PlatformBid: An Auto-Bidding Benchmark from a Unified Advertising Platform's Perspective

Shengtian Yang*
Southeast University
Nanjing, China
yangshengtian@kuaishou.com

Yewen Li*
Kuaishou Technology
Beijing, China
liyewen@kuaishou.com

Peng Jiang
Kuaishou Technology
Beijing, China
jiangpeng07@kuaishou.com

Zhiyi Lyu
Nanyang Technological University
Singapore, Singapore
zhiyi.lyu@ntu.edu.sg

Bo An
Nanyang Technological University
Singapore, Singapore
boan@ntu.edu.sg

Peng Jiang
Unaffiliated
Beijing, China
13126980773@139.com

Qingpeng Cai†
Kuaishou Technology
Beijing, China
caiqingpeng@kuaishou.com

Lei Feng†
Southeast University
Nanjing, China
fenglei@seu.edu.cn

Abstract

Real-time bidding is central to computational advertising, comprising three elements: Supply Side Platform (SSP) selling ad impressions, Demand Side Platform (DSP) bidding for advertisers, and Ad Exchange conducting auctions between them. Traditional auto-bidding algorithms focus solely on the DSP side, maximizing advertiser conversions by adjusting bids against competitors. However, current big ad platforms, such as social media and e-commerce companies, now integrate SSP, DSP, and Ad Exchange functions internally. From such ad platforms' perspective, the goal of the auto-bidding algorithms is not only to maximize the advertisers' conversions, but also the total revenue of the platform. Given the lack of platform-centric evaluation frameworks and the pressing need to advance auto-bidding research, we propose **PlatformBid** - the first comprehensive benchmark designed from a unified ad platform's perspective. To accurately reflect the real-world auto-bidding scenarios, we define three representative settings: (1) homogeneous competition with identical algorithms across advertisers, (2) heterogeneous competition with diverse algorithmic strategies, and (3) promotional competition where some advertisers surge budgets for boosting sales during promotional events like *Black Friday*. We systematically evaluate a broad spectrum of existing auto-bidding methods across these settings, encompassing classical control methods, RL-based methods, and recent generative methods. Besides these methods, we further propose a novel auto-bidding method based on flow-matching, termed **BidFlow**, which leverages the flow-matching method's expressive policy representation to effectively handle dynamic competitive environments. Experimental results demonstrate the effectiveness of our proposed BidFlow on the new settings. Furthermore, online experiment results of BidFlow, especially a significant +2.57% increase of target cost, could also support the offline-online consistency of our PlatformBid.[1]

CCS Concepts

• **Information systems** → **Computational advertising**.

Keywords

Auto-bidding, Benchmark, Flow Matching, Reinforcement Learning, Generative Model

ACM Reference Format:
Shengtian Yang, Yewen Li, Peng Jiang, Zhiyi Lyu, Bo An, Peng Jiang, Qingpeng Cai, and Lei Feng. 2026. PlatformBid: An Auto-Bidding Benchmark from a Unified Advertising Platform's Perspective. In *Proceedings of KDD '26*. ACM, New York, NY, USA, 15 pages. https://doi.org/xxxxx

1 Introduction

Real-time bidding (RTB) has emerged as a dominant paradigm in computational advertising [26], significantly enhancing the efficiency of audience engagement and sales conversion on the Internet. As illustrated in Figure 1, the RTB process operates as follows: when an impression opportunity arises from a user's activity on a supply-side platform (SSP), such as watching videos on a media app, a bid request is broadcast to all competing advertisers via an Ad Exchange. Advertisers then leverage auto-bidding services [31] provided by demand-side platforms (DSPs) to determine bid prices in real-time. The Ad Exchange then selects the highest bidder to display the ad, charging the winning advertiser the market prices, typically the second-highest bid price in a generalized second-price (GSP) auction setting [2]. This paradigm greatly simplifies advertiser operations, as they only need to specify high-level

*Equal contribution. Work done when Shengtian was an intern at Kuaishou.
†Corresponding authors.

KDD '26, Jeju, Korea

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/xxxxx

[1]Code is available at http://anonymous.4open.science/r/PlatformBid-3F10.

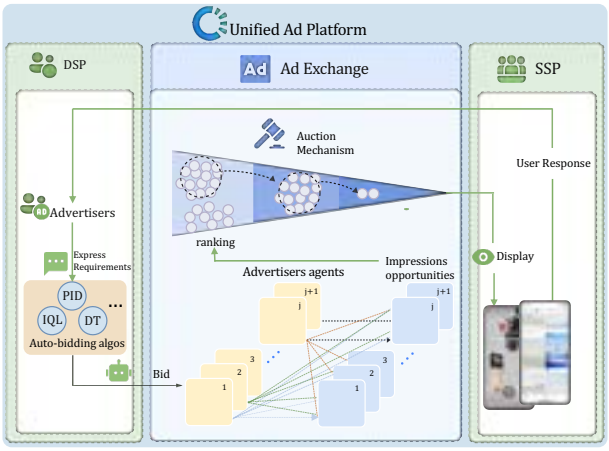

Figure 1: Overview of the unified ad platform. Previous benchmarks primarily focus only on the DSP side. However, as unified ad platforms integrate both DSP and SSP functionalities, a new benchmark from the platform perspective is urgently needed.

objectives and constraints (*e.g.*, target CPA, budget limits) to the DSP, making auto-bidding algorithms indispensable in modern computational advertising. Existing auto-bidding research has predominantly adopted a DSP-centric perspective [28, 34, 38, 39], where DSP companies provide algorithmic bidding solutions designed to optimize individual advertiser objectives only. However, recent industry developments have fundamentally altered this landscape. The emergence and rapid growth of big media and e-commerce companies, such as Meta, TikTok, and Kuaishou, have driven substantial increases in intra-platform advertising activity. Therefore, these companies operate unified ad platforms that consolidate SSP, DSP, and Ad Exchange functionalities. This structural transformation demands a paradigm shift in the auto-bidding research: algorithms must now balance individual advertiser objectives [16, 35] with platform-level revenue optimization.

However, there is still a lack of evaluation benchmarks designed from a unified ad platform's perspective. Existing benchmarks in auto-bidding research have primarily focused on DSP-centric scenarios. A pioneering effort is the iPinYou dataset [40], which provides real-world RTB logs from a DSP serving multiple advertisers across various ad exchanges. Building upon this paradigm, subsequent works have introduced benchmarks with richer features and larger scales. For instance, AuctionNet [32] proposes a large-scale benchmark incorporating diverse advertiser types and competitive auction dynamics. However, all these benchmarks are still designed for the DSP-centric auto-bidding research. Given the pressing need to advance auto-bidding research, we propose the first comprehensive benchmark, termed **PlatformBid**, designed from a unified ad platform's perspective.

To develop a benchmark that accurately and comprehensively captures the essence of platform-level auto-bidding scenarios, we formulate three representative settings in PlatformBid, as illustrated in Figure 2 (b): (1) **Homogeneous competition**. This scenario simulates the platform-wide deployment of a new auto-bidding algorithm, where all advertisers adopt the same strategy. This necessitates evaluating both overall platform revenue and average advertiser performance. We model this through all advertisers employing identical algorithms. (2) **Heterogeneous competition.** Platforms may ensemble multiple auto-bidding algorithms for different purposes, requiring assessment of both aggregate platform performance and inter-algorithm competitive dynamics. We simulate this by having half of the advertisers adopt a baseline method and the other half adopt an alternative method. This setting evaluates whether new algorithms can improve platform-level performance while avoiding adverse impacts on baseline advertisers. (3) **Promotional competition.** Ad platforms derive substantial revenue from promotional events such as *Black Friday*, during which certain advertisers significantly increase their budgets to achieve more conversions. We simulate this by selectively amplifying budgets for specific advertisers, examining auto-bidding performance under budget imbalance.

We systematically evaluate a broad spectrum of auto-bidding methods on PlatformBid across the three representative settings. The evaluated methods encompass: (1) Classical control methods, including PID controllers [4]; (2) Reinforcement learning (RL) methods, particularly offline RL approaches such as BCQ [7] and IQL [18]; and (3) State-of-the-art generative methods, comprising Decision Transformer-based approaches [3] such as GAS [20] and GAVE [8], as well as diffusion-based methods [1, 14, 21, 41] including CBD [19]. Beyond these existing approaches, we propose **BidFlow**, a novel flow-matching-based auto-bidding method specifically designed for dynamic competitive environments. BidFlow leverages flow-matching's expressive policy representation to effectively capture the complex dynamics of ad auctions and employs Q-value-guided distillation to distill a multi-step flow model into an efficient one-step policy. Experimental results demonstrate that BidFlow achieves superior performance across all three settings. Furthermore, online experiment results on the Kuaishou ad platform, *e.g.*, a +2.57% increase of target cost, validate both BidFlow's effectiveness in more complex industrial scenarios and PlatformBid's offline-online consistency, confirming the benchmark's practical value for guiding real-world auto-bidding algorithm development. In summary, the contributions of this work can be summarized as follows:

- We propose PlatformBid, the first comprehensive benchmark designed from a unified ad platform's perspective, formulating three representative settings—homogeneous competition, heterogeneous competition, and promotional competition.
- We evaluate comprehensive auto-bidding methods including classical control approaches, RL methods, and generative approaches, and propose BidFlow, a flow-matching-based method that achieves superior performance through expressive policy representation and Q-value-guided one-step distillation.
- We provide extensive analysis with diverse baselines and validate offline-online consistency through the online experiment, especially a significant +2.57% increase of target cost, confirming PlatformBid's practical value for guiding real-world auto-bidding algorithm development.

2 Related Work

Auto-bidding Benchmarks. Systematic evaluation of auto-bidding algorithms has long relied on proprietary datasets [2, 4, 13], limiting method comparison and reproducibility due to lack of unified standards. The iPinYou dataset [40] pioneered public benchmarking

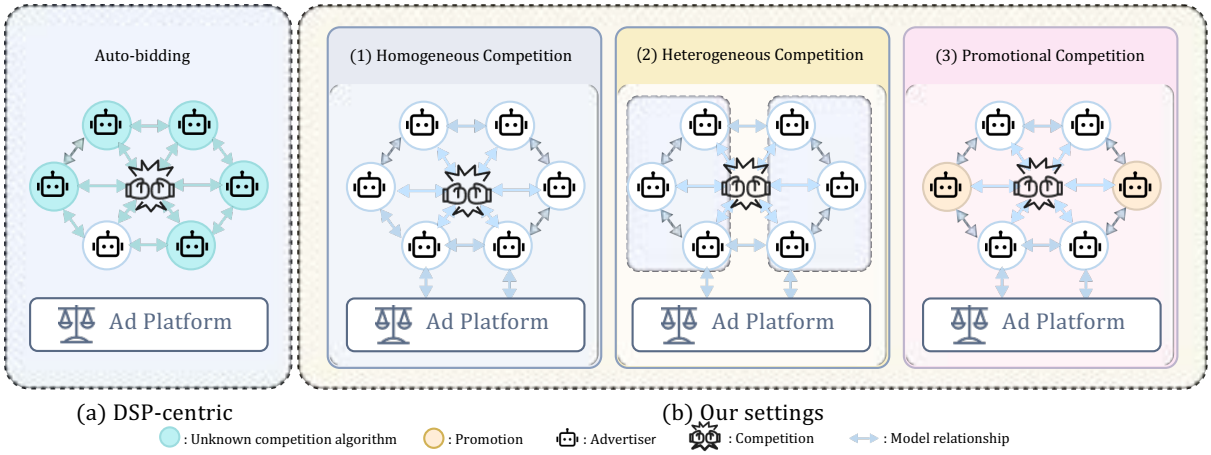

Figure 2: Comparison of PlatformBid settings with existing DSP-centric benchmarks. (a) DSP-centric benchmarks optimize only individual advertiser objectives, ignoring platform-level impacts. (b) Setting 1 (Homogeneous Competition): N advertisers with identical strategies compete dynamically. (c) Setting 2 (Heterogeneous Competition): N/2 baseline advertisers (white) compete against N/2 test advertisers (yellow) simultaneously. (d) Setting 3 (Promotional Competition): Selective budget amplification for specific advertisers (yellow) simulates promotional campaigns. Unlike DSP-centric approaches, PlatformBid jointly considers both advertiser and platform-level objectives.

with real-world RTB logs from a DSP serving multiple advertisers across ad exchanges. However, it primarily focuses on DSP-centric scenarios that optimize individual advertiser objectives. Subsequent benchmarks follow this paradigm while introducing richer features and larger scales—such as AuctionGym [15] and AdCraft [11]—but still optimize individual advertiser strategies in isolation. AuctionNet [32] advanced the field with the first large-scale public dataset and corresponding benchmark. However, its evaluation protocol sequentially replaces advertisers against fixed historical bids by replaying logged ad traffic, failing to capture the competitive dynamics where multiple advertisers simultaneously adjust strategies. In contrast, as illustrated in Figure 2, PlatformBid introduces dynamic multi-advertiser competition across three representative settings, enabling comprehensive assessment of both individual advertiser performance and collective platform revenues.

Auto-bidding Methods. Auto-bidding algorithms have evolved through three main paradigms, including classical control approaches, RL methods, and Generative methods. *Classical control approaches* [4, 10, 17, 22, 25, 36, 37, 42], such as PID controllers [4], provide basic automation through feedback control mechanisms but lack adaptability to complex market dynamics. *Reinforcement learning methods* [29, 33] have significantly advanced the field. Online algorithms like USCB [13] and MAAB [35] optimize bidding strategies through direct environment interaction, while offline methods including BCQ [7] and IQL [18] improve sample efficiency by learning from historical logs without risky online exploration. *Generative methods* [5, 6, 27] offer powerful sequence modeling capabilities. Decision Transformer [3, 24] formulates bidding as conditional sequence generation, avoiding value function estimation instability. GAS [20] extends this with post-training search using multiple critic networks, while GAVE [8] introduces value-guided exploration to address the dataset quality limitation. Diffusion-based methods such as DiffBid [12] and CBD [19] provide trajectory-level planning-based auto-bidding, with CBD achieving notable improvements in sparse-reward scenarios through its completer-aligner framework. However, diffusion models suffer from slow inference due to iterative denoising, limiting deployment in latency-sensitive bidding systems. As these methods have only been validated in DSP-centric benchmarks, we find their performance in PlatformBid is limited due to the increased complexity of environment dynamics. To address this challenge, we propose BidFlow, which leverages flow matching for efficient inference while incorporating Q-learning guidance for optimization.

3 PlatformBid Benchmark

In this section, we introduce our PlatformBid benchmark for platform-centric auto-bidding research. We first formulate the auto-bidding problem from a unified platform's perspective, highlighting key differences from traditional DSP-centric formulations. We then describe the benchmark's data preparation. Subsequently, we present three evaluation settings designed to reflect real-world auto-bidding scenarios. Finally, we detail the evaluation metrics that capture both advertiser-level and platform-level performance.

3.1 Problem Statement

Consider an online advertising scenario involving a sequence of H impression opportunities, *i.e.*, impressions traffic, each identified by index i, where PlatformBid simulates it by replaying a traffic log of impressions. The advertiser secures an opportunity when the bid b_i exceeds competing bids, resulting in an associated cost c_i. PlatformBid employs a Generalized Second-Price (GSP) auction, thus c_i equals the second-highest bid among all competing advertisers. The optimization objective seeks to maximize aggregate value from all opportunities, formulated as $\sum_i o_i v_i$, where v_i represents the value of opportunity i and o_i serves as a binary indicator of

winning status. Beyond budget limitations, the system must satisfy various economic requirements, including constraints on unit costs for specific conversion events such as cost-per-action (CPA) [13]. Consequently, the auto-bidding optimization problem under constraints can be formulated as:

$$\begin{aligned} \max \quad & \sum_i o_i v_i \\ \text{s.t.} \quad & \sum_i o_i c_i \le B, \\ & \frac{\sum_i o_i c_{ij}}{\sum_i o_i p_{ij}} \le C_j, \quad \forall j, \\ & \sum_i e_{il} \le O_l, \quad \forall l, \\ & o_i \in \{0,1\}, \quad \forall i, \end{aligned} \tag{1}$$

where B denotes the advertiser's total budget, p_{ij} represents the performance indicator associated with the j-th advertiser-specified constraint with upper bound C_j, e_{il} quantifies the impact on the l-th platform-level performance metric, and O_l specifies the corresponding platform objective threshold. For instance, O_l may stipulate that the aggregate platform revenue across all H opportunities must not fall below that obtained under a baseline bidding strategy, analogous to the requirement in online A/A testing. A critical distinction from conventional DSP-centric formulations is the explicit incorporation of platform objectives O_l. The subsequent sections examine three distinct problem settings that emerge from different specifications of O_l.

Prior research [12, 13, 20] have predominantly adopted a simplified auto-bidding strategy formulated as:

$$b_i = \lambda_0 v_i + \sum_{j=1}^{J} \lambda_j p_{ij} C_j, \tag{2}$$

where b_i^i denotes the final bid amount for opportunity i. The coefficients λ_j constitute the bidding parameters. In real-world ad platforms, auto-bidding algorithms typically update these bidding parameters at relatively coarse time granularities—such as 30-minute intervals—rather than reacting to individual impression events occurring at millisecond scales. This temporal characteristic provides sufficient computational headroom for sophisticated auto-bidding approaches to process parameter adjustment requests.

3.2 Data Preparation

The auto-bidding task can be formulated as a sequential decision-making task. At each timestep t within a bidding period, the agent observes state $s_t \in \mathcal{S}$ that encodes the current advertising status, and subsequently selects action $a_t \in \mathcal{A}$ to determine bidding parameters. The advertising environment evolves according to an unknown transition dynamic $\mathcal{T}$, where the next state is determined by both historical context and current observations. Upon each state transition, the environment generates reward r_t, which quantifies the value accrued toward the campaign objective during period t. This process repeats until the bidding horizon concludes (*e.g.*, at the end of a day). The key component data of the formulation are detailed below:

- **State** s_t: A collection of campaign-level features describing the current advertising status, including temporal and budget-related information like remaining time, residual budget, expenditure rate, and KPI satisfaction ratios for each constraint.
- **Action** a_t: Adjustments to the bidding parameters λ_j for $j = 0, \ldots, J$ at timestep t, represented as $(a_t^{\lambda_0}, \ldots, a_t^{\lambda_J})$.
- **Reward** r_t: The conversion value accumulated during the interval from t to $t+1$, serving as the optimization signal for the campaign objective.

For benchmarking learning-based auto-bidding methods, we adopt the mainstream offline learning paradigm, which trains policies on historical bidding logs rather than through live online interaction, thereby providing critical safety guarantees.

Our benchmark is compatible with any datasets containing the above essential MDP components, *i.e.*, states, actions, and rewards. Currently, PlatformBid is built upon the AuctionNet [32] dataset, which contains 480K training trajectories collected from 48 advertisers operating under diverse budget and CPA constraints, along with impression traffic logs for testing simulation. The dataset provides two variants based on reward sparsity: Dense and Sparse. While AuctionNet focuses on single-advertiser optimization, we extend it with a critical distinction: simultaneous multi-advertiser policy deployment within a shared auction environment. To prevent advertisers from colluding to submit low bids that would harm the platform's total revenue, a floor price mechanism is adopted. This architectural shift—from isolated single-agent evaluation to interactive multi-agent simulation—enables advertisers to respond to each other's bidding strategies in real time, thereby capturing competitive dynamics in the ad environment. We further introduce platform-level evaluation metrics to assess platform-level performance beyond individual advertiser objectives.

3.3 Evaluation Settings

PlatformBid evaluates auto-bidding algorithms under realistic multi-advertiser competitive dynamics from a platform-centric perspective. A truly effective auto-bidding algorithm must satisfy three criteria: (1) strong performance under universal adoption, ensuring platform-wide efficiency; (2) robustness when competing against diverse strategies without crowding out other participants; and (3) generalization under non-stationary conditions such as promotional events. These criteria motivate our three evaluation settings.

Setting 1: Homogeneous Competition. The first setting, depicted in Figure 2(b)-left, evaluates platform-wide performance when all N=48 advertisers deploy identical bidding policies. Each advertiser uses the same strategy and competes for impressions through the GSP auction mechanism. Performance is measured by averaging scores across all advertisers, capturing aggregate platform outcomes rather than individual advertiser success.

This setting directly mirrors the online experiments where a new auto-bidding algorithm is simultaneously deployed to all advertisers. In such deployments, the platform must evaluate whether the new algorithm improves performance at both the advertiser level and platform level compared to the in-production baseline method. Under homogeneous strategy adoption, emergent competitive dynamics reveal whether the algorithm achieves stable market equilibrium or triggers systemic inefficiencies such as bid inflation or collusive behavior. This homogeneous competition scenario thus

provides a controlled testbed for evaluating platform-wide algorithm launch, measuring both aggregate revenue generation and average advertiser welfare under identical strategy adoption.

Setting 2: Heterogeneous Competition. The second setting, illustrated in Figure 2(b)-middle, evaluates robustness under strategic diversity by partitioning advertisers into two groups: $N/2 = 24$ baseline advertisers running a fixed reference policy (vanilla Decision Transformer [3], a widely adopted method in industry) and $N/2 = 24$ test advertisers running the algorithm under evaluation. This setting captures the deployment reality where new algorithms are rolled out to a subset of advertisers while others continue using existing solutions.

This setting directly mirrors the online test with an ensemble of strategies. In practice, platforms may not deploy a single algorithm to all advertisers simultaneously, as different scenarios have unique requirements. Instead, new algorithms are typically tested on a subset of target advertisers while others continue using existing solutions. This approach enables controlled performance comparison under real competitive dynamics. Only algorithms demonstrating gains for test advertisers without degrading baseline advertiser outcomes satisfy the requirements for production deployment.

Setting 3: Promotional Competition. The third setting, shown in Figure 2(b)-right, evaluates generalization under non-stationary conditions by simulating promotional events. A subset of advertisers (7 out of 48 advertisers in this benchmark) receive doubled budgets, mimicking the budget surges accompanying sales promotions. All advertisers deploy the same algorithm, with metrics reported separately for promotional advertisers, non-promotional advertisers, and platform-wide aggregates.

This setting simulates promotional campaigns that constitute a major revenue source for ad platforms, such as *Black Friday*, where participating advertisers dramatically increase budgets to boost sales. The setting tests two critical capabilities: promotional advertisers must effectively scale their bidding strategies to utilize expanded budgets without violating CPA constraints, as optimal behavior at doubled budget diverges significantly from training distributions; meanwhile, non-promotional advertisers must maintain competitive performance despite intensified competition from budget-rich rivals. This validates whether algorithms can adapt to the non-stationary market dynamics characteristic of major promotional events while ensuring fairness across advertisers with different economic resources.

3.4 Evaluation Metrics

Our evaluation framework employs comprehensive metrics widely adopted in industrial advertising platforms, encompassing measurements at both platform and advertiser levels.

Platform-centric metrics measure aggregate system outcomes:

- **Conversion**: Average conversion value generated across all advertisers.
- **Budget Utilization**: Average fraction of allocated budgets spent, indicating market liquidity and inventory efficiency.

Advertiser-centric metrics assess individual advertiser experience and constraint satisfaction:

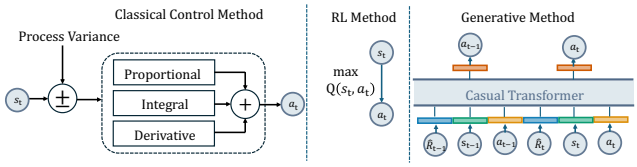

Figure 3: Three categories of auto-bidding algorithms.

- **CPA Ratio**: Average ratio of realized CPA to target CPA across advertisers, where lower values indicate better ROI.
- **CPA Ratio Variance**: Variance of CPA ratios, measuring consistency of constraint satisfaction across advertisers.
- **Exceed Rate**: Fraction of advertisers violating CPA constraints (realized CPA exceeding target CPA).
- **Qualified Rate**: Fraction of advertisers achieving CPA ratios within the acceptable range [0.8, 1.2].

A comprehensive metric balances platform revenue with advertiser satisfaction could be the **Score**, calculated as

$$\text{Score}_i = \begin{cases} \text{Conversion}_i & \text{if } \text{CPA}_i \le \text{Target}_i \\ \text{Conversion}_i \times \left(\frac{\text{Target}_i}{\text{CPA}_i}\right)^2 & \text{otherwise} \end{cases} \tag{3}$$

This formulation reflects the platform's preference for sustainable growth—high conversions achieved through constraint violations are penalized, as such patterns erode advertiser trust and long-term platform health.

More **benchmark implementation details** are in Appendix A.

4 Methods

4.1 Auto-bidding Baselines

As illustrated in Figure 3, existing auto-bidding methods can be categorized into three paradigms based on their inference mechanisms. We select representative methods from each for benchmarking:

(1) Classical Control Methods employ feedback control mechanisms without learning from data. We select PID controllers [4], which compute bidding actions by combining proportional, integral, and derivative terms of the error between actual and target CPA. Given state s_t and process variance, the controller outputs action a_t through a fixed mathematical formula.

(2) Reinforcement Learning Methods learn a policy $\pi(a|s)$ that directly maps states to actions of maximum accumulative value. While RL methods can capture complex state-action relationships in a single forward pass, they typically assume unimodal action distributions (*e.g.*, Gaussian policies) and the MDP assumption, limiting their ability to represent multi-modal bidding strategies required in dynamic auction environments. Following the offline learning paradigm, we include BCQ [7] and IQL [18], representing conservative and implicit Q-learning approaches, respectively.

(3) Generative Methods represent the current state-of-the-art in auto-bidding by formulating the problem as conditional generation. These methods fall into two sub-paradigms:

Decision-Transformer-based approaches capture the joint distribution of return-to-go R, state s, and action a through causal transformer architectures, autoregressively generating actions conditioned on target returns and historical sequences. We testify Decision Transformer (DT) [3] in both standard and score-conditioned

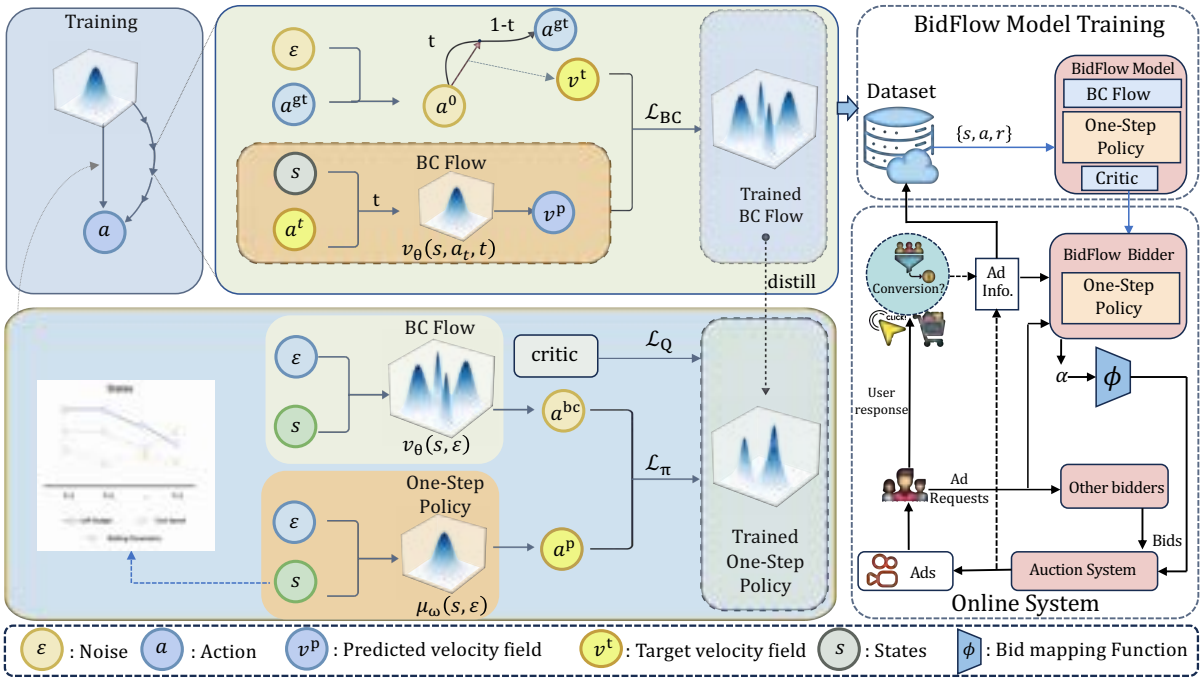

Figure 4: Overview of the BidFlow framework. Training (left): The BC flow policy learns a velocity field to transform noise into bids via flow matching. The one-step policy is trained to distill the BC flow policy while maximizing Q-values. Inference (right): Only the one-step policy is used, enabling efficient single-pass action generation without iterative sampling.

(DT score) variants, GAS [20] with post-training critic search, and GAVE [8] with value-guided exploration. However, these methods typically model action distributions as deterministic or unimodal Gaussian, limiting their expressiveness.

Diffusion-based approaches leverage the distributional modeling capacity of diffusion models to generate future states, then project them to corresponding actions through inverse dynamics models. We include CBD [19] with its completer-aligner framework.

More implementation details can be seen in Appendix B.

4.2 BidFlow: A Flow-matching-based Auto-bidding Method

However, we observe that existing auto-bidding baselines exhibit limited performance in these new settings. This degradation stems from the increasingly critical challenge of modeling complex, multi-modal bidding distributions—a challenge amplified by more dynamic market fluctuations and heightened competitive complexity at the platform level. To address this, we propose advancing auto-bidding methods through flow matching, a state-of-the-art generative modeling paradigm that has demonstrated remarkable success in capturing complex data distributions across multiple domains [9, 23, 30]. Therefore, we propose a novel flow-matching-based auto-bidding method, termed BidFlow.

As shown in Figure 4, BidFlow follows the typical flow-matching framework for offline RL [30], comprising three key components: (1) a critic network $Q_\phi(s,a)$ that estimates state-action values; (2) a behavioral cloning (BC) flow policy $\mu_\theta(s,\varepsilon)$ trained via flow matching to capture the behavioral distribution in the offline dataset; and (3) a one-step policy $\mu_\omega(s,\varepsilon)$ that maximizes Q-values while being regularized through distillation from the BC flow policy. This one-step policy enables faster inference compared to the multi-step BC flow while achieving superior performance through Q-value guidance.

The critic is trained via standard temporal difference learning. Given transitions (s,a,r,s') sampled from the offline dataset $\mathcal{D}$, the critic minimizes:

$$\mathcal{L}_Q(\phi) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\left(Q_\phi(s,a) - r - \gamma\bar{Q}(s',\mu_\omega(s',\varepsilon'))\right)^2\right], \quad (4)$$

where $\bar{Q}$ denotes the target network updated via exponential moving average, and next actions are sampled from the one-step policy with $\varepsilon' \sim \mathcal{N}(0,I)$.

The BC flow policy learns a velocity field $v_\theta(t,s,a^t)$ that transforms Gaussian noise into actions through an ODE. It is trained exclusively with the flow matching objective:

$$\mathcal{L}_{\mathrm{BC}}(\theta) = \mathbb{E}_{\substack{s,a\sim\mathcal{D},a^0\sim\mathcal{N}(0,I),\\ t\sim\mathrm{Unif}([0,1])}}\left[\|v_\theta(t,s,a^t) - (a-a^0)\|_2^2\right], \quad (5)$$

where $a^t = (1-t)a^0 + ta$ is the linear interpolation. At inference, actions are generated by numerically solving the ODE via the Euler method over M steps.

The one-step policy $\mu_\omega(s,\varepsilon)$ learns to directly map noise to actions in a single forward pass. It is trained with a combined objective that balances behavioral regularization and value maximization:

$$\mathcal{L}_\pi(\omega) = \mathbb{E}_{s,\varepsilon}\left[\|\mu_\omega(s,\varepsilon) - \mu_\theta(s,\varepsilon)\|_2^2\right] + \alpha\mathbb{E}_{s,\varepsilon}\left[-Q_\phi(s,\mu_\omega(s,\varepsilon))\right],$$

where the first term distills knowledge from the BC flow policy, the second term maximizes Q-values for outperforming the BC Flow, and α controls the regularization strength.

Table 1: Comparison under PlatformBid setting 1: Homogeneous Competition. Arrow direction "↑" and "↓" denote the performance improvement direction. Best results are bolded.

Dataset	Metric	PID	IQL	BCQ	DT	DT score	GAS	GAVE	CBD	BidFlow
Dense	Conversion ↑	183.98	331.79	236.48	347.79	323.06	353.00	334.52	293.98	**358.90**
	Budget Util ↑	0.54	0.90	0.84	0.96	0.83	0.93	**1.00**	0.98	0.88
	CPA Ratio ↓	1.39	1.00	1.24	1.04	0.91	0.97	1.14	1.28	**0.88**
	CPA Ratio Var ↓	0.69	0.07	0.06	0.06	0.03	0.03	0.12	0.16	**0.03**
	Exceed Rate ↓	0.60	0.48	0.81	0.56	0.31	0.42	0.67	0.75	**0.25**
	Qualified Rate ↑	0.27	0.60	0.42	0.60	**0.73**	0.67	0.44	0.38	0.67
	Score ↑	151.74	289.74	169.60	287.62	311.77	314.27	241.57	189.34	**348.04**
Sparse	Conversion ↑	11.21	28.56	21.56	15.65	27.83	30.15	21.56	32.98	**33.06**
	Budget Util ↑	0.62	0.87	0.44	0.45	0.58	0.65	**1.00**	0.82	0.78
	CPA Ratio ↓	1.94	1.01	0.87	1.35	0.81	**0.80**	1.11	0.97	0.85
	CPA Ratio Var ↓	0.62	**0.09**	0.32	0.89	0.15	0.10	**0.09**	0.16	0.11
	Exceed Rate ↓	0.88	0.42	0.23	0.46	**0.19**	0.27	0.50	0.44	0.25
	Qualified Rate ↑	0.15	0.38	0.31	0.21	0.27	0.31	**0.46**	0.40	0.31
	Score ↑	6.82	22.76	20.98	13.72	26.62	28.45	15.52	28.27	**30.59**

All three components are trained jointly at each iteration. The complete procedure is summarized in Algorithm 1. At deployment, only the one-step policy μ_ω is used for action selection, enabling efficient inference without iterative sampling.

Algorithm 1 Training Procedure of BidFlow

Require: Offline dataset $\mathcal{D}$, BC coefficient α

1: Initialize critic Q_ϕ, target critic $\bar{Q}_{\bar{\phi}}$, BC flow policy v_θ, one-step policy μ_ω
2: **while** not converged **do**
3: Sample batch from $\mathcal{D}$
4: Update Q_ϕ by minimizing TD loss (Eq. 4)
5: Update v_θ by minimizing flow matching loss (Eq. 5)
6: Compute distillation target $\mu_\theta(s, \varepsilon)$ via Euler method
7: Update μ_ω by minimizing $\mathcal{L}_\pi(\omega)$
8: Soft update: $\bar{\phi} \leftarrow \tau\phi + (1-\tau)\bar{\phi}$
9: **end while**
10: **return** One-step policy μ_ω

5 Experiments

5.1 Experimental Setup

We evaluate representative methods spanning the classical control approach, offline RL, and generative approaches. PID [4] serves as the classical control baseline. For offline RL, we include BCQ [7] and IQL [18], which represent conservative and implicit Q-learning paradigms respectively. The generative model family includes Decision Transformer (DT) [3] in both standard and reward-conditioned (DT score) variants, GAS [20] with post-training critic search, GAVE [8] with value-guided exploration, and CBD [19] with its completer-aligner framework. All methods are trained on the AuctionNet datasets and evaluated on both dense and sparse variants of impression traffic test data. Implementation follows the original papers [3, 4, 7, 8, 18–20, 30] with hyperparameters tuned on a held-out validation set. For BidFlow, the BC flow policy uses $M = 10$ Euler steps during training for distillation target computation. All results are averaged over 5 random seeds. More details about implementation are shown in Appendix B.

5.2 Result Analysis

Tables 1, 2, and 3 present the results under key metrics across our three evaluation settings. **Detailed results analysis are provided in Appendix C**. We only highlight three key findings below.

BidFlow outperforms baselines. BidFlow demonstrates superior performance in Setting 1, attaining the highest scores across both dense and sparse conversion configurations while maintaining the lowest CPA exceed rate, indicating effective alignment between advertiser objectives and platform welfare. In Setting 3, BidFlow exhibits advantages for both promotional advertisers and non-promotional advertisers, demonstrating robust generalization under extreme budget imbalance conditions. Nevertheless, our analysis also identifies specific limitations. Under Setting 2 with sparse rewards, BidFlow demonstrates performance degradation, with the baseline DT method struggling to acquire sufficient conversions. These findings provide important insights for developing auto-bidding methods in Setting 2 by explicitly modeling the influence between various auto-bidding methods, highlighting a promising direction for future research.

Algotithm evolution brings benefits. From simple controller methods like PID to complex approaches like DT-based methods, we observe consistent performance improvements across all settings. For instance, in Setting 1, BidFlow achieves a 95% improvement in conversion while maintaining superior cost efficiency compared to PID. The sparse dataset shows similar trends with 3× better performance. In Settings 2 and 3, advanced methods demonstrate superior adaptability. While PID and RL methods exhibit severe performance imbalance, sophisticated algorithms like DT score maintain balanced, high performance. These empirical results demonstrate that algorithmic evolution yields compounding benefits, not only in raw conversion metrics but also in cost efficiency and robustness across diverse competitive environments.

Competitive and cooperative insights. The results in Table 1 demonstrates a strong negative correlation between CPA Ratio Variance and overall Score across both Dense and Sparse datasets. BidFlow, with the lowest variance, achieves the highest Scores, while PID, with the highest variance, shows the poorest performance. This pattern reveals that low-variance strategies exhibit cooperative characteristics rather than aggressive competition, enabling multiple advertisers to achieve their targets simultaneously. Conversely,

Table 2: Key metrics comparison under PlatformBid setting 2: Heterogeneous Competition. Each result pair (Target, Average) shows the average performance of only the target advertisers by the evaluated method and the average performance of all advertisers. Result pairs are bolded if "Average" is the best.

Dataset	Metric	PID	IQL	BCQ	DT score	GAS	GAVE	CBD	BidFlow
Dense	Conversion ↑	170.54, 254.17	309.17, 331.55	231.04, 265.90	333.17, 345.44	340.38, 347.55	355.71, 330.07	313.29, 303.46	**346.88, 347.96**
	CPA Ratio ↓	2.78, 1.91	0.99, 1.03	1.19, 1.19	0.85, 0.96	0.91, 0.97	0.99, 1.06	1.20, 1.13	**0.93, 0.93**
	Score ↑	143.21, 216.74	285.24, 293.86	165.21, 190.03	**321.96, 321.92**	315.81, 315.93	287.78, 262.78	229.86, 233.34	312.69, 308.83
Sparse	Conversion ↑	11.88, 21.65	38.46, 32.59	37.92, 34.42	**44.38, 37.50**	36.96, 33.86	29.30, 32.32	42.71, 31.84	38.08, 28.77
	CPA Ratio ↓	2.08, 1.28	0.94, 0.77	0.63, 0.69	**0.63, 0.65**	0.72, 0.70	1.26, 0.84	0.89, 0.77	0.83, 1.13
	Score ↑	9.52, 20.47	34.42, 30.31	37.78, 33.59	**43.91, 36.26**	36.47, 32.77	21.20, 28.29	38.71, 29.60	35.97, 26.51

Table 3: Key metric comparison under PlatformBid Setting 3: Promotional Competition. "Pro" denotes promotional advertisers, "Non" denotes non-promotional advertisers, and "All" denotes the average performance across all advertisers.

Dataset	Type	Metric	PID	IQL	BCQ	DT	DT score	GAS	GAVE	CBD	BidFlow
Dense	Pro	Conversion ↑	550.71	674.00	543.14	673.14	682.57	690.43	657.71	456.86	**727.29**
		CPA Ratio ↓	1.49	1.10	1.15	1.06	0.98	0.95	1.15	1.32	**0.91**
		Score ↑	519.80	587.70	429.46	555.49	634.28	661.49	456.13	347.37	**701.48**
	Non	Conversion ↑	124.93	294.10	234.17	307.85	316.43	**319.95**	304.32	252.17	310.61
		CPA Ratio ↓	1.71	1.02	1.20	1.12	1.01	0.95	1.20	1.37	**0.89**
		Score ↑	111.46	247.66	171.42	230.79	271.85	277.90	203.71	169.46	**297.25**
	All	Conversion ↑	187.02	349.50	279.23	361.12	371.29	**373.98**	355.85	282.02	371.38
		CPA Ratio↓	1.68	1.03	1.19	1.11	1.01	0.95	1.20	1.36	**0.90**
		Score ↑	171.01	297.25	209.05	278.14	321.73	333.84	240.52	195.41	**356.20**
Sparse	Pro	Conversion ↑	24.86	71.00	54.57	26.00	64.71	**78.86**	48.57	60.86	71.86
		CPA Ratio ↓	**0.47**	0.95	0.67	0.96	0.65	0.71	1.49	0.75	0.68
		Score ↑	24.85	64.24	54.57	25.38	64.71	**78.85**	24.70	59.57	71.86
	Non	Conversion ↑	5.95	27.44	17.49	15.05	23.90	23.93	22.93	**29.27**	28.80
		CPA Ratio ↓	1.37	0.98	**0.81**	1.12	0.86	1.14	1.64	0.99	0.92
		Score ↑	5.86	22.50	16.93	13.00	23.10	21.08	11.79	24.58	**25.67**
	All	Conversion ↑	8.71	33.79	22.90	16.65	29.85	31.94	26.67	33.88	**35.08**
		CPA Ratio ↓	1.24	0.98	**0.79**	1.10	0.83	1.08	1.62	0.95	0.88
		Score ↑	8.63	28.59	22.42	14.81	29.17	29.51	13.67	29.68	**32.41**

high-variance strategies reflect destructive over-competition, causing budget exhaustion and reduced platform revenue. This confirms that variance serves as a reliable proxy for competition intensity, where stability promotes collective welfare.

5.3 Online Experiment Validation

To verify the effectiveness of BidFlow in practice, we deployed it on an online advertising platform. Under the MCB setting, advertisers set budgets with or without CPA/ROI constraints, and we compared BidFlow against the in-production CBD method. The experiment was conducted with identical budget allocations for each method, averaging 80 million daily impressions in two isolated environments. This configuration corresponds to Setting 2, and Table 4 presents the results based on a 4-day online experiment, where "*Target improve*" denotes the improvement on target scenarios and "*All improve*" denotes the effect on the whole MCB ad platform.

The significant improvements in platform revenue (Cost), conversions (Target Cost), and CPA ratio not only demonstrate BidFlow's superiority but also validate the benchmark's practical value for maintaining offline-online consistency. Additionally, BidFlow achieves significantly lower inference latency of 4ms compared to CBD's 12ms when measured on the same computational resources.

Table 4: Online experiment results. Performance improvement of BidFlow compared to CBD.

	Impression ↑	Cost ↑	Target Cost ↑	CPA Ratio ↓
Target improve	+0.69%	+0.79%	+2.57%	-0.70%
All improve	+0.40%	+0.79%	+2.43%	-

6 Limitations and Conclusion

There are still several limitations of this research. First, we focus exclusively on auto-bidding algorithms, while auction mechanism design also significantly impacts bidding performance and warrants further investigation. Second, we do not explicitly model inter-advertiser relationships—a challenging but important research direction given that real-world platforms may host millions of advertisers with complex competitive dynamics. Our benchmark could also be a testbed for these important research directions.

This paper addresses the gap between existing DSP-centric auto-bidding benchmarks and current unified ad platforms' requirements. We propose PlatformBid, the first platform-level auto-bidding benchmark that formulates various representative competitive settings and enables systematic evaluation of diverse methods. We further introduce BidFlow, which leverages expressive flow-matching policies to handle dynamic multi-advertiser competition. BidFlow achieves state-of-the-art performance in most PlatformBid settings, with online experiments confirming strong offline-online consistency, *e.g.*, a +2.57% improvement of target cost. With the continued development of unified ad platforms, this work establishes a critical foundation for advancing platform-centric auto-bidding research.

References

[1] Ajay, A., Du, Y., Gupta, A., Tenenbaum, J. B., Jaakkola, T. S., and Agrawal, P. Is conditional generative modeling all you need for decision making? In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023* (2023), OpenReview.net.

[2] Amin, K., Kearns, M. J., Key, P. B., and Schwaighofer, A. Budget optimization for sponsored search: Censored learning in mdps. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence, Catalina Island, CA, USA, August 14-18, 2012* (2012), N. de Freitas and K. P. Murphy, Eds., AUAI Press, pp. 54–63.

[3] Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. Decision transformer: Reinforcement learning via sequence modeling. In *Advances in Neural Information Processing Systems* (2021), vol. 34, pp. 15084–15097.

[4] Chen, Y., Berkhin, P., Anderson, B., and Devanur, N. R. Real-time bidding algorithms for performance-based display ad allocation. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2011), ACM, pp. 1307–1315.

[5] Dong, Z., Hao, J., Yuan, Y., Ni, F., Wang, Y., Li, P., and Zheng, Y. Diffuserlite: Towards real-time diffusion planning. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024* (2024), A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, Eds.

[6] Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4RL: datasets for deep data-driven reinforcement learning. *CoRR abs/2004.07219* (2020).

[7] Fujimoto, S., Meger, D., and Precup, D. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning* (2019), PMLR, pp. 2052–2062.

[8] Gao, J., Li, Y., Mao, S., Jiang, P., Jiang, N., Wang, Y., Cai, Q., Pan, F., Gai, K., An, B., and Zhao, X. Generative auto-bidding with value-guided explorations. *arXiv preprint arXiv:2504.14587* (2025).

[9] Geng, Z., Deng, M., Bai, X., Kolter, J. Z., and He, K. Mean flows for one-step generative modeling. *CoRR abs/2505.13447* (2025).

[10] Geyik, S. C., Faleev, S., Shen, J., O'Donnell, S., and Kolay, S. Joint optimization of multiple performance metrics in online video advertising. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016* (2016), B. Krishnapuram, M. Shah, A. J. Smola, C. C. Aggarwal, D. Shen, and R. Rastogi, Eds., ACM, pp. 471–480.

[11] Gomrokchi, M., Levin, O., Roach, J., and White, J. Adcraft: An advanced reinforcement learning benchmark environment for search engine marketing optimization. *CoRR abs/2306.11971* (2023).

[12] Guo, J., Huo, Y., Zhang, Z., Wang, T., Yu, C., Xu, J., Zheng, B., and Zhang, Y. Aigb: Generative auto-bidding via conditional diffusion modeling. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2024), ACM, pp. 5038–5049.

[13] He, Y., Chen, X., Wu, D., Pan, J., Tan, Q., Yu, C., Xu, J., and Zhu, X. A unified solution to constrained bidding in online display advertising. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (2021), pp. 2993–3001.

[14] Janner, M., Du, Y., Tenenbaum, J. B., and Levine, S. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA* (2022), K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, and S. Sabato, Eds., vol. 162 of *Proceedings of Machine Learning Research*, PMLR, pp. 9902–9915.

[15] Jeunen, O., Murphy, S., and Allison, B. Off-policy learning-to-bid with auctiongym. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023* (2023), A. K. Singh, Y. Sun, L. Akoglu, D. Gunopulos, X. Yan, R. Kumar, F. Ozcan, and J. Ye, Eds.

[16] Jin, J., Song, C., Li, H., Gai, K., Wang, J., and Zhang, W. Real-time bidding with multi-agent reinforcement learning in display advertising. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018* (2018), A. Cuzzocrea, J. Allan, N. W. Paton, D. Srivastava, R. Agrawal, A. Z. Broder, M. J. Zaki, K. S. Candan, A. Labrinidis, A. Schuster, and H. Wang, Eds., ACM, pp. 2193–2201.

[17] Kitts, B., Krishnan, M., Yadav, I., Zeng, Y., Badeau, G., Potter, A., Tolkachov, S., Thornburg, E., and Janga, S. R. Ad serving with multiple kpis. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017* (2017), ACM, pp. 1853–1861.

[18] Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations* (2022).

[19] Li, Y., Gao, J., Jiang, N., Mao, S., An, R., Pan, F., Zhao, X., An, B., Cai, Q., and Jiang, P. Generative auto-bidding in large-scale competitive auctions via diffusion completer-aligner. *arXiv preprint arXiv:2509.03348* (2025).

[20] Li, Y., Mao, S., Gao, J., Jiang, N., Xu, Y., Cai, Q., Pan, F., Jiang, P., and An, B. Gas: Generative auto-bidding with post-training search. *arXiv preprint arXiv:2412.17018* (2024).

[21] Liang, Z., Mu, Y., Ding, M., Ni, F., Tomizuka, M., and Luo, P. Adaptdiffuser: Diffusion models as adaptive self-evolving planners. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA* (2023), A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202 of *Proceedings of Machine Learning Research*, PMLR, pp. 20725–20745.

[22] Lin, C., Chuang, K., Wu, W. C., and Chen, M. Combining powers of two predictors in optimizing real-time bidding strategy under constrained budget. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016* (2016), S. Mukhopadhyay, C. Zhai, E. Bertino, F. Crestani, J. Mostafa, J. Tang, L. Si, X. Zhou, Y. Chang, Y. Li, and P. Sondhi, Eds., ACM, pp. 2143–2148.

[23] Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023* (2023), OpenReview.net.

[24] Liu, Z., Guo, Z., Yao, Y., Cen, Z., Yu, W., Zhang, T., and Zhao, D. Constrained decision transformer for offline safe reinforcement learning. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA* (2023), A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202 of *Proceedings of Machine Learning Research*, PMLR, pp. 21611–21630.

[25] Maehara, T., Narita, A., Baba, J., and Kawabata, T. Optimal bidding strategy for brand advertising. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden* (2018), J. Lang, Ed., ijcai.org, pp. 424–432.

[26] Muthukrishnan, S. Ad exchanges: Research issues. In *Internet and Network Economics, 5th International Workshop, WINE 2009, Rome, Italy, December 14-18, 2009. Proceedings* (2009), S. Leonardi, Ed., vol. 5929 of *Lecture Notes in Computer Science*, Springer, pp. 1–12.

[27] Ni, F., Hao, J., Mu, Y., Yuan, Y., Zheng, Y., Wang, B., and Liang, Z. Metadiffuser: Diffusion model as conditional planner for offline meta-rl. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA* (2023), A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202 of *Proceedings of Machine Learning Research*, PMLR, pp. 26087–26105.

[28] Ou, W., Chen, B., Dai, X., Zhang, W., Liu, W., Tang, R., and Yu, Y. A survey on bid optimization in real-time bidding display advertising. *ACM Trans. Knowl. Discov. Data 18*, 3 (2024), 58:1–58:31.

[29] Pan, L., Cai, Q., and Huang, L. Softmax deep double deterministic policy gradients. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual* (2020), H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds.

[30] Park, S., Li, Q., and Levine, S. Flow q-learning. *CoRR abs/2502.02538* (2025).

[31] Ren, K., Zhang, W., Chang, K., Rong, Y., Yu, Y., and Wang, J. Bidding machine: Learning to bid for directly optimizing profits in display advertising. *IEEE Trans. Knowl. Data Eng. 30*, 4 (2018), 645–659.

[32] Su, K., Huo, Y., Zhang, Z., Dou, S., Yu, C., Xu, J., Lu, Z., and Zheng, B. Auctionnet: A novel benchmark for decision-making in large-scale games. In *Advances in Neural Information Processing Systems* (2024).

[33] Sutton, R. S., and Barto, A. G. *Reinforcement learning - an introduction, 2nd Edition*. MIT Press, 2018.

[34] Wang, J., Zhang, W., and Yuan, S. Display advertising with real-time bidding (RTB) and behavioural targeting. *Found. Trends Inf. Retr. 11*, 4-5 (2017), 297–435.

[35] Wen, C., Xu, M., Zhang, Z., Zheng, Z., Wang, Y., Liu, X., Rong, Y., Xie, D., Tan, X., Yu, C., Xu, J., Wu, F., Chen, G., Zhu, X., and Zheng, B. A cooperative-competitive multi-agent framework for auto-bidding in online advertising. In *WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022* (2022), K. S. Candan, H. Liu, L. Akoglu, X. L. Dong, and J. Tang, Eds., ACM, pp. 1129–1139.

[36] Yang, X., Li, Y., Wang, H., Wu, D., Tan, Q., Xu, J., and Gai, K. Bid optimization by multivariable control in display advertising. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019* (2019), A. Teredesai, V. Kumar, Y. Li, R. Rosales, E. Terzi, and G. Karypis, Eds., ACM, pp. 1966–1974.

[37] Yang, X., Sun, D., Zhu, R., Deng, T., Guo, Z., Ding, Z., Qin, S., and Zhu, Y. Aiads: Automated and intelligent advertising system for sponsored search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019* (2019), A. Teredesai, V. Kumar, Y. Li, R. Rosales, E. Terzi, and G. Karypis, Eds., ACM, pp. 1881–1890.

[38] Yuan, Y., Wang, F., Li, J., and Qin, R. A survey on real time bidding advertising. In *Proceedings of 2014 IEEE International Conference on Service Operations and Logistics, and Informatics, Qingdao, China, October 8-10, 2014* (2014), IEEE, pp. 418–423.

[39] Zhang, W., Yuan, S., and Wang, J. Optimal real-time bidding for display advertising. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014* (2014), S. A. Macskassy, C. Perlich, J. Leskovec, W. Wang, and R. Ghani, Eds., ACM, pp. 1077–1086.

[40] Zhang, W., Yuan, S., and Wang, J. Real-time bidding benchmarking with ipinyou dataset. *CoRR abs/1407.7073* (2014).

[41] Zhou, S., Du, Y., Zhang, S., Xu, M., Shen, Y., Xiao, W., Yeung, D., and Gan, C. Adaptive online replanning with diffusion models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023* (2023), A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds.

[42] Zhu, H., Jin, J., Tan, C., Pan, F., Zeng, Y., Li, H., and Gai, K. Optimized cost per click in taobao display advertising. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017* (2017), ACM, pp. 2191–2200.

Appendix

A Benchmark Implementation Details

PlatformBid is implemented as an extension to the AuctionNet simulation environment. The simulator processes impression opportunities chronologically, soliciting bids from all active advertisers and resolving auctions according to GSP rules with floor price enforcement.

To prevent collusive bid suppression, which is a phenomenon observed particularly in sparse-reward environments where advertisers may collectively lower bids to reduce costs. To tackle this issue, we introduce a floor price mechanism. The floor price is set at 80% of the original least winning cost from the AuctionNet dataset. If a winning bid falls below the floor price, the impression is not awarded and no transaction occurs. This mechanism preserves market integrity by ensuring that strategic bid reduction cannot extract value below a reasonable threshold, aligning individual incentives with platform revenues.

PlatformBid tackles its modularity goals via defining unifying abstractions over the auto-bidding evaluation pipeline. As depicted in Figure 5, components including agent strategies (`agent.py`), utility functions (`utls.py`), and bidding strategies (`strategy.py`) are gathered into evaluation scripts that are agnostic of the specific auto-bidding algorithm implementations. Structured command-line configurations allow users to easily specify the evaluation setting, dataset variant, competing algorithms, and auction parameters (*e.g.*, floor price rate), making it possible to go directly from one-line inputs to comprehensive evaluation outputs.

B Method Details

For the PID controller, we set $K_p = 0.1$, $K_i = 0.01$, and $K_d = 0.03$ in all settings. For the DT reweight variant, the reweighting coefficient is set to $w = 0.2$ across all settings. For GAS, we employ three critic networks for both dense and sparse datasets across all settings. For BidFlow, the BC coefficient α is set as follows: $\alpha = 10$ for dense data in Settings 1 and 3, $\alpha = 0.3$ for dense data in Setting 2, $\alpha = 1$ for sparse data in Setting 2, and $\alpha = 0.03$ for sparse data in Settings 1 and 3.

C Detailed Experimental Analysis

This section provides comprehensive per-method analysis for each evaluation setting, complementing the main results in Section 5.

C.1 Setting 1: Homogeneous Competition

Table 1 presents results when all 48 advertisers deploy identical strategies.

Classical control method. PID achieves a score of 151.74 on dense data with the lowest budget utilization (0.54) and highest CPA ratio (1.39). When all 48 advertisers deploy PID simultaneously, the resulting bidding dynamics deviate substantially from these calibration conditions, causing persistent overshoot in CPA control. On sparse data, PID deteriorates further (score 6.82) because the infrequent reward signals provide insufficient feedback for its integral and derivative terms to converge.

Generative methods. DT achieves moderate performance (score 287.62 on dense) with reasonable constraint satisfaction (CPA ratio 1.04), benefiting from its return-conditioned generation that implicitly encodes budget-awareness. Its reward-conditioned variant DT score improves constraint control (CPA ratio 0.91, exceed rate 0.31) by enabling explicit targeting of desired cost-performance trade-offs, yielding a better overall score of 311.77. The 24-point score improvement underscores that explicit reward conditioning is a simple yet effective mechanism for constraint awareness in multi-advertiser settings.

GAS further improves upon DT variants by applying critic-based post-training search over generated candidates, achieving a score of 314.27 with an exceed rate of 0.42. The critic effectively filters out constraint-violating actions before execution. However, GAS's search operates over a finite candidate set sampled from the base DT policy, which limits its ability to discover novel bidding strategies that fall outside the DT's generative support. This becomes a bottleneck when the emergent price dynamics in multi-advertiser competition require responses not well-represented in the training distribution.

GAVE attains the highest budget utilization (1.00) but suffers from severe constraint violations (exceed rate 0.67, CPA ratio 1.14), resulting in a penalized score of only 241.57. GAVE's value-guided exploration mechanism becomes counterproductive in homogeneous competition: all 48 advertisers simultaneously explore the same high-value regions, driving up auction prices.

CBD achieves moderate reward (293.98) but poor constraint control (CPA ratio 1.28, exceed rate 0.75), yielding a score of 189.34. The diffusion-based trajectory generation introduces accumulated errors when the denoising process conditions on states that themselves deviate from training distributions, which will be amplified by multi-advertiser interaction.

RL methods. IQL maintains reasonable constraint satisfaction (CPA ratio 1.00, exceed rate 0.48) but achieves lower conversion (331.79) due to its expectile-based conservative value estimation. This conservatism provides a useful safety margin in stable environments; however, in multi-advertiser competition, it causes IQL to systematically undervalue contested impressions and forfeit opportunities to more aggressive competitors.

BCQ performs poorly (score 169.60, exceed rate 0.81) because its behavior-constrained policy generation is anchored to the training data distribution. When 48 BCQ agents compete simultaneously, the realized state-action distribution diverges rapidly from the offline data, yet BCQ continues to generate actions from a support that no longer reflects the actual competitive dynamics. This is a

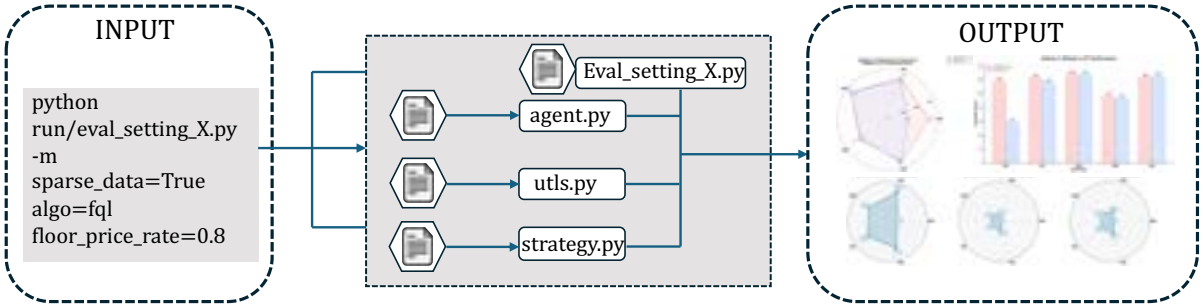

Figure 5: PlatformBid execution diagram. Users run evaluations as configurable experiments, where each setting loads its components from the respective Python modules.

Table 5: Key metrics comparison under PlatformBid setting 2: Heterogeneous Competition. Each result pair (Target, Average) shows the average performance of only the target advertisers by the evaluated method and the average performance of all advertisers. Result pairs are bold if "Average" is the best.

Dataset	Metric	PID	IQL	BCQ	DT score	GAS	GAVE	CBD	BidFlow
Dense	Conversion ↑	170.54, 254.17	309.17, 331.55	231.04, 265.90	333.17, 345.44	340.38, 347.55	355.71, 330.07	313.29, 303.46	**346.88, 347.96**
	Budget Util ↑	0.34, 0.65	0.88, 0.93	0.85, 0.90	0.85, 0.89	0.89, 0.93	**0.99, 0.97**	0.98, 0.94	0.93, 0.95
	CPA Ratio ↓	2.78, 1.91	0.99, 1.03	1.19, 1.19	0.85, 0.96	0.91, 0.97	0.99, 1.06	1.20, 1.13	**0.93, 0.93**
	Qualified Rate ↑	0.17, 0.40	0.83, 0.69	0.54, 0.52	0.63, 0.63	**0.67, 0.67**	0.46, 0.48	0.50, 0.57	0.50, 0.57
	Score ↑	143.21, 216.74	285.24, 293.86	165.21, 190.03	**321.96, 321.92**	315.81, 315.93	287.78, 262.78	229.86, 233.34	312.69, 308.83
Sparse	Conversion ↑	11.88, 21.65	38.46, 32.59	37.92, 34.42	**44.38, 37.50**	36.96, 33.86	29.30, 32.32	42.71, 31.84	38.08, 28.77
	Budget Util ↑	0.51, 0.49	0.99, 0.69	0.68, 0.66	**0.77, 0.69**	0.75, 0.68	1.00, 0.70	0.98, 0.67	0.86, 0.66
	CPA Ratio ↓	2.08, 1.28	0.94, 0.77	0.63, 0.69	**0.63, 0.65**	0.72, 0.70	1.26, 0.84	0.89, 0.77	0.83, 1.13
	Qualified Rate ↑	0.25, 0.15	0.42, 0.25	0.25, 0.29	0.13, 0.13	**0.33, 0.33**	0.33, 0.19	0.17, 0.19	0.33, 0.29
	Score ↑	9.52, 20.47	34.42, 30.31	37.78, 33.59	**43.91, 36.26**	36.47, 32.77	21.20, 28.29	38.71, 29.60	35.97, 26.51

structural limitation of behavior-cloning-constrained methods in non-stationary multi-agent environments.

BidFlow. BidFlow achieves the highest score on both dense (348.04) and sparse (30.59) datasets while maintaining the lowest exceed rate (0.25) and near-lowest CPA ratio variance (0.03 on dense, 0.11 on sparse). The key mechanism is its flow-based policy's capacity to represent multi-modal action distributions conditioned on market state. In homogeneous competition, the optimal bidding strategy is inherently multi-modal: for high-value impressions with few competitors in the current timestep, aggressive bidding maximizes value; for commoditized impressions with intense competition, conservative bids preserve budget for better opportunities. BidFlow's flow-matching architecture captures this conditional complexity through its learned velocity field, whereas Gaussian-policy methods (IQL, BCQ) collapse to unimodal approximations that average over these distinct regimes.

Sparse-reward amplification. The sparse-reward setting amplifies all observed phenomena. CBD achieves competitive conversion (32.98) but with poor constraint control (CPA ratio 0.97, exceed rate 0.44), as the infrequent conversion signals make its trajectory-level return estimation unreliable. BidFlow maintains balanced performance (conversion 33.06, CPA ratio 0.85, exceed rate 0.25), demonstrating that its Q-learning component provides stable value estimation even under sparse feedback. The floor price mechanism proves essential in this setting: without it, we observed collusive bid suppression where all 48 advertisers collectively converged to submitting near-zero bids, extracting impressions at negligible cost.

C.2 Setting 2: Heterogeneous Competition

Table 5 evaluates robustness when 24 test advertisers compete against 24 baseline advertisers running vanilla Decision Transformer. The dual-metric structure (algorithm score, average score) reveals whether performance gains represent genuine market efficiency improvements or mere value redistribution.

Classical control method. PID produces severe performance asymmetry across both data regimes. On dense data (170.54, 254.17), the baseline DT advertisers maintain reasonable performance while the PID test group collapses, with its CPA ratio reaching 2.78—far exceeding the target constraint. The fixed-gain controller, lacking any model of competitor behavior, generates erratic bid fluctuations that fail to win cost-effective impressions against the adaptive DT opponents, resulting in heavy overspending on the few auctions it does win. The sparse setting exhibits a similar pattern (9.52, 20.47) with the PID's CPA ratio at 2.08, confirming that PID's inability to internalize competitive dynamics is a structural limitation rather than a data-regime-specific artifact.

Generative methods. DT score demonstrates strong performance on dense data (321.96, 321.92), with the reward-conditioning mechanism providing effective differentiation from the vanilla DT baseline.

Table 6: Key metric comparison under PlatformBid Setting 3: Promotional Competition. "Pro" denotes promotional advertisers, "Non" denotes non-promotional advertisers, and "All" denotes the average performance across all advertisers.

Dataset	Type	Metric	PID	IQL	BCQ	DT	DT score	GAS	GAVE	CBD	BidFlow
Dense	Pro	Conversion ↑	550.71	674.00	543.14	673.14	682.57	690.43	657.71	456.86	**727.29**
		CPA Ratio↓	1.49	1.10	1.15	1.06	0.98	0.95	1.15	1.32	**0.91**
		Exceed ↓	0.57	0.43	0.71	0.71	0.42	**0.28**	0.86	0.71	0.29
		Score ↑	519.80	587.70	429.46	555.49	634.28	661.49	456.13	347.37	**701.48**
	Non	Conversion ↑	124.93	294.10	234.17	307.85	316.43	**319.95**	304.32	252.17	310.61
		CPA Ratio↓	1.71	1.02	1.20	1.12	1.01	0.95	1.20	1.37	**0.89**
		Exceed ↓	0.68	0.49	0.85	0.61	0.51	0.41	0.66	0.71	**0.34**
		Score ↑	111.46	247.66	171.42	230.79	271.85	277.90	203.71	169.46	**297.25**
	All	Conversion ↑	187.02	349.50	279.23	361.12	371.29	**373.98**	355.85	282.02	371.38
		CPA Ratio↓	1.68	1.03	1.19	1.11	1.01	0.95	1.20	1.36	**0.90**
		Exceed ↓	0.67	0.48	0.83	0.63	0.49	**0.40**	0.69	0.71	0.69
		Score ↑	171.01	297.25	209.05	278.14	321.73	333.84	240.52	195.41	**356.20**
Sparse	Pro	Conversion ↑	24.86	71.00	54.57	26.00	64.71	**78.86**	48.57	60.86	71.86
		CPA Ratio ↓	**0.47**	0.95	0.67	0.96	0.65	0.71	1.49	0.75	0.68
		Exceed ↓	0.38	0.14	**0.00**	0.29	**0.00**	**0.00**	0.86	0.14	**0.00**
		Score ↑	24.85	64.24	54.57	25.38	64.71	**78.85**	24.70	59.57	71.86
	Non	Conversion ↑	5.95	27.44	17.49	15.05	23.90	23.93	22.93	**29.27**	28.80
		CPA Ratio ↓	1.37	0.98	**0.81**	1.12	0.86	1.14	1.64	0.99	0.92
		Exceed ↓	0.44	0.39	**0.20**	0.37	0.32	0.41	0.78	0.44	0.29
		Score ↑	5.86	22.50	16.93	13.00	23.10	21.08	11.79	24.58	**25.67**
	All	Conversion ↑	8.71	33.79	22.90	16.65	29.85	31.94	26.67	33.88	**35.08**
		CPA Ratio ↓	1.24	0.98	**0.79**	1.10	0.83	1.08	1.62	0.95	0.88
		Exceed ↓	0.38	0.35	**0.17**	0.35	0.27	0.35	0.79	0.40	0.25
		Score ↑	8.63	28.59	22.42	14.81	29.17	29.51	13.67	29.68	**32.41**

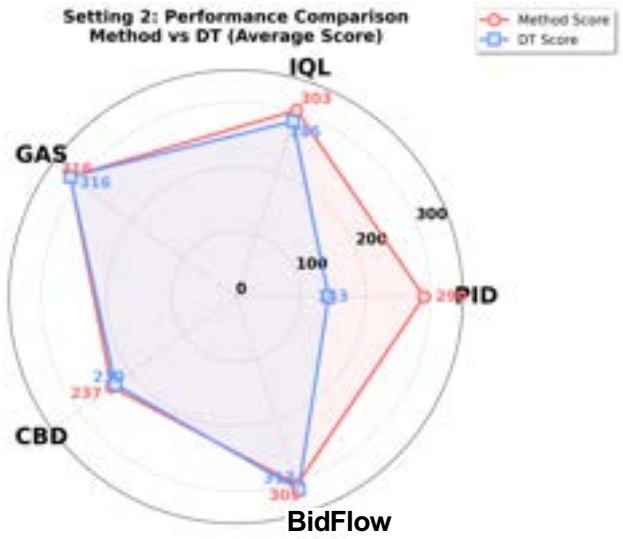

Figure 6: Qualitative comparison of representative methods under Setting 2 (Dense, Heterogeneous Competition). Left: Radar chart showing test group scores (red) and baseline DT group scores (blue) for each method. Right: Bar chart of the same data. Methods like GAS and BidFlow achieve high scores for both groups, reflecting genuine market efficiency gains, whereas PID exhibits severe baseline degradation and CBD shows balanced but overall low performance.

The test group's 10-point advantage reflects genuine strategy improvement, as baseline scores remain near their standalone level. However, this advantage collapses in sparse settings (43.90, 36.26): the test group's strong performance comes at significant baseline cost.

GAS achieves near-perfect parity on dense data (315.81, 315.93)—the most symmetric outcome among all methods. GAS's critic-based search selects actions that are locally optimal *given* the current competitive landscape, rather than actions optimal in the training distribution. The critic effectively acts as an adaptive filter that avoids bidding strategies which, while individually profitable, would displace baseline competitors. This property makes GAS particularly well-suited for AB testing deployments where platforms require strict non-degradation guarantees.

GAVE shows significant crowding-out on dense data (287.78, 262.78). The 50-point baseline degradation indicates that GAVE's value-guided exploration directs test advertisers toward the same high-value impression segments that baseline DT advertisers target, creating a zero-sum competitive dynamic. From a platform perspective, deploying GAVE would reduce aggregate revenues despite appearing to improve the test group.

CBD maintains balance but at low absolute levels (229.86, 233.34 on dense), confirming that its trajectory generation produces suboptimal bidding strategies regardless of the competitive context.

RL methods. IQL achieves reasonable balance on dense data (285.24, 293.86). Its conservative value estimation, while limiting peak performance, prevents the aggressive bidding that drives crowding-out in methods like GAVE. The 17-point gap between baseline and test

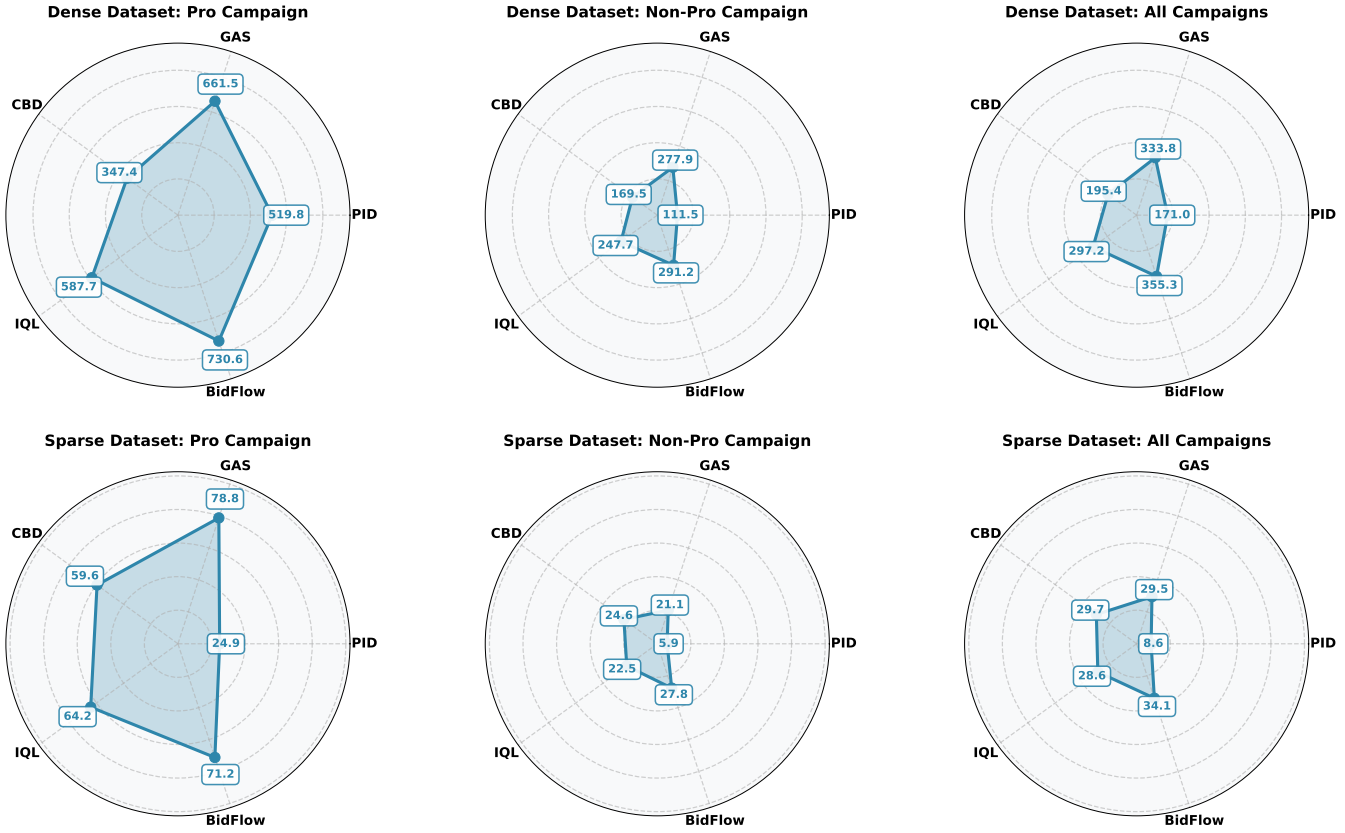

Figure 7: Score comparison across methods under Setting 3 (Promotional Competition). Radar charts display scores for five methods (GAS, PID, BidFlow, IQL, CBD) across promotional (Pro), non-promotional (Non-Pro), and all campaigns on dense (top) and sparse (bottom) datasets. Larger areas indicate better overall performance.

scores is moderate and acceptable for AB testing criteria. On sparse data (34.42, 30.31), IQL maintains this balance with a smaller gap.

BCQ performs poorly across both groups on dense data (165.21, 190.03). The test group's low score indicates that BCQ cannot improve even for its own advertisers, while simultaneously degrading baseline performance.

BidFlow. BidFlow achieves scores of 312.69 and 308.83 (Average) on dense data. Notably, the baseline score (305) remains high, suggesting that BidFlow's bidding behavior creates positive externalities that benefit all market participants rather than extracting value from competitors. However, BidFlow's test group score trails GAS by 11 points (305 vs. 316), indicating that its emphasis on market stability comes at some cost to individual competitiveness. On sparse data (35.97, 26.51), the baseline degradation is more pronounced, revealing that BidFlow's Q-learning component struggles to provide reliable value guidance when conversion signals are rare, causing the policy to fall back on behavioral patterns that do not account well for baseline advertiser dynamics.

C.3 Setting 3: Promotional Competition

Table 6 evaluates generalization under distribution shift when 7 advertisers receive doubled budgets. This creates an asymmetric market where promotional advertisers must scale their bidding to utilize expanded budgets, while non-promotional advertisers face intensified competition from budget-rich rivals.

Classical control method. PID degrades sharply for both advertiser groups on dense data (promotional score 519.80, non-promotional score 111.46). The core issue is a calibration mismatch: PID's integral term was tuned under normal budget levels, so doubling the budget causes persistent accumulation of control error that drives oscillatory, wasteful bidding for promotional advertisers. Non-promotional advertisers fare even worse because the fixed-gain controller has no mechanism to compensate for the elevated clearing prices introduced by budget-rich competitors. On sparse data the promotional conversion falls to 24.86 with a CPA ratio of only 0.47, reflecting extreme budget under-utilization.

Generative methods. DT achieves moderate promotional performance (conversion 673.14, score 555.49 on dense) but with a high exceed rate (0.71). The return-conditioned generation cannot distinguish between "high return from efficient bidding" and "high return from budget overspending," causing frequent CPA constraint violations when the budget doubles.

DT score significantly improves constraint control for promotional advertisers (CPA ratio 0.98, exceed rate 0.42, score 634.28) by

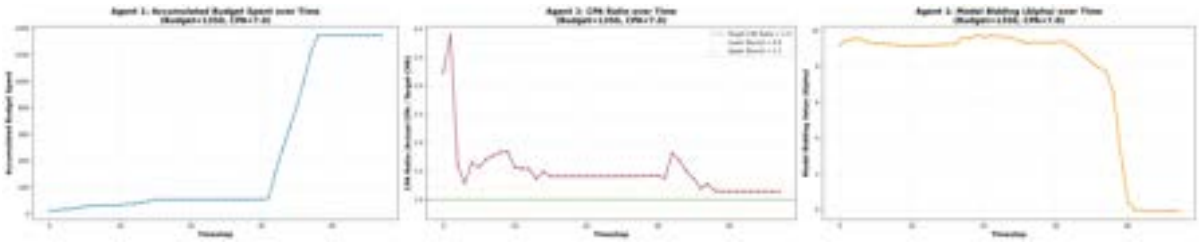

Figure 8: A bad case of BidFlow method.

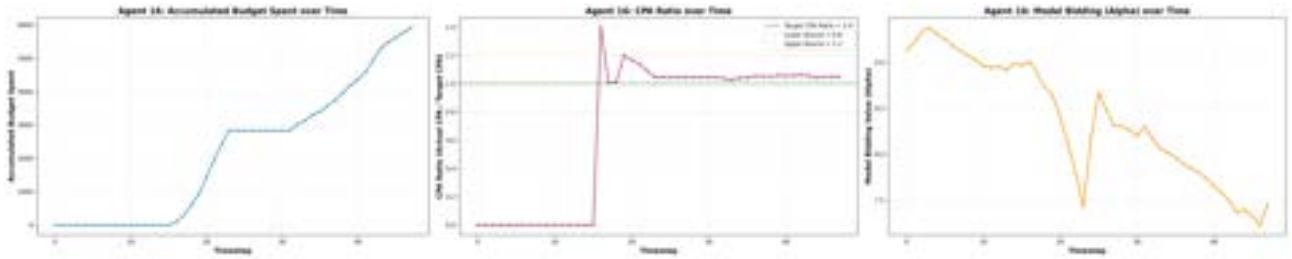

Figure 9: A good case of BidFlow method.

explicitly conditioning on target reward levels that account for the expanded budget capacity. The 79-point score improvement over DT demonstrates that reward conditioning provides meaningful distribution shift robustness.

GAS achieves the strongest promotional performance among generative methods (score 661.49 on dense, conversion 78.86 on sparse). Its critic-based search naturally adapts to the changed competitive landscape by evaluating candidate bids against current market conditions rather than training-time assumptions. The critic effectively recalibrates bid levels to the elevated price environment caused by promotional budget injection.

GAVE struggles severely with constraint control under budget expansion. On dense data, promotional exceed rate reaches 0.86; on sparse data, CPA ratio spikes to 1.49 with the same exceed rate. The value-guided exploration mechanism becomes catastrophic when combined with doubled budgets: GAVE's promotional advertisers aggressively explore high-value impressions with greater purchasing power, driving prices beyond constraint-feasible levels.

CBD reveals its most pronounced limitation in this setting. Promotional conversion of only 456.86 on dense data (versus 673+ for DT, GAS, and IQL) directly reflects the completer module's inability to generate plausible trajectories when budget utilization patterns deviate from training data. The diffusion model was trained on trajectories where budget consumption follows a specific temporal profile; doubling the budget requires a fundamentally different consumption trajectory that falls outside the learned manifold. This makes CBD unsuitable for platforms where promotional events and budget adjustments are routine.

RL methods. IQL achieves balanced promotional performance (conversion 674.00, CPA ratio 1.10, score 587.70 on dense). Its expectile-based value function provides moderate robustness to the budget distribution shift, though conservative estimation limits its ability to fully exploit the expanded budget. The non-promotional score (247.66) is competitive but trails BidFlow and GAS, as IQL's uniform conservatism does not differentiate between heightened competition from promotional rivals and normal market conditions.

BCQ performs poorly for promotional advertisers (score 429.46 on dense). Its behavior-constrained policy generation restricts actions to the support of normal-budget training data, preventing the larger bids needed to effectively deploy doubled budgets. This is the most direct demonstration of behavior cloning's distributional fragility across all three settings.

BidFlow. BidFlow demonstrates the strongest overall adaptation on dense data. Promotional advertisers achieve the highest conversion (727.29), best constraint satisfaction (CPA ratio 0.91, exceed rate 0.29), and highest score (701.48). The flow-based policy's multi-modal representation is particularly advantageous here: it can shift its action distribution toward higher bid levels for promotional advertisers while maintaining the learned constraint-awareness from training. Crucially, BidFlow simultaneously achieves the highest non-promotional score (297.25 vs. GAS's 277.90), a 20-point margin indicating that BidFlow's bidding adjustments expand overall market value rather than merely redistributing it.

On sparse data, BidFlow maintains controlled promotional adaptation (conversion 71.86, CPA ratio 0.68, exceed rate 0.00) and achieves the highest non-promotional score (25.67). The gap to GAS's promotional conversion (78.86 vs. 71.86) suggests that BidFlow's Q-learning component slightly underestimates promotional impression values in sparse settings—a direction for future improvement through adaptive α scheduling.

D Cases Analysis

Bad case analysis. As shown in Figure 8, the case highlights a misalignment between constraint feedback and bid-scaling control. Although the realized CPA ratio remains substantially above the target, the bidding coefficient α maintains at a high value over the same period. This inconsistency is also reflected in the accumulated-budget curve (Left). The agent preserves a large amount of remaining budget in the early stage, but later enters a rapid spending phase where the remaining budget drops sharply. In other words, the policy behaves as if it were compensating for earlier under-delivery by sustaining an aggressive α, even when the CPA signal already indicates overpaying. Such "late catch-up" pacing tends to amplify market-clearing prices under competition, which further worsens CPA. Once the budget becomes tight, α then collapses, leading to a stop-and-go pattern. Overall, this case suggests that, under this budget setting and competitive environment, the method fails to effectively couple CPA control with budget pacing for this agent.

Good case analysis. As shown in Figure 9 (middle), after the agent win the impression, the CPA ratio moves into the moderate range (0.8–1.2). Meanwhile, as illustrated by the α curve in Figure 9 (right), the agent reduced its bidding coefficient α, which keeps the CPA ratio at a desirable level for the remainder of the bidding process. In this case, the budget-spent curve in Figure 9 (left) is also smooth, indicating that the agent does not bid aggressively.