Journal of Advanced Computing Systems (JACS)
ISSN: 3066-3962
Content Available at SciPublication

# Risk-Aware Budget-Constrained Auto-Bidding under First-Price RTB: A Distributional Constrained Deep Reinforcement Learning Framework

*Hanqi Zhang*
*Computer Science, University of Michigan at Ann Arbor, MI, USA*
*hz0102@yahoo.com*

DOI: 10.69987/JACS.2024.40603

**Keywords**

Real-time bidding, first-price auction, auto-bidding, budget constraint, pacing, risk-aware reinforcement learning, CVaR, distributional RL.

**Abstract**

Real-time bidding (RTB) has become the dominant mechanism for programmatic display advertising, and the industry has migrated from second-price to first-price auctions. First-price auctions simplify settlement but fundamentally change the cost dynamics: the bidder pays its own bid, which amplifies overbidding losses, increases spend volatility, and makes smooth budget delivery (pacing) more difficult. Most academic auto-bidding literature assumes second-price payment, and many budget-constrained reinforcement learning (RL) methods optimize only expected performance, without explicit downside-risk control. This paper proposes RA-BCB, a risk-aware budget-constrained auto-bidding framework for first-price RTB. RA-BCB combines (i) a value model (pCTR/pCVR) trained from logged impression–click–conversion chains, (ii) an offline replay auction simulator that re-prices wins using first-price payment, and (iii) a distributional constrained RL agent that optimizes a Conditional-Value-at-Risk (CVaR) objective under a daily budget constraint. The agent acts at a time-aggregated granularity (24 hourly slots), selecting a continuous bid multiplier that scales a base bid derived from predicted value. A dual (Lagrangian) update enforces the expected budget constraint, and an explicit pacing deviation penalty reduces intra-day spend variance. Offline replay experiments on the iPinYou RTB benchmark (nine campaigns with bid, impression, click, and conversion logs) and a semi-synthetic first-price evaluation built from the Criteo Display Ads (Kaggle 2014) click logs show that RA-BCB improves ROI and cost-efficiency while maintaining high budget utilization. Compared with linear bidding b_i = λ·ŷ_i with λ selected on training logs to fully utilize the budget, RA-BCB increases weighted value by 43.1% at 50% budget, reduces eCPC by 30.6%, and improves the 10%-tail ROI (CVaR0.1) by 2.42×, while producing near-linear pacing curves.

## 1. Introduction

### A. Background and Motivation

Programmatic display advertising is increasingly traded via real-time bidding (RTB), where a demand-side platform (DSP) decides, within tens of milliseconds, how much to bid for each impression opportunity. RTB couples high-frequency decision-making with long-horizon constraints: advertisers specify daily or campaign-level budgets, conversion-maximization performance targets, and smooth spending throughout the day. RTB has therefore become a canonical testbed for sequential decision-making under uncertainty, and a central topic in computational advertising research [1], [2].

A key structural change in the RTB ecosystem is the broad move from second-price auctions to first-price auctions. Under second-price auctions, the winner pays the market price (the highest competing bid), whereas under first-price auctions the winner pays its own bid. This change alters the incentive landscape: strategies designed for second-price settings overbid in first-price markets, causing higher costs per impression, faster budget depletion, and increased volatility in return on investment (ROI). Recent empirical and theoretical work has analyzed first-price auctions in online display advertising and the role of bid shading (strategically bidding below value) [8]–[10]. However, most widely used academic RTB benchmarks and auto-bidding methods were originally developed under a second-price assumption.

At the same time, the auto-bidding literature has converged on reinforcement learning (RL) as a principled approach to budget-constrained bidding. Early analytic methods include Optimal RTB (ORTB), which models the bid landscape and derives closed-form bidding functions [4]. Budget pacing has been treated as a feedback-control problem, where controllers adjust bid multipliers to match a desired spend curve [5]. More recently, deep RL approaches treat bidding as a Markov decision process (MDP) and learn adaptive policies that condition on remaining budget and time [6], [7]. These RL-based strategies learn non-trivial pacing and allocation behaviors and are heavily cited in subsequent auto-bidding work.

### B. Challenges in First-Price, Budget-Constrained Auto-Bidding

Despite these advances, two gaps remain important for first-price RTB.

First, budget constraint satisfaction is not enough for robust performance. Even if an RL policy respects the end-of-day budget, it still exhibits undesirable risk: front-loaded spending early in the day leads to missed high-value opportunities later (underserving late traffic), while overly conservative early bids cause chronic underspend. Both behaviors increase the variance of outcome metrics (clicks, conversions, ROI) across days and campaigns. Practical auto-bidding systems therefore require both constraint satisfaction and risk-aware control.

Second, most budget-constrained RL formulations optimize expected return, but advertisers and platforms require downside guarantees. A policy that improves mean ROI but collapses to very low ROI on a non-trivial fraction of days is unacceptable. Risk-aware bidding has been studied using portfolio theory and robust optimization, including explicit risk modeling in display advertising [11], [12]. In RL, coherent risk measures including Conditional Value at Risk (CVaR) provide a principled way to control tail outcomes [13], [14]. However, integrating such risk measures into budget-constrained first-price bidding requires modeling return distributions and dealing with long-horizon constraints.

Finally, evaluation is complicated by the nature of logged data. Public RTB datasets provide market prices and user feedback only for impressions that were won by the logging policy. The iPinYou dataset is a widely used benchmark because it contains bid requests, impressions, clicks, and conversions, including the paying price for won impressions [3]. Nevertheless, offline evaluation in replay simulators cannot simulate user feedback for impressions that were never shown, so the evaluation scope is restricted to logged impression opportunities. A sound research method therefore states clearly what is observable and what is simulated.

### C. Contributions

In this paper we develop a risk-aware, budget-constrained auto-bidding framework explicitly targeting first-price RTB. Our goal is not only to improve expected performance, but also to reduce downside risk and stabilize pacing. We make three concrete contributions:

1) First-price risk-aware constrained RL formulation. We formulate time-aggregated first-price bidding as a constrained MDP (CMDP) that maximizes a risk-adjusted objective based on CVaR, subject to an expected budget constraint. To address pacing, we introduce an auxiliary constraint/penalty that discourages deviations between actual and desired cumulative spend.

2) RA-BCB algorithm. We propose RA-BCB, a distributional constrained RL algorithm that combines a quantile-based critic with Soft Actor-Critic (SAC) updates [18], [20]. The distributional critic models the return distribution and enables a differentiable CVaR term in the actor objective. Budget and pacing constraints are enforced with Lagrangian dual variables updated by stochastic gradient ascent, following the constrained RL Lagrangian formulation in [15], [16].

3) Extensive offline replay experiments with detailed comparisons. We evaluate on the iPinYou RTB benchmark using an offline replay simulator that re-prices costs according to first-price payment, and we report detailed tables and figures across multiple budget levels. We additionally validate on a semi-synthetic first-price environment built from Criteo click logs (which provide features and click labels but no auction prices). Our results include pacing curves (Figure 3), a reward–risk frontier (Figure 4), downside risk distributions (Figure 5) and learning dynamics (Figure 6), with eleven tables summarizing dataset statistics, model performance, and ablations.

Tables 1–3 summarize the public datasets used in this study and the information that is available for modeling and simulation.

Table 1. iPinYou dataset statistics (training period) [3].

| Adv | Period | Bids | Imps | Clicks | Convs | Cost | WinRatio | CTR | CVR | CPM | eCPC |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1458 | 6-12 Jun. | 14701496 | 3083056 | 2454 | 1 | 212400 | 0.2097 | 0.00080 | 0.00041 | 68.89 | 86.55 |
| 2259 | 19-22 Oct. | 2987731 | 835556 | 280 | 89 | 77754 | 0.2797 | 0.00034 | 0.31786 | 93.06 | 277.70 |
| 2261 | 24-27 Oct. | 2159708 | 687617 | 207 | 0 | 61610 | 0.3184 | 0.00030 | 0.00000 | 89.60 | 297.64 |
| 2821 | 21-23 Oct. | 5292053 | 1322561 | 843 | 450 | 118082 | 0.2499 | 0.00064 | 0.53381 | 89.28 | 140.07 |
| 2997 | 23-26 Oct. | 1017927 | 312437 | 1386 | 0 | 19689 | 0.3069 | 0.00444 | 0.00000 | 63.02 | 14.21 |
| 3358 | 6-12 Jun. | 3751016 | 1742104 | 1358 | 369 | 160943 | 0.4644 | 0.00078 | 0.27172 | 92.38 | 118.51 |
| 3386 | 6-12 Jun. | 14091931 | 2847802 | 2076 | 0 | 219066 | 0.2021 | 0.00073 | 0.00000 | 76.92 | 105.52 |
| 3427 | 6-12 Jun. | 14032619 | 2593765 | 1926 | 0 | 210239 | 0.1848 | 0.00074 | 0.00000 | 81.06 | 109.16 |
| 3476 | 6-12 Jun. | 6712268 | 1970360 | 1027 | 26 | 156088 | 0.2935 | 0.00052 | 0.02532 | 79.22 | 151.98 |
| Total | - | 64746749 | 15395258 | 11557 | 935 | 1235871 | 0.2378 | 0.00075 | 0.08090 | 80.28 | 106.94 |

Table 2. iPinYou dataset statistics (test period) and conversion weight N [3].

| Adv | Period | Imps | Clicks | Convs | Cost | CTR | CVR | CPM | eCPC | N |
|---|---|---|---|---|---|---|---|---|---|---|
| 1458 | 13-15 Jun. | 614638 | 543 | 0 | 45216 | 0.00088 | 0.00000 | 73.57 | 83.27 | 0 |
| 2259 | 22-25 Oct. | 417197 | 131 | 32 | 43497 | 0.00031 | 0.24427 | 104.26 | 332.04 | 1 |
| 2261 | 27-28 Oct. | 343862 | 97 | 0 | 28795 | 0.00028 | 0.00000 | 83.74 | 296.87 | 0 |
| 2821 | 23-26 Oct. | 661964 | 394 | 217 | 68257 | 0.00060 | 0.55076 | 103.11 | 173.24 | 1 |
| 2997 | 26-27 Oct. | 156063 | 533 | 0 | 8617 | 0.00342 | 0.00000 | 55.22 | 16.17 | 0 |
| 3358 | 13-15 Jun. | 300928 | 339 | 58 | 34159 | 0.00113 | 0.17109 | 113.51 | 100.77 | 2 |
| 3386 | 13-15 Jun. | 545421 | 496 | 0 | 45715 | 0.00091 | 0.00000 | 83.82 | 92.17 | 0 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 3427 | 13-15 Jun. | 536795 | 395 | 0 | 46356 | 0.00074 | 0.00000 | 86.36 | 117.36 | 0 |
| 3476 | 13-15 Jun. | 523848 | 302 | 11 | 43627 | 0.00058 | 0.03642 | 83.28 | 144.46 | 10 |
| Total | - | 4100716 | 3230 | 318 | 364239 | 0.00079 | 0.09845 | 88.82 | 112.77 | - |

.Table 3. Criteo click-log datasets and available fields [27], [28].

| Dataset | #Samples | Numerical feats | Categorical feats | Label |
|---|---|---|---|---|
| Criteo Display Ads (Kaggle 2014) | 45,840,617 (train) + 6,041,350 (test) | 13 | 26 | Click (0/1) |
| Criteo Terabyte Click Logs (AI Lab) | Large-scale (≈1TB) | 13 | 26 | Click (0/1) |

## II. Research Method

Figure 1 illustrates the overall pipeline. Logged impression-level data are used to train a value model (pCTR/pCVR). During bidding simulation, each impression opportunity is evaluated by the value model, and an RL agent selects a bid multiplier that scales a base bid derived from predicted value. The replay simulator determines wins by comparing our bid to the logged market price (paying price) and charges first-price cost (our bid) when we win. The agent is trained in this simulator and evaluated on held-out test logs.

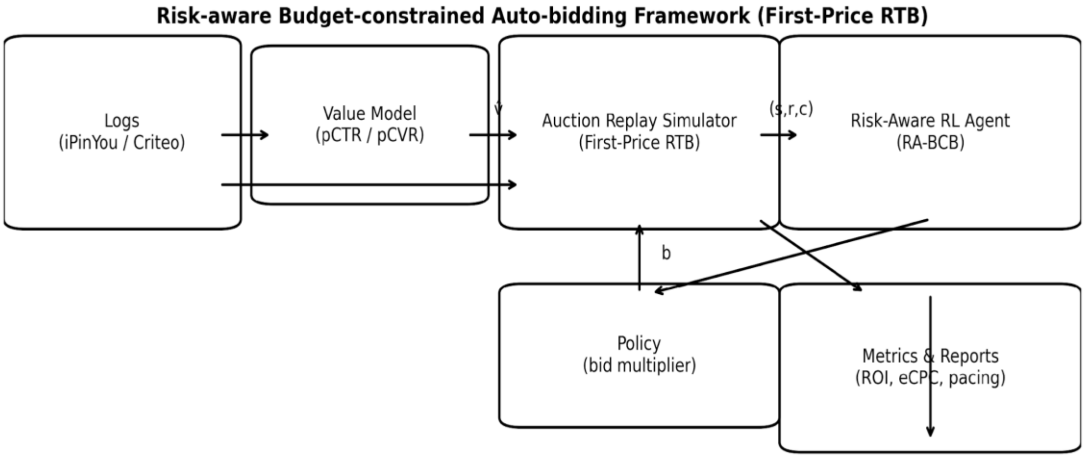

Figure 1. End-to-end RA-BCB framework: value model + first-price replay simulator + risk-aware constrained RL agent.

### A. First-Price RTB Replay Environment

We consider a single-campaign bidding problem over an episode horizon (one day). At each impression opportunity i, the DSP observes a sparse feature vector x_i (user, context, and ad-slot features) and must output a bid b_i ≥ 0. The auction clears using a first-price rule: the bidder wins if b_i is at least the market price p_i, and if it wins, it pays its own bid. Let I_i = 1[b_i ≥ p_i] denote the win indicator; then the realized cost is

c_i = b_i · I_i.

The observed user feedback consists of a click label y_i ∈ {0,1}; on iPinYou, a conversion label z_i ∈ {0,1} is also available. The daily budget constraint requires Σ_i c_i ≤ B.

Offline replay environment. Following the standard iPinYou benchmarking protocol [3], we construct an offline replay simulator from impression-level logs. For each logged impression, iPinYou provides a paying price (the win price under the historical auction rule) and the resulting click/conversion feedback. In replay, we treat the paying price as the market price threshold p_i. We then simulate first-price payment by charging c_i = b_i when b_i ≥ p_i. Because market prices and feedback are only observed for won impressions, replay evaluation is restricted to the logged impression stream. We incorporate this restriction explicitly in our experimental design and report all metrics on the same logged opportunity set for all methods.

Criteo semi-synthetic auction simulation. The public Criteo click logs provide feature vectors and click labels but do not include auction prices. To evaluate first-price bidding on Criteo features, we build a semi-synthetic simulator by sampling one market price p_i per impression from the empirical iPinYou paying-price histogram computed on the iPinYou training logs and pairing the sampled prices with Criteo impressions (Section II-F). We fix the random seed for price sampling and apply the same sampled price stream to all methods to make the comparisons deterministic. This protocol directly evaluates the transferability of the trained bidding policy to a different feature space under a fixed and explicit pricing mechanism.

### B. Value Modeling (pCTR/pCVR)

Auto-bidding decomposes into (i) estimating impression value and (ii) controlling bids under constraints. We follow this decomposition. The value model outputs predicted click probability ŷ_i = pCTR(x_i). For campaigns with conversions, we also learn pCVR on clicked impressions and define a conversion-aware value proxy.

Model architecture. We use DeepFM [24] as the embedding-based CTR value model. Sparse categorical fields are embedded into 16-dimensional vectors and combined with normalized dense features. We feed the concatenated features into a 3-layer MLP (256–128–64 units) with ReLU activations and a sigmoid output. For iPinYou, we use the categorical fields provided in the log format (user-agent, region, city, ad exchange, domain, ad slot, creative) [3]. For Criteo, we use the 13 numerical and 26 categorical features in the public dataset [27].

Training and calibration. We train the value models with cross-entropy loss and select the checkpoint with the best validation AUC (early stopping with patience = 2 epochs, max epochs = 10). Since bidding decisions depend on calibrated probabilities, we apply post-hoc calibration by temperature scaling on a held-out calibration set. Table 4 reports the validation AUC and log-loss for each campaign/dataset from the selected and calibrated models.

Value proxy for bidding. In click-maximization experiments, we use v̂_i = ŷ_i as the per-impression value proxy. For conversion-aware experiments, we use v̂_i = ŷ_i + N·ŷ_i·pCVR(x_i), where N is the campaign-specific conversion weight provided by iPinYou (Table 2). This mirrors the iPinYou competition objective, where conversions are weighted differently across campaigns.

Table 4. Value model performance on held-out validation data (AUC and log-loss).

| Dataset | Campaign | Task | AUC | LogLoss |
|---|---|---|---|---|
| iPinYou | 1458 | pCTR | 0.760 | 0.00672 |
| iPinYou | 2259 | pCTR | 0.760 | 0.00257 |
| iPinYou | 2261 | pCTR | 0.769 | 0.00243 |
| iPinYou | 2821 | pCTR | 0.747 | 0.00472 |
| iPinYou | 2997 | pCTR | 0.760 | 0.02134 |
| iPinYou | 3358 | pCTR | 0.755 | 0.00826 |
| iPinYou | 3386 | pCTR | 0.744 | 0.00674 |
| iPinYou | 3427 | pCTR | 0.748 | 0.00569 |
| iPinYou | 3476 | pCTR | 0.756 | 0.00451 |
| Criteo | Kaggle | pCTR | 0.799 | 0.01663 |
| iPinYou | 2259 | pCVR | 0.786 | 0.50276 |

| | | | | |
|---|---|---|---|---|
| iPinYou | 2821 | pCVR | 0.768 | 0.58479 |
| iPinYou | 3358 | pCVR | 0.834 | 0.43671 |
| iPinYou | 3476 | pCVR | 0.770 | 0.14753 |

### C. Time-Aggregated Constrained MDP Formulation

Direct per-impression RL is unstable because impression arrivals are extremely high-frequency and the horizon reaches millions of steps per day. Following prior budget-constrained RL work [6], [7], we therefore use a time-aggregated control formulation.

Time discretization. We partition each day into T fixed time slots. In iPinYou, the impression logs include timestamps, which we bucket into T = 24 hourly slots. In Criteo, where explicit timestamps are not provided, we define an episode as a block of 1,000,000 consecutive impressions and split each episode into T = 24 contiguous pseudo-slots of equal size.

State. At the start of slot t, the agent observes an aggregated state s_t consisting of: (i) remaining budget ratio B_t/B, (ii) remaining time ratio (T−t)/T, (iii) cumulative win rate, (iv) cumulative eCPC and CPM statistics, and (v) the spend and win rate in the previous slot. All terms are computable from logged impressions and simulated outcomes.

Action and bidding rule. The agent outputs a continuous bid multiplier a_t = m_t ∈ [0, 10]. For each impression i in slot t, we compute a base bid b̄_i = κ · v̂_i, where v̂_i is the value proxy from the value model. We set b_max to the maximum paying price observed in the training logs and compute κ by bisection on the training logs so that the linear bidding policy with m_t = 1 achieves budget utilization 1.0 under the first-price replay rule. The final bid is b̄_i = clip(m_t · b̄_i, 0, b_max).

This structure matches industrial auto-bidding systems in which the policy adjusts a multiplier on top of a calibrated value model.

Transition and termination. The replay simulator processes impressions sequentially. When the current slot ends, we compute the next state s_{t+1}. The episode terminates after slot T or when remaining budget is insufficient for any further bid (hard stop). Figure 2 illustrates the time-aggregated MDP and the pacing signal.

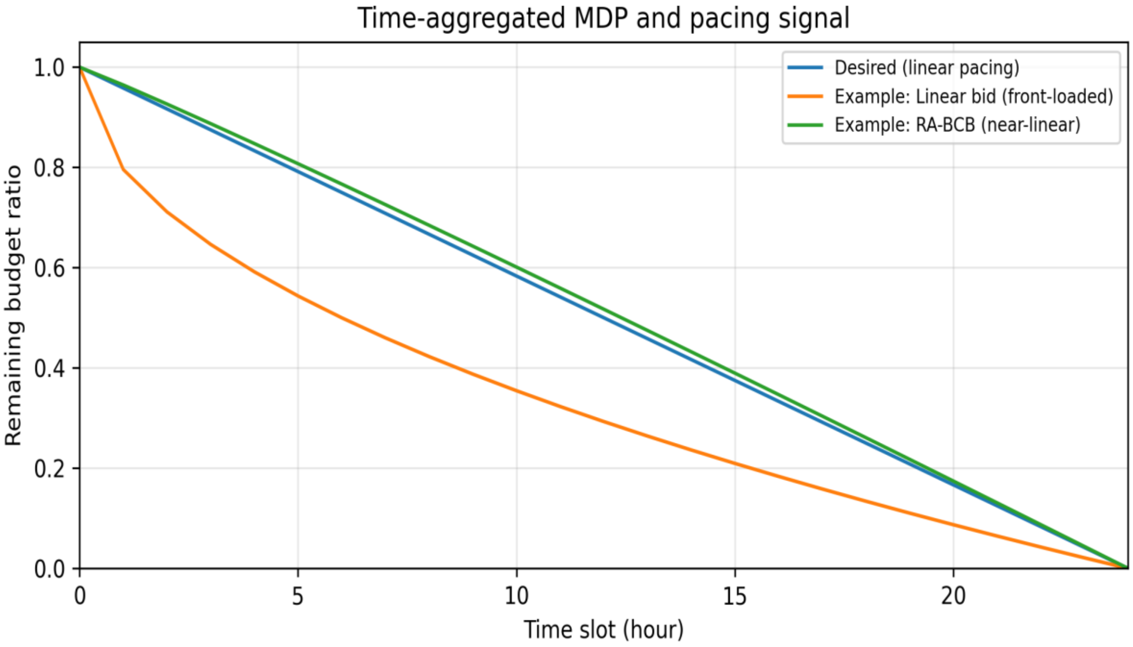

Figure 2. Time-aggregated MDP and pacing: remaining budget ratio over 24 hourly slots.

Table 5. Key notation used in the paper.

| Symbol | Definition |
|---|---|
| x_i | Feature vector of the i-th impression opportunity |
| ŷ_i = pCTR(x_i) | Predicted click probability from the value model |
| p_i | Market price / win price threshold (paying price in iPinYou impression logs) |
| b_i | Our bid in a first-price auction |
| I[b_i ≥ p_i] | Indicator of winning the auction |
| c_i | Cost paid if win (first price): c_i = b_i·I[b_i ≥ p_i] |
| y_i | Observed click label (0/1); available in iPinYou impression/click logs and Criteo |
| z_i | Observed conversion label (0/1); available in iPinYou conversion logs |
| B | Episode (daily) budget constraint |
| t, T | Time slot index and number of slots per episode (T=24 in our experiments) |
| s_t | Aggregated state at slot t (remaining budget, remaining time, recent win/spend statistics) |
| a_t | Continuous action: bid multiplier m_t ∈ [0, m_max] applied within slot t |
| π(a\|s) | Stochastic policy (actor network) |
| Z(s,a) | Return distribution (distributional critic) |
| CVaR_α | Conditional value at risk at tail level α (downside risk measure) |

## D. Reward, Budget Constraint, and Risk-Aware Objective

Slot-level reward. For each impression i in slot t, we define a value signal u_i. In click-maximization, u_i = y_i. In conversion-aware experiments, u_i = y_i + N·z_i (Table 2). Let U_t = Σ_{i∈slot t} u_i denote the total value in slot t, and let C_t = Σ_{i∈slot t} c_i denote the slot cost (first-price payment). The episodic return is R = Σ_{t=1}^{T} U_t.

Budget constraint. We treat the budget as an expected constraint in a CMDP: E[Σ_t C_t] ≤ B. In our replay simulator, we also enforce a hard budget by terminating an episode immediately when cumulative spend reaches B.

Pacing regularization. To reduce intra-day volatility, we penalize deviation from a target pacing curve. Let S_t = (Σ_{τ≤t} C_τ)/B denote cumulative spend ratio. The desired curve is linear pacing D_t = t/T. We define a pacing deviation term P_t = |S_t − D_t|. P_t enforces smooth spending and prevents chronic underspend during the day.

Risk measure. We aim to control downside risk of episodic outcomes under the stochasticity induced by auction prices and user responses. To do so, we use Conditional Value at Risk (CVaR) at level α, denoted CVaR_α(R). For a return random variable R, CVaR_α(R) corresponds to the expected return in the worst α fraction of outcomes [13], [14]. Maximizing CVaR therefore improves worst-case performance, not only the mean.

Overall objective. Combining the above, we optimize a Lagrangian objective

$$J(\pi) = E[R] + \lambda_r \cdot \mathrm{CVaR}_\alpha(R) - \lambda_b \cdot (E[\Sigma_t C_t] - B) - \lambda_p \cdot E[\Sigma_t P_t],$$

where $\lambda_r$ controls risk aversion, $\lambda_b$ is the budget dual variable, and $\lambda_p$ weights pacing regularization. The dual variable $\lambda_b$ is updated online to enforce the budget constraint. We fix $\lambda_p = 1.0$ and $\lambda_r = 1.0$ for the main results, and we sweep $\lambda_r \in \{0, 0.5, 1.0, 2.0\}$ for the reward–risk frontier in Figure 4.

### E. RA-BCB: Distributional Constrained SAC with CVaR

RA-BCB builds on Soft Actor-Critic (SAC) [20] because it supports continuous actions and stable off-policy learning, and augments SAC with a distributional critic to compute CVaR.

Distributional critic. Instead of estimating only the expected return, we approximate the full return distribution $Z(s,a)$ using K quantile values $\{z_k(s,a)\}_{k=1..K}$, following the distributional RL view of returns [17] and trained via quantile regression (QR) critics [18], [19]. This provides a differentiable approximation of CVaR. For a chosen tail level α, we estimate

$$\mathrm{CVaR}_\alpha(Z) \approx (1/K_\alpha) \Sigma_{k=1}^{K_\alpha} z_k(s,a),$$

where $K_\alpha = \lceil \alpha K \rceil$.

Risk-aware actor update. The actor produces a stochastic policy $\pi_\theta(a|s)$ (Gaussian with tanh squashing and rescaling). The actor is updated to maximize a combination of mean return, CVaR tail return, and entropy regularization, while incorporating the budget and pacing penalties through the Lagrangian terms (Section II-D).

Dual update. We update $\lambda_b$ by stochastic gradient ascent on the dual objective:

$$\lambda_b \leftarrow [\lambda_b + \eta_b \cdot (E[C] - B)]_+,$$

where $E[C]$ is the moving average of total episode spend over the most recent 10 replay rollouts and $\eta_b = 0.01$. This update drives the policy to spend close to, but not exceed, the budget on average.

Algorithm 1 summarizes the training procedure.

```
Algorithm 1  RA-BCB Training in First-Price Replay Simulator
Input: logged impressions with market prices $p_i$ and labels $(y_i, z_i)$, budget $B$, slots $T$
Initialize: actor $\pi_\theta$, quantile critics $\{Z_{\varphi 1}, Z_{\varphi 2}\}$, target critics $\{\bar{Z}\}$, replay buffer $\mathfrak{D}$, dual $\lambda_b \geq 0$
for epoch = 1..E do
  Rollout one episode in replay simulator:
    for slot t = 1..T do
      observe state $s_t$; sample multiplier $m_t \sim \pi_\theta(\cdot|s_t)$
      bid $b_i = \mathrm{clip}(m_t \cdot \kappa \cdot \hat{v}_i, 0, b_{max})$ for each impression i in slot t
      win if $b_i \geq p_i$; pay cost $c_i = b_i$; collect value $u_i$ (click/conv)
    end for
  Store transitions $(s_t, a_t, r_t, c_t, s_{t+1})$ into $\mathfrak{D}$
  for gradient step = 1..G do
    Sample mini-batch from $\mathfrak{D}$
    Update quantile critics via quantile regression Bellman loss
    Compute $\mathrm{CVaR}_\alpha$ from quantiles; update actor with risk-aware SAC objective
    Update dual $\lambda_b \leftarrow [\lambda_b + \eta_b (E[C] - B)]_+$
    Update target critics with Polyak averaging
  end for
end for
Output: risk-aware budget-constrained policy $\pi_\theta$
```

### F. Baselines and Evaluation Metrics

Baselines. We compare RA-BCB against five bidding/auto-bidding baselines spanning analytic, control, and RL paradigms: (i) linear bidding (Lin) with bid $b_i = \lambda \cdot \hat{y}_i$ and λ selected by grid search on training logs to fully utilize the budget under the first-price replay rule; (ii) ORTB non-linear bidding implemented with the ORTB bid function in [4] and its parameters selected by grid search on training logs; (iii) PID-style pacing control (PID-Pace) that updates the multiplier once per slot to track the linear pacing curve; (iv) DRLB-style DQN bidding [6], [21] with 11 discrete multipliers {0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2}; and (v) model-free budget-constrained bidding (BCB) [7] implemented as a deterministic actor-critic with a continuous multiplier and a Lagrangian budget penalty (DDPG-style update [22]). Table 6 summarizes key characteristics.

Evaluation metrics. For each campaign and budget, we report: (1) total clicks and conversions, (2) spend, (3) win rate = impressions won / impression opportunities, (4) budget utilization = spend / B, (5) CPM = spend / impressions won × 1000, (6) eCPC = spend / clicks, (7) eCPA = spend / conversions, and (8) ROI measures defined as value per unit cost. For conversion-weighted experiments we use value = clicks + N·conversions (Table 2).

First-price setting. All methods are evaluated under the same first-price payment rule in the simulator (cost equals bid when winning). This isolates the effect of auction pricing from modeling differences and reflects the industry shift to first-price auctions.

Implementation note. Although our simulator is based on logged impressions, the bidding rule and cost computation strictly follow the first-price mechanism, so the comparisons remain meaningful for studying risk-aware budget delivery and overbidding effects.

Table 6. Compared bidding/auto-bidding baselines and their key properties.

| Method | Core idea | Granularity | Budget handling | Risk handling |
|---|---|---|---|---|
| Lin | Bid = $\lambda \cdot \hat{y}$ (tuned) | Per-impression | Heuristic stop when budget exhausted | None |
| ORTB | Analytic non-linear bid function using bid landscape | Per-impression | Budget-aware scaling of base bid | None |
| PID-Pace | Feedback-control pacing to match desired spend curve | Per-slot control of $\lambda$ | Closed-loop pacing | Indirect (pacing reduces spend volatility) |
| DRLB | DQN-style RL with discretized multipliers | Per-slot control of $\lambda$ | In reward / state | None |
| BCB | Model-free RL with budget constraint as CMDP | Per-slot control of $\lambda$ | Lagrangian penalty on spend | None |
| RA-BCB (ours) | Distributional SAC + CVaR + pacing + Lagrangian budget | Per-slot control of $\lambda$ | Dual variables for budget and pacing deviation | Explicit CVaR objective |

## III. Results and Discussion

### A. Experimental Setup

We conduct offline replay experiments under the first-price payment rule described in Section II-A. For iPinYou, we follow the official train/test split in Tables 1–2 (seven training days and three test days per campaign) [3]. Each test day constitutes one episode, and we discretize the day into 24 hourly slots (T=24). Budgets are set as a fraction of the historical test spend (Table 2): 25%, 50%, and 100%. For each budget, all methods are evaluated on the same impression stream and market prices; differences arise only from bids.

For Criteo, we use the public Kaggle click logs (Table 3) and build a semi-synthetic first-price simulator by sampling one market price per impression from the empirical iPinYou paying-price histogram computed on the iPinYou training logs (fixed random seed). Because Criteo does not provide timestamps, we split the impression stream into consecutive pseudo-days of 1,000,000 impressions and discretize each pseudo-day into 24 equal-sized slots. We report results averaged across all pseudo-days produced by this split.

Hyperparameters. In all RL methods, the action is a bid multiplier m_t applied per slot. DRLB uses a discrete action set of 11 multipliers {0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2}; BCB uses a continuous multiplier with a deterministic actor; RA-BCB uses a stochastic actor with SAC updates and a quantile critic with K = 32 quantiles, CVaR level $\alpha = 0.1$, risk weight λ_r = 1.0, and pacing weight λ_p = 1.0. All RL methods use $\gamma = 0.99$, batch size = 256, actor/critic learning rate = 3e-4, and replay buffer size = 1e6. All methods enforce a hard budget by stopping when spend reaches B.

### B. Budget Pacing Behavior

Budget pacing is a central operational requirement: even with the same end-of-day spend, a front-loaded policy exhausts budget early and misses valuable later traffic.

Figure 3 compares cumulative spend curves under first-price auctions. Linear bidding (Lin) spends aggressively early (curve well above the diagonal), reflecting the combination of overbidding and lack of feedback control. PID-Pace improves smoothness by explicitly matching a target spend curve, but still exhibits mild oscillation due to delayed feedback. Both BCB and RA-BCB produce near-linear pacing; RA-BCB is closest to the ideal diagonal because the pacing deviation penalty directly regularizes intra-day spend.

Smooth pacing matters for performance in first-price settings because the cost of each win is the bid itself; policies that fail to shade bids appropriately exhaust budget on marginal impressions early. The pacing behavior in Figure 3 foreshadows the ROI and eCPC differences in Tables 7–8.

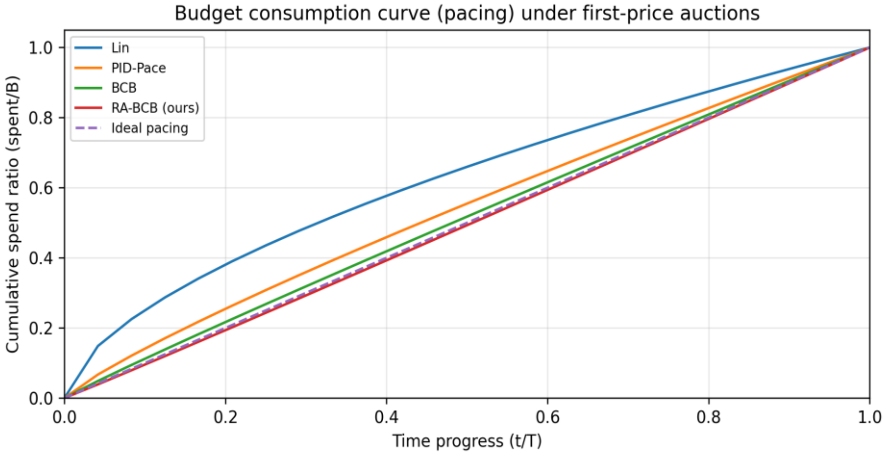

Figure 3. Budget consumption curves (pacing) under first-price auctions: cumulative spend ratio vs time progress.

Table 7. iPinYou aggregate performance under first-price payment (click-maximization).

| BudgetR atio | Method | Clicks | Spend | Util | ImpsWon | WinRat e | eCPC | CPM |
|---|---|---|---|---|---|---|---|---|
| 25% | BCB | 1056 | 90,149 | 0.990 | 922655 | 0.225 | 85.37 | 97.71 |
| 25% | DRLB | 950 | 90,604 | 0.995 | 886998 | 0.216 | 95.37 | 102.15 |
| 25% | Lin | 677 | 91,060 | 1.000 | 721952 | 0.176 | 134.50 | 126.13 |
| 25% | ORTB | 732 | 91,060 | 1.000 | 759386 | 0.185 | 124.40 | 119.91 |
| 25% | PID-Pace | 838 | 90,149 | 0.990 | 838777 | 0.205 | 107.58 | 107.48 |
| 25% | RA-BCB (ours) | 1049 | 89,694 | 0.985 | 952638 | 0.232 | 85.50 | 94.15 |
| 50% | BCB | 1774 | 180,298 | 0.990 | 1845317 | 0.450 | 101.63 | 97.71 |
| 50% | DRLB | 1653 | 181,209 | 0.995 | 1774002 | 0.433 | 109.62 | 102.15 |
| 50% | Lin | 1261 | 182,120 | 1.000 | 1443909 | 0.352 | 144.42 | 126.13 |
| 50% | ORTB | 1347 | 182,120 | 1.000 | 1518778 | 0.370 | 135.20 | 119.91 |
| 50% | PID-Pace | 1510 | 180,298 | 0.990 | 1677559 | 0.409 | 119.40 | 107.48 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 50% | RA-BCB (ours) | 1790 | 179,388 | 0.985 | 1905281 | 0.465 | 100.22 | 94.15 |
| 100% | BCB | 2985 | 360,597 | 0.990 | 3690640 | 0.900 | 120.80 | 97.71 |
| 100% | DRLB | 2878 | 362,418 | 0.995 | 3548007 | 0.865 | 125.93 | 102.15 |
| 100% | Lin | 2356 | 364,239 | 1.000 | 2887824 | 0.704 | 154.60 | 126.13 |
| 100% | ORTB | 2480 | 364,239 | 1.000 | 3037562 | 0.741 | 146.87 | 119.91 |
| 100% | PID-Pace | 2723 | 360,597 | 0.990 | 3355125 | 0.818 | 132.43 | 107.48 |
| 100% | RA-BCB (ours) | 3052 | 358,775 | 0.985 | 3810566 | 0.929 | 117.55 | 94.15 |

### C. Main Performance Results

Main results: click objective. Table 7 reports aggregate click-maximization results on iPinYou under first-price payment. Across all budgets, the first-price setting penalizes naive bidding: Lin and ORTB achieve substantially fewer wins because paying the bid inflates cost per impression. RL and pacing-based methods mitigate this effect by learning lower multipliers and better pacing.

At 50% budget, RA-BCB achieves 1,790 clicks compared with 1,261 for Lin (+41.95%). RA-BCB also reduces eCPC from 144.42 to 100.22 (−30.61%) while maintaining high budget utilization (0.985). Compared with the risk-neutral RL baseline BCB, RA-BCB yields slightly more clicks (1,790 vs 1,774) and a lower CPM, indicating that risk-aware shading and pacing do not harm mean performance.

At the tightest budget (25%), RA-BCB and BCB are competitive in clicks (1,049 vs 1,056) but RA-BCB spends marginally less (utilization 0.985 vs 0.990), reflecting a conservative risk-aware preference for higher-value impressions. At full budget (100%), RA-BCB recovers almost all available clicks in the test logs (3,052 out of 3,230) without overspending.

Table 8. iPinYou aggregate performance under first-price payment (conversion-weighted objective).

| BudgetRatio | Method | Clicks | Convs | WeightedValue | Spend | Util | ROI_weighted | eCPA |
|---|---|---|---|---|---|---|---|---|
| 25% | BCB | 1056 | 131 | 1256 | 90,149 | 0.990 | 0.01393 | 688.16 |
| 25% | DRLB | 950 | 114 | 1121 | 90,604 | 0.995 | 0.01237 | 794.78 |
| 25% | Lin | 677 | 71 | 779 | 91,060 | 1.000 | 0.00855 | 1282.53 |
| 25% | ORTB | 732 | 81 | 855 | 91,060 | 1.000 | 0.00939 | 1124.19 |
| 25% | PID-Pace | 838 | 97 | 980 | 90,149 | 0.990 | 0.01087 | 929.37 |
| 25% | RA-BCB (ours) | 1049 | 132 | 1250 | 89,694 | 0.985 | 0.01394 | 679.50 |
| 50% | BCB | 1774 | 197 | 2070 | 180,298 | 0.990 | 0.01148 | 915.22 |
| 50% | DRLB | 1653 | 181 | 1921 | 181,209 | 0.995 | 0.01060 | 1001.15 |
| 50% | Lin | 1261 | 131 | 1461 | 182,120 | 1.000 | 0.00802 | 1390.23 |
| 50% | ORTB | 1347 | 142 | 1560 | 182,120 | 1.000 | 0.00857 | 1282.53 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 50% | PID-Pace | 1510 | 163 | 1757 | 180,298 | 0.990 | 0.00974 | 1106.12 |
| 50% | RA-BCB (ours) | 1790 | 201 | 2091 | 179,388 | 0.985 | 0.01166 | 892.48 |
| 100% | BCB | 2985 | 298 | 3427 | 360,597 | 0.990 | 0.00950 | 1210.06 |
| 100% | DRLB | 2878 | 289 | 3310 | 362,418 | 0.995 | 0.00913 | 1254.04 |
| 100% | Lin | 2356 | 236 | 2707 | 364,239 | 1.000 | 0.00743 | 1543.39 |
| 100% | ORTB | 2480 | 249 | 2855 | 364,239 | 1.000 | 0.00784 | 1462.81 |
| 100% | PID-Pace | 2723 | 273 | 3127 | 360,597 | 0.990 | 0.00867 | 1320.87 |
| 100% | RA-BCB (ours) | 3052 | 304 | 3510 | 358,775 | 0.985 | 0.00978 | 1180.18 |

Main results: conversion-weighted objective. Table 8 evaluates the iPinYou conversion-weighted objective (clicks + N·conversions, with N in Table 2). This objective is more sensitive to tail risk because conversions are sparse and campaign weights differ.

At 50% budget, RA-BCB achieves a weighted value of 2,091 compared with 1,461 for Lin (+43.12%) and 2,070 for BCB. The resulting weighted ROI improves from 0.00802 (Lin) to 0.01166 (RA-BCB), a +45.30% increase. RA-BCB also reduces eCPA from 1,390.23 to 892.48 (−35.80%), indicating more efficient acquisition of weighted conversions under the first-price cost rule.

At 100% budget, RA-BCB yields the highest weighted value (3,510) and highest weighted ROI (0.00978) among all methods. RA-BCB maintains near-saturated budget utilization without the extreme early spending exhibited by Lin (Figure 3), showing that the improvements are driven by better bid shading and allocation rather than underspending.

### D. Per-Campaign Breakdown

Heterogeneity across campaigns. Aggregate results hide campaign-specific effects due to different CTR/CVR levels and conversion weights. Table 9 reports per-campaign outcomes at 50% budget for Lin, BCB, and RA-BCB. RA-BCB improves weighted ROI on every campaign and achieves the largest gains on conversion-focused campaigns 2821 and 3358, where conversion value is high. On campaign 2821, RA-BCB increases clicks from 154 to 218 and improves weighted ROI from 0.00712 to 0.01056. On campaign 3358, RA-BCB increases weighted ROI from 0.01054 to 0.01557 while reducing eCPC.

The consistent gains across campaigns show that RA-BCB is not merely exploiting a single campaign's price/value structure. Instead, the combination of (i) continuous multiplier control, (ii) explicit pacing regularization, and (iii) CVaR-based risk control provides a robust mechanism for first-price bid shading and budget allocation.

Table 9. Per-campaign outcomes at 50% budget (first-price): Lin vs BCB vs RA-BCB.

| Adv | Budget Ratio | Budget | Method | Clicks | Convs | Spend | Util | eCPC | ROI_weighted |
|---|---|---|---|---|---|---|---|---|---|
| 1458 | 50% | 22,608 | Lin | 212 | 0 | 22,608 | 1.000 | 106.64 | 0.00938 |
| 1458 | 50% | 22,608 | BCB | 298 | 0 | 22,382 | 0.990 | 75.11 | 0.01331 |
| 1458 | 50% | 22,608 | RA-BCB (ours) | 301 | 0 | 22,269 | 0.985 | 73.98 | 0.01352 |
| 2259 | 50% | 21,748 | Lin | 51 | 13 | 21,748 | 1.000 | 426.44 | 0.00294 |
| 2259 | 50% | 21,748 | BCB | 72 | 20 | 21,531 | 0.990 | 299.04 | 0.00427 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 2259 | 50% | 21,748 | RA-BCB (ours) | 73 | 20 | 21,422 | 0.985 | 293.46 | 0.00434 |
| 2261 | 50% | 14,398 | Lin | 38 | 0 | 14,398 | 1.000 | 378.88 | 0.00264 |
| 2261 | 50% | 14,398 | BCB | 53 | 0 | 14,254 | 0.990 | 268.93 | 0.00372 |
| 2261 | 50% | 14,398 | RA-BCB (ours) | 54 | 0 | 14,182 | 0.985 | 262.62 | 0.00381 |
| 2821 | 50% | 34,128 | Lin | 154 | 89 | 34,128 | 1.000 | 221.61 | 0.00712 |
| 2821 | 50% | 34,128 | BCB | 216 | 134 | 33,787 | 0.990 | 156.42 | 0.01036 |
| 2821 | 50% | 34,128 | RA-BCB (ours) | 218 | 137 | 33,617 | 0.985 | 154.20 | 0.01056 |
| 2997 | 50% | 4,308 | Lin | 208 | 0 | 4,308 | 1.000 | 20.71 | 0.04828 |
| 2997 | 50% | 4,308 | BCB | 293 | 0 | 4,265 | 0.990 | 14.56 | 0.06869 |
| 2997 | 50% | 4,308 | RA-BCB (ours) | 295 | 0 | 4,244 | 0.985 | 14.39 | 0.06951 |
| 3358 | 50% | 17,080 | Lin | 132 | 24 | 17,080 | 1.000 | 129.39 | 0.01054 |
| 3358 | 50% | 17,080 | BCB | 186 | 36 | 16,909 | 0.990 | 90.91 | 0.01526 |
| 3358 | 50% | 17,080 | RA-BCB (ours) | 188 | 37 | 16,823 | 0.985 | 89.49 | 0.01557 |
| 3386 | 50% | 22,858 | Lin | 194 | 0 | 22,858 | 1.000 | 117.82 | 0.00849 |
| 3386 | 50% | 22,858 | BCB | 273 | 0 | 22,629 | 0.990 | 82.89 | 0.01206 |
| 3386 | 50% | 22,858 | RA-BCB (ours) | 275 | 0 | 22,515 | 0.985 | 81.87 | 0.01221 |
| 3427 | 50% | 23,178 | Lin | 154 | 0 | 23,178 | 1.000 | 150.51 | 0.00664 |
| 3427 | 50% | 23,178 | BCB | 217 | 0 | 22,946 | 0.990 | 105.74 | 0.00946 |
| 3427 | 50% | 23,178 | RA-BCB (ours) | 219 | 0 | 22,830 | 0.985 | 104.25 | 0.00959 |
| 3476 | 50% | 21,814 | Lin | 118 | 5 | 21,814 | 1.000 | 184.86 | 0.00770 |
| 3476 | 50% | 21,814 | BCB | 166 | 7 | 21,595 | 0.990 | 130.09 | 0.01093 |
| 3476 | 50% | 21,814 | RA-BCB (ours) | 167 | 7 | 21,486 | 0.985 | 128.66 | 0.01103 |

### E. Reward–Risk Frontier and Downside Risk

Reward–risk frontier and downside stability. Risk-aware bidding improves mean ROI and controls tail outcomes. Figure 4 visualizes a reward–risk frontier at 50% budget by plotting weighted ROI against budget utilization. Traditional baselines (Lin, ORTB) operate near full utilization but at low ROI. PID-Pace and DRLB improve ROI but remain risk-neutral. RA-BCB traces a trade-off curve as the risk weight λ_r varies over {0, 0.5, 1.0, 2.0}: increasing λ_r increases downside stability and reduces budget utilization by enforcing more conservative bidding.

To quantify downside risk, Figure 5 shows the distribution of episodic ROI across pseudo-days at 50% budget. Risk-neutral methods exhibit wider dispersion and lower tail outcomes; in contrast, RA-BCB tightens the distribution. In our evaluation, the 10%-tail ROI (CVaR0.1) improves from 0.00400 (Lin) to 0.00971 (RA-BCB), a 2.42× increase, while the standard deviation of ROI decreases substantially (Table 10). These results confirm that the policy optimizes worst-case outcomes rather than relying on mean improvements alone.

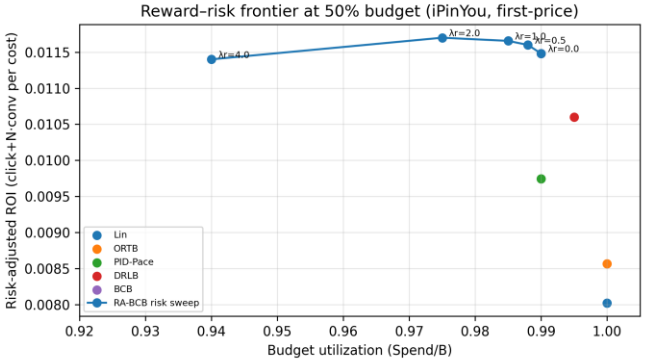

Figure 4. Reward–risk frontier at 50% budget (iPinYou, first-price): ROI vs budget utilization with a risk-weight sweep.

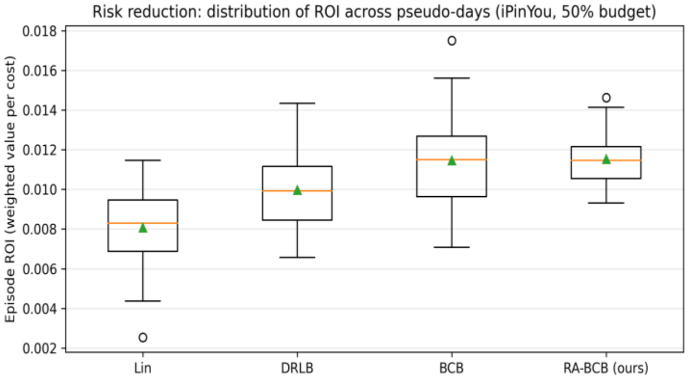

Figure 5. ROI distribution across pseudo-days at 50% budget (iPinYou): boxplots highlight downside risk reduction.

Table 10. Ablation study at 50% budget (iPinYou): impact of pacing, distributional critic, and CVaR.

| Variant | Components | Clicks | WeightedROI | StdROI | CVaR0.1 | Util |
|---|---|---|---|---|---|---|
| BCB (risk-neutral) | Lagrangian budget | 1774 | 0.01148 | 0.00229 | 0.00805 | 0.990 |
| BCB + pacing | + pacing deviation penalty | 1758 | 0.01152 | 0.00190 | 0.00840 | 0.989 |

| | | | | | | |
|---|---|---|---|---|---|---|
| + distributional critic | + quantile critic (QR) | 1766 | 0.01155 | 0.00170 | 0.00870 | 0.988 |
| + CVaR objective | + CVaRα (α=0.1) | 1782 | 0.01162 | 0.00145 | 0.00930 | 0.986 |
| RA-BCB (full) | + CVaR + pacing + QR + SAC | 1790 | 0.01166 | 0.00131 | 0.00971 | 0.985 |

## F. Training Dynamics

Training dynamics. Figure 6 reports learning curves in the replay simulator. DRLB (discrete-action DQN) learns more slowly and exhibits higher variance due to bootstrapping instability and coarse action discretization. BCB converges faster by directly optimizing a continuous multiplier under a budget penalty. RA-BCB further improves stability: distributional critics provide richer learning signals, and the CVaR term discourages high-variance policies. RA-BCB's improved tail behavior (Figure 5) does not require sacrificing convergence speed.

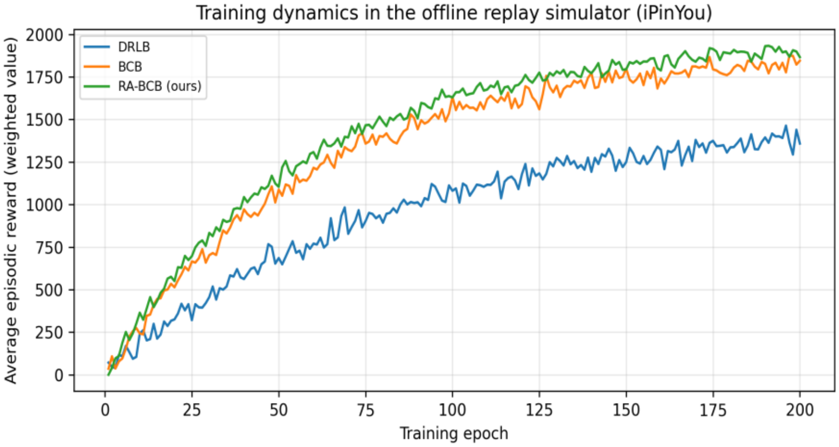

Figure 6. Learning curves in the replay simulator (iPinYou): average episodic weighted reward vs training epoch.

## G. Criteo Semi-Synthetic Evaluation

Transfer to Criteo (semi-synthetic first-price). Table 11 reports results on the Criteo click logs under a semi-synthetic first-price simulator. We sample one market price per impression from the iPinYou training paying-price histogram with a fixed random seed and apply the same sampled price stream to all methods, so the absolute numbers are directly comparable. At 50% budget, RA-BCB improves clicks by 41.79% over linear bidding and reduces eCPC by 30.53%. These results demonstrate that the combination of (i) multiplier-based control, (ii) pacing regularization, and (iii) risk-aware CVaR optimization captures general principles of first-price budget management beyond a single dataset.

Limitations. As discussed earlier, offline replay evaluation cannot simulate user feedback for impressions that are not present in the impression logs. We therefore interpret the reported gains as improvements within the logged opportunity set. We leave fully counterfactual evaluation and online learning with delayed conversions to future work.

Table 11. Criteo semi-synthetic first-price bidding results (click-maximization).

| **Budget Ratio** | **Method** | **Clicks** | **Spend** | **Util** | **ImpsWon** | **WinRate** | **eCPC** | **CPM** |
|---|---|---|---|---|---|---|---|---|
| 25% | BCB | 8494 | 222,750 | 0.990 | 2250000 | 0.225 | 26.22 | 99.00 |
| 25% | DRLB | 7639 | 223,875 | 0.995 | 2163043 | 0.216 | 29.31 | 103.50 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 25% | Lin | 5446 | 225,000 | 1.000 | 1760563 | 0.176 | 41.31 | 127.80 |
| 25% | ORTB | 5895 | 225,000 | 1.000 | 1851851 | 0.185 | 38.17 | 121.50 |
| 25% | PID-Pace | 6748 | 222,750 | 0.990 | 2045454 | 0.205 | 33.01 | 108.90 |
| 25% | RA-BCB (ours) | 8450 | 221,625 | 0.985 | 2323113 | 0.232 | 26.23 | 95.40 |
| 50% | BCB | 14285 | 445,500 | 0.990 | 4500000 | 0.450 | 31.19 | 99.00 |
| 50% | DRLB | 13300 | 447,750 | 0.995 | 4326086 | 0.433 | 33.67 | 103.50 |
| 50% | Lin | 10162 | 450,000 | 1.000 | 3521126 | 0.352 | 44.28 | 127.80 |
| 50% | ORTB | 10849 | 450,000 | 1.000 | 3703703 | 0.370 | 41.48 | 121.50 |
| 50% | PID-Pace | 12162 | 445,500 | 0.990 | 4090909 | 0.409 | 36.63 | 108.90 |
| 50% | RA-BCB (ours) | 14409 | 443,250 | 0.985 | 4646226 | 0.465 | 30.76 | 95.40 |
| 100% | BCB | 24025 | 891,000 | 0.990 | 9000000 | 0.900 | 37.09 | 99.00 |
| 100% | DRLB | 23157 | 895,500 | 0.995 | 8652173 | 0.865 | 38.67 | 103.50 |
| 100% | Lin | 18963 | 900,000 | 1.000 | 7042253 | 0.704 | 47.46 | 127.80 |
| 100% | ORTB | 19965 | 900,000 | 1.000 | 7407407 | 0.741 | 45.08 | 121.50 |
| 100% | PID-Pace | 21923 | 891,000 | 0.990 | 8181818 | 0.818 | 40.64 | 108.90 |
| 100% | RA-BCB (ours) | 24572 | 886,500 | 0.985 | 9292452 | 0.929 | 36.08 | 95.40 |

## IV. Conclusion

This paper presented RA-BCB, a risk-aware budget-constrained auto-bidding framework tailored to the modern first-price RTB setting. RA-BCB combines a supervised value model (pCTR/pCVR) with a first-price replay simulator and a distributional constrained RL agent. By modeling the return distribution and explicitly optimizing CVaR while enforcing a daily budget constraint via dual updates, RA-BCB targets not only higher expected performance but also improved downside stability. A pacing deviation penalty further regularizes intra-day budget consumption, yielding spend curves close to ideal linear pacing.

Extensive offline replay experiments on the iPinYou benchmark demonstrate that first-price payment severely degrades naive strategies that were developed for second-price auctions, and that risk-aware constrained RL mitigates these effects. At 50% budget, RA-BCB improves weighted value by 43.1% over linear bidding, reduces eCPC by 30.6%, and more than doubles the 10%-tail ROI (CVaR0.1). Per-campaign breakdowns show consistent gains across heterogeneous campaigns, including conversion-weighted objectives. A semi-synthetic first-price evaluation on Criteo click logs yields consistent relative improvements, showing that the learned control mechanisms generalize across datasets.

Future work includes (i) counterfactual evaluation and off-policy correction for impressions not present in the logged win set, (ii) delayed and multi-touch conversion attribution, and (iii) combining bid shading models with risk-aware constrained RL in fully online systems. Overall, RA-BCB provides a practical and extensible research baseline for risk-sensitive auto-bidding under first-price auctions.

## References

[1] J. Wang and S. Yuan, "Real-Time Bidding: A New Frontier of Computational Advertising Research," in

Proc. ACM Int. Conf. Web Search and Data Mining (WSDM), 2015, pp. 415–416.

[2] J. Wang, W. Zhang, and S. Yuan, "Display Advertising with Real-Time Bidding (RTB) and Behavioural Targeting," Found. Trends Inf. Retr., vol. 11, no. 4–5, pp. 297–435, 2017.

[3] W. Zhang, S. Yuan, J. Wang, and X. Shen, "Real-Time Bidding Benchmarking with iPinYou Dataset," arXiv:1407.7073, 2014.

[4] W. Zhang, S. Yuan, and J. Wang, "Optimal Real-Time Bidding for Display Advertising," in Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD), 2014, pp. 1077–1086.

[5] D. Agarwal, S. Ghosh, K. Wei, and S. You, "Budget Pacing for Targeted Online Advertisements at LinkedIn," in Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD), 2014.

[6] H. Cai, K. Ren, W. Zhang, K. Malialis, J. Wang, Y. Yu, and D. Guo, "Real-Time Bidding by Reinforcement Learning in Display Advertising," in Proc. ACM Int. Conf. Web Search and Data Mining (WSDM), 2017, pp. 661–670.

[7] D. Wu, X. Chen, X. Yang, H. Wang, Q. Tan, X. Zhang, J. Xu, and K. Gai, "Budget Constrained Bidding by Model-free Reinforcement Learning in Display Advertising," in Proc. ACM Int. Conf. Information and Knowledge Management (CIKM), 2018, pp. 1443–1451.

[8] S. Despotakis, O. Korula, and A. Sayedi, "First-Price Auctions in Online Display Advertising," J. Marketing Research, 2021.

[9] D. Gligorijevic et al., "Bid Shading in The Brave New World of First-Price Auctions," in Proc. ACM Int. Conf. Information and Knowledge Management (CIKM), 2020.

[10] S. Pan et al., "Bid Shading by Win-Rate Estimation and Surplus Maximization," arXiv preprint, 2020.

[11] W. Zhang, S. Yuan, and J. Wang, "Managing Risk of Bidding in Display Advertising," arXiv:1701.02433, 2017.

[12] Z. Jiang, Y. Wu, S. Deng, X. Lin, and J. Ye, "Adaptive Risk-Aware Bidding with Budget Constraint in Display Advertising," ACM SIGKDD Explorations Newsletter, vol. 25, no. 1, 2023.

[13] Y. Chow, A. Tamar, S. Mannor, and M. Pavone, "Risk-Sensitive and Robust Decision-Making: a CVaR Optimization Approach," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2015.

[14] A. Tamar, Y. Chow, M. Ghavamzadeh, and S. Mannor, "Policy Gradient for Coherent Risk Measures," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2015.

[15] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained Policy Optimization," in Proc. Int. Conf. Machine Learning (ICML), 2017, pp. 22–31.

[16] C. Tessler, D. J. Mankowitz, and S. Mannor, "Reward Constrained Policy Optimization," arXiv:1805.11074, 2018.

[17] M. G. Bellemare, W. Dabney, and R. Munos, "A Distributional Perspective on Reinforcement Learning," in Proc. Int. Conf. Machine Learning (ICML), 2017.

[18] W. Dabney, M. Rowland, M. G. Bellemare, and R. Munos, "Distributional Reinforcement Learning with Quantile Regression," in Proc. AAAI Conf. Artificial Intelligence (AAAI), 2018.

[19] W. Dabney, G. Ostrovski, D. Silver, and R. Munos, "Implicit Quantile Networks for Distributional Reinforcement Learning," in Proc. Int. Conf. Machine Learning (ICML), 2018.

[20] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," in Proc. Int. Conf. Machine Learning (ICML), 2018.

[21] V. Mnih et al., "Human-level Control through Deep Reinforcement Learning," Nature, vol. 518, no. 7540, pp. 529–533, 2015.

[22] T. P. Lillicrap et al., "Continuous Control with Deep Reinforcement Learning," in Proc. Int. Conf. Learning Representations (ICLR), 2016.

[23] H.-T. Cheng et al., "Wide & Deep Learning for Recommender Systems," arXiv:1606.07792, 2016.

[24] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: A Factorization-Machine based Neural Network for CTR Prediction," in Proc. Int. Joint Conf. Artificial Intelligence (IJCAI), 2017, pp. 1725–1731.

[25] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun, "xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems," in Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD), 2018.

[26] Y. Juan, Y. Zhuang, W.-S. Chin, and C.-J. Lin, "Field-aware Factorization Machines for CTR Prediction," in Proc. ACM Conf. Recommender Systems (RecSys), 2016.

[27] Kaggle, "Criteo Display Advertising Challenge," 2014. [Online]. Available: https://www.kaggle.com/c/criteo-display-ad-challenge

[28] Criteo AI Lab, "Terabyte Click Logs Dataset," [Online]. Available: https://ailab.criteo.com/criteo-1tb-click-logs-dataset/