = \operatorname{argmax}\int_0^t w(b(v(z)))v(z)p(z)dz \tag{A8}$$

$$\text{s.t.}\int_0^t w(b(v(z)))b(v(z))p(z)dz = \frac{B}{N}$$

Among them, the prior distribution of bid requests $p(z)$, the value assessment of ad display opportunities $v(z)$, and the distribution of winning bids $w(b)$ can be obtained by fitting historical market prices. The total budget of B and the total number of bid requests are determined manually by operators. Therefore, the only variable in the model is the bid strategy function $b(v(z))$. The problem finally turns into a conditional extreme value problem as in Equation (2).

Appendix D. Implementation Details

Table A1. Features and examples of iPinYou data set.

Feature	Description	Example
IP	user's IP address	121.225.158.3
adexchange	the platform of the advertiser participating in the bidding	1
advertiser	participating advertiser ID	1458
bidid	the unique identifier of the bid record	72879b068fec2d3c2afd51
bidprice	advertiser's bid price	300
city	user's city	84
click	the number of clicks on the ad	0
creative	Ad creative logo	48f2e9ba1570a5e1dd653caa
domain	user's area	trqRTuqbjoFf1mKYUV
hour	time	0
ipinyouid	user's ID	Vhk7ZAnyPIc9tbE
keypage	URL of advertiser landing page	befa5efe83be5e7c5085b
logtpye	log type	1
payprice	the transaction price of the ad display opportunity	55
region	user administrative area information	80
slotformat	format of the opportunity	1
slotheight	the height of the opportunity	90
slotid	the slot ID of the opportunity	mm_34955955_11267874
slotprice	the reserved price of the opportunity	0
slotvisibility	the visibility of the opportunity	0
slotwidth	width of the advertising display opportunity	728
timestamp	the timestamp of the bid	20130606000105500.0
url	URL address of the ad display opportunity	de0cca5e4ff921ca803b
useragent	the user's browser information	windows_chrome
usertag	the user's tag	10063130450003707586 1504
weekday	indicates what day of the week it is	4

Table A2. The industry types of advertisers.

Advertiser ID	Category
1458	E-commerce
2259	Milk powder
2261	Communication
2821	Footwear
2997	M-Commerce
3358	Software Development
3386	International E-Commerce
3427	Oil
3476	Tires

Table A3. Statistics of training data. Win Rate $\overline{WR}$ and Click Rate $\overline{C}$ are presented in average.

Adv.ID	Impression	Clicks	Cost ($)	$\overline{WR}$	$\overline{C}$	$\overline{eCPC}$
1458	3,083,056	2454	212,400.20	20.97%	0.08%	86.55
2259	835,556	280	77,754.90	27.97%	0.03%	277.70
2261	687,617	207	61,610.94	31.84%	0.03%	297.64
2821	1,322,561	843	118,082.30	24.99%	0.06%	140.07
2997	312,437	1386	19,689.07	30.69%	0.44%	14.21
3358	1,657,692	1358	160,943.10	46.44%	0.08%	118.51
3386	2,847,802	2076	219,066.90	20.21%	0.07%	105.52
3427	2,512,439	1926	210,239.90	29.35%	0.08%	109.16
3476	1,945,007	1027	156,088.50	23.78%	0.05%	151.98
total	15,204,167	11,557	1,235,875.81	28.47%	0.10%	144.59

Table A4. Statistics of test data. Win Rate $\overline{WR}$ and Click Rate $\overline{C}$ are presented in average.

Adv.ID	Impression	Clicks	Cost ($)	$\overline{WR}$	$\overline{C}$	$\overline{eCPC}$
1458	614,638	543	45,216.45	100%	0.09%	83.27
2259	417,197	131	43,497.56	100%	0.03%	332.04
2261	343,862	97	28,796.00	100%	0.03%	296.87
2821	661,964	394	68,257.10	100%	0.06%	173.24
2997	156,063	533	8617.15	100%	0.34%	16.17
3358	261,001	339	34,159.77	100%	0.13%	100.77
3386	545,421	496	45,715.53	100%	0.09%	92.17
3427	514,559	395	46,356.52	100%	0.08%	117.36
3476	514,560	302	43,627.58	100%	0.06%	144.46
total	4,029,265	3230	364,243.66	100%	0.1%	150.71

References

1. Yuan, S.; Wang, J.; Zhao, X. Real-time bidding for online advertising: Measurement and analysis. In Proceedings of the Seventh International Workshop on Data Mining for Online Advertising, Chicago, IL, USA, 11 August 2013; pp. 1–8.
2. Muthukrishnan, S. Ad exchanges: Research issues. In *International Workshop on Internet and Network Economics*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 1–12.
3. Wu, W.C.-H.; Yeh, M.-Y.; Chen, M.-S. Predicting winning price in real time bidding with censored data. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 10–13 August 2015; pp. 1305–1314.
4. Cui, Y.; Zhang, R.; Li, W.; Mao, J. Bid landscape forecasting in online ad exchange marketplace. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 21–24 August 2011; pp. 265–273.
5. Zhang, W.; Yuan, S.; Wang, J. Optimal real-time bidding for display advertising. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 1077–1086.
6. Zhu, W.-Y.; Shih, W.-Y.; Lee, Y.-H.; Peng, W.-C.; Huang, J.-L. A gamma-based regression for winning price estimation in real-time bidding advertising. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; pp. 1610–1619.
7. Wu, W.; Yeh, M.-Y.; Chen, M.-S. Deep censored learning of the winning price in the real time bidding. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 2526–2535.
8. Wang, T.; Yang, H.; Yu, H.; Zhou, W.; Liu, Y.; Song, H. A revenue-maximizing bidding strategy for demand-side platforms. *IEEE Access* **2019**, *7*, 68692–68706. [CrossRef]
9. Wang, T.; Yang, H.; Jiang, S.; Shi, Y.; Li, Q.; Tang, X.; Yu, H.; Song, H. Kaplan–meier markov network: Learning the distribution of market price by censored data in online advertising. *Knowl.-Based Syst.* **2022**, *251*, 109248. [CrossRef]
10. Ren, K.; Zhang, W.; Chang, K.; Rong, Y.; Yu, Y.; Wang, J. Bidding machine: Learning to bid for directly optimizing profits in display advertising. *IEEE Trans. Knowl. Data Eng.* **2017**, *30*, 645–659. [CrossRef]
11. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 2672–2680.
12. Wang, K.; Gou, C.; Duan, Y.; Lin, Y.; Zheng, X.; Wang, F.-Y. Generative adversarial networks: Introduction and outlook. *IEEE/CAA J. Autom. Sin.* **2017**, *4*, 588–598. [CrossRef]
13. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.
14. Zhang, W.; Zhou, T.; Wang, J.; Xu, J. Bid-aware gradient descent for unbiased learning with censored data in display advertising. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 665–674.
15. Wang, Y.; Ren, K.; Zhang, W.; Wang, J.; Yu, Y. Functional bid landscape forecasting for display advertising. In Proceedings of the Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, 19–23 September 2016; Proceedings, Part I 16; Springer: Berlin/Heidelberg, Germany, 2016; pp. 115–131.
16. Wang, T.; Yang, H.; Liu, Y.; Yu, H.; Song, H. A multimodal approach for improving market price estimation in online advertising. *Knowl.-Based Syst.* **2023**, *266*, 110392. [CrossRef]
17. Cai, H.; Ren, K.; Zhang, W.; Malialis, K.; Wang, J.; Yu, Y.; Guo, D. Real-time bidding by reinforcement learning in display advertising. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, Cambridge, UK, 6–10 February 2017; pp. 661–670.
18. Perlich, C.; Dalessandro, B.; Hook, R.; Stitelman, O.; Raeder, T.; Provost, F. Bid optimizing and inventory scoring in targeted online advertising. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing, China, 12–16 August 2012; pp. 804–812.

19. Lin, C.-C.; Chuang, K.-T.; Wu, W.C.-H.; Chen, M.-S. 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, Indianapolis, IN, USA, 24–28 October 2016; pp. 2143–2148.
20. Chen, Y.; Berkhin, P.; Anderson, B.; Devanur, N.R. Real-time bidding algorithms for performance-based display ad allocation. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 21–24 August 2011; pp. 1307–1315.
21. Chapelle, O. Modeling delayed feedback in display advertising. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 1097–1105.
22. Xu, J.; Lee, K.-c.; Li, W.; Qi, H.; Lu, Q. Smart pacing for effective online ad campaign optimization. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 10–13 August 2015; pp. 2217–2226.
23. Lee, K.-C.; Jalali, A.; Dasdan, A. Real time bid optimization with smooth budget delivery in online advertising. In Proceedings of the Seventh International Workshop on Data Mining for Online Advertising, Chicago, IL, USA, 11 August 2013; pp. 1–9.
24. Afshar, R.R.; Zhang, Y.; Kaymak, U. Dynamic ad network ordering method using reinforcement learning. *Int. J. Comput. Intell. Syst.* **2022**, *15*, 27. [CrossRef]
25. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
26. Wang, Y.; Liu, J.; Liu, Y.; Hao, J.; He, Y.; Hu, J.; Yan, W.P.; Li, M. Ladder: A human-level bidding agent for large-scale real-time online auctions. *arXiv* **2017**, arXiv:1708.05565.
27. Liu, M.; Liu, J.; Hu, Z.; Ge, Y.; Nie, X. Bid optimization using maximum entropy reinforcement learning. *Neurocomputing* **2022**, *501*, 529–543. [CrossRef]
28. Wang, H.; Du, C.; Fang, P.; Yuan, S.; He, X.; Wang, L.; Zheng, B. Roi constrained bidding via curriculum-guided bayesian reinforcement learning. *arXiv* **2022**, arXiv:2206.05240.
29. Lu, J.; Yang, C.; Gao, X.; Wang, L.; Li, C.; Chen, G. Reinforcement learning with sequential information clustering in real-time bidding. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 1633–1641.
30. Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*; PMLR: London, UK, 2016; pp. 1928–1937.
31. Jin, J.; Song, C.; Li, H.; Gai, K.; Wang, J.; Zhang, W. Real-time bidding with multi-agent reinforcement learning in display advertising. In Proceedings of the 27th ACM International Conference on Information and knowledge Management, Torino, Italy, 22–26 October 2018; pp. 2193–2201.
32. Lu, Y.; Lu, C.; Bandyopadhyay, N.; Kumar, M.; Gupta, G. Functional optimization reinforcement learning for real-time bidding. *arXiv* **2022**, arXiv:2206.13939.
33. Abbeel, P.; Ng, A.Y. Apprenticeship learning via inverse reinforcement learning. In Proceedings of the Twenty-First International Conference on Machine Learning, Banff, AB, Canada, 4–8 July 2004; p. 1.
34. Hussein, A.; Gaber, M.M.; Elyan, E.; Jayne, C. Imitation learning: A survey of learning methods. *ACM Comput. Surv. (CSUR)* **2017**, *50*, 1–35. [CrossRef]
35. Pomerleau, D.A. Efficient training of artificial neural networks for autonomous navigation. *Neural Comput.* **1991**, *3*, 88–97. [CrossRef] [PubMed]
36. Ross, S.; Gordon, G.; Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, Ft. Lauderdale, FL, USA, 11–13 April 2011; pp. 627–635.
37. Ho, J.; Gupta, J.; Ermon, S. Model-free imitation learning with policy optimization. In *International Conference on Machine Learning*; PMLR: London, UK, 2016; pp. 2760–2769.
38. Pathak, D.; Agrawal, P.; Efros, A.A.; Darrell, T. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning*; PMLR: London, UK, 2017; pp. 2778–2787.
39. Hafner, D.; Lillicrap, T.; Ba, J.; Norouzi, M. Dream to control: Learning behaviors by latent imagination. *arXiv* **2019**, arXiv:1912.01603.
40. Ren, K.; Qin, J.; Zheng, L.; Yang, Z.; Zhang, W.; Yu, Y. Deep landscape forecasting for real-time bidding advertising. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019, pp. 363–372.
41. Wu, D.; Chen, X.; Yang, X.; Wang, H.; Tan, Q.; Zhang, X.; Xu, J.; Gai, K. Budget constrained bidding by model-free reinforcement learning in display advertising. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino, Italy, 22–26 October 2018; pp. 1443–1451.
42. Ausubel, L.M. A generalized vickrey auction. *Econo0 Metr.* **1999**, 23, 493-505.
43. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv* **2013**, arXiv:1312.5602.
44. Wang, S.; Jia, D.; Weng, X. Deep reinforcement learning for autonomous driving. *arXiv* **2018**, arXiv:1811.11329.

45. Zhou, F.; Luo, B.; Hu, T.; Chen, Z.; Wen, Y. A combinatorial recommendation system framework based on deep reinforcement learning. In Proceedings of the 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 15–18 December 2021; pp. 5733–5740.
46. Cui, B.; Sun, R.; Su, J. A novel deep reinforcement learning strategy in financial portfolio management. In Proceedings of the 2022 7th International Conference on Big Data Analytics (ICBDA), Guangzhou, China, 4–6 March 2022; pp. 341–348.
47. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef] [PubMed]
48. Ross, S.; Bagnell, D. Efficient reductions for imitation learning. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, Sardinia, Italy, 13–15 May 2010; pp. 661–668.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.