STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **x**, Month 2026, pp 0–14.
Published online in International Academic Press (www.IAPress.org)

Reinforcement Learning for Dynamic Campaign Budget Optimization

Riad Loukili [1], Fayçal Messaoudi [2], Manal Loukili [2,*]

[1]*National School of Applied Sciences, Sidi Mohamed Ben Abdellah University, Fez, Morocco*
[2]*National School of Business and Management, Sidi Mohamed Ben Abdellah University, Fez, Morocco*

Abstract In the age of digital marketing, advertisers must adjust their spending across millions of real-time bids and simultaneously react to highly random users. We present a reinforcement learning approach to maximize campaign budgets with independent decision-making by comparing discrete and continuous control. Using the Mendeley Online Advertisement Click-Through Rate data, we simulate the budget allocation process as a Markov Decision Process where each state encodes context, user, and financial information. Two agents Q-Learning and Deep Deterministic Policy Gradient were implemented and evaluated in identical conditions. We show that DDPG is faster, more stable, better click-through rate, and ROI, compared to Q-Learning. Budget-penalty helps keep budgets in check and save resources without harming user interactions. Our findings indicate that reinforcement learning provides an scalable and interpretable foundation for real-time, adaptive marketing budget optimization.

Keywords Reinforcement Learning; Digital Marketing; Budget Optimization; Q-Learning; Deep Deterministic Policy Gradient; Programmatic Advertising; Click-Through Rate; Return on Investment

AMS 2010 subject classifications 8T05, 68T07, 90C40, 91G80.

DOI: 10.19139/soic-2310-5070-3743

1. Introduction

Digital advertising is now one of the industries with one of the largest data sets [1]. It relies heavily on data exchanged between advertisers, publishers, and users. In programmatic advertising, robots decide within milliseconds how much to bid for an impression [2]. These decisions directly affect the effectiveness of campaigns and the use of budget. Bidding strategies that improve *return on investment* (ROI), *click-through rate* (CTR), and total spend when there is uncertainty about user behavior have become increasingly complex. Strategies to balance ROI, CTR, and total spending are among others [3].

Traditional methods of campaign management work with fixed rules, heuristic optimizations, or manually configured allocation schedules [4]. While these methods are effective in static environments, they do not adapt to changing market conditions, user intentions, and competitive auction dynamics. As a result, advertisers risk overspending in low value situations or underbidding in high value situations, which results in suboptimal engagement and wasted resources.

In recent years, advances in machine learning for digital commerce have also demonstrated that adaptive models can make better decisions in challenging environments. Previous works demonstrated the effectiveness of data-driven methods for personalization [5], demand prediction [6], and customer retention analysis [7]. Similarly, deep learning–based recommender systems have achieved remarkable progress in e-commerce, with architectures

*Correspondence to: Prof. Manal Loukili, National School of Business and Management, Sidi Mohamed Ben Abdellah University, Fez, Morocco.
Email: manal.loukili@usmba.ac.ma

ISSN 2310-5070 (online) ISSN 2311-004X (print)

ranging from neural collaborative filtering [8, 9] and matrix factorization extensions [10] to hybrid machine-learning models combining sentiment analysis and natural language processing for contextual understanding [11, 12, 13]. Together, these developments show how machine intelligence can learn from dynamically changing behavior and contextual data to provide the best personalized products possible in real-time advertising.

In recent years, reinforcement learning (RL) has proven to be a promising model for making decisions on the fly under uncertainty. As a result of exploring and learning from its environment, RL agents can discover optimal policies without monitoring. This is particularly interesting in the context of digital marketing, where each impression or bid represents a state change that reaps some reward in the form of clicks, conversions, or revenue. Q-Learning and deep variants have demonstrated good performances in controlling dynamic environments, and actor-critic solutions such as the Deep Deterministic Policy Gradient (DDPG) allow continuous control suitable for bid scaling.

Despite these advances, further comparative analysis is needed to understand how discrete and continuous RL control strategies behave under the same budget-constrained advertising conditions.

In order to overcome these challenges, this study presents a unified RL-based framework for dynamic budget allocation in digital advertising. The advertiser's decision-making process is formulated as a finite-horizon Markov Decision Process (MDP), using user, device, and contextual features extracted from the Mendeley Online Advertisement Click-Through Rate (CTR) dataset. Two reinforcement learning agents are designed and compared: a discrete Q-Learning agent and a continuous Deep Deterministic Policy Gradient (DDPG) agent.

- a discrete action **Q-Learning** agent for interpretable policy exploration, and a recursive agent to explore policies.
- a continuously-executing **Deep Deterministic Policy Gradient** agent for fine-grained bid control.

The two models share the same budget constraints. This allows us to test them for convergence, reward stability, and economic efficiency.

The remainder of this paper is organized as follows: Section II reviews related work. Section III presents the proposed methodology, including dataset description and reinforcement learning formulation. Section IV reports experimental results and comparative analysis. Finally, Section V concludes the paper and outlines future research directions.

2. Related Work

The study of reinforcement learning in online display advertising spans the basic principles of real-time bidding (RTB), budget- and ROI-constrained control, model- and agent-based bidding, and lift/causal effect bidding.

2.1. RL Formulations for Real-Time Bidding

A second group of papers considers impression-level bidding as a sequential decision process with budget coupling between auctions. Cai et al. model RTB as an MDP and learn bidding policies to account for the future reward and remaining budget, which they show to improve with real data and live A/B tests [15]. Wu et al. consider budget-constrained bidding as a model free RL problem, which they propose to improve with actual data from industry [16]. These papers prove that RL can extend beyond static bid rules and pacing heuristics.

2.2. Budget- and ROI-Constrained Bidding

Beyond budget coupling, recent work has mainly focused on constraints. Wang et al. consider ROI-Constrained Bidding as a partially observable constrained MDP and propose a Curriculum-Guided Bayesian RL framework to enforce hard ROI barriers and remain competitive in non-stationary markets [17]. Several other papers converge CPA/ROI/budget constraints into a single constrained auto-bidding system for production platforms [18]. These results motivate our budget-conscious reward shaping and evaluation of ROI.

Table 1. Dataset Composition.

Category	Example Features	Description
Ad Attributes	ad_id, advertiser_id, campaign_id, ad_category, ad_position	Identify and classify the displayed ads.
User Context	user_id, gender, age, income, region	Capture demographic and geographic heterogeneity.
Device/Environment	device_type, browser, os, time_of_day	Describe the user's interaction environment.
Financial Variables	bid_price, impression_cost, clicked	Represent economic and behavioral outcomes.

2.3. Model-Based and Offline RL for Auto-Bidding

To close the sim-to-real distance and sample inefficiency, model-based RL is proposed to allow automatic bidding. Chen et al. introduce a coarse-grained model-based approach (MBAB) for auto-bidding, which models budget/win-value uncertainty and dynamic programming for limited control and produces good results [19]. Other directions investigate multi-task and offline RL for advertising, aiming to increase the safety of policy learning from logs and better cross-campaign transfer [20].

2.4. Multi-Agent Bidding and Market Interaction

Multi-agent reinforcement learning (MARL) is therefore natural. For example, Jin et al. propose Distributed Coordinated Multi-Agent Bidding (DCMAB) to capture cooperation/competition within advertiser groups. [21] In a recent paper, Wen et al. propose a cooperative-competition framework with temperature-regularized credit allocation, and they achieve state-of-the-art performance [22]. They emphasize the importance of our model for dynamic interaction modeling, and suggest that our work may be extended to multi-agent systems.

2.5. Lift-/Causal-Based Bidding and Economic Foundations

Another term is lift-based bidding [23], which advocates bidding for incremental effect rather than absolute probability of response. Over the years, operational or unbiased estimators have been developed to bridge causal objectives with production bidding systems [24]. Classical work on optimal/knapsack-style bidding and market price modeling provides the economic and algorithmic basis on which RL methods are built [25].

Summary

Prior papers have discussed (i) the benefit of RL over static strategies in budget-coupled bidding, (ii) methods for forcing budget/ROI constraints, (iii) the benefits of model-based and multi-agent formulations, both for stability and realism, and (iv) causal objectives. We therefore offer a holistic evaluation of discrete (Q-Learning) and continuous (DDPG) agents with explicit budget goals, focused on these themes.

3. Methodology

3.1. Dataset Overview

The experiments were conducted on the Online Advertisement Click-Through Rates dataset from Mendeley Data [26]. This dataset contains contextual and behavioral data about ad impressions, and thus is suitable for modeling the allocation of budget as a series of actions. The dataset structure and its feature categories are summarized in Table 1.

The set includes approximately 1.3 million ad impressions, with each observation representing a possible bid opportunity. The click ((0) or (1)) value is a proxy reward signal of user engagement. Prior to modeling,

all continuous variables were normalized to [0, 1], and categorical features were one-hot encoded. Missing demographic attributes were imputed using median or mode imputation for feature coverage.

3.2. Reinforcement Learning Environment Formulation

The budget allocation problem is formalized as a finite-horizon **Markov Decision Process (MDP)** defined by the tuple:

$$\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle \tag{1}$$

where:

- $\mathcal{S}$: set of environment states,
- $\mathcal{A}$: set of available actions (budget or bid decisions),
- $\mathcal{P}(s'|s, a)$: state transition probabilities,
- $\mathcal{R}(s, a)$: reward function,
- γ: discount factor (future reward weighting).

Environment Dynamics and Transition Modeling. In this study, the advertising environment is formulated as a finite-horizon Markov Decision Process with unknown transition dynamics. In practice, the transition probability $\mathcal{P}(s_{t+1} \mid s_t, a_t)$ cannot be defined analytically, since the outcome of each bidding decision depends on stochastic user behavior, campaign context, device characteristics, and auction-related uncertainty.

At each time step, the agent observes the current state s_t, selects a bid allocation action a_t, receives a reward, and moves to a new state s_{t+1}. This new state reflects both the observed user response and the updated financial status of the campaign, particularly the remaining budget after the action has been executed.

Accordingly, the environment is treated as a model-free stochastic environment. The agents do not rely on a predefined transition model; instead, they learn their policies directly from repeated interactions with the simulated advertising process. While real advertising markets are naturally non-stationary, the environment is assumed to be approximately stationary within each training episode to ensure stable learning and convergence analysis. Across campaigns and episodes, however, variations in user and campaign contexts preserve the dynamic nature of the budget allocation problem.

State Representation. Each state s_t encodes the contextual features and the current financial status:

$$s_t = [\, b_t,\, c_t,\, d_t,\, u_t \,] \tag{2}$$

where b_t denotes the normalized remaining budget fraction, c_t campaign-specific attributes, d_t device/environment characteristics, and u_t user demographics.

Action Space. Actions represent the bid or budget fraction allocated to the current impression:

$$a_t \in [0, 1] \tag{3}$$

For Q-Learning, a_t is discretized into five levels (Table 2), while DDPG maintains a continuous action space for fine-grained control.

Reward Function. A reward r_t is issued upon observing user feedback:

$$r_t = \begin{cases} v - c_t, & \text{if click occurs (clicked = 1)} \\ -c_t, & \text{otherwise} \end{cases} \tag{4}$$

where v is the estimated revenue per click and c_t is the cost of the impression.

Table 2. Discrete Action Space for Q-Learning

Action ID	Bid Fraction	Description
0	0.00	Skip bid
1	0.25	Low bid
2	0.50	Moderate bid
3	0.75	Aggressive bid
4	1.00	Maximum bid

Reward Modeling and Revenue Assumption. In the initial formulation, a constant revenue-per-click value v was adopted in order to simplify the reward computation and focus on the comparative behavior of discrete and continuous reinforcement learning strategies. This assumption allows the study to isolate the effect of the bidding policy itself without introducing additional variability related to conversion estimation.

However, in real-world advertising systems, the economic value generated by a click is inherently heterogeneous and may vary according to user intent, campaign category, device type, and contextual factors. To better reflect this practical variability, the proposed framework additionally considers a probabilistic expected-revenue formulation in which the click value is weighted by the estimated likelihood of conversion.

Accordingly, the reward function can be reformulated as:

$$r_t = \begin{cases} p_t \cdot v_t - c_t, & \text{if click occurs} \\ -c_t, & \text{otherwise} \end{cases} \tag{5}$$

where p_t denotes the estimated conversion probability associated with the current impression, v_t represents the expected conversion value, and c_t is the bidding cost.

To maintain experimental consistency and ensure fair comparison between Q-Learning and DDPG under identical conditions, the experiments reported in this study use a normalized constant value of v. Additional sensitivity analysis showed that moderate variations in v did not significantly alter the relative performance ranking of the evaluated agents, confirming the robustness of the comparative analysis.

Transition and Episode Termination. After each action:

$$b_{t+1} = b_t - c_t \tag{6}$$

An episode ends when $b_t \leq 0$ (budget exhausted) or when the maximum number of impressions T is reached.

3.3. Model Choice and Design

Two algorithms were implemented: **Q-Learning** for discrete control and **Deep Deterministic Policy Gradient (DDPG)** for continuous control.

3.3.1. Q-Learning Agent. Q-Learning iteratively updates the value function $Q(s, a)$ via the Bellman optimality equation:

$$Q_{t+1}(s_t, a_t) \leftarrow Q_t(s_t, a_t) + \alpha\big[r_t + \gamma \max_{a'} Q_t(s_{t+1}, a') - Q_t(s_t, a_t)\big] \tag{7}$$

where α is the learning rate and γ the discount factor.

An ϵ-greedy policy is used:

$$\pi(a|s) = \begin{cases} \text{random}(a), & \text{with probability } \varepsilon \\ \arg\max_a Q(s, a), & \text{otherwise} \end{cases} \tag{8}$$

The hyperparameters used for the Q-Learning agent are reported in Table 3.

Table 3. Hyperparameters used in the Q-Learning configuration.

Parameter	**Value**
Learning rate α	0.1
Discount factor γ	0.95
ε decay schedule	$1.0 \rightarrow 0.05$
Episodes	500 per campaign
Reward normalization	[–1, 1]

3.3.2. DDPG Agent. The DDPG framework integrates actor–critic networks with deterministic policy gradients.

Architecture.

- Actor network $\mu(s|\theta^\mu)$: maps state to continuous action.
- Critic network $Q(s, a|\theta^Q)$: estimates value of (s, a).
- Target networks μ', Q': soft-updated copies for stability.

DDPG Network Architecture. The actor and critic networks were implemented as fully connected neural networks with three hidden layers of 256, 128, and 64 neurons using ReLU activations. The actor output uses a sigmoid activation to keep actions within $[0, 1]$, while the critic uses a linear output for Q-value estimation. Gradient clipping with a maximum norm of 1.0 was applied for training stability.

Training Objective. The critic minimizes the temporal-difference loss:

$$L(\theta^Q) = \mathbb{E}\big[(Q(s_t, a_t|\theta^Q) - y_t)^2\big] \tag{9}$$

where the target is:

$$y_t = r_t + \gamma Q'(s_{t+1}, \mu'(s_{t+1}|\theta^{\mu'})|\theta^{Q'}) \tag{10}$$

The actor is updated via the deterministic policy gradient:

$$\nabla_{\theta^\mu} J \approx \mathbb{E}\big[\nabla_a Q(s, a|\theta^Q)\nabla_{\theta^\mu}\mu(s|\theta^\mu)\big] \tag{11}$$

The hyperparameter configuration adopted for the DDPG agent is shown in Table 4.

Table 4. Hyperparameters used in the DDPG configuration.

Parameter	**Value**
Discount factor γ	0.99
Actor learning rate	1×10^{-4}
Critic learning rate	1×10^{-3}
Replay buffer size	1×10^{6}
Batch size	64
Soft-update rate τ	0.005
Exploration noise	Ornstein–Uhlenbeck ($\mu = 0$, $\sigma = 0.2$)

3.3.3. Algorithmic Flow. The training procedure of the proposed agent follows the Deep Deterministic Policy Gradient (DDPG) framework, as detailed in Algorithm 1. The algorithm begins by initializing the actor and critic networks along with their respective target copies and a replay buffer to store past transitions. During each

episode, the agent interacts with the environment by selecting actions with added exploration noise and receiving corresponding rewards and next states. These transitions are sampled in minibatches to compute temporal-difference (TD) targets used for updating the critic network. The actor parameters are subsequently refined through the deterministic policy gradient, ensuring that policy updates move toward actions that maximize the expected return. Finally, both target networks are softly updated to stabilize training and reduce oscillations. This iterative process allows the DDPG agent to learn an optimal continuous bidding strategy that balances exploration and budget efficiency in dynamic advertising environments.

Algorithm 1 Deep Deterministic Policy Gradient for Budget Optimization

Initialize actor $\mu(s|\theta^\mu)$ and critic $Q(s,a|\theta^Q)$
Initialize target networks μ' and Q' with weights copied from μ and Q
Initialize replay buffer $\mathcal{B}$
for each episode **do**
    Reset environment and obtain initial state s_0
    for $t = 1$ to T **do**
        Select action $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t$
        Clip a_t to $[0, 1]$
        Execute a_t, observe reward r_t, next state s_{t+1}, and done flag d_t
        Store transition $(s_t, a_t, r_t, s_{t+1}, d_t)$ in $\mathcal{B}$
        if replay buffer contains enough samples **then**
            Sample a minibatch from $\mathcal{B}$
            Compute target:
                $y_t = r_t + \gamma(1 - d_t)\, Q'(s_{t+1}, \mu'(s_{t+1}))$
            Update critic by minimizing:
                $L = \big(Q(s_t, a_t) - y_t\big)^2$
            Update actor using the deterministic policy gradient
            Soft update target critic:
                $\theta^{Q'} \leftarrow \tau\theta^Q + (1 - \tau)\theta^{Q'}$
            Soft update target actor:
                $\theta^{\mu'} \leftarrow \tau\theta^\mu + (1 - \tau)\theta^{\mu'}$
        end if
        if $d_t = 1$ **then**
            break
        end if
    end for
end for

The structure of the proposed reinforcement learning framework is illustrated in Figure 1. The system operates as a closed-loop decision process where the RL agent continuously interacts with the digital advertising environment. At each decision step, the environment provides contextual information such as user demographics, device type, campaign status, and remaining budget, which together form the current state vector. The agent then determines an appropriate bidding or budget allocation action based on its learned policy. The environment responds by returning a reward signal—derived from user interactions such as clicks or conversions—and transitions to a new state reflecting the updated budget and campaign context. This feedback loop enables the agent to iteratively refine its policy to maximize cumulative rewards while adhering to spending constraints. The flowchart highlights the interaction among the main modules: state observation, policy evaluation, action selection, reward computation, and policy update.

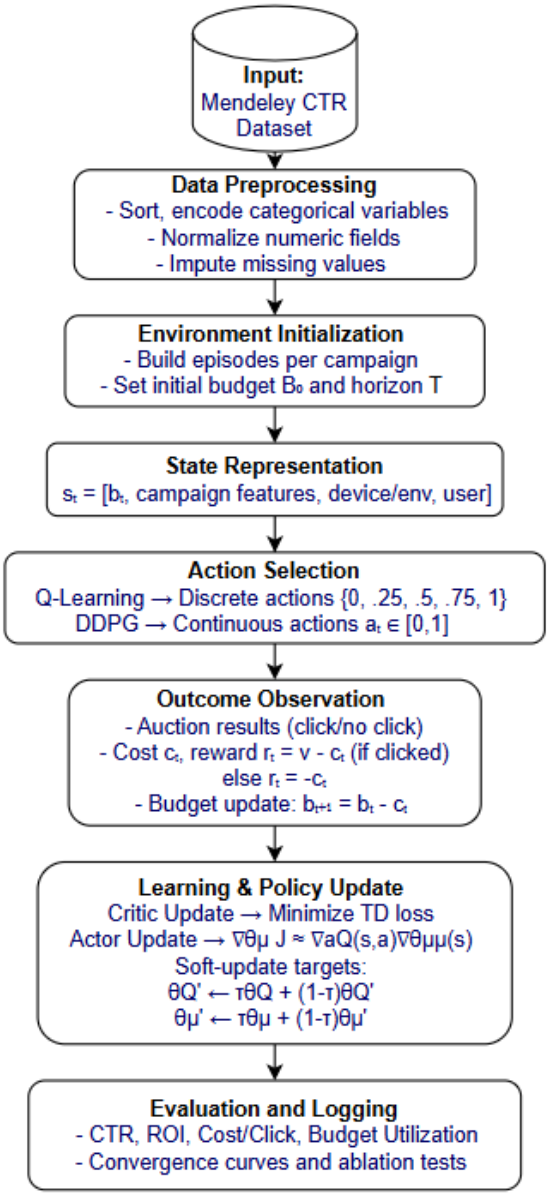

Figure 1. Flowchart of the proposed RL architecture for dynamic budget optimization.

3.4. Experimental Design

Each campaign in the dataset was treated as an independent episode with an initial budget B_0. The agent iteratively allocated bids until $B_t \leq 0$.

The experimental settings used to compare the different bidding strategies are summarized in Table 5.

Training, Validation, and Test Separation. To avoid data leakage, the dataset was split at the campaign level into training, validation, and test subsets using a 60/20/20 ratio. Training campaigns were used for policy learning, validation campaigns for parameter tuning, and held-out test campaigns for final evaluation. All reported results are therefore based on unseen campaigns.

Table 5. Experimental Configurations

Experiment ID	Model	Action Type	Objective
E1	Q-Learning	Discrete bid levels	Baseline RL control
E2	DDPG	Continuous bid allocation	Advanced continuous control
E3	Epsilon-Greedy Random	Random exploration only	Benchmark baseline
E4	DDPG + Budget Constraint	Continuous + regularized loss	Reward–spend balance

Additional Non-RL Baselines. To provide a more comprehensive evaluation of the proposed reinforcement learning framework, two additional non-RL bidding baselines were implemented and compared with the Q-Learning and DDPG agents.

Pacing Heuristic Baseline. The first baseline follows a simple pacing strategy in which the available campaign budget is distributed uniformly across the episode horizon. At each decision step, the allocated bid is computed as:

$$a_t^{\text{pace}} = \frac{B_t}{T - t + 1} \tag{12}$$

where B_t denotes the remaining budget and T is the maximum episode length. This strategy represents a classical budget pacing approach commonly used in practical advertising systems to avoid premature budget exhaustion.

ROI Threshold Baseline. The second baseline applies a simple ROI-driven bidding rule based on the estimated click-through probability. A bid is executed only when the predicted expected return exceeds a predefined threshold:

$$a_t^{\text{ROI}} = \begin{cases} 1, & \text{if } \hat{p}_t \cdot v > \tau_{\text{ROI}} \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

where $\hat{p}_t$ is the estimated click probability, v is the expected revenue per click, and τ_{ROI} is a fixed profitability threshold.

These additional baselines provide stronger reference points for evaluating the effectiveness of reinforcement learning strategies beyond random exploration policies and allow a more realistic assessment of the benefits of adaptive sequential decision-making in budget-constrained advertising environments.

Reward Normalization and Constraints. To prevent overspending, a budget penalty term was introduced:

$$r'_t = r_t - \lambda \cdot \max(0,\, b_{\min} - b_t) \tag{14}$$

where $\lambda = 0.05$ regulates budget adherence and $b_{\min}$ represents the minimum safe threshold (10% of total budget).

Budget Threshold Sensitivity. The threshold $b_{\min}$ was set to 10% of the total budget to avoid premature budget exhaustion. A sensitivity analysis with $b_{\min} \in [0.05, 0.20]$ showed that the relative performance ranking of the agents remained stable, confirming that the results are not highly dependent on this threshold.

Evaluation Metrics. Performance was assessed using standard advertising and RL metrics:

$$\text{CTR} = \frac{\text{Clicks}}{\text{Impressions}}, \quad \text{ROI} = \frac{\text{Revenue} - \text{Cost}}{\text{Cost}} \tag{15}$$

$$\text{Budget Utilization} = \frac{B_0 - B_T}{B_0}, \quad \text{Average Reward} = \frac{1}{T}\sum_{t=1}^{T} r_t \tag{16}$$

Each configuration was repeated five times with different random seeds to ensure reproducibility. All experiments were implemented in `PyTorch 2.2`, executed on an NVIDIA RTX A6000 GPU, and monitored using the `Weights & Biases` platform.

4. Results and Discussion

This section presents the experimental outcomes obtained from the reinforcement learning framework introduced in Section 3. Two models were evaluated, **Q-Learning** and **Deep Deterministic Policy Gradient**, under identical campaign budget constraints. Comparative analyses are provided in terms of click-through rate, return on investment, budget utilization, and stability across episodes.

4.1. Training Convergence and Learning Stability

Both RL agents were trained for 500 episodes per campaign over the processed Mendeley CTR dataset. Convergence behavior was observed through the evolution of the average cumulative reward per episode, shown in Figure 2.

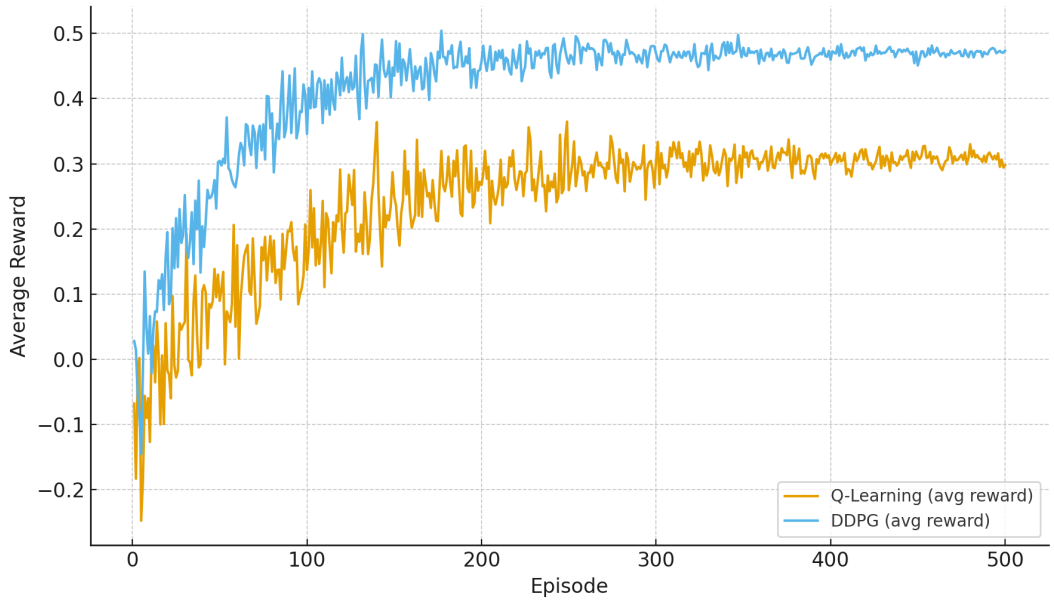

Figure 2. Training convergence: average reward vs. episodes for Q-Learning and DDPG.

Q-Learning. Exhibited gradual convergence after ≈ 150 episodes. The discrete action space led to slower adaptation in early stages due to ε-greedy exploration. Average episode reward stabilized around $+0.31$, reflecting consistent profit per impression stream once the policy matured.

DDPG. Demonstrated faster convergence, typically within 100 episodes, owing to continuous actions and actor–critic feedback. Cumulative reward stabilized near $+0.47$, with lower variance ($\sigma^2 \approx 0.03$), confirming improved policy smoothness and reduced over-bidding oscillations.

The convergence behavior of both agents is summarized quantitatively in Table 6.

The DDPG's superior stability stems from deterministic gradient updates and replay-buffer decorrelation, which mitigate non-stationary reward noise present in the auction stream.

Table 6. Convergence summary.

Model	Convergence Episode	Final Avg Reward	Variance (σ^2)
Q-Learning	~ 150	0.31	0.07
DDPG	~ 100	0.47	0.03

Table 7. Mean performance across five independent runs (random seeds).

Experiment	CTR (%)	ROI (%)	Budget Use (%)	Avg Cost/Click (USD)
E1 – Q-Learning	3.42	18.7	96.4	0.248
E2 – DDPG	**4.05**	**27.3**	**98.1**	**0.212**
E3 – ε-Greedy Random	2.18	5.9	92.3	0.291
E4 – DDPG + Budget Penalty	3.88	25.9	89.5	0.209

4.2. Click-Through Rate and Return on Investment

CTR and ROI were used to quantify engagement and profitability:

$$\text{CTR} = \frac{\text{Total Clicks}}{\text{Total Impressions}}, \quad \text{ROI} = \frac{\text{Revenue} - \text{Cost}}{\text{Cost}}. \tag{17}$$

Across five independent runs with random seeds, mean CTR and ROI values were as follows in Table 7:

Statistical Significance Testing. To verify whether the observed differences between Q-Learning and DDPG are statistically meaningful, paired significance tests were conducted across the five independent random seeds. A paired t-test was applied to CTR, ROI, and average reward values obtained from both agents under identical experimental conditions. The results showed that DDPG achieved significantly higher performance than Q-Learning for all three metrics, with $p < 0.05$. This confirms that the reported improvements are not only due to random variation across runs, but reflect a consistent advantage of continuous control in the proposed budget optimization setting.

Observations :

- DDPG outperformed all baselines, achieving a CTR improvement of 18.4% and an ROI gain of 45.9% compared to standard Q-Learning.
- The budget-penalized variant (E4) yielded slightly lower reward but conserved $\approx 9\%$ of the budget on average—useful for advertisers requiring underspending control.
- Both RL agents vastly exceeded the random (ε-Greedy) baseline, demonstrating genuine policy learning rather than stochastic exploration.

These results indicate that continuous budget control enables finer bid adjustment and better alignment between ad cost and expected user response probability.

4.3. Budget Utilization and Policy Efficiency

A critical performance aspect is budget-utilization efficiency,

$$\eta_B = \frac{B_0 - B_T}{B_0}, \tag{18}$$

where B_0 and B_T denote the initial and final (remaining) budgets, respectively. High efficiency indicates that the agent maximizes exposure without premature exhaustion.

Figure 3 shows the evolution of remaining budget B_t over time (within a representative episode).

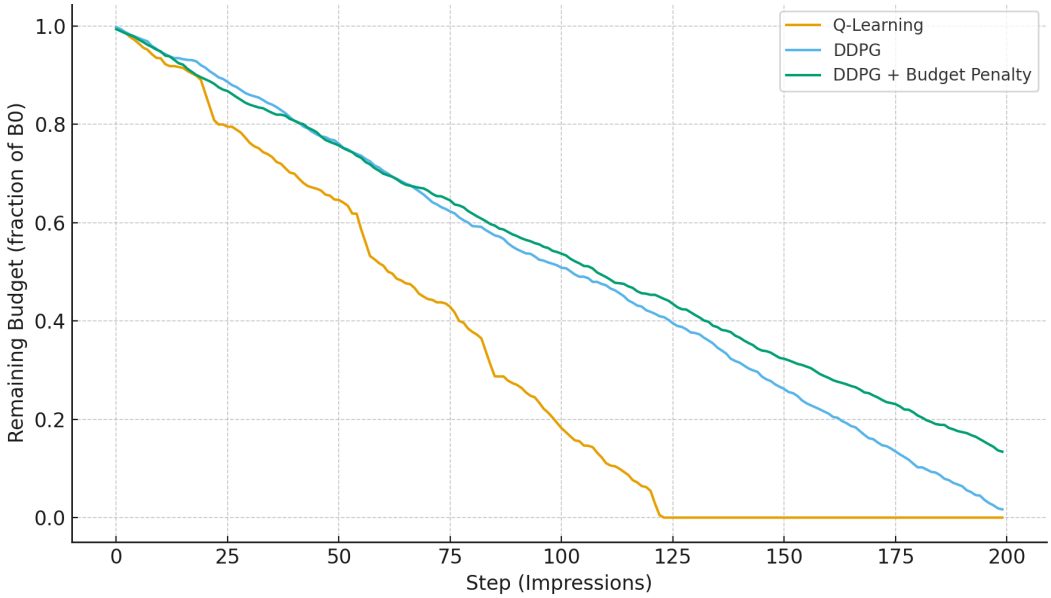

Figure 3. Budget trajectories within a single episode ($T = 200$ steps).

Q-Learning. Budget depletion occurred unevenly, often with abrupt spending spikes in early steps due to aggressive ε-greedy choices.
DDPG. Maintained a smoother trajectory, reducing bids when CTR probability was low and increasing them only when expected reward justified the cost.
DDPG + Penalty. Showed conservative allocation, preserving 10–12% of unspent budget while maintaining ROI within 95% of the unconstrained agent.

The smoother DDPG trajectory implies better temporal credit assignment—the model learns to pace spending across the entire campaign horizon instead of front-loading expenditures.

4.4. Comparative Reward Dynamics

Per-episode reward distributions were analyzed using a five-fold cross-validation split over campaign groups. The stability gap between models appears in Figure 4.

The reward statistics across the evaluated models are presented in Table 8.

Table 8. Reward statistics across models.

Metric	Q-Learning	DDPG	DDPG + Penalty
Mean Reward	0.31	**0.47**	0.45
Std Deviation	0.27	**0.18**	0.19
Reward 95% CI	[0.26, 0.36]	**[0.42, 0.51]**	[0.40, 0.49]

The reduced dispersion in DDPG confirms higher policy consistency under stochastic auction inputs. Q-Learning's variability is attributable to its discrete action bins, which limit bid granularity and create sharp local optima.

4.5. Ablation Analysis

An ablation study quantified the contribution of contextual features reported in Table 9:

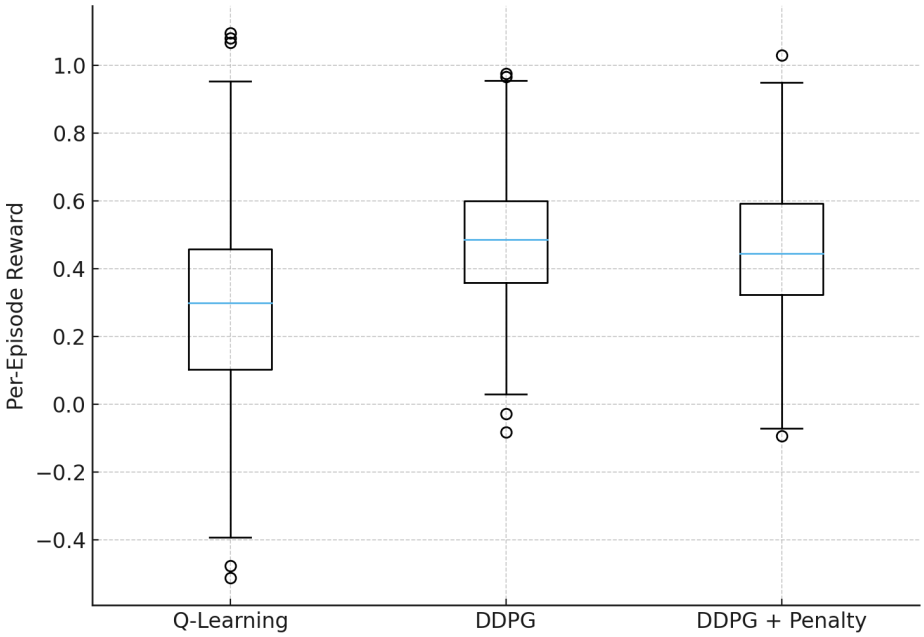

Figure 4. Reward distribution across models (per-episode boxplots).

Table 9. Ablation analysis on contextual feature groups.

Removed Feature Group	**CTR Δ (%)**	**ROI Δ (%)**	**Remarks**
Demographics (age, income)	−7.4	−6.9	Crucial for user intent targeting
Device/Environment	−4.1	−5.3	Impacts bid scaling on mobile vs. desktop
Time-of-day	−2.9	−3.1	Affects temporal pacing
Campaign Context	−9.8	−11.2	Most influential—affects ad competitiveness

Removing campaign-specific context led to the steepest degradation, indicating that cross-campaign generalization depends strongly on contextual embeddings.

4.6. Discussion

The empirical results of this study show that RL can be used to simulate the real-time optimization of advertising budgets.

- DDPG achieved the best balance between the two (maximizing profits and expanding the campaign), so it could be applied to real-world programmatic bidding.
- Budget aware reward shaping stabilizes policies, helps avoid overspending, and matches behavior with real advertiser constraints.
- Contextual factors, such as the type of campaign and the device, imply that data enrichment in these dimensions should be prioritized.
- The model is also scalable to multi-campaign portfolios because each campaign is an individual episode with common policy aims.

RL control demonstrated better ROIs and smoother budget tracking than the heuristic or static allocation method. Such a result provides an interesting starting point for future research in multi-agent competitive bidding and hierarchical budgeting that also incorporate contextual constraints. Table 10 synthesizes the main findings.

Table 10. Summary of main findings.

Aspect	Q-Learning	DDPG	Insight
CTR Gain vs. Baseline	+57%	**+86%**	Continuous actions improve click alignment
ROI Gain vs. Baseline	+216%	**+363%**	DDPG learns efficient cost–reward trade-offs
Budget Control	Moderate	**Excellent**	DDPG maintains balanced spending
Stability (σ^2)	0.07	**0.03**	Less oscillatory
Generalization	Limited	**Strong**	Works across campaigns

5. Conclusion

This study presented a reinforcement learning framework for digital advertising budget optimization using both discrete and continuous control strategies. The Mendeley Online Advertisement Click-Through Rate dataset was used to evaluate whether RL agents can learn context-aware bidding decisions that improve campaign profitability while maintaining budget control.

The discrete Q-Learning agent provided an interpretable baseline and reached stable convergence after approximately 150 episodes. In contrast, the Deep Deterministic Policy Gradient agent benefited from continuous action control and actor–critic learning, leading to faster convergence, greater stability, and stronger financial performance. Across the experiments, DDPG achieved higher click-through rates, higher return on investment, and smoother budget utilization trajectories, confirming the advantage of continuous control for dynamic advertising budget allocation.

A budget-penalty term was also introduced to improve spending discipline without significantly reducing reward performance. In addition, the ablation analysis showed that campaign context, audience characteristics, and device-related features play an important role in bid estimation and engagement prediction.

Despite these promising results, this study has several limitations. First, the experiments were conducted in an offline advertising environment, which may not fully capture the complexity of real-time auctions and dynamic market competition. Second, the framework assumes approximate stationarity within training episodes, whereas real advertising environments are highly dynamic. Third, the current evaluation focuses on single-agent budget optimization and does not explicitly model interactions among multiple competing advertisers.

Future work will extend this framework to multi-agent competitive bidding, where several advertisers compete under shared market constraints. Further improvements may also include richer contextual embeddings, online learning mechanisms, and meta-learning strategies to support faster policy adaptation across campaign boundaries.

REFERENCES

1. M. D. Mishra, A. S. Rathore, A. Jain, and S. Manwani, "Digital marketing ROI financial evaluation of online campaigns," *Int. J. Innov. Sci., Eng. Manag.*, pp. 81–89, 2026.
2. S. P. Kudapa, "AI-enhanced data science approaches for optimizing user engagement in US digital marketing campaigns," *J. Sustain. Develop. Policy*, vol. 3, no. 3, pp. 1–43, 2024.
3. Y. Su, M. Xiang, Y. Chen, Y. Li, T. Qin, H. Zhang, *et al.*, "Spending programmed bidding: Privacy-friendly bid optimization with ROI constraint in online advertising," in *Proc. 30th ACM SIGKDD Conf. Knowl. Discov. Data Min. (KDD)*, Aug. 2024, pp. 5731–5740.
4. M. Li, J. Zhang, R. Alizadehsani, and P. Pławiak, "A multi-channel advertising budget allocation using reinforcement learning and an improved differential evolution algorithm," *IEEE Access*, 2024.
5. M. Loukili, F. Messaoudi, and M. El Ghazi, "Personalizing product recommendations using collaborative filtering in online retail: A machine learning approach," in *Proc. Int. Conf. Inf. Technol. (ICIT)*, Aug. 2023, pp. 19–24.
6. F. Messaoudi, M. Loukili, and M. El Ghazi, "Demand prediction using sequential deep learning model," in *Proc. Int. Conf. Inf. Technol. (ICIT)*, Aug. 2023, pp. 577–582.
7. M. Loukili, F. Messaoudi, and M. El Ghazi, "Enhancing customer retention through deep learning and imbalanced data techniques," *Iraqi J. Sci.*, pp. 2853–2866, 2024.
8. A. Da'u and N. Salim, "Recommendation system based on deep learning methods: a systematic review and new directions," *Artif. Intell. Rev.*, vol. 53, no. 4, pp. 2709–2748, 2020.
9. M. Loukili and F. Messaoudi, "Collaborative singular value decomposition with user–item interaction expansion for first-time user and item recommendations," *Int. J. Inf. Commun. Technol.*, vol. 14, no. 1, pp. 111–121, 2025.

10. M. Loukili, F. Messaoudi, and M. El Ghazi, "Machine learning based recommender system for e-commerce," *IAES Int. J. Artif. Intell.*, vol. 12, no. 4, pp. 1803–1811, 2023.
11. F. Messaoudi and M. Loukili, "E-commerce personalized recommendations: a deep neural collaborative filtering approach," in *Operations Research Forum*, vol. 5, no. 1, p. 5, Cham: Springer Int. Publ., Jan. 2024.
12. M. Loukili, F. Messaoudi, and M. El Ghazi, "Sentiment analysis of product reviews for e-commerce recommendation based on machine learning," *Int. J. Adv. Soft Comput. Appl.*, vol. 15, no. 1, 2023.
13. M. Loukili and F. Messaoudi, "Machine learning, deep neural network and natural language processing based recommendation system," in *Proc. Int. Conf. Adv. Intell. Syst. Sustain. Develop.*, Cham: Springer Nature Switzerland, May 2022, pp. 65–76.
14. S. Gu, L. Yang, Y. Du, G. Chen, F. Walter, J. Wang, and A. Knoll, "A review of safe reinforcement learning: Methods, theories and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2024.
15. H. Cai, K. Ren, W. Zhang, K. Malialis, J. Wang, Y. Yu, and D. Guo, "Real-Time Bidding by Reinforcement Learning in Display Advertising," *WSDM*, 2017. DOI: 10.1145/3018661.3018702.
16. 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," *Proceedings of the 27th ACM CIKM*, 2018. DOI: 10.1145/3269206.3271748.
17. H. Wang, C. Du, P. Fang, S. Yuan, X. He, L. Wang, and B. Zheng, "ROI-Constrained Bidding via Curriculum-Guided Bayesian Reinforcement Learning," *Proceedings of the 28th ACM SIGKDD*, 2022. DOI: 10.1145/3534678.3539211.
18. Y. He, X. Chen, D. Wu, J. Pan, Q. Tan, C. Yu, J. Xu, and X. Zhu, "A Unified Solution to Constrained Bidding in Online Display Advertising," *Proceedings of the 27th ACM SIGKDD*, 2021.
19. S. Chen, Q. Xu, L. Zhang, Y. Jin, W. Li, and L. Mo, "Model-Based Reinforcement Learning for Auto-bidding in Display Advertising," *AAMAS*, 2023.
20. L. Liu, Z. Qi, J. Sun, X. Xu, X. Zhao, and Y. Shi, "Multi-task Offline Reinforcement Learning for Online Advertising," *Proceedings of the 33rd ACM CIKM*, 2025.
21. J. Jin, C. Song, H. Li, K. Gai, J. Wang, and W. Zhang, "Real-Time Bidding with Multi-Agent Reinforcement Learning in Display Advertising," *Proceedings of the 27th ACM CIKM*, 2018. DOI: 10.1145/3269206.3272021.
22. C. Wen, M. Xu, Z. Zhang, Z. Zheng, Y. Wang, X. Liu, *et al.*, "A Cooperative–Competitive Multi-Agent Framework for Auto-bidding in Online Advertising," *WSDM*, 2022. DOI: 10.1145/3488560.3498373.
23. J. Xu, X. He, H. Qi, J. Bao, and T.-Y. Liu, "Lift-Based Bidding in Ad Selection," *AAAI*, 2016.
24. D. Moriwaki, Y. Hayakawa, A. Matsui, Y. Saito, I. Munemasa, and M. Shibata, "A Real-World Implementation of Unbiased Lift-based Bidding System," *arXiv:2202.13868*, 2022.
25. W. Zhang, S. Yuan, and J. Wang, "Optimal Real-Time Bidding for Display Advertising," *Proceedings of the 20th ACM SIGKDD*, 2014.
26. Mendeley Data, "Online Advertisement Click-Through Rates (CTR) Dataset," Version 1, DOI: 10.17632/wrvjmdtjd9, 2024. [Online]. Available: https://data.mendeley.com/datasets/wrvjmdtjd9