Online Appendices to "Cold Start to Improve Market Thickness on Online Advertising Platforms: Data-Driven Algorithms and Field Experiments"

Zikun Ye, Dennis J. Zhang, Heng Zhang, Renyu Zhang, Xin Chen, Zhiwei Xu

Appendix A: Table of Notations

We provide a list of the notations in Table 4.

Appendix B: Supporting Argument for Regret Analysis

We devote this section to the proof of Theorem 1. Supporting analysis for justifying the prediction oracle assumption (Assumption 2) for the case of neural networks can be found in Appendix G. Before presenting the full-fledged proof of Theorem 1, we first give the proof of Lemma 1, followed by some preliminaries.

B.1. Proof of Lemma 1.

Let y^* denote the optimal solution of the optimization model in Lemma 1. Consider an arbitrary policy π, we have the following observation:

$$\begin{aligned}\frac{1}{T}\cdot\mathbb{E}_{\mathcal{D}^T,\pi}\big[\Gamma(V)\big] &= \frac{1}{T}\mathbb{E}_{\mathcal{D}^T,\pi}\left[\sum_{j=1}^K b_j\sum_{t=1}^T v_{tj}+\sum_{j=1}^K\beta_j\min\left\{\sum_{t=1}^T v_{tj},\alpha T\right\}\right]\\ &= \mathbb{E}_{\mathcal{D}^T,\pi}\left[\sum_{j=1}^K b_j\sum_{t=1}^T v_{tj}/T+\sum_{j=1}^K\beta_j\min\left\{\sum_{t=1}^T v_{tj}/T,\alpha\right\}\right]\\ &\le \sum_{j=1}^K b_j\mathbb{E}_{\mathcal{D}^T,\pi}\left[\sum_{t=1}^T v_{tj}/T\right]+\sum_{j=1}^K\beta_j\min\left\{\mathbb{E}_{\mathcal{D}^T,\pi}\left[\sum_{t=1}^T v_{tj}/T\right],\alpha\right\},\end{aligned}$$

in which the inequality follows from the Jensen's inequality.

Let us use $\mathcal{A}_{tj}$ to denote the event that ad j is displayed for user t and $\mathcal{D}_{ij}$ to denote the distribution that ad j is clicked when the context is i and ad j is displayed. It then follows that

$$\begin{aligned}&\mathbb{E}_{\mathcal{D}^T,\pi}\left[\sum_{t=1}^T v_{tj}/T\right]=\sum_{t=1}^T\mathbb{E}_{i\sim\mathcal{D}_X}\left[\mathbb{E}_{\mathcal{H}^t,\pi,\mathcal{D}_{ij}}[v_{tj}|i]/T\right]=\sum_{t=1}^T\mathbb{E}_{i\sim\mathcal{D}_X}\left[\mathbb{P}_\pi(\mathcal{A}_{tj}|i)/T\cdot c_{ij}\right]\\ =&\mathbb{E}_{i\sim\mathcal{D}_X}\left[\sum_{t=1}^T\mathbb{P}_\pi(\mathcal{A}_{tj}|i)/T\cdot c_{ij}\right]=\mathbb{E}_{i\sim\mathcal{D}_X}[c_{ij}y_{ij}^\pi],\end{aligned}$$

in which we define $y_{ij}^\pi=\sum_{t=1}^T\mathbb{P}_\pi(\mathcal{A}_{tj}|i)/T$. Note that for any fixed i, it must be that $\sum_{j=1}^K y_{ij}^\pi=1$. Therefore,

$$\begin{aligned}\frac{1}{T}\cdot\mathbb{E}_{\mathcal{D}^T,\pi}\big[\Gamma(V)\big]&\le\sum_{j=1}^K b_j\mathbb{E}_{i\sim\mathcal{D}_X}[c_{ij}y_{ij}^\pi]+\sum_{j=1}^K\beta_j\min\left\{\mathbb{E}_{i\sim\mathcal{D}_X}[c_{ij}y_{ij}^\pi],\alpha\right\},\\ &\le\sum_{j=1}^K b_j\mathbb{E}_{i\sim\mathcal{D}_X}\left[c_{ij}y_{ij}^*\right]+\sum_{j=1}^K\beta_j\min\left\{\mathbb{E}_{i\sim\mathcal{D}_X}[c_{ij}y_{ij}^*],\alpha\right\}=\mathsf{OPT}.\end{aligned}$$

This concludes the proof. □

B.2. Preliminaries for Regret Analysis

We first make several additional assumptions in our proof, purely for the ease and clarity of exposition. First of all, instead of solving

$$\mathsf{OPT}^t=\min_{\lambda_j\in[0,\beta_j],\ \forall j\in A}\sum_{i\in\mathcal{I}}\hat{p}_i^t\max_{j=1,2,\dots,K}\left(\hat{c}_{ij}^t(b_j+\lambda_j)\right)+\alpha\sum_{j=1}^K(\beta_j-\lambda_j),\tag{11}$$

Table 4 Table of Notations

Notations in the Allocation Model and SBL Algorithm	
Notation	Description
K	The number of ads
$A=\{1,2,\dots,K\}$	The set of ads
T	The total number of the user views, namely, ad impressions
X	The finite or countably infinite context set
x_{tj}	x_{tj} is the feature vector associated with round t and ad j
$x_t=(x_{t1},\dots,x_{tK})\in X$	The context associated with round t
$\mathcal{I}=\{1,2,3,\dots\}$	The index set of context types
$a_t\in A$	The ad which is chosen to be displayed to the user view t
$v_t(a_t)\in\{0,1\}^K$	The K-dimensional binary vector representing whether each ad is clicked
$\mathcal{D}$	The distribution of $(x_t,\{v_t(a)\}_{a\in K})$ over $X\times\{0,1\}^{K\times K}$
$\mathcal{D}_X$	The marginal distribution of $\mathcal{D}$ over context types $[m]$
c_{ij}	The CTR of ad j under the context i
$V:=\sum_{t=1}^{T}v_t(a_t)$	The accumulated click-through vector
$b_j\in[0,1]$	The bid per click of ad $j\in A$
$\beta_j\in(0,1]$	The cold start reward per click of ad $j\in A$
$\alpha\in(0,1)$	The target click per round
$\Gamma(V)$	The objective value
$\mathcal{H}_t=\bigcup_{s=1,\dots,t-1}\{(x_s,a_s,v_s(a_s))\}$	The history update to round t
$\Delta_A=\{y\in\mathbb{R}^{\lvert A\rvert}:y_j\geq 0,\forall j\in A,\sum_{j\in A}y_j\leq 1\}$	The distribution over arms which defines the feasible ad allocation plan
π	The policy mapping from $\mathcal{H}_t$ to Δ_A
$\hat{c}_{ij}^t$	The predicted CTR based on $\mathcal{H}_t$ of ad j under context i
λ^{t*}	The empirically optimal shadow bidding prices at round t
λ^t	The shadow bidding prices generated by the SBL algorithm at round t
p_i	The probability that context $i\in\mathcal{I}$ occurs
$\hat{p}_i^t$	The empirically estimated probability that context $i\in\mathcal{I}$ occurs at round t
$\tau_1<\tau_2<\dots$	The epoch schedule to update λ
$f(x)=O(g(x))$	There exists a positive constant c such that $f(x)\leq c\cdot\lvert g(x)\rvert$ for sufficiently large x
$f(x)=\tilde{O}(g(x))$	There exists a positive constant c such that $f(x)\leq c\cdot\lvert g(x)\rvert\cdot\log^k(\lvert g(x)\rvert)$ for some $k>0$ and sufficiently large x
$f(x)=\Omega(g(x))$	There exists a positive constant c such that $f(x)\geq c\cdot\lvert g(x)\rvert$ for sufficiently large x
$f(x)=\Theta(g(x))$	$\lvert f(x)\rvert/\lvert g(x)\rvert$ converges to 1 as x goes to infinity.
Notations in the Neural Network Prediction Oracle	
Notation	Description
$\mathcal{X}$	The set of functions $(X\times A\mapsto[0,1])$ to estimate CTR
w_0	The dimension of the context $x_{ij}\in\mathbb{R}^{w_0}$
w	The number of hidden nodes of the neural network
L	The number of hidden layers of the neural network
d	The prediction error term of the regressor
$\theta\in\mathbb{R}^d$	The coefficients of parameters in the neural network
$\theta_0\in\mathbb{R}^d$	The initialized coefficients of parameters in the neural network
$\theta^t\in\mathbb{R}^d$	The updated coefficients of parameters in the neural network at round t
$H_j(x_{ij},\theta)$	The output of the neural network parameterized by θ given the feature input x_{ij} associated with context i and ad j
$\theta^*\in\mathbb{R}^d$	The coefficients such that $c_{ij}=\langle\nabla_\theta H_j(x_i,\theta_0),\theta^*-\theta_0\rangle$
λ_0	The regularization parameter in training the neural network
η	The step size in training the neural network
U	The number of descent steps in training the neural network
I_d	The identity matrix of dimension d
$\boldsymbol{H}$	The neural tangent kernel matrix defined by Zhou et al. (2020)
γ	The scalar such that $\boldsymbol{H}\succeq\gamma I$, where $M_1\succeq M_2$ refers to that M_1-M_2 is semi-positive-definite
Notations in Proofs	
Notation	Description
N_j^t	The set of contexts i for which $j\in\arg\max_{j'}\hat{c}_{ij'}^t(b_{j'}+\lambda_{j'})$
$\mathcal{T}_j^t$	The time periods before the round t when ad j is displayed
$n_j^t=\lvert\mathcal{T}_j^t\rvert$	The cardinality of the set $\mathcal{T}_j^t$, i.e, the number of displays of ad j before round t
$g_{ij}=\nabla_\theta H_j(x_i,\theta_0)$	The gradients of the function $H_j(x_i,\theta_0)$
$\hat{\gamma}_j^t=\sum_{i\in N_j^t}p_i\hat{c}_{ij}^t$	The empirical estimated probability of click-through for ad j at round t
$\gamma_j^t=\sum_{i\in N_j^t}p_ic_{ij}^t$	The expected probability of click-through for ad j at round t
$n(j)$	The number of times that ad j is clicked over the whole horizon

in the analysis, we assume that we solve

$$\mathsf{OPT}^t = \min_{\lambda_j \in [0,\beta_j],\ \forall j \in A} \sum_{i \in \mathcal{I}} p_i \max_{j=1,2,\ldots,K} \left(\hat{c}_{ij}^t (b_j + \lambda_j) \right) + \alpha \sum_{j=1}^{K} (\beta_j - \lambda_j). \tag{12}$$

That is, we assume that we observe p_i for any $i \in \mathcal{I}$, instead of only having access to its empirical estimate. In the meanwhile, we replace Assumption 1 with $p_i \hat{c}_{ij} \le O(T^{-\frac{1}{3}} (\log T)^{\frac{1}{3}} K^{-\frac{5}{3}})$ for all $i \in \mathcal{I}$. This is without loss of generality, because all the argument presented in this section still follows without this assumption. Indeed, with McDiarmid's inequality and the bound of Rademacher complexity term (Boucheron et al. 2013) with countably many contexts, we can still uniformly bound the error of the empirical probability estimate $\hat{p}_i^t$. Specifically, with probability at least $1 - t^{-4}$, for any context $i \in \mathcal{I}$, we have $|\hat{p}_i^t - p_i| \le O(\sqrt{\log t / t})$, where t is the total number of occurrences for context i. As a result, this introduces a lower order error than $\tilde{O}(t^{-1/3})$, which can be ignored for our regret analysis. We will discuss more about this point at the end of the proof of Theorem 1 (see Appendix B.3).

Second, we assume that, when we solve (12), the inputs are in a *general position*. In other words, for any shadow bidding prices λ and round t when deciding which ad to display given a context, namely, $\left| \left\{ i : |\arg\max_k \{\hat{c}_{ik}^t (\lambda_k + b_k)\}| > 1 \right\} \right| \le K$. This assumption is introduced to avoid too many ties in Step 1 of the SBL algorithm, thus bounding the gap between primal and dual solutions in a lower order compared to the total regret. Similar assumptions are also made in the online matching and online linear program literature, e.g., Devanur and Hayes (2009), Agrawal et al. (2014). As argued by Devanur and Hayes (2009), when the general position assumption does not hold, an infinitesimal permutation ξ_{ij} can be added to each $\hat{c}_{ij}^t$ without affecting much of the objective function value for any λ, where ξ_{ij} is chosen independently and uniformly at random from a tiny interval $[-\varepsilon, \varepsilon]$ with ε being arbitrarily small. Hence, it is without loss of generality to assume the total number of ties is bounded by K with probability one. As will be clear in the proof of Lemma 2 below, the general position assumption helps us bound the total number of entries for the allocation decision constructed from the dual that are different from the primal solution by $O(K^2)$.

Next, some definitions and notations are in order. Throughout the proof of Theorem 1, we define the reward process (for any policy) $\{r(x_t, a_t)\}_{t=1}^T$, where the reward at round t is denoted by:

$$r(x_t, a_t) = \begin{cases} 0, & \text{if } \ell_t(a_t) = 0, \\ b_{a_t}, & \text{if } \sum_{s=1}^{t-1} \ell_s(a_t) \ge \alpha T \text{ and } \ell_t(a_t) = 1, \\ b_{a_t} + \beta_{a_t}, & \text{if } \sum_{s=1}^{t-1} \ell_s(a_t) < \alpha T \text{ and } \ell_t(a_t) = 1, \end{cases}$$

where $\ell_t(a) = 1$ if and only if $a = a_t$ and a click-through occurs. Notice that the objective value is given by $\Gamma(V) = \sum_{t=1}^T r(x_t, a_t)$. By Lemma 1, the expected reward satisfies $\mathbb{E}[\sum_{t=1}^T r(x_t, a_t)] \le T \cdot \mathsf{OPT}$. We note that $r(\cdot, \cdot)$ corresponds to the real reward collection process, which is hard to work with, because it depends on the click-through history of each ad so far. To overcome this challenge, we instead work with an auxiliary reward process, defined by

$$\tau(x_t, a_t) = \begin{cases} 0, & \text{if } \ell_t(a_t) = 0, \\ b_{a_t} + \beta_{a_t}, & \text{if } \ell_t(a_t) = 1. \end{cases}$$

Note $\tau(x_t, a_t) - r(x_t, a_t) = \beta_{a_t}$ when $\sum_{s=1}^{t-1} \ell_s(a_t) \ge \alpha T$ and a click-through occurs in round t. Otherwise, $\tau(x_t, a_t) = r(x_t, a_t)$.

Furthermore, we define N_j^t as the set of contexts i for which ad j is chosen to be displayed, with tie-breaking resolved, in round t. Thus, if $i \in N_j^t$, it holds that $j \in \arg\max_{j'} \hat{c}_{ij'}^t(b_{j'} + \lambda_{j'})$. Let λ^{t*} be the empirical optimal dual solution to (12), y^{t*} be the corresponding empirical optimal primal solution, and we use $\hat{y}$ to represent the primal integer allocation decision to any shadow bidding strategy λ with arbitrary tie-breaking. Hence, $N_j^t = \{i : \hat{y}_{ij}^t = 1\}$.

Before formally presenting the full-fledged analysis, we need to bound the regret induced by the tie-breaking problem and dual mirror descent in SBL algorithms. Clearly, complementary slackness implies that if $y_{i\ell}^{t*} > 0$, then $\ell \in \arg\max_j \hat{c}_{ij}^t(b_j + \lambda_j^t)$. Hence, if $\arg\max_j \hat{c}_{ij}^t(b_j + \lambda_j^t)$ returns a unique solution ℓ, then the optimal primal allocation y^{t*} and the integer solution $\hat{y}^{t*}$ constructed from the dual λ^{t*} are the same. Because we use the dual-based solution to make the ad allocation decision in the primal space, the tie-breaking in the SBL algorithm can induce a difference between empirically optimal objective value and the objective value by the dual-based allocation. One may expect that the solution to (12) still yields a good performance due to complementary slackness, the general position assumption, and Assumption 1. Formally, the following lemma holds.

LEMMA 2 **(Approximate Complementary Slackness)**. *Under Assumption 1, we have:*

(a) Suppose the SBL-RS algorithm is applied. There exist a family of non-negative constants $\{\eta_j \geq 0 : j \in A\}$ *with* $\sum_{j\in A} \eta_j \leq O(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}} K^{\frac{1}{3}})$*, such that the following approximate complementary slackness condition holds for each* $j \in A$*: (i) If* $\lambda_j^t \in [0, \beta_j)$*, we have* $\sum_{i\in N_j^t} p_i \hat{c}_{ij}^t \geq \alpha - \eta_j$*. (ii) If* $\lambda_j^t \in (0, \beta_j]$*, we have* $\sum_{i\in N_j^t} p_i \hat{c}_{ij}^t \leq \alpha + \eta_j$*.*

(b) Suppose the SBL-DMD algorithm is applied. Define $s_t(\lambda) = -\sum_{j\in[K]\setminus a_t} \alpha\lambda_j + (\hat{c}_{i_t a_t}^t - \alpha)\lambda_{a_t}$*. There exist a family of non-negative constants* $\{\eta_j \geq 0 : j \in A\}$ *with* $\sum_{j\in A} \eta_j \leq O(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}} K^{\frac{1}{3}}) + \mathbb{E}_{i\sim\mathcal{D}_X}\left[s_t(\lambda) + \frac{2\eta}{\sigma} + \frac{1}{\eta} D_\varphi(\lambda, \lambda^t) - \frac{1}{\eta} D_\varphi(\lambda, \lambda^{t+1})\right]$ *such that for all feasible* λ*, the approximate complementary slackness condition defined in part (a) holds.*

Proof of Lemma 2.

We first prove **Part (a)**, i.e., the approximate complementary slackness for the SBL-RS algorithm. To analyze the non-smooth convex dual (9), we first write down the corresponding primal linear program (13) and its dual (14) as follows,

$$\begin{aligned} \text{(Primal)} \quad \mathsf{OPT}^t = \max_{y\geq 0, u\geq 0} \quad & \sum_{i\in\mathcal{I}} \sum_{j=1}^{K} p_i \hat{c}_{ij}^t b_j y_{ij} + \sum_{j=1}^{K} \beta_j(\alpha - u_j) \\ \text{s.t.} \quad & \sum_{j=1}^{K} y_{ij} \leq 1, \ \forall i \in \mathcal{I}, \ \sum_{i\in\mathcal{I}} p_i \hat{c}_{ij}^t y_{ij} + u_j \geq \alpha, \ \forall j \leq K, \end{aligned} \tag{13}$$

and,

$$\begin{aligned} \text{(Dual)} \quad \mathsf{OPT}^t = \min_{\lambda\geq 0, \mu} \quad & \sum_{i\in\mathcal{I}} p_i \mu_i + \alpha \sum_{j=1}^{K} (\beta_j - \lambda_j) \\ \text{s.t.} \quad & \lambda_j \leq \beta_j \ \forall j \leq K \\ & \mu_i - \hat{c}_{ij}^t \lambda_j \geq \hat{c}_{ij}^t b_j \ \forall i \in \mathcal{I}, \ \forall j \leq K. \end{aligned} \tag{14}$$

Let (y^{t*}, u^{t*}) and (λ^{t*}, μ^{t*}) be the optimal solution to (13) and (14) respectively. Clearly, the following complementary slackness conditions

$$\lambda_j^{t*}(\alpha - u_j^{t*} - \sum_{i\in\mathcal{I}} p_i \hat{c}_{ij}^t y_{ij}^{t*}) = 0,\ u_j^{t*}(\lambda_j^{t*} - \beta_j) = 0,\ \text{and}\ y_{ij}^{t*}(\mu_i^{t*} - \hat{c}_{ij}^{t*}\lambda_j^{t*} - \hat{c}_{ij}^{t*} b_j) = 0,$$

hold for all $i \in \mathcal{I}$ and $j \in A$. To highlight the intuition, let us first consider the case in which there is no tie in $\arg\max_{j'} \hat{c}_{ij'}^t(b_{j'} + \lambda_{j'}^{t*})$ for any $i \in \mathcal{I}$. Notice that if $\ell \notin \arg\max_{j'} \hat{c}_{ij'}^t(b_{j'} + \lambda_{j'}^{t*})$, then the complementary condition implies that $y_{i\ell}^{t*} = 0$. Since the primal program is increasing in y, it must be that $y_{ij}^{t*} = 1$ if $j = \arg\max_{j'} \hat{c}_{ij'}^t(b_{j'} + \lambda_{j'}^{t*})$. In this case, $i \in N_j^t$ if and only if $y_{ij}^{t*} = 1$, and $y_{ij}^{t*} = 0$ otherwise . If $\lambda_j^{t*} < \beta_j$, then $u_j^{t*} = 0$. As a result, $\sum_{i\in N_j^t} p_i \hat{c}_{ij}^t = \sum_{i\in N_j^t} p_i \hat{c}_{ij}^t y_{ij}^{t*} = \sum_{i\in\mathcal{I}} p_i \hat{c}_{ij}^t y_{ij}^{t*} + u_j^{t*} \geq \alpha$. Similarly, if $\lambda_j^{t*} > 0$, then $\alpha - u_j^{t*} - \sum_{i\in\mathcal{I}} p_i \hat{c}_{ij}^t y_{ij}^{t*} = 0$, which implies that $\sum_{i\in N_j^t} p_i \hat{c}_{ij}^t \leq \alpha$. So, if there is no tie, the lemma holds true with $\eta_j = 0$, $\forall j \in A$.

Next, we bound the gap in the case under tie-breaking under the primal allocation induced by the empirical optimal dual solution, i.e., gap induced by difference between $\hat{y}^{t*}$ and y^{t*}. By the general position assumption, there are at most K ties and, thus, the tie-breaking introduces at most K^2 different entries between a primal optimal allocation y^{t*} and the corresponding integer solution $\hat{y}^{t*}$. This is because, with an argument similar to the one in the previous argument, for $i \in \mathcal{I}$ such that there is no tie, the K-dimension decision vector must satisfy $\hat{y}_{i\cdot}^{t*} = y_{i\cdot}^{t*}$ and all entries in the vector belong to the set $\{0, 1\}$. Hence,

$$\begin{aligned}
&\sum_{j=1}^{K} \left| \sum_{i\in N_j^{t*}} p_i \hat{c}_{ij}^t - \sum_{i\in\mathcal{I}} p_i \hat{c}_{ij}^t y_{ij}^{t*} \right| = \sum_{j=1}^{K} \left| \sum_{i\in\mathcal{I}} p_i \hat{c}_{ij}^t \hat{y}_{ij}^{t*} - \sum_{i\in\mathcal{I}} p_i \hat{c}_{ij}^t y_{ij}^{t*} \right| \\
\leq &\sum_{j=1}^{K} \sum_{i\in\mathcal{I}} p_i \hat{c}_{ij}^t \left| \hat{y}_{ij}^{t*} - y_{ij}^{t*} \right| \leq O(K^2 (T^{-\frac{1}{3}} (\log T)^{\frac{1}{3}} K^{-\frac{5}{3}})) = O(T^{-\frac{1}{3}} (\log T)^{\frac{1}{3}} K^{\frac{1}{3}}),
\end{aligned}$$

in which the first inequality follows from the definition of $\hat{y}^{t*}$ and the second inequality is due to Assumption 1. Let us define $\eta_j := |\sum_{i\in N_j^{t*}} p_i \hat{c}_{ij}^t - \sum_{i\in\mathcal{I}} p_i \hat{c}_{ij}^t y_{ij}^{t*}|$. If $\lambda_j^{t*} < \beta_j$ (i.e., Part (i) of the approximate complementary slackness condition), it holds that $u_j^{t*} = 0$ and $\sum_{i\in N_j^{t*}} p_i \hat{c}_{ij}^t y_{ij}^{t*} = \sum_{i\in\mathcal{I}} p_i \hat{c}_{ij}^t y_{ij}^{t*} + u_j^{t*} \geq \alpha$. Therefore, $\sum_{i\in N_j^{t*}} p_i \hat{c}_{ij}^t \geq p_i \hat{c}_{ij}^t y_{ij}^{t*} - \eta_j \geq \alpha - \eta_j$. The argument for the case $\lambda_j^{t*} > 0$ (i.e., Part (ii) of the approximate complementary slackness condition) follows similarly. This complete the proof of Part (a).

Finally, we prove **part (b)**, i.e., to bound the regret induced by dual mirror descent in SBL-DMD by using a standard result on online mirror descent, i.e., Proposition 1 in Section B.4. We define the primal objective function $\mathsf{Obj}^t(y, u)$ of the optimization model (13), and the dual objective function $\mathsf{Obj}^t(\lambda) := \sum_{i\in\mathcal{I}} p_i \max_{j'=1,2,\ldots,K} \left(\hat{c}_{ij'}^t (b_{j'} + \lambda_{j'}) \right) + \alpha \sum_{j=1}^{K} (\beta_j - \lambda_j)$ after using the optimal $\mu_i^t = \max_{j'\in[K]} \hat{c}_{ij'}^t (b_{j'} + \lambda_{j'})$, for all $i \in \mathcal{I}$, and $s_t(\lambda) = -\sum_{j\in[K]\setminus a_t} \alpha\lambda_j + (\hat{c}_{i_t a_t}^t - \alpha)\lambda_{a_t}$. In SBL-DMD, we update λ^t over periods and (y^t, u^t) denotes the corresponding primal decision, i.e., $(y^t, u^t) = \arg\max_{y\geq 0, u\geq 0} \sum_{i\in\mathcal{I}} \sum_{j=1}^{K} p_i \hat{c}_{ij}^t b_j y_{ij} + \sum_{j=1}^{K} \beta_j(\alpha - u_j) + \sum_{i\in\mathcal{I}} \mu_i^t (1 - \sum_{j=1}^{K} y_{ij}) + \sum_{j=1}^{K} \lambda_j^t (\sum_{i\in\mathcal{I}} p_i \hat{c}_{ij}^t y_{ij} + u_j - \alpha)$. One can check, for any realized i_t at period t, the corresponding played arm a_t with primal decision $y_{i_t a_t}^t > 0$ is exactly the decision $a_t = \arg\max_{j\in[K]} \hat{c}_{i_t j}^t (b_j + \lambda_j^t)$ defined in SBL-DMD. Then, we have the following inequality,

$$\begin{aligned}
\mathsf{Obj}^t(y^{t*},u^{t*})-\mathsf{Obj}^t(y^t,u^t) &\le O(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}})+\mathsf{Obj}^t(y^{t*},u^{t*})-\big(\mathsf{Obj}^t(\lambda^t)-\sum_{j=1}^{K}\lambda_j^t(\sum_{i\in\mathcal{I}}p_i\hat{c}_{ij}^t y_{ij}^t-\alpha)\big)\\
&\le O(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}})+\sum_{j=1}^{K}\lambda_j^t(\sum_{i\in\mathcal{I}}p_i\hat{c}_{ij}^t y_{ij}^t-\alpha)\\
&= O(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}})+\mathbb{E}_{i\sim\mathcal{D}_X}\left[s_t(\lambda^t)\right]\\
&\le O(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}})+\mathbb{E}_{i\sim\mathcal{D}_X}\left[s_t(\lambda)+\frac{2\eta}{\sigma}+\frac{1}{\eta}D_\varphi(\lambda,\lambda^t)-\frac{1}{\eta}D_\varphi(\lambda,\lambda^{t+1})\right],
\end{aligned}$$

where the first inequality follows from tie-breaking error induced by the primal (y^t,u^t) and dual λ^t shown in Part (a), as well as the definition of the Lagrange dual. The second inequality follows from the weak duality, the equality follows from the definition of $s_t(\lambda^t)$, and the third inequality from Proposition 1, which holds for any λ. Thus, we complete the proof. □

B.3. Proof of Theorem 1

In this part, we present the main argument for the proof of Theorem 1. Note that N_j^t and $\hat{c}_{ij}^t$ are random variables measurable with respect to the history $\mathcal{H}_t$. We define $\hat{\gamma}_j^t := \sum_{i\in N_j^t} p_i\hat{c}_{ij}^t$ and $\gamma_j^t := \sum_{i\in N_j^t} p_i c_{ij}^t$. One can show that for the SBL algorithms, the expected number of any ad j being sampled before round t is $\sum_{s=1}^{t}\frac{\epsilon_s}{K} = \Theta\left(t^{\frac{2}{3}}K^{-\frac{2}{3}}(\log t)^{\frac{1}{3}}\right)$. By Hoeffding's inequality, by round t, ad j has been sampled $\Theta\left(t^{\frac{2}{3}}K^{-\frac{2}{3}}(\log t)^{\frac{1}{3}}\right)$ times with probability $1-t^{-4}$. This implies that, by Assumption 2 and the union bound, the CTR estimate satisfies $|\hat{c}_{ij}^t - c_{ij}^t| = O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$ for all ads with probability at least $1-t^{-3}$. Combining the above observations, we will show the following lemma.

LEMMA 3 **(Per Period Gap of the Alternative Reward Process).**

(*a*) *Conditioned on exploitation at round t of the SBL-RS algorithm, it holds that*

$$\mathbb{E}\left[\tau(x_t,a_t)\right] \ge \mathsf{OPT}+\mathbb{E}[\sum_{j=1}^{K}\beta_j(\gamma_j^t-\alpha)^+]-O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)-O\left(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}\right). \tag{15}$$

(*b*) *Conditioned on exploitation at round t of the SBL-DMD algorithm, it holds that, for all feasible λ*

$$\begin{aligned}
\mathbb{E}\left[\tau(x_t,a_t)\right] \ge \mathsf{OPT}+\mathbb{E}[\sum_{j=1}^{K}\beta_j(\gamma_j^t-\alpha)^+]-O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)-O\left(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}\right)\\
-\mathbb{E}_{i\sim\mathcal{D}_X}\left[s_t(\lambda)+\frac{2\eta}{\sigma}+\frac{1}{\eta}D_\varphi(\lambda,\lambda^t)-\frac{1}{\eta}D_\varphi(\lambda,\lambda^{t+1})\right].
\end{aligned} \tag{16}$$

Proof of Lemma 3

We first show **part (b)**. Applying the approximate complementary slackness (Lemma 2(b)), we bound the expected empirical auxiliary reward process under the implementation of the dual-solution in the primal space:

$$\sum_{i\in N_j^t} p_i\hat{c}_{ij}^t(b_j+\beta_j).$$

Fixing the history $\mathcal{H}_t$ and an arbitrary ad j, we have the following equality:

$$\begin{aligned}\mathsf{OPT}^t &= \sum_{i\in\mathcal{I}} p_i \max_{j'=1,2,\ldots,K}\left(\hat{c}^t_{ij'}(b_{j'}+\lambda^t_{j'})\right)+\alpha\sum_{j=1}^{K}(\beta_j-\lambda^t_j)\\ &= \sum_{j=1}^{K}\sum_{i\in N^t_j} p_i\hat{c}^t_{ij}(b_j+\lambda^t_j)+\alpha\sum_{j=1}^{K}(\beta_j-\lambda^t_j)\\ &= \sum_{j=1}^{K}\sum_{i\in N^t_j} p_i\hat{c}^t_{ij}(b_j+\beta_j)-\sum_{j=1}^{K}(\hat{\gamma}^t_j-\alpha)(\beta_j-\lambda^t_j),\end{aligned}$$

where the first equality follows from the definition of OPT^t, the second from the definition of N^t_j, and the third from the identity $\sum_{i\in N^t_j} p_i\hat{c}^t_{ij}=\hat{\gamma}^t_j$. Thus, we have

$$\sum_{j=1}^{K}\sum_{i\in N^t_j} p_i\hat{c}^t_{ij}(b_j+\beta_j)=\mathsf{OPT}^t+\sum_{j=1}^{K}(\hat{\gamma}^t_j-\alpha)(\beta_j-\lambda^t_j). \tag{17}$$

Hence, if the (exact) complementary slackness condition holds with $\eta_j=0$ for all $j\in A$ (see Lemma 2(b)), we have $(\hat{\gamma}^t_j-\alpha)(\beta_j-\lambda^t_j)=\beta_j(\hat{\gamma}^t_j-\alpha)^+$. In this case,

$$\sum_{j=1}^{K}\sum_{i\in N^t_j} p_i\hat{c}^t_{ij}(b_j+\beta_j)=\mathsf{OPT}^t+\sum_{j=1}^{K}(\hat{\gamma}^t_j-\alpha)(\beta_j-\lambda^t_j)=\mathsf{OPT}^t+\sum_{j=1}^{K}\beta_j(\hat{\gamma}^t_j-\alpha)^+.$$

Otherwise, $\eta_j>0$ for some $j\in A$ in Lemma 2(b). In this case, we show the following bound for the SBL-DMD algorithm:

$$\sum_{j=1}^{K}(\hat{\gamma}^t_j-\alpha)(\beta_j-\lambda^t_j)\ge\sum_{j=1}^{K}\beta_j(\hat{\gamma}^t_j-\alpha)^+-O\left(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}\right)-\mathbb{E}_{i\sim\mathcal{D}_X}\left[s_t(\lambda)+\frac{2\eta}{\sigma}+\frac{1}{\eta}D_\varphi(\lambda,\lambda^t)-\frac{1}{\eta}D_\varphi(\lambda,\lambda^{t+1})\right]. \tag{18}$$

To obtain the inequality in (18), we observe, by Lemma 2(b), that $\sum_{j\in A}\eta_j\le O(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}})+\mathbb{E}_{i\sim\mathcal{D}_X}\left[s_t(\lambda)+\frac{2\eta}{\sigma}+\frac{1}{\eta}D_\varphi(\lambda,\lambda^t)-\frac{1}{\eta}D_\varphi(\lambda,\lambda^{t+1})\right]$, so it suffices to show that

$$(\hat{\gamma}^t_j-\alpha)(\beta_j-\lambda^t_j)\ge\beta_j(\hat{\gamma}^t_j-\alpha)^+-\eta_j \text{ for all } j\in A.$$

More specifically, we consider three cases: (a) $\lambda^t_j=0$, (b) $\lambda^t_j\in(0,\beta_j)$, and (c) $\lambda^t_j=\beta_j$.

If $\lambda^t_j=0$, we have $(\hat{\gamma}^t_j-\alpha)(\beta_j-\lambda^t_j)=\beta_j(\hat{\gamma}^t_j-\alpha)$. If $\hat{\gamma}^t_j>\alpha$, clearly, $\beta_j(\hat{\gamma}^t_j-\alpha)^+=\beta_j(\hat{\gamma}^t_j-\alpha)$. Otherwise, $(\hat{\gamma}^t_j-\alpha)^+=0$, and $\lambda^t_j=0$ implies that $\eta_j\ge\alpha_j-\hat{\gamma}^t_j$. Therefore,

$$\beta_j(\hat{\gamma}^t_j-\alpha)=\beta_j(\hat{\gamma}^t_j-\alpha)^+-\beta_j(\alpha-\hat{\gamma}^t_j)\ge\beta_j(\hat{\gamma}^t_j-\alpha)^+-\beta_j\eta_j\ge\beta_j(\hat{\gamma}^t_j-\alpha)^+-\eta_j,$$

where the last inequality follows from $\beta_j\in[0,1]$.

If $\lambda^t_j=\beta_j$, we have $(\hat{\gamma}^t_j-\alpha)(\beta_j-\lambda^t_j)=0$. Furthermore, the following inequality holds

$$0=\beta_j(\hat{\gamma}^t_j-\alpha)^+-\beta_j(\hat{\gamma}^t_j-\alpha)^+\ge\beta_j(\hat{\gamma}^t_j-\alpha)^+-\beta_j\eta_j\ge\beta_j(\hat{\gamma}^t_j-\alpha)^+-\eta_j,$$

where the first inequality follows from $\hat{\gamma}^t_j-\alpha\le\eta_j$ (see Lemma 2(b)) and $\eta_j\ge0$, which together imply $(\hat{\gamma}^t_j-\alpha)^+\le\eta_j$, and the second from $\beta_j\in[0,1]$. It then follows that $(\hat{\gamma}^t_j-\alpha)(\beta_j-\lambda^t_j)\ge\beta_j(\hat{\gamma}^t_j-\alpha)^+-\eta_j$ for the case where $\lambda^t_j=\beta_j$.

If $\lambda_j \in (0, \beta_j)$, Lemma 2(b) suggests that $|\hat{\gamma}_j^t - \alpha| \le \eta_j$. In the case where $\hat{\gamma}_j^t > \alpha$, we have $\hat{\gamma}_j^t - \alpha \le \eta_j$ and, therefore,

$$(\hat{\gamma}_j^t - \alpha)(\beta_j - \lambda_j^t) = (\hat{\gamma}_j^t - \alpha)^+ \beta_j - (\hat{\gamma}_j^t - \alpha)\lambda_j^t \ge (\hat{\gamma}_j^t - \alpha)^+ \beta_j - \eta_j \lambda_j^t \ge (\hat{\gamma}_j^t - \alpha)^+ \beta_j - \eta_j,$$

where the first inequality follows from $\hat{\gamma}_j^t - \alpha \le \eta_j$, and the second from $\lambda_j^t < \beta_j \le 1$. In the case where $\hat{\gamma}_j^t \le \alpha$, we have

$$(\hat{\gamma}_j^t - \alpha)(\beta_j - \lambda_j^t) = (\hat{\gamma}_j^t - \alpha)^+ \beta_j - (\alpha - \hat{\gamma}_j^t)(\beta_j - \lambda_j^t) \ge (\hat{\gamma}_j^t - \alpha)^+ \beta_j - \eta_j \beta_j \ge (\hat{\gamma}_j^t - \alpha)^+ \beta_j - \eta_j,$$

where the first equality follows from $(\hat{\gamma}_j^t - \alpha)^+ = 0$, the first inequality from $0 \le \beta_j - \lambda_j^t \le \beta_j$ and $0 \le \alpha - \hat{\gamma}_j^t \le \eta_j$, and the second inequality from $\beta_j \in [0,1]$. Therefore, $(\hat{\gamma}_j^t - \alpha)(\beta_j - \lambda_j^t) \ge \beta_j(\hat{\gamma}_j^t - \alpha)^+ - \eta_j$ for all $j \in A$ and, hence, inequality (18) follows.

Finally, we evaluate $\mathbb{E}[\tau(x_t, a_t)|\mathcal{H}_t]$ and bound the terms $\mathsf{OPT}^{\mathsf{t}}$ and $(\hat{\gamma}_j^t - \alpha)^+$. Consider two cases: (a) $|\hat{c}_{ij}^t - c_{ij}^t| = O\left(t^{-\frac{1}{3}} (\log t)^{\frac{1}{3}} K^{\frac{1}{3}} d^{\frac{1}{2}}\right)$ for all j, which occurs with probability at least $1 - t^{-3}$ (see the discussions before Lemma 3); and (b) $|\hat{c}_{ij}^t - c_{ij}^t| \ne O\left(t^{-\frac{1}{3}} (\log t)^{\frac{1}{3}} K^{\frac{1}{3}} d^{\frac{1}{2}}\right)$ for some $j \in A$, which occurs with probability less than t^{-3}.

We first consider the case where $|\hat{c}_{ij}^t - c_{ij}^t| = O\left(t^{-\frac{1}{3}} (\log t)^{\frac{1}{3}} K^{\frac{1}{3}} d^{\frac{1}{2}}\right)$ for all ad j (which occurs with probability at least $1 - t^{-3}$). It follows from the definition of OPT (see Lemma 1) that

$$\mathsf{OPT} = \min_{0 \le \lambda_j \le \beta_j, \forall j \in A} \sum_{i \in \mathcal{I}} p_i \max_{j=1,2,\ldots,K} \left(c_{ij}(b_j + \lambda_j)\right) + \alpha \sum_{j=1}^{K} (\beta_j - \lambda_j).$$

Because $|\hat{c}_{ij}^t - c_{ij}| = O\left(t^{-\frac{1}{3}} (\log t)^{\frac{1}{3}} K^{\frac{1}{3}} d^{\frac{1}{2}}\right)$, by the definitions of OPT and OPT^t, we have

$$|\mathsf{OPT}^t - \mathsf{OPT}| \le O\left(t^{-\frac{1}{3}} (\log t)^{\frac{1}{3}} K^{\frac{1}{3}} d^{\frac{1}{2}}\right). \tag{19}$$

Similarly, we bound $\mathbb{E}[\tau(x_t, a_t)|\mathcal{H}_t]$ by

$$\mathbb{E}[\tau(x_t, a_t)|\mathcal{H}_t] = \sum_{j=1}^{K} \sum_{i \in N_j^t} p_i c_{ij}(b_j + \beta_j) \ge \sum_{j=1}^{K} \sum_{i \in N_j^t} p_i \hat{c}_{ij}^t (b_j + \beta_j) - O\left(t^{-\frac{1}{3}} (\log t)^{\frac{1}{3}} K^{\frac{1}{3}} d^{\frac{1}{2}}\right). \tag{20}$$

Furthermore, by Jensen's inequality, we observe that

$$(\hat{\gamma}_j^t - \alpha)^+ \ge \left(\gamma_j^t - \alpha - O\left(t^{-\frac{1}{3}} (\log t)^{\frac{1}{3}} K^{\frac{1}{3}} d^{\frac{1}{2}}\right) \sum_{i \in N(j)} p_i\right)^+ \ge (\gamma_j^t - \alpha)^+ - O\left(t^{-\frac{1}{3}} (\log t)^{\frac{1}{3}} K^{\frac{1}{3}} d^{\frac{1}{2}}\right). \tag{21}$$

Collecting the terms of (17), (18), (19), (20), and (21) above, we have that

$$\begin{aligned} \mathbb{E}\left[\tau(x_t, a_t) \middle| \mathcal{H}_t\right] \ge \mathsf{OPT} + \sum_{j=1}^{K} \beta_j (\gamma_j^t - \alpha)^+ - O\left(t^{-\frac{1}{3}} (\log t)^{\frac{1}{3}} K^{\frac{1}{3}} d^{\frac{1}{2}}\right) - O\left(T^{-\frac{1}{3}} (\log T)^{\frac{1}{3}} K^{\frac{1}{3}}\right) \\ - \mathbb{E}_{i \sim \mathcal{D}_X}\left[s_t(\lambda) + \frac{2\eta}{\sigma} + \frac{1}{\eta} D_\varphi(\lambda, \lambda^t) - \frac{1}{\eta} D_\varphi(\lambda, \lambda^{t+1})\right]. \end{aligned} \tag{22}$$

For the case $|\hat{c}_{ij}^t - c_{ij}^t| \ne O\left(t^{-\frac{1}{3}} (\log t)^{\frac{1}{3}} K^{\frac{1}{3}} d^{\frac{1}{2}}\right)$, which occurs with probability less than t^{-3}, we can bound the expected gap between $\mathbb{E}[\tau(x_t, a_t)]$ and OPT by $O(t^{-3})$, which is a lower order term compared to the gap

in the case where $|\hat{c}_{ij}^t - c_{ij}^t| = O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$. Integrating over the distribution of history $\mathcal{H}_t$, we have established inequality (16) for the SBL-DMD algorithm.

To **part (a)**, we adopt the same argument as the proof of inequality (18) together with Lemma 2(a) to show that, under the SBL-RS algorithm, the following inequality holds:

$$\sum_{j=1}^{K}(\hat{\gamma}_j^t - \alpha)(\beta_j - \lambda_j^t) \geq \sum_{j=1}^{K}\beta_j(\hat{\gamma}_j^t - \alpha)^+ - O\left(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}\right). \tag{23}$$

Hence, for the case $|\hat{c}_{ij}^t - c_{ij}^t| = O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$ for all ad j, the inequalities (17), (23), (19), (20), and (21) together imply that

$$\mathbb{E}\left[\tau(x_t,a_t)\middle|\mathcal{H}_t\right] \geq \mathsf{OPT} + \sum_{j=1}^{K}\beta_j(\gamma_j^t - \alpha)^+ - O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right) - O\left(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}\right).$$

Therefore, similar to the proof of part (a), integrating over $\mathcal{H}_t$ will establish inequality (15) for the SBL-RS algorithm. □

Summing inequality (15) over the whole T periods (i.e., $t = 1, 2, ..., T$) and invoking Jensen's Inequality, we have the following regret bound on the expected auxiliary reward process $\mathbb{E}\left[\sum_{t=1}^{T}\tau(x_t,a_t)\right]$ under the SBL-RS algorithm:

$$\mathbb{E}\left[\sum_{t=1}^{T}\tau(x_t,a_t)\right] \geq T\cdot\mathsf{OPT} + \mathbb{E}\left[\sum_{j=1}^{K}\beta_j\left(\sum_{t=1}^{T}\gamma_j^t - \alpha T\right)^+\right] - O\left(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right). \tag{24}$$

Similarly, summing inequality (16) over the whole T periods and invoking Jensen's Inequality implies that under the SBL-DMD algorithm:

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{T}\tau(x_t,a_t)\right] \geq T\cdot\mathsf{OPT} + \mathbb{E}\left[\sum_{j=1}^{K}\beta_j\left(\sum_{t=1}^{T}\gamma_j^t - \alpha T\right)^+\right] - O\left(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)\\ - \mathbb{E}\left[\sum_{t=1}^{T}s_t(\lambda)\right] - \frac{2\eta}{\sigma}T - \frac{1}{\eta}D_\varphi(\lambda,\lambda^1).\end{aligned} \tag{25}$$

Next, we bound the difference between the auxiliary reward process $\tau(x_t,a_t)$ and the true reward process $r(x_t,a_t)$. Denote the total number of clicks for ad j in all T rounds as $n(j)$.

LEMMA 4 **(Difference between Two Reward Processes)**. *Under both SBL-RS and SBL-DMD algorithms, it holds that*

$$\mathbb{E}\left[\sum_{j=1}^{K}\left(n(j) - \sum_{t=1}^{T}\gamma_j^t\right)^+\right] \leq O\left(\sqrt{KT\log T}\right). \tag{26}$$

Proof of Lemma 4.

It suffices to establish a high probability bound: With probability at least $1 - T^{-4}$, the following inequality holds:

$$\sum_{j=1}^{K}\left(n(j) - \sum_{t=1}^{T}\gamma_j^t\right)^+ \leq O\left(\sqrt{KT\log T}\right). \tag{27}$$

We now show that for any subset of ads, denoted by $\mathcal{S}$,

$$\sum_{j\in\mathcal{S}}n(j) - \sum_{j\in\mathcal{S}}\sum_{t=1}^{T}\gamma_j^t \leq O\left(\sqrt{KT\log T}\right) \text{ with probability at least } 1 - T^{-4}. \tag{28}$$

Note that given the history by round t, $\mathcal{H}_t$, the expected total number of clicks for ad j is given by γ_j^t. One can use Azuma-Hoeffding inequality to show that for a fixed subset $\mathcal{S}$, we have with probability at most T^{-4K},

$$\sum_{j\in\mathcal{S}} n(j) - \sum_{j\in\mathcal{S}}\sum_{t=1}^{T}\gamma_j^t \geq O\left(\sqrt{KT\log T}\right).$$

Take a union bound over all subsets and notice that $2^K T^{-4K} \leq T^{-4}$. Hence, (28) holds with probability at least $1-T^{-4}$.

We now show (27) by contradiction. Suppose that (27) does not hold with probability at least T^{-4}. Define $\mathcal{S}'$ as the set of ads such that $n(j) > \sum_{t=1}^{T}\gamma_j^t$. It follows that

$$\sum_{j\in\mathcal{S}'} n(j) - \sum_{j\in\mathcal{S}'}\sum_{t=1}^{T}\gamma_j^t = \sum_{j\in\mathcal{S}'}\left(n(j)-\sum_{t=1}^{T}\gamma_j^t\right) = \sum_{j=1}^{K}\left(n(j)-\sum_{t=1}^{T}\gamma_j^t\right)^+ > O\left(\sqrt{KT\log T}\right),$$

with probability at least T^{-4}, which contradicts inequality (28). Thus, inequality (27) holds with probability at least $1-T^{-4}$. Finally, we take the expectation of (27) and the inequality (26) follows immediately. □

We are now ready to prove Theorem 1.

Proof of Theorem 1.

First, we prove **part (a)** for the SBL-DMD algorithm. Note that

$$\begin{aligned}\sum_{t=1}^{T}\tau(x_t,a_t) &= \sum_{t=1}^{T} r(x_t,a_t) + \sum_{j=1}^{K}\beta_j\left(n(j)-\alpha T\right)^+ \\ &\leq \sum_{t=1}^{T} r(x_t,a_t) + \sum_{j=1}^{K}\beta_j\left(n(j)-\sum_{t=1}^{T}\gamma_j^t\right)^+ + \sum_{j=1}^{K}\beta_j\left(\sum_{t=1}^{T}\gamma_j^t-\alpha T\right)^+, \end{aligned} \tag{29}$$

where the inequality follows from $(X+Y)^+ \leq X^+ + Y^+$ for any $X, Y \in \mathbb{R}$. Putting the inequalities (25), (26), and (29) together, we obtain, for the exploitation rounds of the SBL-DMD algorithm,

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{T} r(x_t,a_t)\right] &\geq \mathbb{E}\left[\sum_{t=1}^{T}\tau(x_t,a_t)\right] - \mathbb{E}\left[\sum_{j=1}^{K}\beta_j\left(n(j)-\sum_{t=1}^{T}\gamma_j^t\right)^+\right] - \mathbb{E}\left[\sum_{j=1}^{K}\beta_j\left(\sum_{t=1}^{T}\gamma_j^t-\alpha T\right)^+\right] \\ &\geq \mathbb{E}\left[\sum_{t=1}^{T}\tau(x_t,a_t)\right] - O\left(\sqrt{KT\log T}\right) - \mathbb{E}\left[\sum_{j=1}^{K}\beta_j\left(\sum_{t=1}^{T}\gamma_j^t-\alpha T\right)^+\right] \\ &\geq T\cdot\mathsf{OPT} - O\left(\sqrt{KT\log T}\right) - O\left(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right) - \mathbb{E}\left[\sum_{t=1}^{T}s_t(\lambda)\right] - \frac{2\eta}{\sigma}T - \frac{1}{\eta}D_\varphi(\lambda,\lambda^1) \\ &= T\cdot\mathsf{OPT} - O\left(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right) - \mathbb{E}\left[\sum_{t=1}^{T}s_t(\lambda)\right] - \frac{2\eta}{\sigma}T - \frac{1}{\eta}D_\varphi(\lambda,\lambda^1).\end{aligned}$$

For the last equality, we drop the regret term $\tilde{O}(\sqrt{KT})$ which might dominate when K is large. However, in the online advertising practice, T is several orders of magnitude higher than K. For Platform O, the number of ad impressions in the cold-start period T is 10,000 times larger than the number of ads K. Hence, $\tilde{O}(\sqrt{KT})$ is negligible compared with $\tilde{O}(T^{\frac{2}{3}}K^{\frac{1}{3}})$ and can be safely dropped.

Next, we relax the assumption that p_i is known to the algorithm for each i by demonstrating that observing $\hat{p}_i$ only will only incur an additional regret of an order lower than $O\left(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$. We show that using the empirical probability $\hat{p}_i$ (instead of the true one p_i) only incurs an additional regret of an order lower than

the one in Lemma 3(b), i.e., $O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)+O\left(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}\right)$. Note that, by Assumption 2, the estimate satisfies $|\hat{c}_{ij}^t - c_{ij}^t| = O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$ for all ads with probability at least $1-t^{-3}$. Furthermore, the empirical distribution estimate $\hat{p}_i^t$ satisfies that with probability at least $1-t^{-4}$, for any context $i\in\mathcal{I}$, we have $|\hat{p}_i^t - p_i| \le O(\sqrt{\log t/t})$ (see the discussions in Appendix B.2). Combining the above two error estimation bounds on $\hat{c}_{ij}^t$ and $\hat{p}_j$, we have, by the definitions of OPT^t and OPT,

$$\mathsf{OPT}^t \ge \mathsf{OPT} - O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right) - O\left(t^{-\frac{1}{2}}(\log t)^{\frac{1}{2}}\right) - O\left(t^{-\frac{5}{6}}(\log t)^{\frac{5}{6}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right) = \mathsf{OPT} - O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right). \tag{30}$$

Hence, Lemma 3(b) and inequality (16) continue to hold if we replace p_i with $\hat{p}_i^t$ for each context i and each round t. The rest of Theorem 1's proof remains the same. Summarizing our argument above, we have shown that the expected regret of the SBL-DMD algorithm is bounded by

$$O\left(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)+\mathbb{E}\left[\sum_{t=1}^{T}s_t(\lambda)\right]+\frac{2\eta}{\sigma}T+\frac{1}{\eta}D_\varphi(\lambda,\lambda^1),$$

i.e., part (b) holds.

Finally, we prove **part (a)** for the SBL-RS algorithm. We follow exactly the same argument as the proof for part (b) and derive, from inequalities (24), (26), and (29), that, under SBL-RS,

$$\mathbb{E}\left[\sum_{t=1}^{T}r(x_t,a_t)\right] \ge T\cdot\mathsf{OPT} - O\left(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$$

Together with (30), we have, for $\tau_m - \tau_{m-1} = 1$, the expected regret of the SBL-RS algorithm is bounded by

$$O\left(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right),$$

i.e., part (a) holds if the algorithm re-solves the empirical dual in each period.

We next show that it suffices to solve the empirical dual problem with a fixed epoch of size $O(T^{\frac{2}{3}})$. Since the regret bound is of order $\tilde{O}(T^{\frac{2}{3}})$, we can discard the first $T^{\frac{2}{3}}$ periods without affecting the order of the regret bound. After the first $T^{\frac{2}{3}}$ periods, with a fixed epoch schedule such that $\tau_{m+1}-\tau_m = O(T^{\frac{2}{3}})$, we have $\tau_m \ge (1/2)\tau_{m+1}$. Therefore, at round t the additional regret it incurs to solve the empirical dual program is at most a constant multiplication of $O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$, which is still of order $O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$. Therefore, summing this bound over t from 1 to T, we have that the total additional regret from setting $\tau_m - \tau_{m-1} = O(T^{\frac{2}{3}})$ is of order $O\left(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$. This completes the proof of Theorem 1(a). □

B.4. Online Mirror Descent

To make our paper self-contained, we present a standard result on online mirror descent, which is inherited from Proposition 5 of Balseiro et al. (2021).

PROPOSITION 1 **(Online Mirror Descent).** *With the sequence of convex functions*

$$s_t(\lambda) = -\sum_{j\in[K]\setminus a_t}\alpha\lambda_j + (\hat{c}_{ia_t}^t - \alpha)\lambda_{a_t},$$

let $z_t \in \partial_\lambda s_t(\lambda)$ be a subgradient and

$$\lambda^{t+1} = \operatorname*{arg\,min}_{0\le\lambda_j\le\beta_j,\forall j\in A}\langle z_t,\lambda\rangle + \frac{1}{\eta}D_\varphi(\lambda,\lambda^t).$$

By the definition of $s_t(\lambda)$*, the subgradients are bounded by* $||z_t||_\infty \leq \alpha + 1 \leq 2$*. Suppose the reference function* φ *is* σ*-strongly convex with respect to* L_1*-norm. Then, for any* $\lambda_j \in [0, \beta_j]$ $(1 \leq j \leq K)$*, we have*

$$s^t(\lambda^t) - s^t(\lambda) \leq \frac{2\eta}{\sigma} + \frac{1}{\eta} D_\varphi(\lambda, \lambda^t) - \frac{1}{\eta} D_\varphi(\lambda, \lambda^{t+1}),$$

and

$$\sum_{t=1}^{T} [s^t(\lambda^t) - s^t(\lambda)] \leq \frac{2\eta}{\sigma} T + \frac{1}{\eta} D_\varphi(\lambda, \lambda^1).$$

The proof of Proposition 1 mainly follows from the first order conditions and the Three-Point Property of the Bregman projection. We refer interested readers to Proposition 5 in Balseiro et al. (2021) for proof details.

Appendix C: Additional Empirical Analysis

In this section, we present the following additional empirical analysis: (a) Validation for the causality of Figure 1; (b) randomization check for our field experiment; (c) verification of SUTVA for the experiment; and (d) regression analysis as robustness checks for the short-term impact of oSBL (i.e., Table 2).

C.1. Validation for the Causality of Figure 1

In this section, we casually estimate the effect of cold start performance on ad retention using two different methods. We first conduct a propensity score matching (PSM) analysis using pre-experiment data. Specifically, we access the data of all new ads created between May 1, 2020 and May 7, 2020, and examine their performance before our two-sided experiment which started on May 23, 2020. Specifically, we use PSM to construct the treatment and control groups of ads. The treatment (resp. control) group of ads correspond to those ads whose conversions is greater or equal to (resp. below) 10 during their cold start period. We include all potential confounding variables we are aware of, such as bidding price, budget, industry, and target strategy, to match the control sample with the treatment sample using logistic regression. Note here the target strategy is set by the advertiser before the ad campaign, which (based on age, gender, location, phone brand and so on) chooses a subset of eligible platform users for displaying ads.

There are in total 97,273 new ads created between May 1, 2020 and May 7, 2020. We first construct a balanced subsample of 22,994 new ads in the treatment and control groups with PSM. Then, assuming that the data sample constructed by PSM satisfies the conditional independence assumption (CIA), we run a regression to estimate the causal effect of obtaining at least 10 conversions during the cold start period. See Chapter 3 of Angrist and Pischke (2008) for details. We then run a linear regression with the indicator variable for obtaining at least 10 conversions as the treatment variable and controlling for other features used in matching, i.e.,

$$\text{PSM}: y_j = \beta_0 + \beta_1 \text{Treatment}_j + X_j + \epsilon_j,$$

where y_j corresponds to whether ad j is retained in the two week course after its cold-start period, Treatment_j corresponds to whether ad j is the treatment group, and X_j are the features. Our regression result shows that if a new ad gains more than 10 conversions during the cold start period, the retention rate is significantly increased by 15.03% (p-value<0.0001), which is reported in the column (2) of Table 5. Thus, we have partially established the causality for Figure 1 that reaching the cold-start conversion threshold during the first few

days could significantly boost the retention rate of an ad. We also report the naive regression result with PSM, i.e.,

$$\text{Linear Regression}: y_j = \beta_0 + \beta_1 \text{Treatment}_j + \epsilon_j,$$

in column (1) of Table 5 for comparison. Note that this corresponds to a straightforward t-test.

One may still worry whether some other unobservable confounding variables may invalidate the above matching-based result. To further justify the validity of Figure 1, we leverage our experimental data to casually estimate the effect of cold-start success on ad retention. Specifically, we use our two-sided experiment as an instrumental variable (IV) to identify the effect of whether a new ad gaining more than 10 conversions during the cold start period on its retention (encoded as the binary variable representing whether the ad remains active on the platform for everyday in the following two weeks after the cold start period). We adopt the two-stage least squares (2SLS) specification given by (31).

$$\begin{aligned} &\text{IV-First Stage: } s_j = \alpha_0 + \alpha_1 \text{Treatment}_j + X_j + \epsilon_j \\ &\text{IV-Second Stage: } y_j = \beta_0 + \beta_1 \hat{s}_j + X_j + \epsilon_j \end{aligned} \tag{31}$$

To estimate the impact of cold start success on ad retention, we denote $s_j = 1$ if ad j gains more than 10 conversions during the experiment; otherwise $s_j = 0$. X_j is ad-specific features, such as bidding prices, budget, industry, and target strategy, for ad j. $\text{Treatment}_j = 1$ if ad j is in the treatment group; otherwise $\text{Treatment}_j = 0$. We use $y_j = 1$ to denote that ad j stays on the platform in the next two weeks after the cold-start period; otherwise $y_j = 0$. We remark that Treatment is a valid instrument in this setting. On the one hand, the p-values of the weak instrument tests are smaller than 10^{-5} so the strong first-stage assumption holds. On the other hand, it seems unlikely that our experiment could impact the retention of an ad through a channel other than conversions, so exclusion restriction also holds.

Under the 2SLS specification (31), we further validate the causality of Figure 1 and demonstrate that gaining 10 conversions during the cold start period will significantly increase the ad retention rate by 15.20% (p-value is less than 0.0001), which is reported in the column (3) of Table 5. Note this result effectively matches that in the column (2). We also remark that the relative effect sizes reported in Table 5 are not directly comparable with that in line 1 of Table 3 Panel A. This is because the former uses the binary variable representing whether the ad is retained as the outcome variable, whereas the latter uses the number of active days in the ad's life time as the outcome variable.

C.2. Randomization Check of the Field Experiment

To confirm the success of our randomization in the two-sided experiment, we check the randomization on both the ad side and the UV side before oSBL coming into effect. For the ad side randomization check, we report the ad side randomization check results in Table 6 Panel A, where the numbers are re-scaled to protect the sensitive data. Table 6 Panel A shows that treatment and control ads in our sample were similar in bidding prices, the proportion of ads targeting iOS users, the proportion of ads targeting UI Version X, and the proportion of ads in various industries. We remark that these features are all submitted by the advertiser once s/he launches a new ad on the DSP and, therefore, are not affected by the algorithm of choice. Similarly, the UV side randomization check results are reported in Table 6 Panel B, where the numbers are

Table 5 Effect of Cold Start Success on Ad Retention

	Methodology:		
	Benchmark	Matching	Instrumental Variable
	(1)	(2)	(3)
Absolute Effect Size	17.01%****	15.03%****	15.20%****
Standard Error	(0.004)	(0.006)	0.003
Experimental Data	No	No	Yes
Observations	97,273	22,994	49,544
Industry Fixed Effects	No	Yes	Yes
Bidding Price	No	Yes	Yes
Budget	No	Yes	Yes
Target Strategy	No	Yes	Yes

Note: *p<0.1; **p<0.01; ***p<0.001; ****p<0.0001.

Table 6 Randomization Check of the Experiment

Panel A: Randomization Check on the Ad side

		Treatment ads	Control ads	p-value of t-test
Statistics during the Experiment	Number of New Ads	34,605	34,076	
	Bidding Price	48.14	48.17	0.91
		(52.24)	(51.45)	
	Proportion of Ads for iOS Users	24.1%	24.2%	0.98
		(0.427)	(0.428)	
	Proportion of Ads for UI Version X	30.3%	28.3%	0.69
		(0.459)	(0.450)	
	Proportion of Ads in Game Industry	13.8%	13.7%	0.98
		(0.086)	(0.081)	
	Proportion of Ads in Education Industry	0.75%	0.67%	0.93
		(0.086)	(0.082)	
	Proportion of Ads in Finance Industry	1.75%	1.87%	0.93
		(0.131)	(0.135)	

Panel B: Randomization Check on the UV side

		Treatment UV	Control UV	p-value of t-test
Statistics during the Experiment	Number of Users	197,460,792	197,401,621	
	Male Proportion	0.540	0.540	>0.99
		(0.491)	(0.491)	
	Average Revenue per User	0.95	0.95	>0.99
		(27.15)	(27.14)	
	Average Impressions per User	23.36	23.24	0.95
		(17900)	(17864)	
	Average Clicks per User	3.195	3.20	0.99
		(4455)	(4458)	
	Average Conversions per User	0.041	0.040	0.88
		(32.80)	(32.25)	

Note: Standard deviations in Panel A are clustered at the ad level and reported in the parentheses. Standard deviations in Panel B are clustered at the user level and reported in the parentheses. To protect sensitive data, the reported metrics are rescaled.

also rescaled to protect the sensitive data. As we can see from Table 6 Panel B, treatment UVs and control UVs generate similar revenues, ad impressions, ad clicks, and ad conversions per hour. The proportion of female users in the treatment group is also similar to that in the control group. We have thus confirmed that the treatment ads (resp. UVs) and control ads (resp. UVs) in our sample are comparable, implying that

any difference between groups after the experiment started should be attributed to whether our new oSBL algorithm has been implemented.

C.3. Verification of SUTVA for the Experiment

To verify SUTVA for our two-sided experiment, we examine two additional assumptions during our experiment time period: (a) The CTR and CVR distributions of the mature ads are not affected by the cold start algorithms applied to different UVs; and (b) The total number of ad impressions displayed to a user is not affected by the cold start algorithms applied to different ads. To test the first assumption, we sample 13,337 mature ads one day before the experiment and compare their empirical CTR before and during the experiment. The average CTR before the experiment is 13.11% with standard deviation 0.099, while the average CTR during the experiment is 13.19% with standard deviation 0.100. The p-value of the pairwise t-test is 0.284 and 0.481 for CTR and CVR, respectively, implying that our algorithm does not substantially change the CTR and CVR of mature ads *during* the experiment. To test the second assumption, we conduct a t-test of average ad impressions per user for the treatment and control ad impressions in our experiment. We find that the p-value is 0.96. Hence, our algorithm does not change the number of ad impressions significantly. Therefore, SUTVA holds for our two-sided experiment.

C.4. Robustness Check for Regression-Based Results

In this subsection, we replicate our main results for the short-term impact of oSBL (i.e., Table 2) using the following linear regression specification which controls for the ad features to improve the efficiency of our estimators:

$$\text{Performance Indicator}_j = \alpha_0 + \alpha_1 \text{Treatment}_j + X_j + \epsilon \tag{32}$$

For the impact of the new algorithm on cold start reward and cold start success rate, Treatment_j is 1 if ad j is in the treatment group, otherwise 0; X_j is ad-specific features including the industry category (of the advertiser), bidding price, budget, and the to target strategy. The target strategy means that advertisers can predetermine whom to display the ad based on users' age, gender, location, phone, device features, and so on. We use the specification (32) to check the robustness of results on the revenue implications of our oSBL algorithm, where Treatment_j is 1 if ad j is in the treatment group, otherwise 0; For conciseness, we only report the three most important metrics of the platform, namely the cold start success rate and the cold start reward, as well as the revenue. The result of specification (32) is presented in Table 7. The regression-based results indicate that, after controlling for ad-specific characteristics, our algorithm can significantly increase the cold start success rate by 41.22% and the cold start reward by 53.87%, which are similar to the model-free results on the short-term impact of oSBL (see Section 6.1) in both directions and magnitudes.

Appendix D: Additional Simulation Analysis

In this section, we first introduce the details of the simulation system, the code and data of which is released to the public on GitHub.[13] Presenting the simulation system helps us better explain in detail our algorithm and experiment (see Appendix D.1). Then, we present comprehensive simulation analysis to complement

[13] See https://github.com/zikunye2/cold_start_to_improve_market_thickness_simulation for details.

Table 7 Regression-Based Effects of the Algorithm

	Dependent Variable:			
	Cold Start Reward	Cold Start Success Rate	Revenue Per User	Objective Value
	(1)	(2)	(3)	(4)
Treatment−Control	73.0	0.0146	-0.0072	556,607
	(5.04)	(<0.001)	(<0.001)	
Relative Effect Size	41.22%****	53.87%****	-0.592%**	0.157%****
Model-Free Relative Effect Size	47.71%****	61.62%****	-0.717%**	0.147%****
Industry Fixed Effects	Yes	Yes	Yes	Yes
Bidding Price	Yes	Yes	Yes	Yes
Budget	Yes	Yes	Yes	Yes
Target Strategy	Yes	Yes	Yes	Yes

Note: *p<0.1; **p<0.01; ***p<0.001; ****p<0.0001. Standard errors in column (1) and (2) are at the ad level and reported in the parentheses. Standard error in column (3) is at the user level.

our theoretical and empirical results. In Appendix D.2, we quantify the estimation biases under single-sided experiments (see also Section 5). Appendix D.3 numerically illustrates the expected regret of our algorithm (Theorem 1). Appendix D.4 leverage the simulation model built in Section 6.3 to numerically show that for a wide range of cold start reward parameter β_j, our oSBL algorithm can substantially boost the long-term total advertising revenue of the platform.

D.1. Simulation System

To begin with, we describe the simulation model built upon on the online advertising practice and real data of Platform O. Interested readers are referred to our GitHub repository (see Footnote 13) for the data inputs and implementation details of the simulation system. The goal of open-sourcing this simulation system is to help scholars interested in our paper better understand the online advertising platforms in practice, the implementation details of our algorithm and the two-sided experiment, and how our two-sided experiment framework enables to reduce the estimation bias under the violation of SUTVA.

Advertising System and Cold Start Setting. To capture the core advertising mechanism without getting trapped in engineering details, our simulator assumes the CPC billing option and first-price auction, and adopts a threshold on the number of clicks, instead of conversions, as our cold start target (i.e., αT in our theoretical model). Analogously, we convert each ad's cost-per-action (CPA) to the corresponding CPC by multiplying the average conversion rate of ad. The hyper-parameters of our simulation system, including the cold start target parameter α, cold start reward β_j's, the number of user impressions during the cold start period T, the proportion of new and mature ads on the DSP, and the budget of each ad, are all fine-tuned to provide a close approximation of the real DSP on Platform O. The exact values of these parameters are provided in the GitHub repository (see Footnote 13). Readers may choose their own hyper-parameters as appropriate in the problem context they are studying.

Ground-Truth CTR Model and Data Inputs: The simulator is equipped with a "ground-truth" click-through model for each ad, which is assumed to be a (linear) fixed-effects models constructed with the

historical user impression and click-through data. Specifically, when displaying ad j to a user impression i, the user will click through the ad with probability

$$\mathsf{CTR}_{i,j} = \mathsf{User\text{-}Invariant\text{-}CTR}_j + \mathsf{Coefficient\text{-}Vector}_j^T \times \mathsf{Feature}_i. \tag{33}$$

As is clear from ground-truth data-generating process (33), the CTR for ad j and user impression i consists of two parts, one capturing the ad fixed-effect $\mathsf{User\text{-}Invariant\text{-}CTR}_j$ and the other capturing the impact of user feature $\mathsf{Coefficient\text{-}Vector}_j^T \times \mathsf{Feature}_i$. The user-invariant CTR only depends on ad j and is independent of the feature of the user i. As parameter inputs for our simulation system, we randomly sample 300 data points from the joint distribution of bid prices (which we linearly scale from CPC using a random multiplier) and the user-invariant ad CTR based on the real data of Platform O.[14] To include the user-feature related information into the ground truth CTR model (i.e., the second term of Eq. (33)), we incorporate the demographic feature of each user i, $\mathsf{Feature}_i$, sampled from the real marginal distribution of user gender, location, and age on Platform O. The gender feature follows a binary distribution on *male*, and *female* with equal probability. The location feature follows a discrete distribution on *large city*, *medium city*, and small city with probability densities equal to 0.22, 0.46, and 0.32, respectively. The age feature follows a discrete distribution on *young*, *mid-age*, and *old* with probability densities equal to 0.46, 0.34, and 0.20, respectively. When generating the feature of each user view in the simulation, we assume the above three features (*gender*, *age*, and *location*) are independent. This is consistent with the practice of Platform O, and it is straightforward to adjust the simulation system that accounts for dependent user features. For each ad j, the 3-dimensional ad-specific coefficient vector for user-features (i.e., $\mathsf{Coefficient-Vector}_j$ in Eq. (33)) is randomly sampled from a uniform distribution on the support $[-0.5, 0.5] \times [-0.5, 0.5] \times [-0.5, 0.5]$. As before, interested readers may change the ground-truth CTR model and/or its data inputs in accordance to the specific problem they work on.

Machine Learning Oracle to Predict CTR: As discussed in Section 4, an indispensable infrastructure for a DSP in practice is the machine learning system (usually DNNs) to generate the pCTR for each ad j and user i. To simulate the ML oracle for predicting CTRs, we distinguish the new ads from the mature ones. For a mature ad j, the ML oracle has already accurately learned the true user-invariant CTR $\mathsf{User\text{-}Invariant\text{-}CTR}_j$ and the true ad-specific $\mathsf{Coefficient-Vector}_j$. In this case, the predicted CTR is identical to the true CTR, i.e., $\mathsf{pCTR}_{i,j} = \mathsf{CTR}_{i,j}$ for any user i. If ad j is new, however, the ground-truth model parameters are unknown to the ML oracle. In this case, the ML oracle initialize the prediction algorithm with $\mathsf{Coefficient\text{-}Vector}_j \leftarrow (0,0,0)'$, and

$$\mathsf{User\text{-}Invariant\text{-}CTR}_j \leftarrow 0.5 \times \text{ground-truth } \mathsf{User\text{-}Invariant\text{-}CTR}_j + 0.5 \times \text{average CTR of all ad impressions},$$

which approximates the ML system on a real DSP in practice which uses the average CTR of the ad category as the initial pCTR. Without loss of generality, one can fine-tune this hyper-parameter in the simulation system to strengthen its CTR prediction. To train the CTR prediction model (33) and produce a pCTR upon

[14] This data sample is provided at https://github.com/zikunye2/cold_start_to_improve_market_thickness_simulation/blob/main/ctr_bid_data.npy.

the arrival of each user impression, we adopt the online stochastic gradient descent algorithm to minimize the (empirical) mean squared loss.

We remark that the real ML system of a DSP to predict CTRs and CVRs usually comprise of very large-scale DNNs leveraging millions of features which are typically sparse embeddings of both the ads and users (see, e.g., Covington et al. 2016). For data security and clarity reasons, we are unable to publicize such detailed individual user- and ad- level data of Platform O. Instead, we take the above compromised route to simulate the ground-truth CTR by combining ad-specific user-invariant CTRs and the linear dependence on user features.

Ad Delivery Algorithm: As discussed in Section 3.1, the core logic for the ad delivery algorithm of a DSP is, upon the arrival of a user impression, to display the ad with the highest eCPM, which thus maximizes the expected advertising revenue from this ad impression. In our simulation model, the eCPM of displaying ad j to user i is given by $\mathsf{eCPM}_{i,j} = \mathsf{pCTR}_{i,j} \times b_j$. Following the practice of Platform O, in addition to maximizing the eCPM of each ad impression, the baseline benchmark ad delivery algorithm for the simulation system also adopts the PID controller (see Appendix H.2, Eq. (41)) to adaptively adjusts the bidding prices during the cold start period. Specifically, for each new ad, the algorithm uniformly increases its bidding price to the upper-bound and then adopts the PID controller Eq. (41) to adaptively adjust the real-time bidding prices until the end of the cold start period of the ad. The hyper-parameters of the PID system, (k_p, k_i, k_d) in Eq. (41), are fine-tuned to match the moments of the simulation system with those of the real DSP on Platform O. If our new SBL algorithm is adopted, we implement its SBL-DMD version for all new ads on the simulation system. For a mature ad, regardless of the cold start algorithm applied to new ads, the ad delivery algorithm keeps the real-time bidding prices the same as the CPC submitted by its advertiser at the beginning of the campaign, i.e., the bid of mature ad j remains b_j. Of course, the readers are free to implement other ad delivery algorithms for both new and mature ads in our simulation system.

Our Field Experiment: Finally, we introduce how our different experiment designs are implemented on the simulation system. In our simulation system, we implement 3 different experimental designs: (i) parallel simulations, (ii) one-sided (UV-side or ad-side) experiment, and (iii) two-sided experiment. We consider the estimates for 2 outcome variables: (i) cold start success rate and (ii) advertising revenue. For the parallel simulations, we run two simulations separately, one with all ads and UVs under the control condition and the other all under the treatment condition. Therefore, the difference between the respective outcomes of the 2 parallel simulations generate the "ground-truth" treatment effect of interest, which is the benchmark to evaluate the treatment effect estimation accuracy of other (one- and two- sided) experiment designs in our simulation system.

Treatment and Intervention of Experiments. As discussed in Section 5.1, one-sided experiment may randomize the subject on the UV-side or the ad-side. The implementation of UV-side, ad-side and two-sided experiments on the simulation system follows the design described in Section 5.1, in Figures 4 and 5 in particular. These 3 experiment designs can be formalized under a unified UV-ad pair framework. Denote the set of

all UVs as $\mathcal{U}$ and the set of all new ads as $\mathcal{A}$. We first randomly assign UVs into the non-overlapping treatment group $\mathcal{T}_u$ and control group $\mathcal{C}_u$. Analogously, the new ads are randomly and independently assigned into the non-overlapping treatment group $\mathcal{T}_a$ and control group $\mathcal{C}_a$. For the UV-side experiment design, $|\mathcal{T}_u| = |\mathcal{C}_u|$, $\mathcal{T}_a = \mathcal{A}$ and $\mathcal{C}_a = \emptyset$. For the ad-side experiment design, $|\mathcal{T}_a| = |\mathcal{C}_a|$, $\mathcal{T}_u = \mathcal{U}$ and $\mathcal{C}_u = \emptyset$. For the two-sided experiment design, $|\mathcal{T}_u| = |\mathcal{C}_u|$ and $|\mathcal{T}_a| = |\mathcal{C}_a|$. For all 3 experiment designs, the algorithmic intervention and treatment condition are applied at the UV-ad pair level, as illustrated by Figures 4 and 5. For any UV $i \in \mathcal{U}$ and ad $j \in \mathcal{A}$, the UV-ad pair will be under the treatment condition, which we denote as $(i,j) \in \mathcal{T}_{ua}$ if and only if both the UV and the ad are in the respective treatment groups, i.e., $i \in \mathcal{T}_u$ and $j \in \mathcal{T}_a$. Otherwise, $j \notin \mathcal{T}_a$ or $i \notin \mathcal{T}_u$, either the control condition will be applied to the UV-ad pair, which we denote as $(i,j) \in \mathcal{C}_{ua}$, or this pair will be completely blocked in the case of the two-sided experiment design (see Figure 5), which we denote as $(i,j) \in \mathcal{B}_{ua}$. For the (UV-side, ad-side and two-sided) experiment designs considered in this paper, the bidding strategy of any UV-ad pair under the treatment condition (i.e., $(i,j) \in \mathcal{T}_{ua}$), will follow the SBL-DMD algorithm (i.e., the bid of ad j on user impression i in period t is $b_j + \lambda_j^t$), whereas that of a UV-ad pair under the control condition (i.e., $(i,j) \in \mathcal{C}_{ua}$) will follow the benchmark PID-based algorithm introduced above. Finally, if the UV-ad pair is blocked in the two-sided experiment design, i.e., $(i,j) \in \mathcal{B}_{ua}$, the bid of ad j on user impression i will remain 0 throughout the cold start period.

Experiment Outcomes. To have a fair comparison with the ground-truth treatment effect produced by the parallel simulations, we emphasize that both outcome metrics of interest (i.e., the cold start success rate and advertising revenue) should be carefully scaled according to the UV traffic assigned into different experiment groups. For completeness, we introduce how to evaluate both metrics under the 3 experiment designs separately.

UV-side Experiment Design. For this design, we assume ω of UVs are assigned into the treatment and control groups respectively, i.e., $|\mathcal{T}_u| = |\mathcal{C}_u| = \omega|\mathcal{U}|$. The cold start success rate under the UV-side experiment design is measured as follows. For each new ad $j \in \mathcal{A}$, we denote by $V_j^{\mathcal{T}}$ (resp. $V_j^{\mathcal{C}}$) the total number of click-throughs of ad j by users in the treatment (resp. control) group. Then, we scale the cold start success threshold by the user traffic ratio of the treatment and control groups, i.e., ω. Hence, $\mathbb{I}_{\{V_j^{\mathcal{T}} \geq \omega\alpha T\}}$ (resp. $\mathbb{I}_{\{V_j^{\mathcal{C}} \geq \omega\alpha T\}}$) is the binary indicator for whether ad j is successfully cold started by treatment (control) UVs. Therefore, the cold start success rate of the treatment condition is

$$\frac{\sum_{j\in\mathcal{A}} \mathbb{I}_{\{V_j^{\mathcal{T}} \geq \omega\alpha T\}}}{|\mathcal{A}|},$$

whereas that of the control condition is

$$\frac{\sum_{j\in\mathcal{A}} \mathbb{I}_{\{V_j^{\mathcal{C}} \geq \omega\alpha T\}}}{|\mathcal{A}|}.$$

To evaluate the advertising revenue for the treatment and control conditions under the UV-side experiment design, we need to consider that generated by mature ads, the set of which we denote as $\mathcal{A}_m$. Therefore, the total revenue of the treatment condition is

$$\sum_{j\in\mathcal{A}\cup\mathcal{A}_m} b_j V_j^{\mathcal{T}},$$

whereas that of the control condition is

$$\sum_{j\in\mathcal{A}\cup\mathcal{A}_m} b_j V_j^{\mathcal{C}}.$$

Ad-side Experiment Design. For this design, the cold start success rate of the treatment condition is

$$\frac{\sum_{j\in\mathcal{T}_a}\mathbb{I}_{\{V_j\geq\alpha T\}}}{|\mathcal{T}_a|},$$

whereas that of the control condition is

$$\frac{\sum_{j\in\mathcal{C}_a}\mathbb{I}_{\{V_j\geq\alpha T\}}}{|\mathcal{C}_a|}.$$

The total revenue of the treatment condition is

$$\sum_{j\in\mathcal{T}_a} b_j V_j,$$

whereas that of the control condition is

$$\sum_{j\in\mathcal{C}_a} b_j V_j.$$

Two-sided Experiment Design. Finally, similar to the UV-side experiment design, the cold start success rate under the two-sided experiment design should be evaluated with the cold start success threshold re-scaled by the user traffic of treatment and control groups. Therefore, the cold start success rate of the treatment condition is

$$\frac{\sum_{j\in\mathcal{T}_a}\mathbb{I}_{\{V_j^{\mathcal{T}}\geq\omega\alpha T\}}}{|\mathcal{T}_a|},$$

whereas that of the control condition is

$$\frac{\sum_{j\in\mathcal{C}_a}\mathbb{I}_{\{V_j^{\mathcal{C}}\geq\omega\alpha T\}}}{|\mathcal{C}_a|}.$$

Under the two-sided experiment design, the total revenue should take into account that generated by the mature ads. Therefore, the revenue of the treatment condition is given by

$$\sum_{j\in\mathcal{T}_a\cup\mathcal{A}_m} b_j V_j^{\mathcal{T}},$$

whereas that of the control condition is given by

$$\sum_{j\in\mathcal{C}_a\cup\mathcal{A}_m} b_j V_j^{\mathcal{C}}.$$

Finally, as discussed in Section 5.1, blocking those treatment (resp. control) ads in auctions for control (resp. treatment) UVs in two-sided experiment design (see Figure 5) will reduce the competition in the auctions, which may result in overestimation of the cold start success rate for both the treatment and control conditions. This may not be a problem for a large-scale marketplace with ad targeting like Platform O (blocking 20% of the experimented new ads only decreases the number of competing ads in the auctions by 6% in our experiment). However, for our relatively small-scale simulation system, such reduction in the competition of the auctions may create substantial biases in the estimation. To de-bias the estimates in our simulation, we randomly add the same number of mature ads as the blocked new ads into the experimented auctions. Readers interested in applying our simulation system for studying other experiment designs should use their own judgment for their specific problem contexts whether mature ads should be re-sampled and added back to the auctions so as to counter the estimation bias from the reduced competition.

Table 8 Bias Analysis of Different Estimators

	Ad-Side Experiment		UV-Side Experiment		Two-Sided Experiment	
	Treatment	Control	Treatment	Control	Treatment	Control
Cold-Start Success Rate	0.068 (0.011)	0.024 (0.017)	0.050 (0.007)	0.038 (0.013)	0.060 (0.007)	0.038 (0.008)
Value of Estimator	4.4%**		1.2%*		2.2%**	
Bias/Global Treatment Effect	120%		-40%		10%	

Note: *p<0.05; **p<0.01; ***p<0.001; ****p<0.0001. Standard errors are reported in the parentheses.

D.2. Estimation Biases with One-Sided Experiments

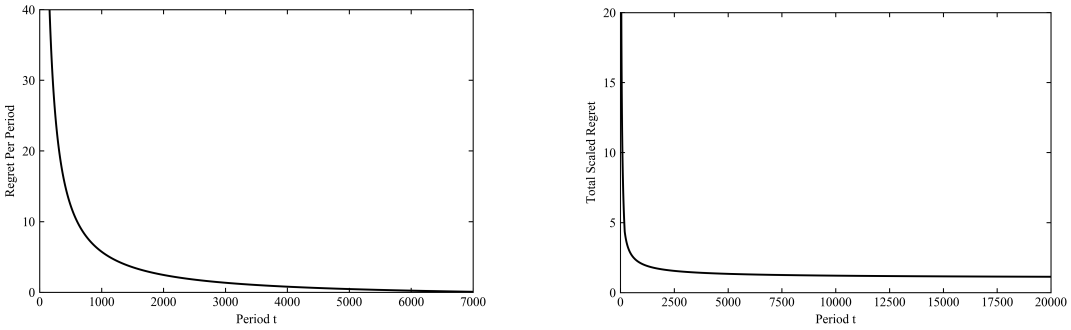
In this simulation study, we quantify the estimation biases under single-sided experiments using our simulation system described in Appendix D.1. This simulation is based on the simulation system built in Appendix D.1, with much more input data and finer granularity. Specifically, we randomly sample 100 new ads, 200 mature ads, and 1,000,000 user page-views during the cold start phase as the data inputs. Moreover, after the cold start phase, consistent with Figure 1, we assume that a new ad with fewer accumulated clicks have a higher probability to leave the advertising platform. The new ads who stay on the platform together with the 200 mature ads proceed to the stationary phase with another 1,000,000 user page-views.

To illustrate the potential estimation biases induced by the violation of SUTVA with Ad-side and UV-side experiments, we run three numerical simulations (Ad-side randomization experiment (see Figure 4(a), UV-side randomization experiment (see Figure 4(b), and two-sided experiment (see Figure 5). We fix $\alpha = 0.001$ and $\beta = 2b$ for all experiments in this set of simulations. In all these simulations, 50% of UVs/Ads are randomly assigned into the treatment condition, and the other 50% into the control condition.

We replicate the simulation for each randomized experiment for five times and report corresponding estimation results of cold start success rate in Table 8. Our simulation results demonstrate that the ad-side experiment significantly overestimates the treatment effect of the proposed algorithm, whereas the user-side experiment underestimates the effect. Furthermore, the two-sided equips us with an unbiased estimate. Therefore, our simulation results necessitate and validate our two-sided experiment design by showing that whereas one-sided experiments are likely to produce substantially biased estimates, our novel two-sided design helps correct such biases.

D.3. Regret Illustration

To numerically illustrate the regret of our algorithm, we consider a pure cold start setting with new ads only. We randomly select 100 new ads from the data set as well as 20,000 impressions to be allocated. The target number of clicks for each ad per one period is $\alpha = 0.001$, which is scaled in consistency with the cold start success criteria of 10 conversions in the three-day cold start horizon. We set the cold start reward coefficient $\beta_j = 2b_j$ for each ad j. We run the simulation for 50 times and compute the average regret. The results are plotted in in Figure 8, which confirms our theoretical result (Theorem 1) that the regret of SBL algorithms is bounded by $O(t^{2/3}(\log t)^{1/3})$.



(a) Average Regret (b) Average Regret Scaled by $t^{-2/3}(\log t)^{-1/3}$

Figure 8 Average Regret and Scaled Regret in the Simulation with $\alpha = 0.001$ and $\beta_j = 2b_j$

D.4. Impact of the Cold Start Reward Coefficient on Long-Term Revenue

As shown in Section 5.2, the online implementation of our algorithm sets the cold start reward coefficient at $\beta_j = 2b_j$ for each ad j. In this subsection, we leverage the simulation model built in Section 6.3 to demonstrate that for a wide range of choices of β_j, our oSBL algorithm can successfully boost the long-term total advertising revenue of the platform. Similar to the simulation setting in Section 6.3, to estimate the global treatment effect of the cold start reward coefficient β/b on long-term revenue, we use the data with 12 million impressions from those during April 9, 2020 and April 30, 2020. The specific simulation setting and the model validation has been documented in Section 6.3, we randomly sample 1.2 million impressions with replacement for each simulation and replicate 10 times via Bootstrap to estimate the long-term revenue of the oSBL. All the following results pass the t-test with p-value smaller than 10^{-3}.

In this subsection, however, we emphasize on the robustness of the choice of cold start coefficient β/b, which boosts the positive long term revenue within a wide range. In the regard, we conduct a sensitivity analysis with three varying parameters in the simulation. Δ_r, which refers to the average relative increase of the retention length for mature ads, Δ_{CTR}, which refers to the average relative increase of the CTR for mature ads, and the cold start reward coefficient β/b. To obtain a complete picture on the global treatment effect of our algorithm, we conduct a sensitivity analysis with our simulation model by varying $\Delta_r \in \{1\%, 2\%, 3\%\}$, varying Δ_{CTR} from 0 to 15%, and varying β/b from 0 to 3, assuming that oSBL is applied to all ads and UVs. The baseline revenue is denoted by R_0 and the revenue under the oSBL algorithm denoted by $R(\Delta_r, \Delta_{CTR}, \beta/b)$ (so $R_0 = R(0,0)$). We are interested in the relative advertising revenue increase associated with oSBL:

$$\Xi(\Delta_r, \Delta_{CTR}, \beta/b) = \frac{R(\Delta_r, \Delta_{CTR}, \beta/b) - R_0}{R_0} \times 100\%$$

The results of the sensitivity analysis presented in Figures 9, 10, and 11 demonstrate that for a wide range of Δ_r and Δ_{CTR}, our algorithm oSBL can successfully boost the long-term revenue with a flexible choice of β_j's.

Appendix E: Performance of Subgradient Descent Algorithm for Solving Duals

To demonstrate the effectiveness of subgradient descent to obtain the shadow bids λ for ad allocation, we compare our shadow-bidding-price-based ad allocation, where λ is computed by subgradient descent method

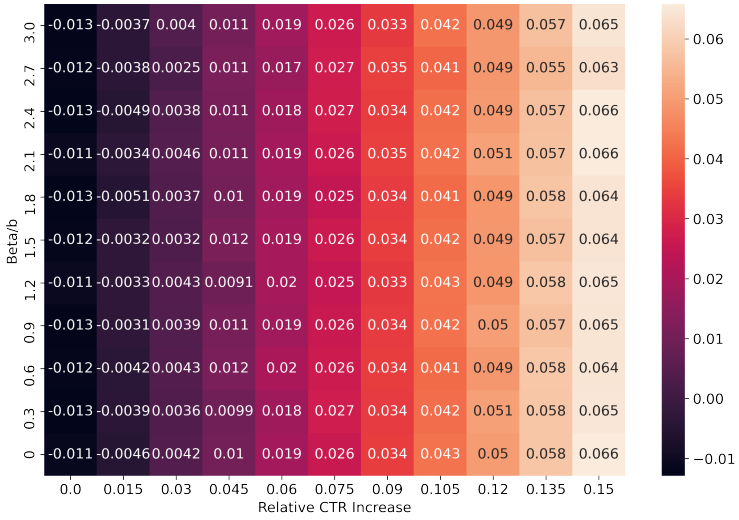


Figure 9 **Global Treatment Effect of oSBL on Advertising Revenue with $\Delta_r = 0.01$**

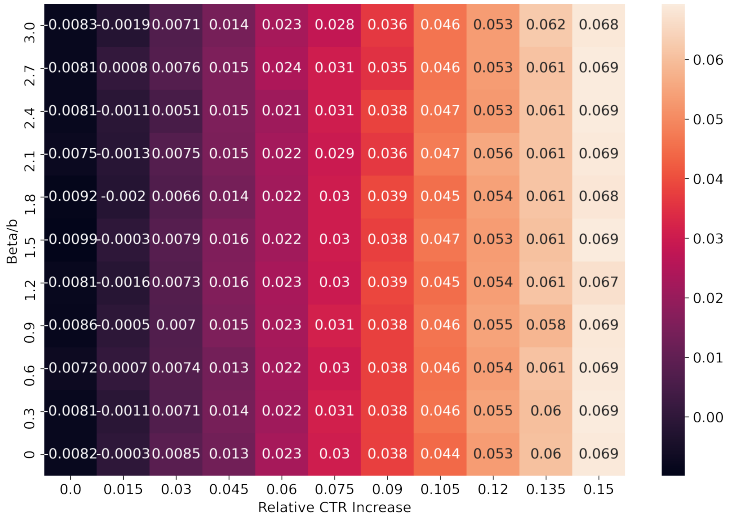


Figure 10 **Global Treatment Effect of oSBL on Advertising Revenue with $\Delta_r = 0.02$**

with (a) the Simplex method which solves the primal directly, (b) another gradient-based method SHALE (Bharadwaj et al. 2012, Hojjat et al. 2017), and (c) the current practice of Platform O, namely showing the ad with maximum eCPM without considering the cold start reward. We examine a small-scale instance with 100 Ads and 10,000 UVs in an offline setting. However, our online implementation solves the dual instances with more than 10,000,000 UVs, which is impossible to solve in a reasonable time by the standard Simplex approach. The stopping condition of our subgradient descent algorithm is when the duality gap is less than

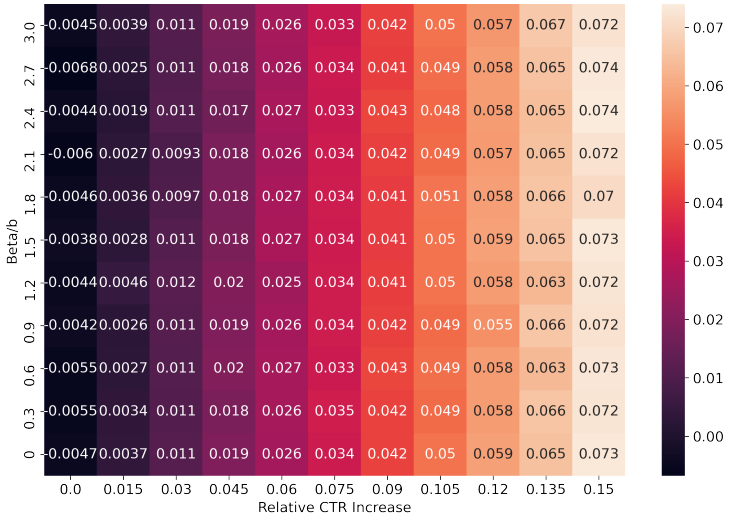


Figure 11 Global Treatment Effect of oSBL on Advertising Revenue with $\Delta_r = 0.03$

$O(10^{-4})$, which is consistent with our online implementation. Other parameters such as bidding prices b_j and pCTRs are directly from the real data. The computational results are summarized in Table 9.

Table 9 Objective Value Comparison

	Current Practice (1)	SHALE (2)	Subgradient Descent (3)	Simplex (4)
Revenue	288,556	284,913	278,598	278,588
Cold Start Reward	67,747	76,171	98,429	98,679
Total Objective Value	356,303	361,084	377,026	377,267

The objective value is the sum of the revenue and cold start reward. The relative difference between our dual-based subgradient descent approach and the optimal objective value is less than 0.07%, which suggests that the gap induced by integer round-offs and the stopping condition in our algorithm is negligible. Moreover, our algorithm performs substantially better than the SHALE algorithm (Bharadwaj et al. 2012).

Appendix F: Robustness Check of the UV Sampling Rate in oSBL

In this section, we conduct robustness check for the UV sampling rate, which shows that even a low sampling rate of 1% for user views could already cover most of the new ads and produce robust dual solutions. Moreover, considering that both memory and computational time increase linearly with the sampling rate, we choose 4% sampling rate for our online implementation, which strikes a good balance of sample representativeness and computational time.

Table 10 Robustness Check of the UV Sampling Rate

	Sampling Rate of UV r		
	$r=0.04$ (1)	$r=0.02$ (2)	$r=0.01$ (3)
The number of new Ads	6216	6216	6216
Mean of λ	64.21	64.22	64.25
Standard deviation of λ	62.96	62.96	63.05
25^{th} *percentile of* λ	15.00	15.00	15.00
50^{th} *percentile of* λ	57.60	57.60	57.21
75^{th} *percentile of* λ	90.00	90.00	90.00

Note: The differences between λ's calculated by different sampling rates are not significant. P-values of t-tests between (1) and (2), (1) and (3), and (2) and (3) are, respectively, 0.716, 0.155, and 0.280.

Appendix G: Training with Neural Networks to Predict CTR

In this section, we show that the there exists a fully connected neural network satisfying *Prediction Oracle* with high probability (i.e., Assumption 2 holds) under either (a) the lazy training regime or (b) the training algorithm with gradient descent.

Before presenting the formal results and their proofs, we first introduce the fully connected neural network and its initialization procedure. In recent years, deep-learning-based recommender systems are flourishing and widely used in practice (see the review paper Zhang et al. 2019). A large-scale DSP like Platform O is also armed with deep neural networks to predict the CTR and CVR of ads. In practice, due to the limited computational resource and high requirement on fast response, the "funnel" structure is widely adopted. For example, YouTube's recommender system (see Covington et al. 2016) uses a rough deep learning model which is very efficient but less accurate, to select hundreds out of millions of videos. Then, it uses a more sophisticated deep-learning-based ranking model with more feature inputs to choose dozens of videos from the hundreds selected in the previous step. Platform O and other video sharing platforms adopt a similar recommendation strategy. Specifically, for Platform O's DSP, there are two stages before an ad enters the final auction: filtering and pre-ranking, both of which adopt rough deep neural network models to rule out the ads not suitable for the user impression. Then, at the final auction stage, Platform O uses a set of fully connected neural networks with the ReLU activation function, i.e., $\sigma(\cdot)=\max\{\cdot,0\}$, to predict the CTR and CVR. Since there are only around 150 ads joining the auction, Platform O typically uses an individualized neural network for predicting the CTR of each ad rather than a unified model for all ads. Without loss of generality, we assume all hidden layers of the neural network have the same number of nodes. And we denote $L\geq 2$ as the network depth, w as the number of nodes in each hidden layer, and w_0 as the dimension of the context/feature vector, i.e., $x_{ij}\in\mathbb{R}^{w_0}$ for all $i\in\mathcal{I},\ j\in A$. Following the convention of the neural network literature (e.g., Cao and Gu 2019, Chizat et al. 2019), we parameterize the neural network by $\theta\in\mathbb{R}^d$, where the prediction error term $d=w^2(L-2)+ww_0+w$. Then, we can use the function $H_j(x_{ij},\theta)=\sqrt{w}W_L\sigma(W_{L-1}\sigma(\cdots\sigma(W_1x_{ij})))$ to represent the output of the neural network given the parameter θ, for any ad j and context i, where $\theta=[\mathrm{vec}(W_1),\cdots,\mathrm{vec}(W_L)]$, $W_1\in\mathbb{R}^{w_0\times w}$, $W_i\in\mathbb{R}^{w\times w}$, $2\leq i\leq L-1$, $W_L\in\mathbb{R}^{w\times 1}$ and the operator $\mathrm{vec}(\cdot)$ refers to representing the matrix as a vector.

In practice, the initialization procedure may take into account the domain knowledge of the context. In our analysis that follows, we adopt the initialization procedure in He et al. (2015), known as the *He Initialization* to set θ_0. For each layer $1 \le l \le L-1$, we set W_l to be $\begin{pmatrix} W & 0 \\ 0 & W \end{pmatrix}$, where each entry of this matrix W is randomly and independently drawn from a normal distribution $N(0, 2/w)$. The parameter of the last layer is initialized as $(w^T, -w^T)$ where each entry of vector w is randomly and independently drawn from distribution $N(0, 1/w)$. One can verify that under this initialization procedure of θ_0, it holds that $H_j(x_{ij}, \theta_0) = 0$ for all context i and ad j (see Cao and Gu 2019, He et al. 2015).

To validate Assumption 2 for fully connected neural networks, we make additional technical assumptions as follows, which are mild and commonly made in the related literature (e.g., Cao and Gu 2019, Zhou et al. 2020).

ASSUMPTION 3 **(Finite and nonparallel Contexts)**. *(a) The number of context type i is finite, i.e., the cardinality of the context set $|X| = m < +\infty$, and m is bounded by a fixed polynomial of T, i.e., $m \le O(T^k)$ for a some k. (b) For any pair of contexts $x_{ij}, x_{i'j'} \in X$ $(i \ne i'$ or $j \ne j')$, x_{ij} and $x_{i'j'}$ are not parallel. (c) The L_2-norm of each context is normalized to 1, i.e., $\|x_{ij}\|_2 = 1$ for any context $i \in [m]$ and ad $j \in A$. (d) The j^{th} component of x is equal to the $(j + d/2)^{th}$ component for any context $x \in X$ and $j \le d/2$*

The parts (a) and (b) of Assumption 3 are mild, while parts (c) and (d) are just for the convenience of analysis. Notice that part (d) can always be satisfied by transforming any context x to a new one $x' = [x, x]/\sqrt{2}$.

G.1. Lazy Training Regime

The recent progress of *Neural Tangent Kernels* (e.g., Cao and Gu 2019, Jacot et al. 2018, Arora et al. 2019, Zhou et al. 2020) theoretically characterizes the representation power of a neural network. Following this literature, we use $\boldsymbol{H}$ to denote the Neural Tangent Kernel Matrix in the same way as Definition 4.1 of Zhou et al. (2020). As discussed in Zhou et al. (2020) Assumption 4.2, $\boldsymbol{H} \succeq \gamma I$ always holds for some $\gamma > 0$, where I is the identity matrix under Assumption 3. This ensures that the Neural Tangent Kernel Matrix $\boldsymbol{H}$ is always nonsingular. Lemma 5 below shows that, as long as the ground truth CTR can be represented as a bounded function of user contexts and ads, then the fully connected neural network with a large width w can accurately predict this CTR with high probability in terms of the *He Initialization*.

LEMMA 5 **(Lemma 5.1 in Zhou et al. (2020))**. *Under Assumption 3, for any $j \in A$ there exists a constant $C > 0$ such that, if $w \ge Cm^4L^6 \log(m^2L/\delta)/\gamma^4$, then with probability $1-\delta$ over the He Initialization of the parameter θ_0, there exists a $\theta^* \in \mathbb{R}^d$ such that, for any $i \in \mathcal{I}$,*

$$c_{ij} = \langle \nabla_\theta H_j(x_{ij}, \theta_0), \theta^* - \theta_0 \rangle, \sqrt{w}\, \|\theta^* - \theta_0\|_2 \le \sqrt{2c^T \boldsymbol{H} c},$$

where $c := [c_{ij}]_{i \in \mathcal{I}} \in \mathbb{R}^m$.

Notice that the approximation of ground truth CTR in Lemma 5 is linear in the gradient $\nabla_\theta H_j(x_{ij}, \theta_0)$ parametrized by $\theta^* - \theta_0$. The original neural network for the ad j mapping $H_j(\cdot, \cdot)$ is now divided into two steps. First, it maps the context x_{ij} to the gradient $\nabla_\theta H_j(x_{ij}, \theta_0)$. This is a static mapping that depends

on the initialization θ_0, but independent of the parameter θ. The second step linearly maps the gradient $\nabla_\theta H_j(x_{ij},\theta_0)$ to the true CTR, c_{ij}. As a consequence, to train the neural network under this regime, it suffices to fit a linear function parameterized by θ. This training method is referred to as *Lazy Training* in the literature (Chizat et al. 2019). Specifically, lazy training with the regularized least square loss function at round t for ad j is equivalent to solving the following minimization problem:

$$\min_{\theta} w\lambda_0\|\theta-\theta_0\|^2+\sum_{\tau\in\mathcal{T}_j^t}(z_\tau-\langle\nabla_\theta H_j(x_{i_\tau a_\tau},\theta_0),\theta-\theta_0\rangle)^2, \tag{34}$$

where $\mathcal{T}_j^t$ denotes the time periods until round t in which ad j is played, x_{i_τ,a_τ} represents the context vector associated with context i_τ and ad a_τ realized at round τ. λ_0 is the regularized parameter, z_τ denotes the corresponding click-through outcome of round τ. With lazy training, we effectively linearize the neural network for CTR prediction, thus reducing it to a linear regression model. Lazy training facilitates us to focus on cold start algorithm design, without delving into the details of how a neural network shall be trained. Similar approaches, usually referred to as *Kernelized Contextual Bandits*, have been adopted in the contextual bandit literature (e.g., Valko et al. 2013). As identified by Chizat et al. (2019), the lazy training phenomenon, where a neural network behaves similarly to a linear model when the parameter θ is close to the initialization parameter θ_0, will occur when the neural network is over-parameterized. In addition, Chizat et al. (2019) also show that the gradient flows of the lazy training process and the gradient descent training process (see Appendix G.2) are close to each other for over-parameterized neural networks. We also remark that the real online training procedure of Platform O's CTR/CVR prediction model is neither pure supervised learning nor lazy training, but a substantial compromise under limited computational resources. In this regard, incorporating the exact online training process into our regret analysis is unnecessary and beyond the scope of this paper. Although Chizat et al. (2019) empirically shows that lazy training might not perform well in some cases with biased gradients, this training method still provides a good theoretical understanding of how the CTR/CVR estimate is produced by neutral networks, and inspires us to validate Assumption 2 for neural networks with gradient descent training (see Appendix G.2). We are now ready to validate Assumption 2 for neural networks under the lazy training regime.

Proposition 2 **(Prediction Oracle with Lazy Training)**. *Under Assumption 3, for any ad $j\in A$ with the prediction model (34) trained on n_j^t i.i.d samples drawn from $\mathcal{D}_X$ and the corresponding click-through outcome before round t, then we have that for any δ there exists a constant $C>0$, such that, if $w\geq Cm^4L^6\log(m^2L/\delta)/\gamma^4$ and $n_j^t\geq\Omega(d\log(dT))$, it holds that, with probability at least $1-\delta-T^{-4}$, for any context $i\in\mathcal{I}$ the following inequality holds:*

$$|\hat{c}_{ij}^t-c_{ij}|\leq O\left(\sqrt{\frac{d\log T}{n_j^t}}\right).$$

where $\hat{c}_{ij}^t$ is the predicted CTR at round t via model (34) with $0<\lambda_0\leq O(\sqrt{1/(2wc^T\boldsymbol{H}c)})$.

Proof of Proposition 2. We first introduce some definitions. We use I_d to denote the identity matrix with dimension d. We define $g_{ij} := \nabla_\theta H_j(x_{ij}, \theta_0)$, for all $i \in \mathcal{I}$ and $j \in A$. Following the standard lazy training with regularized squared loss (34), we can compute θ^t in closed form at each round t as follows:

$$\begin{aligned} A_j^t &:= w\lambda_0 I_d + \sum_{\tau \in \mathcal{T}_j^t} g_{i_\tau a_\tau} g_{i_\tau a_\tau}^T, \quad & D_j^t &:= [g_{i_\tau a_\tau}^T]_{\tau \in \mathcal{T}_j^t} \\ b_j^t &:= \sum_{\tau \in \mathcal{T}_j^t} v_{i_\tau a_\tau} g_{i_\tau a_\tau}, & V_j^t &:= [v_{i_\tau a_\tau}]_{\tau \in \mathcal{T}_j^t} \\ \theta^t &:= (A_j^t)^{-1} b_j^t + \theta_0, & s_{ij}^t &:= \sqrt{g_{ij}^T (A_j^t)^{-1} g_{ij}} \end{aligned}$$

By Lemma 5, we consider the case, with high probability $1-\delta$, where CTR c_{ij} can be perfectly predicted via a linear mapping. Thus, at round t after observing n_j^t i.i.d samples, we have for any realized context $i \in \mathcal{I}$ at round t, with probability at least $1-\delta$:

$$\begin{aligned} |\hat{c}_{ij}^t - c_{ij}| &= |g_{ij}^T(\theta^t - \theta_0) - g_{ij}^T(\theta^* - \theta_0)| \\ &= |g_{ij}^T (A_j^t)^{-1} b_j^t - g_{ij}^T (A_j^t)^{-1} (w\lambda_0 I_d + (D_j^t)^T D_j^t)(\theta^* - \theta_0)| \\ &= |g_{ij}^T (A_j^t)^{-1} (D_j^t)^T (V_j^t - D_j^t(\theta^* - \theta_0)) - w\lambda_0 g_{ij}^T (A_j^t)^{-1} (\theta^* - \theta_0)| \\ &\le |g_{ij}^T (A_j^t)^{-1} (D_j^t)^T (V_j^t - D_j^t(\theta^* - \theta_0))| + w\lambda_0 M \left\| (A_j^t)^{-1} g_{ij} \right\|_2, \end{aligned} \tag{35}$$

where the first equality follows from the lazy training process $\hat{c}_{ij}^t = g_{ij}^T(\theta^t - \theta_0)$ (by Lemma 5), and the second from the identity $A_j^t = w\lambda_0 I_d + (D_j^t)^T D_j^t$ and $b_j^t = (D_j^t)^T V_j^t$. The inequality of (35) follows from the triangular inequality and $\|\theta^* - \theta_0\|_2 \le M$ (by Lemma 5, we take the value of M at $M = \sqrt{2c^T \boldsymbol{H} c / w}$). Because $\mathbb{E}[V_j^t - D_j^t(\theta^* - \theta_0)] = 0$, Azuma–Hoeffding inequality implies the following concentration inequality on the first term of (35).

$$\begin{aligned} \mathbb{P}\left[|g_{ij}^T (A_j^t)^{-1} (D_j^t)^T (V_j^t - D_j^t(\theta^* - \theta_0))| \ge \sqrt{\frac{1}{2}\log\frac{2}{\Delta}} s_{ij}^t \right] &\le 2\exp\left(- \frac{\log(2/\Delta)(s_{ij}^t)^2}{\left\| D_j^t (A_j^t)^{-1} g_{ij} \right\|_2^2} \right) \\ &\le 2\exp(-\log(2/\Delta)) = \Delta, \end{aligned} \tag{36}$$

where the second inequality follows from

$$\begin{aligned} \left\| D_j^t (A_j^t)^{-1} g_{ij} \right\|_2^2 &= (D_j^t (A_j^t)^{-1} g_{ij})^T D_j^t (A_j^t)^{-1} g_{ij} \\ &\le g_{ij}^T (A_j^t)^{-1} (I_d + (D_j^t)^T D_j^t)(A_j^t)^{-1} g_{ij} \\ &= g_{ij}^T (A_j^t)^{-1} g_{ij} = (s_{ij}^t)^2 \end{aligned} \tag{37}$$

Similarly, we have the bound $\left\| (A_j^t)^{-1} g_{ij} \right\|_2 \le s_{ij}^t$. Combining the above two inequalities (36) and (37), we have, with probability at least $1-\Delta$, $|\hat{c}_{ij}^t - c_{ij}| \le \left(w\lambda_0 M + \sqrt{\frac{1}{2}\log\frac{2}{\Delta}} \right) s_{ij}^t$. Notice that the gradient satisfies that $\|g_{ij}\|_2 \le \sqrt{wL}$ for all i and j (see Cao and Gu 2019), and the regularization parameter satisfies $\lambda_0 \le O(\sqrt{1/(2wc^T \boldsymbol{H} c)})$. Therefore, the regularization term satisfies

$$w\lambda_0 M \left\| (A_j^t)^{-1} g_{ij} \right\|_2 \le w \cdot O(\sqrt{1/(2wc^T \boldsymbol{H} c)}) \cdot \sqrt{2c^T \boldsymbol{H} c / w} \cdot s_{ij}^t = O(s_{ij}^t),$$

where the inequality follows from that $\lambda_0 \le O(\sqrt{1/(2wc^T \boldsymbol{H} c)})$ and $M = \sqrt{2c^T \boldsymbol{H} c / w}$. Let $\Delta := T^{-4}$, we have that, with probability at least $1 - T^{-4}$ and a fixed context i,

$$|\hat{c}_{ij}^t - c_{ij}| \le O(\sqrt{\log T} s_{ij}^t),$$

where $s_{ij}^t = \sqrt{g_{ij}^T(w\lambda_0 I_d + \sum_{\tau\in\mathcal{T}_j^t} g_{i_\tau a_\tau} g_{i_\tau a_\tau}^T)^{-1} g_{ij}}$. Next, we show that with probability at least $1-T^{-4}$, it holds that $s_{ij}^t \le O(\sqrt{d/n_j^t})$ with samples $n_j^t \ge \Omega(d\log(dT))$. Let $\hat{\Sigma} := w\lambda_0 I_d + \sum_{\tau\in\mathcal{T}_j^t} g_{i_\tau a_\tau} g_{i_\tau a_\tau}^T$, and $\Sigma := w\lambda_0 I_d + n_j^t \mathbb{E}[gg^T]$, then we have,

$$\begin{aligned}(s_{ij}^t)^2 &= g_{ij}^T \hat{\Sigma}^{-1} g_{ij} \\ &\le |g_{ij}^T \hat{\Sigma}^{-1}(\Sigma - \hat{\Sigma})\Sigma^{-1} g_{ij}| + g_{ij}^T \Sigma^{-1} g_{ij} \\ &\le O\left(\frac{d}{n_j^t}\right) \left\| \frac{1}{n_j^t} \sum_{\tau\in\mathcal{T}_j^t} g_{i_\tau a_\tau} g_{i_\tau a_\tau}^T - \mathbb{E}[gg^T] \right\|_2 + O\left(\frac{d}{n_j^t}\right),\end{aligned} \tag{38}$$

where the first inequality follows from the triangle inequality, and the second from the bound on gradient $\|g\|_2 \le \sqrt{wL} \le \sqrt{d}$. Next, we need to bound the term $\|\frac{1}{n_j^t}\sum_{\tau\in\mathcal{T}_j^t} g_{i_\tau a_\tau} g_{i_\tau a_\tau}^T - \mathbb{E}[gg^T]\|_2 \le O(1)$. Because, the gradients are bounded with $\|g\|_2 \le \sqrt{wL} \le \sqrt{d}$, with high probability $1-T^{-4}$, we have the following inequality based on Theorem 1.6.2 (the Matrix Bernstein inequality) in (Tropp 2015),

$$\left\| \frac{1}{n_j^t} \sum_{\tau\in\mathcal{T}_j^t} g_{i_\tau a_\tau} g_{i_\tau a_\tau}^T - \mathbb{E}[gg^T] \right\|_2 \le \sqrt{\frac{2d\log(dT)}{n_j^t}} + \frac{\sqrt{d}\log(dT)}{3n_j^t} \le O(1),$$

where the second inequality follows from the condition $n_j^t \ge \Omega(d\log(dT))$. Therefore, after taking the union bound for all context $i \in \mathcal{I}$ together with that c_{ij} can be perfectly predicted via the linear function, we obtain that, with probability at least $1-\delta-T^{-4}$, it holds

$$|\hat{c}_{ij}^t - c_{ij}| \le O\left(\sqrt{\frac{d\log T}{n_j^t}}\right),$$

where the inequality follows from the assumption that m is smaller than the polynomials of T. This concludes the proof of Proposition 2. □

Notice that, for the DNN predictor in our analysis, we require a very wide feed-forward neural network with width $O(m^4)$, which implies an impractical prediction error parameter $d = O(m^8)$. However, the network width's dependence on m can be significantly reduced to the effective dimension of the neural tangent kernel matrix shown by Zhou et al. (2020). Moreover, Zhou et al. (2020) claim that this effective dimension only depends logarithmically on the number of contexts m in several special cases. Furthermore, imposing some structural assumptions on the click-through rate (as a function of user context x_t and ad j) $c_{tj} = \mathbb{E}[v_{tj}(a_t)|a_t = j] \in [0,1]$ also reduces the width of the DNN to estimate it. For example, Yarotsky (2017) show that DNNs of width $O\left(\epsilon^{-\frac{dim(x)}{\beta}}(\log(1/\epsilon)+1)\right)$ suffice to achieve an ϵ-approximation error uniformly for all contexts under the β-Sobolev smoothness assumption. Farrell et al. (2021) show a novel convergence rate for this class of DNNs. However, it is worth mentioning that the convergence result in Farrell et al. (2021) only implies our prediction oracle assumption when solving the empirical estimation to the optimality. This represents a deviation from our first-order optimization method presented in the next subsection. We leave it as future research to obtain a tighter dependence of d on m for general feed-forward neural networks.

G.2. Training with the Gradient Descent Algorithm

We now consider the gradient-based training procedure for neural networks to validate Assumption 2. In fact, one can devise a gradient descent training algorithm that achieves the same convergence rate as lazy

training (i.e., $|\hat{c}_{ij}^t - c_{ij}| \le O(\sqrt{d\log T/n_j^t})$), because the training trajectory path, $\{\theta^t\}_{t=1}^T$, of the gradient descent procedure is close to that of lazy training. Formally, we propose the gradient-based training of a neural network as follows. This gradient descent procedure is an approximation of SGD. This training method can also be replaced by stochastic gradient descent with a more involved analysis such as Allen-Zhu et al. (2019).

Training a Neural Network with Gradient Descent at the Round t for ad j

Input: Step size η, number of gradient descent steps U, network width w, regularization parameter λ_0.

Loss function: $\mathcal{L}(\theta) := \sum_{\tau\in\mathcal{T}_j^t}(H_j(x_{i_\tau a_\tau},\theta) - v_{i_\tau a_\tau})^2/2 + w\lambda_0\|\theta-\theta_0\|_2^2/2$

For $u = 0,1,2,\ldots,U-1$ **do**

$\theta^{u+1} = \theta^u - \eta\nabla\mathcal{L}(\theta^u)$

The following proposition shows that Assumption 2 holds for a neural network if trained with the gradient descent algorithm described above.

PROPOSITION 3 **(Prediction Oracle with Gradient-based Training)**. *Under Assumption 3 and all the conditions of Proposition 2, for any ad $j \in A$, the predicted CTR at round t, $\hat{c}_{ij}^t$, is obtained by the gradient descent algorithm. For any δ, there exist a family of constants $\{C_i\}_{i=0}^5 > 0$ such that, if for all $t \in [T]$, the regularization parameter λ_0, training step size η, number of steps U, and network width w satisfy*

$$w \ge C_0 m^4 L^6 \log(m^2L/\delta)/\gamma^4$$
$$2\sqrt{t/(w\lambda_0)} \ge C_1 w^{-3/2}L^{-3/2}[\log(mL^2/\delta)]^{3/2}$$
$$2\sqrt{t/(w\lambda_0)} \le C_2 \min\{L^{-6}[\log w]^{-3/2}, (w(\lambda_0\eta)^2L^{-6}t^{-1}(\log w)^{-1})^{3/8}\}$$
$$\eta \le C_3(w\lambda_0 + twL)^{-1}$$
$$U > C_4 \log(d\log(T))/\log(1-\eta w\lambda_0)$$
$$w^{1/6} \ge C_5\sqrt{\log w}L^{7/2}t^{7/6}\lambda_0^{-7/6}(1+\sqrt{t/\lambda_0}),$$

it holds that, if ad j is displayed $n_j^t \ge \Omega(d\log(dT))$ times and the random click-through outcomes $\{v_s(a_s) \in \{0,1\} : 1 \le s \le t\}$ are observed, then for all context $i \in \mathcal{I}$, with probability at least $1-\delta-T^{-4}$, the following inequality holds:

$$|\hat{c}_{ij}^t - c_{ij}| \le O\left(\sqrt{\frac{d\log T}{n_j^t}}\right).$$

Before proving Proposition 3, we first introduce Lemma 6 and Lemma 7 below to bound the training trajectory $\{\theta^t : t = 1,2,...,T\}$ and the gradient $\nabla_\theta H_j(x_{ij},\hat{\theta})$, respectively.

LEMMA 6 **(Lemma B.2 in Zhou et al. (2020))**. *For any ad $j \in A$, there exist a family of constants $\{C_i\}_{i=1}^5 > 0$ such that for any $\delta \in (0,1)$, if for each $t \in [T]$, η and w satisfy*

$$2\sqrt{t/(w\lambda_0)} \ge C_1 w^{-3/2}L^{-3/2}[\log(mL^2/\delta)]^{3/2}$$
$$2\sqrt{t/(w\lambda_0)} \le C_2 \min\{L^{-6}[\log w]^{-3/2}, (w(\lambda_0\eta)^2L^{-6}t^{-1}(\log w)^{-1})^{3/8}\}$$
$$\eta \le C_3(w\lambda_0 + twL)^{-1}$$
$$w^{1/6} \ge C_4\sqrt{\log w}L^{7/2}t^{7/6}\lambda_0^{-7/6}(1+\sqrt{t/\lambda_0})$$

then, with probability at least $1-\delta$ *over the He Initialization of* θ_0*, we have, for any* $t \in [T]$, $\|\theta^t - \theta_0\|_2 \le 2\sqrt{t/w\lambda_0}$ *and*

$$\|\theta^t - (A_j^t)^{-1}b_j^t - \theta_0\|_2 \le (1-\eta w\lambda_0)^{U/2}\sqrt{t/(w\lambda_0)} + C_5 w^{-2/3}\sqrt{\log w}L^{7/2}t^{5/3}\lambda_0^{-5/3}(1+\sqrt{t/\lambda_0}). \tag{39}$$

LEMMA 7 **(Lemma B.4 in Zhou et al. (2020))**. *For any ad* $j \in A$*, there exist a family of constants* $\{C_i\}_{i=1}^3 > 0$ *such that for any* $\delta \in (0,1)$*, if* τ *satisfies that*

$$C_1 w^{-3/2}L^{-3/2}[\log(mL^2/\delta)]^{3/2} \le \tau \le C_2 L^{-6}[\log w]^{-3/2}.$$

then, with probability at least $1-\delta$ *over the He Initialization of* θ_0*, for all* $\hat{\theta}$ *and* $\tilde{\theta}$ *satisfying* $\|\hat{\theta}-\theta_0\|_2 \le \tau$ *and* $\|\tilde{\theta}-\theta_0\|_2 \le \tau$*, we have, for any context* i*,*

$$\left|H_j(x_{ij},\tilde{\theta}) - H_j(x_{ij},\hat{\theta}) - \langle \nabla_\theta H_j(x_{ij},\hat{\theta}), \tilde{\theta}-\hat{\theta}\rangle\right| \le C_3\tau^{4/3}L^3\sqrt{w\log w}.$$

With Lemma 6 and Lemma 7, we are now ready to prove Proposition 3.

Proof of Proposition 3. It suffices to consider the union of the high probability cases in Proposition 2, Lemma 6, and Lemma 7. Let us set $\tau = 2\sqrt{t/w\lambda_0}$ in Lemma 7. At round t, after observing n_j^t i.i.d samples of each ad j, for a fixed context i, we have the following inequality:

$$\begin{aligned}
|\hat{c}_{ij}^t - c_{ij}| &= |H_j(x_{ij},\theta^t) - \langle \nabla_\theta H_j(x_{ij},\theta_0), \theta^* - \theta_0\rangle| \\
&\le |H_j(x_{ij},\theta^t) - \langle \nabla_\theta H_j(x_{ij},\theta_0), (A_j^t)^{-1}b_j^t\rangle| + |\langle \nabla_\theta H_j(x_{ij},\theta_0), (A_j^t)^{-1}b_j^t\rangle - \langle \nabla_\theta H_j(x_{ij},\theta_0), \theta^*-\theta_0\rangle| \\
&\le |H_j(x_{ij},\theta^t) - \langle \nabla_\theta H_j(x_{ij},\theta_0), (A_j^t)^{-1}b_j^t\rangle| + O(\sqrt{d\log(T)/n_j^t}) \\
&\le |H_j(x_{ij},\theta^t) - H_j(x_{ij},(A_j^t)^{-1}b_j^t+\theta_0) + H_j(x_{ij},\theta_0)| + C_3\tau^{4/3}L^3\sqrt{w\log w} + O(\sqrt{d\log(T)/n_j^t}) \\
&\le |\langle \nabla_\theta H_j(x_{ij},\theta_0), -(A_j^t)^{-1}b_j^t - \theta_0 + \theta^t\rangle| + 2C_3\tau^{4/3}L^3\sqrt{w\log w} + O(\sqrt{d\log(T)/n_j^t}) \\
&\le (1-\eta w\lambda_0)^{U/2}\sqrt{wL}\sqrt{t/w\lambda_0} + O(t^{2/3}w^{-1/6}\lambda_0^{-2/3}L^3\sqrt{\log w}) + O(\sqrt{d\log(T)/n_j^t}),
\end{aligned} \tag{40}$$

where the equality follows from Lemma 5. The first inequality of (40) follows from the triangular inequality. The second inequality of (40) follows from Proposition 2. The third and fourth inequalities of (40) follow from Lemma 7, the fact that $\|(A_j^t)^{-1}b_j^t\|_2 \le \tau$ (see Lemma C.4 in Zhou et al. 2020), the triangular inequality, and $H_j(x_{ij},\theta_0) = 0$ by the *He Initialization* of θ_0. The last inequality of (40) follows from Lemma 6 which bounds $\|\theta^t - (A_j^t)^{-1}b_j^t - \theta_0\|_2$ using inequality (39), and the bound on the gradient $\|\nabla_\theta H_j(x_{ij},\theta_0)\|_2 \le \sqrt{wL}$ (see Cao and Gu 2019, Zhou et al. 2020). With a sufficiently large neural network width w, the second term of the last inequality (40), $O(t^{2/3}w^{-1/6}\lambda_0^{-2/3}L^3\sqrt{\log w})$, can be bounded by $O(\sqrt{d\log(T)/n_j^t})$. The first term of (40), $(1-\eta w\lambda_0)^{U/2}\sqrt{wL}\sqrt{t/w\lambda_0}$ converges to 0 at an exponential rate with respect to the number of training steps U. Because $U > \Omega(\log(d\log(T))/\log(1-\eta w\lambda_0))$, $(1-\eta w\lambda_0)^{U/2}\sqrt{wL}\sqrt{t/w\lambda_0}$ is also bounded by $O(\sqrt{d\log(T)/n_j^t})$. Following the same argument as the proof of Proposition 2, we obtain that, with probability $1-\delta-T^{-4}$, for any context i,

$$|\hat{c}_{ij}^t - c_{ij}| \le O\left(\sqrt{\frac{d\log(1/\delta)}{n_j^t}}\right).$$

This concludes the proof of Proposition 3. □

Appendix H: Details of the Online Advertising System for Platform O

In this section, we describe the institutional details of the online advertising system for Platform. In particular, the auction mechanisms, billing options, and the PID system are introduced.

H.1. Auction Mechanisms and Billing Options

For a large scale online platform, the DSP allocates billions of ad impressions to hundreds of thousands of ads each day. In order not to ruin the user experience, the DSP needs to efficiently match the tremendous number of ads and ad impressions within milliseconds. Before entering the auction stage, the DSP quickly downscales the size of the ad pool from hundreds of thousands to hundreds by simple filtering rules set by advertisers and predictive models. At the auction stage, hundreds of ads compete to win an ad impression based on advertisers' bids. The ad impression is allocated to the ad with the highest *estimated Cost per Mille* (eCPM) of the match, which measures the expected revenue of displaying the ad to the respective platform user for a thousand times. Such an allocation rule ensures that each ad impression generates the highest *ex ante* revenue in expectation.

The eCPM of a match between an ad and an ad impression depends on what the advertiser bids on (impression, click, or conversion). More specifically, if the advertiser bids on *impression*, eCPM is the bid itself (eCPM=bid). If the advertiser bids on *click*, eCPM equals the bid multiplied by the predicted CTR (pCTR) of the ad (eCPM=bid×pCTR). Finally, if the advertiser bids on *conversion*, eCPM equals the bid multiplied by the product of pCTR and the predicted conversion rate (pCVR) of the ad (eCPM=bid×pCTR×pCVR), where pCVR is defined as the rate of the user being converted after clicking the ad. Here, conversion means that, upon clicking an ad, a user eventually becomes the advertiser's customer. A typical conversion, sometimes also called an "action" of the user, may take different forms, such as app installation or deposit in a game.

Typically, there are several different billing options for advertisers to choose from, including *Cost per Mille* (CPM), *Cost per Click* (CPC), *Cost per Action* (CPA), *Optimized Cost per Mille* (oCPM) and *Optimized Cost per Click* (oCPC). We summarize the differences between these billing options in Table 11. Under the CPM, CPC, and CPA billing option, advertisers bid on the impressions, clicks, and conversions, respectively, and are directly charged after their ads are displayed, clicked, or converted. Due to the intrinsic uncertainty of user clicks and conversions, the advertiser bears a high risk under the CPM scheme. On the other hand, if an ad is displayed to a platform user, it already makes a negative impact on user experience and causes losses to the platform. As a consequence, in the current online advertising market (such as Facebook and Platform O), oCPM and oCPC under which the DSP and the advertiser *share* the click and conversion uncertainty risks are the most popular billing options. More specifically, under both oCPM and oCPC, the advertisers bid on conversion. However, the advertiser will be charged by the expected cost per impression, bid_conversion×pCTR×pCVR (resp. the expected cost per click, bid_conversion×pCVR), when the ad is displayed to (resp. clicked by) a user under oCPM (resp. oCPC). One should note that, because of the randomness in click-through and conversion rates, the actual payment of the advertiser per conversion is not necessarily the same as its bid for a conversion under oCPM or oCPC. For each billing option, the auction may run in a first-price or second-price fashion, under which the winning advertiser pays its own bid, or the bid with the second-highest eCPM.

Table 11 Billing Options

Payment Scheme	Bid Price	Charged upon	Fee Deduction	eCPM (Rank by)
CPM	bid_impression	impression	bid_impression	bid_impression
CPC	bid_click	click	bid_click	bid_click×pCTR
oCPM	bid_conversion	impression	bid_conversion×pCTR×pCVR	bid_conversion×pCTR×pCVR
oCPC	bid_conversion	click	bid_conversion×pCVR	bid_conversion×pCTR×pCVR
CPA	bid_conversion	conversion	bid_conversion	bid_conversion×pCTR×pCVR

Note: This table is mainly for the first-price auction. Also, bid_impression, bid_click, bid_conversion are the bids on impressions, clicks, and conversions given by advertisers, respectively. The column Fee Deduction gives the budget depleted upon each impression, click, or conversion.

H.2. PID-Based Bidding System

As introduced in Section 3, the PID controller is a feedback control device widely used in online advertising platforms, especially for the oCPC and oCPM billing options. The PID controller aims to gear the realized CPA of each ad as close to the target CPA as possible, so it increases the bid price thus boosting its eCPM and the chance of winning the auction, if the actual cost per conversion of an ad falls below the target cost, and vice versa.

Both billing option oCPM and oCPC suffer from the issue that the actual cost (per conversion) of an advertiser is different from its bid (also known as the target cost). Such a cost-control issue is exacerbated by the following: (a) second-price auction, under which the winning advertiser pays the expected cost per impression/click of the bidder with the second-highest eCPM; and (b) biased estimation of pCTR and pCVR, under which for each ad and ad impression pair the DSP could not accurately estimate the CTR and CVR. In practice, the PID controller is widely adopted on online advertising platforms (Yang et al. 2019, Zhang et al. 2016) to control the gap between the actual cost and the target cost for an advertiser. Under PID, advertisers authorize the DSP to adaptively change their real-time bid prices to address the aforementioned *cost control problem*. The core of the PID controller is a simple feedback control idea: If the actual cost per conversion falls below the target cost, the DSP will increase the bid price thus boosting its eCPM and the chance of winning the auction, and vice versa. This real-time bid price given by the PID controller is referred to as the *System Bidding Price* throughout this paper.

Formally, the PID controller is formulated as Eq. (41).

$$\begin{aligned}
&\mathsf{error}_t = \mathsf{targetBid} - \mathsf{realCost}_t/\mathsf{realConversion}_t,\\
&\mathsf{P}_t = k_p \times \mathsf{error}_t\\
&\mathsf{I}_t = k_i \times \sum_{t' \le t} \mathsf{error}_{t'},\\
&\mathsf{D}_t = k_d \times (\mathsf{error}_t - \mathsf{error}_{t-1})\\
&\mathsf{PID}_t = \mathsf{P}_t + \mathsf{I}_t + \mathsf{D}_t,\\
&\mathsf{systemBid}_{t+1} = \mathsf{systemBid}_t + \mathsf{PID}_t \times \mathsf{systemBid}_t
\end{aligned} \tag{41}$$

It is clear from the above formulation that the PID controller changes the system bidding price $\mathsf{systemBid}_t$ after accumulating the feedback data for each ad within a fixed amount of time. The first equation quantifies the gap between the target cost and the real cost (the total actual cost divided by the total actual conversions).

P_t, I_t, D_t represents the Proportional, Integral, Derivative (PID) term in the PID system respectively. And the corresponding coefficients k_p, k_i, k_d are hyper-parameters to be fine-tuned. Readers interested in more details about the PID system are referred to, e.g., Yang et al. (2019) and Zhang et al. (2016).