ENHANCING GENERATIVE AUTO-BIDDING WITH OFFLINE REWARD EVALUATION AND POLICY SEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

Auto-bidding serves as a critical tool for advertisers to improve their advertising performance. Recent progress has demonstrated that AI-Generated Bidding (AIGB), which learns a conditional generative planner from offline data, achieves superior performance compared to typical offline reinforcement learning (RL)-based auto-bidding methods. However, existing AIGB methods still face a performance bottleneck due to their inherent inability to explore beyond the static offline dataset. To address this, we propose AIGB-Pearl (*Planning with EvaluAtor via RL*), a novel method that integrates generative planning and policy optimization. The core of AIGB-Pearl lies in constructing a trajectory evaluator for scoring generation quality and designing a provably sound KL-Lipschitz-constrained score maximization scheme to ensure safe and efficient exploration beyond the offline dataset. A practical algorithm incorporating the synchronous coupling technique is further devised to ensure the model regularity required by the proposed scheme. Extensive experiments on both simulated and real-world advertising systems demonstrate the state-of-the-art performance of our approach.

1 INTRODUCTION

The increasing demand for commercial digitalization has facilitated the development of the auto-bidding technique in online advertising. Distinguished from traditional manual bidding products, auto-bidding provides advertisers with an efficient and flexible scheme to automatically optimize bids in dynamic and competitive environments (Balseiro et al., 2021a; Deng et al., 2021; Balseiro et al., 2021b). Technically, auto-bidding can be viewed as an offline sequential decision-making problem, which aims to maximize the advertising performance over the bidding episode, relying solely on a static offline dataset due to system limitations (Mou et al., 2022), such as safety concerns.

As a standard approach to offline decision-making problems, offline reinforcement learning (RL) (Kumar et al., 2020) is widely adopted to solve the auto-bidding problem. By employing conservative policy search schemes, offline RL mitigates the infamous *out-of-distribution* (OOD) problem (Fujimoto et al., 2019), enabling reliable generalization beyond the offline dataset. However, due to the employment of bootstrapped value estimations, offline RL methods typically suffer from the training instability issue (Peng et al., 2024), risking policy performance degradation and instability.

Recent advances in generative models shed new light on offline decision-making problems (Zhu et al., 2023; Kang et al., 2023). Specifically, AI-generated bidding (AIGB) models the auto-bidding as a trajectory generation task and employs a generative model to approximate the conditional trajectory distribution of the offline dataset (Guo et al., 2024). AIGB avoids the requirement for bootstrapping and exhibits more stable training and superior performance. However, the modeling approach in AIGB does not explicitly align with the performance optimization objective of the auto-bidding problem, preventing it from actively improving toward higher performance. As a result, AIGB primarily imitates well-behaved trajectories from the offline dataset, lacking the ability to explore higher-quality trajectories beyond the offline data (Ajay et al., 2023).

Hence, there arises a question: *built on AIGB, the latest state-of-the-art auto-bidding method, can we devise a plausible scheme to involve policy optimization in its generative model?* To this end, a natural idea is to integrate offline RL methods into AIGB. However, it is nontrivial to implement in the auto-bidding problem since (i) there is a lack of reward signals in AIGB to guide the generative model. Specifically, the generation quality of the generative model remains unknown during

training, making it infeasible to explore new trajectories beyond the offline dataset; (ii) no dedicated offline RL algorithm exists for AIGB. In particular, theoretical analysis that guarantees safe generalization and mitigates OOD issues for generative models in auto-bidding remains largely unexplored.

To address these critical challenges, we propose **AIGB-Pearl** (Planning with EvaluAtor via RL), an RL-enhanced version of AIGB that learns a *trajectory evaluator* to score generation quality and drive exploration of the generative model through continuous interaction. The evaluator is trained through supervised learning on the offline dataset. Crucially, to mitigate the OOD problem and ensure reliable evaluator utilization, we examine the theoretical upper bound on the evaluator's bias. Then, guided by this analysis, we establish a KL-Lipschitz-constrained score maximization objective with a provable sub-optimality bound, enabling safe and effective exploration beyond the offline data. Moreover, to perform constrained score maximization, we design a practical algorithm incorporating the *synchronous coupling* technique, which helps ensure the Lipschitz requirement of the generative model. In addition, we note that AIGB-Pearl operates without the need for bootstrapping, exhibiting greater training stability compared to offline RL methods.

To summarize, our contributions in this paper are fourfold: (i) we propose a novel generative auto-bidding method, AIGB-Pearl, that enables continuous improvement in generation quality through exploration beyond the offline dataset; (ii) we propose a provable KL-Lipschitz constrained score maximization objective with a sub-optimality bound, ensuring a safe and effective generalization beyond the offline dataset; (iii) we devise a practical algorithm with synchronous coupling that effectively ensures the Lipschitz requirement for the generative model; (iv) extensive simulated and real-world experiments demonstrate that AIGB-Pearl achieves SOTA performance and verify the effectiveness of the developed techniques in enhancing safe and effective generalization.

2 PRELIMINARIES

2.1 PROBLEM STATEMENT

This work studies the auto-bidding problem for a single advertiser subject to a budget constraint. The auction mechanism follows a sealed-bid, second-price rule. The objective is to devise a bidding policy that maximizes the cumulative value of the impressions won over a finite bidding episode (e.g., a day) within a budget $B > 0$. As established in (He et al., 2021), the optimal bid for each impression i is proportional to its intrinsic value $v_i > 0$, scaled by a non-negative factor $a \geq 0$ that remains consistent across all impressions. Under this strategy, the advertiser wins an impression i if $av_i \geq p_i$ and pays p_i upon winning, where $p_i > 0$ is the market price. The Return on Investment (ROI) of impression i is defined as v_i/p_i, and we denote its upper bound as $R_m \triangleq \max_i v_i/p_i$.

However, the scaling factor is unknown in advance, and the volatility of impressions drives its continual change throughout the bidding process. Hence, a standard practice involves recalibrating the scaling factor a at fixed intervals of $T \in \mathbb{N}_+$ time steps (Guo et al., 2024; He et al., 2021; Mou et al., 2022). This casts the auto-bidding to a sequential decision-making problem.

Specifically, the auto-bidding problem can be modeled as a Markov Decision Process (MDP) $< \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P} >$. The state $s_t \triangleq [t, \bar{c}_{t-1}, x] \in \mathcal{S}$ is composed of the current time step $t \in [T]$, the cost ratio $\bar{c}_{t-1} = c_{t-1}/B > 0$ where c_{t-1} is the advertiser's cost for impressions won between time step $t-1$ and t, and a static advertiser-specific feature x that includes the budget and many other individual information. The action $a_t \in \mathcal{A}$ denotes the calibrated scaling factor at time step t. The reward $r_t \geq 0$ describes the value of the impressions won between time steps t and $t+1$, and $\mathcal{P}$ denotes the state transition rule. The auto-bidding problem can be formulated as:

$$\max_{a_1, a_2, \cdots, a_T} \mathbb{E}_{s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)} \left[\sum_{t=1}^{T} r_t \right], \quad \text{s.t.} \sum_{t=1}^{T} c_t \leq B. \tag{1}$$

Offline Setting. Due to safety concerns—common in real-world advertising systems—we are restricted to learning the optimal bidding policy from a static *offline dataset* $\mathcal{D}$ composed of historical states and actions along with associated rewards. This makes the considered auto-bidding problem an offline sequential decision-making task.

2.2 OFFLINE RL METHODS

RL constitutes a standard approach for auto-bidding problems, seeking an optimal bidding policy $\pi : \mathcal{S} \to \mathcal{A}$ that maximizes cumulative rewards. Specifically, this is typically achieved by learning a Q-value function, $Q(s_t, a_t) \triangleq \mathbb{E}_\pi[\sum_{t'=t}^{T} r_{t'}]$, through temporal difference (TD) error minimization:

$$\min_Q \mathbb{E}_{(s_t,a_t,r_t,s_{t+1})\sim\mathcal{D}}[Q(s_t, a_t) - r_t - \max_{a_{t+1}} \hat{Q}(s_{t+1}, a_{t+1})]^2, \tag{2}$$

where $\hat{Q}$ is a target Q-value function with parameters updated via Polyak averaging (Mnih et al., 2015). Upon convergence, the optimal bidding policy is derived as $\pi(s_t) = \arg\max_{a_t} Q(s_t, a_t)$.

Due to the offline setting of the considered auto-bidding problem, directly employing Eq. 2 results in the infamous *out-of-distribution* (OOD) problem (Fujimoto et al., 2019), making the policy erroneously deviate from the offline dataset $\mathcal{D}$. As a standard solution, offline RL (Yu et al., 2020; Kumar et al., 2020; Kidambi et al., 2020; Wang et al., 2022) constrains the policy's behavior near $\mathcal{D}$ during TD learning, enabling reliable generalization beyond the offline dataset.

However, offline RL methods notoriously suffer from training instability caused by TD-learning (Peng et al., 2024), where the bootstrapped value of the Q function serves as its training label, resulting in an erroneous ground truth. Training stability is critical in auto-bidding due to the absence of an accurate offline policy evaluation method and the high cost of online policy examination in a real-world advertising system (Mou et al., 2022).

2.3 GENERATIVE AUTO-BIDDING METHODS

Definition 1 (Trajectory and Trajectory Quality)**.** *The* trajectory *is formalized as the state sequence throughout the bidding episode, i.e.,* $\tau \triangleq [s_1, s_2, \cdots, s_T]$*. The trajectory quality is defined as the normalized cumulative reward of the trajectory, i.e.,* $y(\tau) \triangleq \sum_{t=1}^{T} \bar{r}_t$ [1]*, where* $\bar{r}_t = r_t/B$*.*

Unlike RL methods, the AI-generated auto-bidding (AIGB) (Guo et al., 2024) treats the auto-bidding problem as a sequence generation task. Specifically, a conditional generative model is employed to fit the conditional trajectory distribution $p_\theta(\tau|y(\tau))$ within the offline dataset $\mathcal{D}$, i.e.,

$$\max_\theta \mathbb{E}_{(\tau,y(\tau))\sim\mathcal{D}}[\log p_\theta(\tau|y(\tau))], \tag{3}$$

where θ denotes the parameter. Let $y_m > 0$ be the maximum trajectory quality in $\mathcal{D}$, we have $\forall y \in \mathcal{D}, y \in [0, y_m]$. During inference, AIGB follows a *planning-and-control* architecture. Specifically, at each time step, a trajectory is sampled from the trained generative model that acts as the *planner*, with a manually set condition $y^* \triangleq (1 + \epsilon)y_m$, where $\epsilon > 0$ is a hyper-parameter with usually a small value. Then, an extra off-the-shelf inverse dynamic model (Agrawal et al., 2016), acting as the *controller*, is employed to compute the action. See Appendix B for detailed descriptions. AIGB avoids TD learning and generally outperforms offline RL methods (Guo et al., 2024).

However, the modeling approach in AIGB modeling does not explicitly align with the performance optimization objective of the auto-bidding problem. As a result, AIGB primarily relies on imitating trajectories from the offline dataset, lacking the ability to actively explore higher-quality trajectories and improve its generation quality from the performance feedback. This limitation imposes a fundamental performance ceiling. Particularly, as generative models (e.g., diffusion models) are prone to overfitting with limited data (Ajay et al., 2022), their generalization ability beyond the offline dataset can be restricted, especially when there is a lack of high-quality trajectories in the offline dataset.

3 METHOD

Enabling AIGB to explore higher-quality trajectories actively beyond the offline dataset can enhance its performance and generalization ability. To this end, we propose **AIGB-Pearl** (**P**lanning with **E**valu**A**tor via **RL**) that constructs a *trajectory evaluator* (referred to as the evaluator for simplicity)

[1]Note that, as in real-world advertising systems, the bidding process will automatically suspend once the advertiser's budget runs out, and thereby, any action sequence will not violate the budget constraint. Hence, the trajectory quality can be directly defined as the cumulative reward of the trajectory.

to integrate RL methods into AIGB's planner training. Specifically, the evaluator learns a *score* $\hat{y}_\phi(\tau)$ to estimate the trajectory quality $y(\tau)$ via supervised learning based on the offline dataset $\mathcal{D}$, i.e., $\min_\phi \mathbb{E}_{\tau\sim\mathcal{D}}[(\hat{y}_\phi(\tau) - y(\tau))^2]$, where ϕ denotes the evaluator parameter. Then, with ϕ fixed, the planner tries to maximize the score of its generation through iterative interactions with the evaluator, as shown in Fig. 1. Formally, this can be formulated as:

$$\max_\theta \; L(\theta) \triangleq \mathbb{E}_{\tau\sim p_\theta(\tau|y^*)}[\hat{y}_\phi(\tau)], \tag{4}$$

where the condition is fixed to y^* in both training and inference stages to ensure consistency.

As can be seen, the effectiveness of AIGB-Pearl fundamentally hinges on the evaluator's reliability. However, given the offline nature of the considered auto-bidding problem, evaluator training is confined to the fixed dataset $\mathcal{D}$. Directly solving Eq. 4 can induce the infamous OOD problem due to the evaluator's generalization limits, potentially degrading the planner's true performance. Importantly, there is a notable lack of theoretical approaches to this OOD problem.

To address this challenge, we examine the theoretical bounds on the evaluator's bias. Then, guided by this analysis, we propose a KL-Lipschitz-constrained score maximization objective for the planner in Section 3.1 to ensure reliable utilization of the evaluator. Notably, this objective is theoretically justified by a sub-optimality bound established in Section 3.1.1. Finally, a practical algorithm is presented in Section 3.2, where a synchronous coupling method is employed to realize the planner's Lipschitz constraint.

3.1 KL-Lipschitz-constrained score maximization

This section focuses on the reliable exploitation of the evaluator-guided score maximization.

Our basic idea is to optimize θ within a domain where the gap between the planner's score $L(\theta)$ and its true performance $J(\theta) \triangleq \mathbb{E}_{\tau\sim p_\theta(\tau|y^*)}[y(\tau)]$ is bounded by a small certifiable upper bound. This ensures the score maximization occurs only in regions where the evaluator is reliable.

$$|J(\theta) - L(\theta)| = |\mathbb{E}_{\tau\sim p_\theta(\tau|y^*)}[y(\tau)] - \mathbb{E}_{\tau\sim p_\theta(\tau|y^*)}[\hat{y}_\phi(\tau)]|. \tag{5}$$

In the following, we investigate this gap. Specifically, we find that the trajectory quality $y(\tau)$ is a Lipschitz continuous function as stated in Theorem 1, and the proof is given in Appendix C.1.

Theorem 1 (Lipschitz Continuous of $y(\tau)$.)**.** *The trajectory quality $y(\tau)$ is $\sqrt{T}R_m$-Lipschitz continuous with respect to the Frobenius norm.*

Motivated by Theorem 1, we enforce a $\sqrt{T}R_m$-Lipschitz regularity on the evaluator's training to inherit the Lipschitz continuity of the true trajectory quality $y(\tau)$ (as described in Section 3.2.1). Since the Lipschitz constant of the trained evaluator may not be exactly $\sqrt{T}R_m$, we denote its value as $k\sqrt{T}R_m$, where $k \geq 0$ reflects the violation degree. Note that a tighter satisfaction of the Lipschitz constraint by the evaluator results in k closer to 1.

Equipped with Theorem 1 and the Lipschitz property of the evaluator, we derive the following upper bound on the performance gap between $J(\theta)$ and $L(\theta)$, and the proof is given in Appendix C.2.

Theorem 2 (Evaluator Bias in Planning Performance Bound)**.** *Let the upper bound of the evaluator's bias on its training dataset $\mathcal{D}$ be $\delta_D > 0$, i.e., $\mathbb{E}_{\tau\sim D}|y(\tau) - \hat{y}_\phi(\tau)| \leq \delta_D$, and let the Lipschitz constant of $\hat{y}_\phi(\tau)$ be $k\sqrt{T}R_m$. The gap between the planner's score $L(\theta)$ and its true performance $J(\theta)$ can be bounded by:*

$$|J(\theta) - L(\theta)| \leq \delta_D + (1+k)\sqrt{T}R_m \mathbb{E}_{y\sim p_D(y)}\Big[\underbrace{W_1(p_\theta(\tau|y^*), p_\theta(\tau|y))}_{\text{Lipschitz sensitivity to } y} + \underbrace{W_1(p_\theta(\tau|y), p_D(\tau|y))}_{\text{imitation error on } \mathcal{D}}\Big],$$

where W_1 denotes the 1-Wasserstein distance.

Note that δ_D could be regulated to a small value via supervised training of the evaluator, and k depends on the Lipschitz property of the resulting evaluator [2]. Consequently, bounding the evaluator bias in the planner's performance requires constraining the following two factors:

[2]Note that k cannot approach zero without compromising δ_D, as excessively small k prevents the evaluator from fitting the offline dataset $\mathcal{D}$.

- the planner's Lipschitz sensitivity to condition y (the first Wasserstein term)
- the planner's imitation error on the offline dataset (the second Wasserstein term).

Formally, we establish that the expectation of the first Wasserstein term can be bounded by the Lipschitz constant $\text{Lip}_{W_1}(p_\theta(\tau|y))$ of the planner with respect to the condition y measured by W_1:

$$\mathbb{E}_{y\sim p_D(y)}[W_1(p_\theta(\tau|y^*), p_\theta(\tau|y))] \leq (1+\epsilon) y_m \text{Lip}_{W_1}(p_\theta(\tau|y)). \tag{6}$$

The proof is given in Appendix C.3. Therefore, we constrain the planner's Lipschitz constant to a positive value $L_p > 0$ to ensure the boundedness of the first Wasserstein term, where L_p is a hyperparameter whose lower bound is provided later in Eq. 10.

Moreover, we establish that a constrained KL divergence $\mathbb{E}_{y\sim p_D(y)}[D_{\text{KL}}(p_D(\tau|y)\|p_\theta(\tau|y))] \leq \delta_K$ could bound the expectation of the second Wasserstein distance term as follows, where $\delta_K > 0$ is a hyperparameter and can be set to a small value, close to zero. See Appendix C.4 for the proof. Note that the KL divergence constraint here inherently makes the planner perform conditional behavior cloning on the offline dataset $\mathcal{D}$ (Guo et al., 2025).

$$\mathbb{E}_{y\sim p_D(y)}[W_1(p_\theta(\tau|y), p_D(\tau|y))] \leq \sqrt{\delta_K}. \tag{7}$$

Collectively, to effectively perform score maximization with a small and certifiable evaluator bias, we enforce Lipschitz continuity of the planner with respect to the condition y while preserving its behavior cloning fidelity to the offline dataset $\mathcal{D}$. Formally, Eq. 4 is transformed to:

$$\max_\theta \quad L(\theta) \qquad \text{(Score Maximization)} \tag{8}$$

$$\text{s.t.} \quad \mathbb{E}_{y\sim p_D(y)}[D_{\text{KL}}(p_D(\tau|y)\|p_\theta(\tau|y))] \leq \delta_K \quad \text{(KL Constraint)} \tag{8a}$$

$$\text{Lip}_{W_1}(p_\theta(\tau|y)) \leq L_p \qquad \text{(Lipschitz Constraint)} \tag{8b}$$

Eq. 8 forms the score maximization objective in AIGB-Pearl.

Remark 1. *Intuitively, the KL and Lipschitz constraints jointly ensure the planner's generation under condition y^* remains within a certified neighborhood of the offline dataset $\mathcal{D}$. Meanwhile, the evaluator trained on $\mathcal{D}$ maintains high accuracy within this $\mathcal{D}$-proximal region, and the Lipschitz regularization on the evaluator bounds its sensitivity to input perturbations, preventing drastic value fluctuations in OOD regions and promoting more reliable extrapolation. Therefore, Eq. 8 with a Lipschitz-regularized evaluator can realize reliable evaluator-guided score maximization.*

3.1.1 SUB-OPTIMALITY GAP BOUND

This section focuses on presenting and analyzing the sub-optimality bound of the solution to the proposed Eq. 8. Specifically, denote the solution to the true performance $J(\theta)$ as $\theta^* \triangleq \arg\max_\theta J(\theta)$, and denote the solution to the proposed Eq. 8 as $\hat{\theta}$. The following theorem gives the sub-optimality bound of the planner's performance, and the proof is given in Appendix C.5.

Theorem 3 (Sub-optimality Gap Bound)**.** *Let $\delta_M \triangleq \mathbb{E}_{y\sim p_D(y)}[D_{KL}(p_D(\tau|y)\|p_{\theta^*}(\tau|y))]$ be the expected distance between the optimal trajectory distribution and the trajectory distribution of the offline dataset $\mathcal{D}$. The true performance gap between the optimal parameter θ^* and the solution $\hat{\theta}$ to Eq. 8 is bounded by:*

$$J(\theta^*) - J(\hat{\theta}) \leq 2\delta_D + (1+2k)\sqrt{T}R_m\left[\sqrt{\delta_M} + \sqrt{\delta_K} + (1+\epsilon)y_m L_p\right]. \tag{9}$$

Theoretical Result Analysis. In Theorem 3, the constants $\delta_M, R_m, T, \epsilon, y_m$ characterize domain-specific properties of the auto-bidding task and the offline dataset $\mathcal{D}$. Nonetheless, a lower training error δ_D of the evaluator and a closer k to 1 correspond to a smaller sub-optimality gap. Note that k cannot be smaller than 1 without compromising δ_D, as excessively small k prevents the evaluator from fitting the offline dataset $\mathcal{D}$.

Moreover, in Theorem 3, a lower behavior cloning error δ_K and a lower Lipschitz constant L_p of the planner lead to a smaller sub-optimality gap. However, an excessively small L_p prevents the planner from behavior cloning the offline dataset $\mathcal{D}$ (as required by the KL constraint), resulting in a large

δ_K. Actually, a theoretical lower bound for L_p is given by the Lipschitz constant of the conditional trajectory distribution of the offline dataset $p_D(\tau|y)$:

$$L_p \geq \sup_{y_1 \neq y_2} \frac{W_1(p_D(\tau|y_1), p_D(\tau|y_2))}{|y_1 - y_2|}. \tag{10}$$

where $y_1, y_2 \in \mathcal{D}$. Consequently, we leverage this lower bound of L_p in AIGB-Pearl.

3.2 Practical Algorithm Design

This section focuses on the practical algorithm implementation of Eq. 8. Section 3.2.1 first presents our reliability-enhanced evaluator architecture, followed by the synchronous-coupling-based Lipschitz planner design in Section 3.2.2.

3.2.1 Lipschitz Trajectory Evaluator

As shown in Fig. 1, the evaluator processes the trajectory τ to predict a score $\hat{y}_\phi(\tau)$ for quality estimation. The evaluator is trained via supervised learning using the offline dataset $\mathcal{D}$. Besides, to satisfy $\sqrt{T}R_m$-Lipschitz constraint requirement according to Theorem 2, we add Lipschitz regularization term to the training loss of the evaluator, which can be expressed as:

$$l_e(\phi) = \underbrace{\mathbb{E}_{\tau\sim\mathcal{D}}\left[(\hat{y}_\phi(\tau) - y(\tau))^2\right]}_{\text{fitting the ground truth}} + \beta_1 \underbrace{\mathbb{E}_{\tau_1,\tau_2}\left[|\hat{y}_\phi(\tau_1) - \hat{y}_\phi(\tau_2)| - \sqrt{T}R_m \left\|\tau_1 - \tau_2\right\|_F\right]_+}_{\text{Lipschitz penalty}}, \tag{11}$$

where $\beta_1 > 0$ is a hyper-parameter, $[\cdot]_+ \triangleq \max\{0, \cdot\}$. Moreover, to further enhance the accuracy of the evaluator, we devise two specific techniques, including the LLM Embedding enhancement and pair-wise learning, whose details are given in Appendix D.1.

3.2.2 Lipschitz Planner With Synchronous Coupling

As shown in Fig. 1, the planner is implemented by a Causal Transformer (Chen et al., 2021b) that generates trajectories in an auto-regressive manner. Specifically, the model takes the condition y and history states $s_{1:t}$ as input tokens, and predicts the next state as a Gaussian distribution, $p_\theta(s_{t+1}|s_{1:t}, y) = \mathcal{N}(\mu_\theta(s_{1:t}, y, t), \sigma^2_\theta(s_{1:t}, y, t))$, where μ_θ denotes the mean and $\sigma_\theta > 0$ the standard deviation. During the auto-regressive generation process, each output state is sampled from the Gaussian distribution using the reparameterization trick, i.e., $s_{t+1} = \mu_\theta(s_{1:t}, y, t) + \sigma_\theta(s_{1:t}, y, t) \cdot \eta_t$, where $\eta_t \sim \mathcal{N}(0, I)$ [3].

Regularized Planner Training Loss. To perform the score maximization in Eq. 8, we involve two regularization terms in the planner's training loss $l_p(\theta)$, including a conditional behavior cloning loss, corresponding to the KL constraint Eq. 8a, and a Lipschitz penalty loss, corresponding to the Lipschitz constraint Eq. 8b, i.e.,

$$\begin{aligned} l_p(\theta) = &- \underbrace{\mathbb{E}_{\tau\sim p_\theta(\tau|y^*)}[\hat{y}_\phi(\tau)]}_{\text{planner score } L(\theta)} - \beta_2 \underbrace{\mathbb{E}_{(\tau,y)\sim p_D}[\log p_\theta(\tau|y)]}_{\text{conditional behavior clone}} \\ &+ \beta_3 \underbrace{\mathbb{E}_{y_1,y_2\in\mathcal{D}\cup\{y^*\}}\left[W_1(p_\theta(\tau|y_1), p_\theta(\tau|y_2)) - L_p|y_1 - y_2|\right]_+}_{\text{Lipschitz penalty, where } W_1(p_\theta(\tau|y_1), p_\theta(\tau|y_2)) \text{ is replaced by } \hat{W}_1(y_1, y_2; \theta)}, \end{aligned} \tag{12}$$

where $\beta_2, \beta_3 > 0$ are two hyper-parameters. With prior RL works (Sutton et al., 1999), we can derive the closed-form expression of planner's score gradient $\nabla_\theta L(\theta)$ as shown in Appendix C.6. The core of the planner loss lies in the computation of $W_1(p_\theta(\tau|y_1), p_\theta(\tau|y_2))$.

Wasserstein Upper Bound as Surrogate. Accurate computation of this Wasserstein distance term is challenging, as it requires finding the optimal coupling between $p_\theta(\tau|y_1)$ and $p_\theta(\tau|y_2)$

[3] Note that the time step t and the static advertiser feature x in the state $s_t = [t, c_{t-1}, x]$ do not need to be generated. We only generate the next cost ratio c_t in practice.

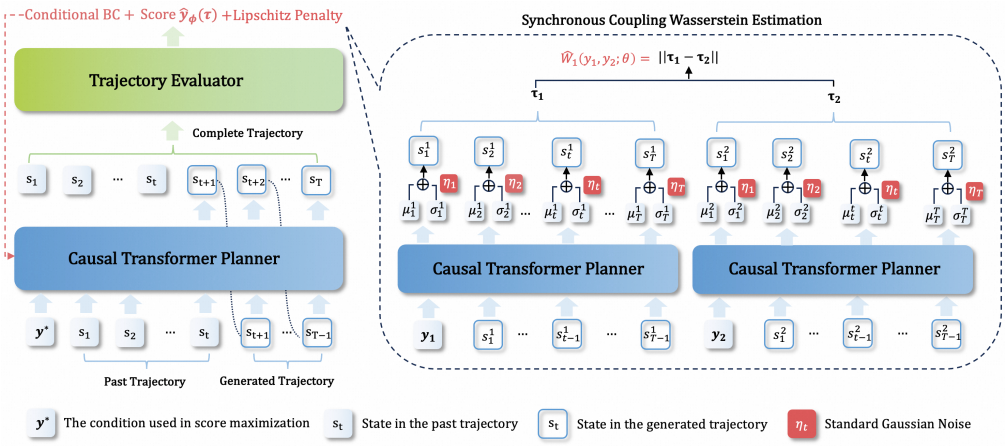

Figure 1: **AIGB-Pearl** *(Planning with EvaluAtor via RL)* constructs a trajectory evaluator to score the trajectory quality and let the planner maximize the obtained score under the KL-Lipschitz constraint through continuous interaction with the evaluator. A synchronous coupling method is used to estimate the Wasserstein term in the Lipschitz penalty.

that minimizes the expected transportation cost. Nonetheless, we can choose a certain coupling $\gamma \in \Gamma(p_\theta(\tau|y_1), p_\theta(\tau|y_2))$ to obtain an upper bound of this Wasserstein term, i.e.,

$$W_1(p_\theta(\tau|y_1), p_\theta(\tau|y_2)) \triangleq \inf_{\gamma \in \Gamma(p_\theta(\tau|y_1), p_\theta(\tau|y_2))} \mathbb{E}_{(\tau_1,\tau_2)\sim\gamma}\left[\sum_t \|s_t^1 - s_t^2\|\right]$$
$$\leq \mathbb{E}_{\eta_{1:T}\sim\mathcal{N}(0,I)}\left[\sum_t \|s_t^1 - s_t^2]\|\right] \triangleq \hat{W}_1(y_1, y_2; \theta). \tag{13}$$

where $\hat{W}_1(y_1, y_2; \theta)$ denotes the upper bound, and s_t^i is the t-th state in trajectory τ_i. It can be seen that $\hat{W}_1(y_1, y_2; \theta) \leq L_p|y_1 - y_2|$ acts as a sufficient condition to make the planner L_p-Lipschitz continuous. Thus, we replace $W_1(p_\theta(\tau|y_1), p_\theta(\tau|y_2))$ by this upper bound in the planner loss.

Synchronous Coupling Wasserstein. Instead of using random couplings, we employ a *synchronous coupling* γ_{sync} to make the upper bound tighter. Specifically, two trajectories τ_1 and τ_2—conditioned on y_1 and y_2, respectively—are generated using the same sequence of Gaussian noise $\{\eta_1, \eta_2, ...\eta_T\}$. The definition of $\hat{W}_1(y_1, y_2; \theta)$ is given in Eq. 13. Compared to random couplings, the synchronous coupling reduces spurious variance in the trajectory comparison by aligning stochasticity through shared noise, resulting in a tighter upper bound on the Wasserstein distance (Lindvall, 2002).

Moreover, if we make the predicted variance σ_θ of the planner as a fixed constant, then the expression of $\hat{W}_1(y_1, y_2; \theta)$ can be further simplified to $\hat{W}_1(y_1, y_2; \theta) = \sum_t \|\mu_\theta(s_{1:t}^1, y_1, t) - \mu_\theta(s_{1:t}^2, y_2, t)\|$. The overall AIGB-Pearl algorithm is summarized in Algorithm 1 in Appendix D due to page limits.

4 Experiments

We conduct both simulated and real-world experiments to validate the effectiveness of our approach. In the experiments, we mainly investigate the following Research Questions (**RQ**s): (1) Does enhancing AIGB with policy optimization improve overall performance, and can it generalize better to unseen data compared to existing AIGB methods? (Section 4.2) (2) How does the KL-Lipschitz constraint affect the performance of the planner? (Section 4.3) (3) Can the proposed method guarantee the Lipschitz property of the evaluator and the planner? (Section 4.4). (4) What is the evaluator's accuracy on the training data, and how well does it generalize to unseen data? (Section 4.5). The training stability of AIGB-Pearl is studied in Appendix E.5.

4.1 Experiment Setup

Experiment Environment. We conduct simulated experiments in an open-source offline advertising system with 30 advertisers of four budget levels (1.5k, 2.0k, 2.5k, and 3.0k), as in (Mou et al., 2022; Guo et al., 2024). The offline dataset comprises 5k trajectories generated by 20 advertisers.

Table 1: Overall performance (GMV) in simulated experiments with 30 advertisers. Δ indicates the relative improvement of AIGB-Pearl against the most competitive baseline (which is underlined). Note that the absolute values are normalized without specific meanings; only Δ matters.

Budget	**USCB**	**BCQ**	**CQL**	**IQL**	**Diff-QL**	**MOPO**	**DT**	**DiffBid**	**AIGB-Pearl**	**Δ**
1.5k	454.25	454.72	461.82	456.80	469.73	470.38	477.39	<u>480.76</u>	502.98	**+4.62%**
2.0k	482.67	483.50	475.78	486.56	487.91	489.27	507.30	<u>511.17</u>	521.84	**+2.09%**
2.5k	497.66	498.77	481.37	518.27	510.83	523.91	527.88	<u>531.29</u>	545.03	**+2.59%**
3.0k	500.60	501.86	491.36	549.19	552.73	549.01	550.66	<u>556.32</u>	574.17	**+3.21%**

Table 2: Overall performance in real-world A/B tests, involving 6k advertisers over 19 days.

Methods	GMV	BuyCnt	ROI	Cost	Methods	GMV	BuyCnt	ROI	Cost
DiffBid	76,390,174	650,962	5.31	14,395,290	**USCB**	52,182,805	516,994	4.92	10,598,486
AIGB-Pearl	78,676,009	665,173	5.41	14,551,054	**AIGB-Pearl**	53,973,101	520,796	5.13	10,515,772
Δ	**+3.00%**	**+2.20%**	**+1.89%**	+1.10%	**Δ**	**+3.43%**	**+0.74%**	**+4.24%**	-0.78%
Methods	GMV	BuyCnt	ROI	Cost	Methods	GMV	BuyCnt	ROI	Cost
DT	34,808,665	341,995	5.61	6,205,665	**MOPO**	51,674,071	579,332	3.08	16,771,892
AIGB-Pearl	35,957,933	344,194	5.77	6,246,512	**AIGB-Pearl**	53,292,945	591,741	3.23	16,475,670
Δ	**+3.30%**	**+0.64%**	**+0.16%**	+0.66%	**Δ**	**+3.13%**	**+2.14%**	**+4.87%**	-1.77%

Extra detailed settings of simulated experiments are given in Appendix E.1. For real-world experiments, we conduct online A/B tests on one of the world's largest E-commerce platforms, TaoBao. The offline dataset comprises 200k trajectories of 10k advertisers. See Appendix E.2 for extra detailed settings of real-world experiments. In both simulated and real-world experiments, we employ the same inverse dynamics model from Agrawal et al. (2016) as used in AIGB to serve as the controller. Moreover, the evaluator is trained on the entire offline dataset, and its generalization ability is evaluated using K-fold cross-validation with $K = 5$.

Baselines. We compare our method with the state-of-the-art AIGB methods, including **DiffBid** (Guo et al., 2024) and **DT** (Chen et al., 2021a) that learn from conditional behavior cloning of the offline dataset with a diffusion model and a Causal Transformer, respectively. We also compare our method with RL auto-bidding methods, including **USCB** (He et al., 2021) that learns the auto-bidding policy in a manually constructed advertising system with DDPG (Silver et al., 2014); and offline RL auto-bidding methods, including model-free offline RL methods **BCQ** (Fujimoto et al., 2019), **CQL** (Kumar et al., 2020), **IQL** (Kostrikov et al., 2022) and **Diff-QL** (Wang et al., 2022), and model-based offline RL method **MOPO** (Yu et al., 2020).

Performance Index. The objective in the auto-bidding problem Eq. 1, i.e., the cumulative rewards over the bidding episode, acts as the main performance index in our experiments and is referred to as the *gross merchandise volume*, **GMV**. In addition, we utilize three other metrics commonly used in the auto-bidding problem to evaluate the performance of our approach. The first metric is the total number of impressions won over the bidding episode, referred to as the **BuyCnt**. The second metric is the **Cost** over the bidding episode, and the third one is the *return on investment* **ROI** defined as the ratio between the GMV and the Cost. Note that larger values of GMV, BuyCnt, and ROI with a Cost oscillating within an acceptable tolerance ($\pm 2\%$) indicate a better performance.

4.2 Overall Performance

To answer RQ(1): Table 1 shows that our method consistently outperforms all baselines in GMV across all four budget levels in simulated experiments. In real-world experiments, the results in Table 2 show that our method also achieves superior performance in GMV, BuyCnt, and ROI, with Cost fluctuations within 2%. Notably, both simulated and real-world experiments consistently demonstrate that AIGB-Pearl achieves a **+3%** improvement in GMV over the AIGB, the state-of-the-art auto-bidding method. Since our method and DiffBid share the same controller, the performance gain stems solely from the planner. This provides strong empirical evidence that the proposed conservative RL learning for score maximization effectively enhances overall performance.

Notably, we also apply AIGB-Pearl to another important auto-bidding problem, named TargetROAS. As shown in Appendix E.3, real-world experiments show that AIGB-Pearl achieves a **+5%** improvement in GMV compared to AIGB.

Table 3: Generalization performance in real-world A/B tests with unseen advertisers against AIGB methods, involving 4k advertisers over 19 days.

Methods	GMV	BuyCnt	ROI	Cost	Methods	GMV	BuyCnt	ROI	Cost
DiffBid	67,092,973	553,020	5.39	12,444,306	**DT**	30,562,007	300,271	5.61	5,450,573
AIGB-Pearl	69,252,539	565,776	5.53	12,534,379	**AIGB-Pearl**	31,502,309	305,202	5.74	5,484,473
Δ	**+3.32%**	**+2.31%**	**+2.48%**	+0.72%	**Δ**	**+3.08%**	**+1.64%**	**+2.32%**	+0.62%

Table 4: Ablation Study. The effectiveness of the KL constraint and the Lipschitz constraint in Real-world A/B tests, involving 6k advertisers over 8 days.

AIGB-Pearl	GMV	BuyCnt	ROI	Cost	Methods	GMV	BuyCnt	ROI	Cost
w/o KL	30,906,963	292,605	4.25	7,269,018	**w/o Lipschitz**	32,284,972	268,551	5.73	5,634,304
with KL	31,243,688	292,783	4.26	7,342,485	**with Lipschitz**	32,869,329	281,979	5.79	5,678,252
Δ	**+1.09%**	**+0.06%**	**+0.08%**	+1.01%	**Δ**	**+1.81%**	**+0.50%**	**+1.05%**	+0.78%

It is worth noting that in auto-bidding systems, a $+3\%$ GMV uplift is highly significant, translating to **millions** of RMB in additional **daily** GMV on Taobao-scale advertising platforms.

Generalization Ability. We examine the performance of AIGB-Pearl on advertisers that are not used to generate trajectories in the offline dataset, comparing it against existing AIGB methods. For simplicity, we refer to these advertisers as *advertisers outside the offline dataset*. Table 3 reports the performance on 4k advertisers outside the offline dataset in real-world experiments. AIGB-Pearl demonstrates consistently better results in terms of GMV (**+3%**), BuyCnt, and ROI, while maintaining Cost fluctuations within 2% compared to baselines. These results indicate that the proposed method has better generalization ability than existing AIGB methods.

4.3 Ablation Study

To answer RQ(2): We remove the KL constraint and the Lipschitz constraint from AIGB-Pearl individually and evaluate the model's performance in each ablated variant with real-world A/B tests. The results are presented in Table 4. It can be seen that the KL constraint contributes **+1.1%** improvement in GMV, and the Lipschitz constraint provides **+1.8%** improvement in GMV, demonstrating their respective roles in enhancing AIGB-Pearl's performance.

Visualization. Three AIGB-Pearl-generated trajectory examples are presented in Fig. 2. As can be observed, the trajectories generated by AIGB-Pearl are plausible. In contrast, the ablated variant without the KL and Lipschitz constraints produces trajectories that significantly deviate from the optimal trajectory in the offline dataset [4] and exhibit clear pathological behaviors—such as excessive budget consumption, backward-trending pacing, and underutilization of available budgets (see Appendix E.4 for explanation)—which further validate the KL-Lipschitz constraint necessity.

4.4 Lipschitz Value Examination

To answer RQ(3): We report that the Lipschitz value of the trajectory quality $y(\tau)$ and the conditional trajectory distribution p_D of the offline dataset are 1.62 and 0.38, respectively. We set $L_p = 0.50$, which is near its lower bound estimation 0.38 [5]. To calculate the Lipschitz constants of the evaluator and the planner, we sample $8,000$ pairs of trajectories and compute their Lipschitz constants. The results are shown in Fig. 3 and Fig. 4. It can be observed that most sample values satisfy the Lipschitz constraint, and the Lipschitz constants of the evaluator $\hat{y}_\phi(\tau)$ and planner $p_\theta(\tau|y)$ are 2.2 and 0.56, respectively, near 1.62 and 0.50. This indicates that the Lipschitz constraints of the evaluator and the planner are successfully satisfied.

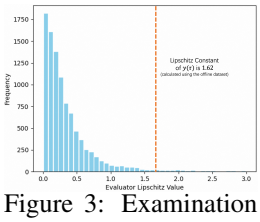

Figure 3: Examination of Evaluator Lipschitz.

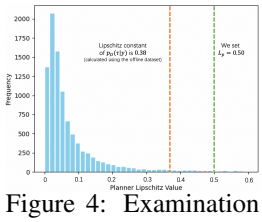

Figure 4: Examination of Planner Lipschitz.

[4] The optimal trajectory in the offline dataset can act as a reference trajectory, and empirically, optimal trajectories should not deviate from this reference largely.

[5] Considering that the lower bound estimation is computed solely in a data-driven manner and may be underestimated, we conservatively increase its value accordingly.

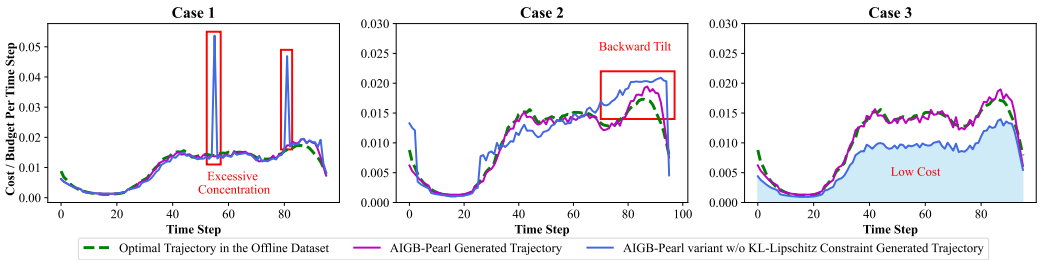

Figure 2: Trajectory Generation Visualization. Three cases are presented. Here, the AIGB-Pearl generates plausible trajectories, whereas its variant without the KL-Lipschitz constraint produces generations that significantly deviate from the reference and exhibit evident issues.

Table 5: Evaluator accuracy for simulated and real-world experiments. Results are reported for training data and OOD data evaluated using 5-fold cross-validation.

Simulated Exp	Training Data	OOD Data (Cross-Validation)	**Real-world Exp**	Training Data	OOD Data (Cross-Validation)
MAE ↓	0.6	0.7 ± 0.06	MAE ↓	1.0	1.2 ± 0.03
AUC ↑	89.9%	$85.5\% \pm 0.5\%$	AUC ↑	77.4%	$75.1\% \pm 0.2\%$

4.5 Evaluator Accuracy Examination

Accuracy Metrics. The evaluator's accuracy is assessed along two dimensions, including the *absolute accuracy* measured by mean absolute error (MAE) metrics, reflecting how close the predicted scores are to ground truth scores, and the *ranking accuracy* by AUC metric, reflecting the correctness of relative rankings between trajectory pairs. Note that the MAE of each advertiser's data sample is normalized by its budget to ensure comparability across advertisers. A lower MAE, together with a higher AUC, indicate better evaluator accuracy.

To answer RQ(4): We report the accuracy of the trained evaluator in both simulated and real-world experiments in Table 5. We evaluate the evaluator's accuracy on the training data and its generalization ability using K-fold cross-validation, where $K = 5$. To the best of our knowledge, we are the first to introduce the trajectory evaluator into the generative auto-bidding framework. The reasonableness of our evaluator is evidenced by its pairwise ranking accuracy of 86% AUC and 75% AUC on OOD trajectories in the simulated and real-world experiments, respectively, which are substantially above the 50% random chance level, despite the high complexity and dynamic nature of the bidding environment. Importantly, with the guidance of the trained evaluator, the planner outperforms state-of-the-art AIGB methods even on OOD data, as demonstrated in the Table. 3.

5 Conclusions

This paper proposes AIGB-Pearl to enhance AIGB with reward evaluation and policy optimization. By introducing a trajectory evaluator and a provable KL-Lipschitz-constrained score maximization objective, our approach ensures safe and efficient generalization beyond the offline dataset, supported by theoretical guarantees. Extensive simulated and real-world experiments validate the SOTA performance of our approach.

References

Pulkit Agrawal, Ashvin V Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. *Advances in neural information processing systems*, 29, 2016.

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*, 2022.

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua B. Tenenbaum, Tommi S. Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision making? In *The Eleventh International Conference on Learning Representations*, 2023.

Santiago Balseiro, Yuan Deng, Jieming Mao, Vahab Mirrokni, and Song Zuo. Robust auction design in the auto-bidding world. *Advances in Neural Information Processing Systems*, 34:17777–17788, 2021a.

Santiago R Balseiro, Yuan Deng, Jieming Mao, Vahab S Mirrokni, and Song Zuo. The landscape of auto-bidding auctions: Value versus utility maximization. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pp. 132–133, 2021b.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021a.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021b.

Yuan Deng, Jieming Mao, Vahab Mirrokni, and Song Zuo. Towards efficient auctions in an auto-bidding world. In *Proceedings of the Web Conference 2021*, pp. 3965–3973, 2021.

Yuan Deng, Negin Golrezaei, Patrick Jaillet, Jason Cheuk Nam Liang, and Vahab Mirrokni. Multi-channel autobidding with budget and roi constraints. In *International Conference on Machine Learning*, pp. 7617–7644. PMLR, 2023.

Zhijian Duan, Yusen Huo, Tianyu Wang, Zhilin Zhang, Yeshu Li, Chuan Yu, Jian Xu, Bo Zheng, and Xiaotie Deng. An adaptable budget planner for enhancing budget-constrained auto-bidding in online advertising. *arXiv preprint arXiv:2502.05187*, 2025.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pp. 2052–2062. PMLR, 2019.

Djordje Gligorijevic, Tian Zhou, Bharatbhushan Shetty, Brendan Kitts, Shengjun Pan, Junwei Pan, and Aaron Flores. Bid shading in the brave new world of first-price auctions. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 2453–2460, 2020.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Ziyu Guan, Hongchang Wu, Qingyu Cao, Hao Liu, Wei Zhao, Sheng Li, Cai Xu, Guang Qiu, Jian Xu, and Bo Zheng. Multi-agent cooperative bidding games for multi-objective optimization in e-commercial sponsored search. *arXiv preprint arXiv:2106.04075*, 2021.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Jiayan Guo, Yusen Huo, Zhilin Zhang, Tianyu Wang, Chuan Yu, Jian Xu, Bo Zheng, and Yan Zhang. Generative auto-bidding via conditional diffusion modeling. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, pp. 5038–5049, 2024.

Yue He, Xiujun Chen, Di Wu, Junwei Pan, Qing Tan, Chuan Yu, Jian Xu, and Xiaoqiang Zhu. A unified solution to constrained bidding in online display advertising. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, pp. 2993–3001, 2021.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Junqi Jin, Chengru Song, Han Li, Kun Gai, Jun Wang, and Weinan Zhang. Real-time bidding with multi-agent reinforcement learning in display advertising. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 2193–2201, 2018.

Bingyi Kang, Xiao Ma, Chao Du, Tianyu Pang, and Shuicheng Yan. Efficient diffusion policies for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36:67195–67212, 2023.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. *Advances in neural information processing systems*, 33: 21810–21823, 2020.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2022.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33:1179–1191, 2020.

Yinchuan Li, Xinyu Shao, Jianping Zhang, Haozhi Wang, Leo Maxime Brunswic, Kaiwen Zhou, Jiqian Dong, Kaiyang Guo, Xiu Li, Zhitang Chen, Jun Wang, and Jianye Hao. Generative models in decision making: A survey. *arXiv preprint arXiv:2502.17100*, 2025.

Torgny Lindvall. *Lectures on the coupling method*. Courier Corporation, 2002.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Zhiyu Mou, Yusen Huo, Rongquan Bai, Mingzhou Xie, Chuan Yu, Jian Xu, and Bo Zheng. Sustainable online reinforcement learning for auto-bidding. *Advances in Neural Information Processing Systems*, 35:2651–2663, 2022.

Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.

Ling Pan, Nikolay Malkin, Dinghuai Zhang, and Yoshua Bengio. Better training of gflownets with local credit and incomplete trajectories. In *International Conference on Machine Learning*, pp. 26878–26890. PMLR, 2023.

Zhiyong Peng, Yadong Liu, and Zongtan Zhou. Deadly triad matters for offline reinforcement learning. *Knowledge-Based Systems*, 284:111341, 2024.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241. Springer, 2015.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International conference on machine learning*, pp. 387–395. Pmlr, 2014.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015.

Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

Alexandre B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer Publishing Company, Incorporated, 1st edition, 2008. ISBN 0387790519.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Cédric Villani. *Topics in optimal transportation*, volume 58. American Mathematical Soc., 2021.

Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2008.

Hao Wang, Bo Tang, Chi Harold Liu, Shangqin Mao, Jiahong Zhou, Zipeng Dai, Yaqi Sun, Qianlong Xie, Xingxing Wang, and Dong Wang. Hibid: A cross-channel constrained bidding system with budget allocation by hierarchical offline deep reinforcement learning. *IEEE Transactions on Computers*, 73(3):815–828, 2023.

Weixun Wang, Shaopan Xiong, Gengru Chen, Wei Gao, Sheng Guo, Yancheng He, Ju Huang, Jiaheng Liu, Zhendong Li, Xiaoyang Li, et al. Reinforcement learning optimization for large-scale learning: An efficient and user-friendly scaling library. *arXiv preprint arXiv:2506.06122*, 2025.

Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. *arXiv preprint arXiv:2208.06193*, 2022.

Chao Wen, Miao Xu, Zhilin Zhang, Zhenzhe Zheng, Yuhui Wang, Xiangyu Liu, Yu Rong, Dong Xie, Xiaoyang Tan, Chuan Yu, et al. A cooperative-competitive multi-agent framework for auto-bidding in online advertising. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pp. 1129–1139, 2022.

Wush Chi-Hsuan Wu, Mi-Yen Yeh, and Ming-Syan Chen. Predicting winning price in real time bidding with censored data. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1305–1314, 2015.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33:14129–14142, 2020.

Zhengbang Zhu, Hanye Zhao, Haoran He, Yichao Zhong, Shenyu Zhang, Haoquan Guo, Tingting Chen, and Weinan Zhang. Diffusion models for reinforcement learning: A survey. *arXiv preprint arXiv:2311.01223*, 2023.

A Related Works

A.1 RL-based Auto-bidding Methods

Auto-bidding plays a critical role in online advertising by automatically placing bids, allowing advertisers to participate efficiently in real-time auctions (Balseiro et al., 2021a; Deng et al., 2021; Balseiro et al., 2021b). The auto-bidding problem can be modeled as a Markov Decision Process and addressed using reinforcement learning techniques. USCB (He et al., 2021) proposes a unified solution to the constrained bidding problem, employing an RL method, DDPG (Silver et al., 2014), to dynamically adjust parameters to an optimal bidding strategy. Mou et al. (2022) design a sustainable online reinforcement learning framework that iteratively alternates between online explorations and offline training, alleviating the sim2rel problem. A few studies explore multi-agent RL for auto-bidding (Jin et al., 2018; Guan et al., 2021; Wen et al., 2022), while several focus on budget allocation and bidding strategies in multi-channel scenarios using RL-based approaches (Wang et al., 2023; Deng et al., 2023; Duan et al., 2025). Importantly, offline RL methods such as BCQ (Fujimoto et al., 2019), CQL (Kumar et al., 2020), IQL (Kostrikov et al., 2022), and MOPO (Yu et al., 2020) have demonstrated significant potential in this domain. These methods allow policy learning from pre-collected datasets without requiring online interaction. Moreover, offline RL, such as Diffusion-QL (Wang et al., 2022), adopts generative models as the policy model architecture for better expressive capacity.

However, RL-based methods often suffer from training instability caused by bootstrapping and alternating training paradigms between critics and actors. Training instability typically deteriorates policy performance (Sutton et al., 1998). Moreover, training stability is even more critical in auto-bidding considering two domain-specific challenges: the absence of an accurate offline policy evaluation method and the high cost of online policy examination in a real-world advertising system (Mou et al., 2022). Therefore, stable convergence to a well-performed policy is essential to ensure deployment reliability and system safety.

A.2 Generative Auto-bidding Methods

Generative models exhibit strong capabilities in capturing and replicating the underlying data distributions across a wide range of fields (Kingma & Welling, 2022; Goodfellow et al., 2020; Pan et al., 2023; Sohl-Dickstein et al., 2015; Ho et al., 2020; Vaswani et al., 2017). They can be effectively incorporated into decision-making systems by generating complete trajectories to guide agents toward high reward behaviors (Zhu et al., 2023; Kang et al., 2023; Li et al., 2025). In particular, Decision Transformer (DT) (Chen et al., 2021a) reframes RL as a conditional sequence modeling problem and leverages transformer architectures to generate actions conditioned on desired returns, historical states, and actions. AIGB (Guo et al., 2024) extends the generative perspective to the auto-bidding domain by formulating auto-bidding as a conditional generative modeling problem. DiffBid generates a state trajectory based on the desired return utilizing a conditional diffusion model, and then generates actions aligned with the optimized trajectory. These methods achieve superior performance in auto-bidding and offer distinct advantages over traditional RL methods. They do not rely on the bootstrapping mechanism commonly used in RL, thereby avoiding the instability caused by the deadly triad. Even so, these generative auto-bidding methods still encounter a performance bottleneck due to their neglect of fine-grained generation quality evaluation and inability to explore beyond static datasets. In contrast, our method facilitates both reward evaluation and policy search through a learned trajectory evaluator.

B AIGB Method Details

AIGB models the sequential decision-making problem through conditional diffusion modeling, enabling effective trajectory generation for auto-bidding scenarios. Specifically, AIGB utilizes the denoising diffusion probabilistic model (DDPM) (Ho et al., 2020) for generation. The forward and reverse processes are modeled as:

$$q(\tau_{k+1}|\tau_k), \quad p_\theta(\tau_k|\tau_{k+1}, y(\tau)), \tag{14}$$

respectively, where q represents the forward noising process while p_θ the reverse denoising process.

Forward Process. In the forward process, the noise is gradually added to the latent variable by a Markov chain with pre-defined variance schedule β_k:

$$q(\tau_k|\tau_{k-1}) = \mathcal{N}(\tau_k; \sqrt{1-\beta_k}\tau_{k-1}, \beta_k I) \tag{15}$$

where $k \in [K]$ refers to the diffusion step, $\tau_k \triangleq [s_1, s_2, \cdots, s_T]_k$ represents the latent variable in the k-th diffusion step, and τ_0 is the original trajectory. A notable property of the forward process is that τ_k at an arbitrary time-step k can be sampled in closed form as:

$$q(\tau_k|\tau_0) = \mathcal{N}(\tau_k; \sqrt{\bar{\alpha}_k}\tau_0, (1-\bar{\alpha}_k)I), \tag{16}$$

where $\alpha_k = 1 - \beta_k$ and $\bar{\alpha}_k = \prod_{i=1}^{k} \alpha_k$. When $k \to \infty$, τ_k approaches a standard Gaussian distribution. In particular, AIGB employs a cosine noise schedule (Nichol & Dhariwal, 2021) to control the schedule β_k.

Reverse Process. In the reverse process, diffusion models aim to remove the added noise on τ_K and recursively recover τ_0. This process is governed by the conditional model $p_\theta(\tau_{k-1}|\tau_k, y(\tau))$, which is parameterized through a noise prediction model $\epsilon_\theta(\tau_k, y(\tau), k)$. AIGB adopts a temporal U-Net (Ronneberger et al., 2015) for the noise prediction model, a common choice in diffusion-based decision-making methods (Ajay et al., 2022).

B.1 Training Stage

The training of the diffusion model is typically formulated as minimizing the mean squared error between the predicted noise ϵ_θ and the true noise applied during the forward diffusion process. Specifically, during each iteration, we randomly sample a trajectory from the offline dataset $\mathcal{D}$ and pick a time step $t \in [T]$. We recursively add the Gaussian noise ϵ to the states in τ with time steps bigger than t and predict the added noises with $\epsilon_\theta(\tau_k, y(\tau), k)$, where the states between 0 and t in τ_k are set to real history states $s_1, s_2, \cdots, s_t$. In addition to this standard objective, AIGB also incorporates a supervised loss that measures the discrepancy between the true actions and the actions predicted by an inverse dynamics model $\hat{f}_\phi(s_t, \hat{s}_{t+1})$. Overall, the complete training objective of AIGB can be expressed as:

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{k,\tau\in\mathcal{D}}[||\epsilon - \epsilon_\theta(\tau_k, y(\tau), k)||^2] + \mathbb{E}_{(s_t, a_t, \hat{s}_{t+1})\in\mathcal{D}}[||a_t - \hat{f}_\phi(s_t, \hat{s}_{t+1})||^2]. \tag{17}$$

During training, the condition $y(\tau)$ is randomly dropped to enhance model robustness. This technique ensures that both the unconditional model $\epsilon_\theta(\tau_k, k)$ and the conditional model $\epsilon_\theta(\tau_k, y(\tau), k)$ are effectively trained together.

B.2 Inference Stage

Starting with Gaussian noise, trajectories are iteratively generated through a series of denoising steps. Specifically, AIGB uses a classifier-free guidance strategy (Ho & Salimans, 2021) to guide the generation of bidding and extract high-likelihood trajectories in the dataset. During generation, AIGB combines conditional and unconditional score estimates linearly:

$$\hat{\epsilon}_k := \epsilon_\theta(\tau_k, k) + \omega\left(\epsilon_\theta\left(\tau_k, y(\tau), k\right) - \epsilon_\theta\left(\tau_k, k\right)\right), \tag{18}$$

where ω is the guidance scale that controls the influence of the condition $y(\tau)$. This formulation effectively steers the trajectory generation towards regions of the data distribution most consistent with the given condition. The predicted state at each step is sampled from $p_\theta(\tau_{k-1}|\tau_k, y(\tau))$:

$$\tau_{k-1} \sim \mathcal{N}\left(\tau_{k-1}|\mu_\theta\left(\tau_k, y(\tau), k\right), \Sigma_\theta\left(\tau_k, k\right)\right), \tag{19}$$

with mean and variance defined as $\mu_\theta(\tau_k, y(\tau), k) = \frac{1}{\sqrt{\alpha_k}}(\tau_k - \frac{\beta_k}{\sqrt{1-\bar{\alpha}_k}}\hat{\epsilon}_k)$ and $\Sigma_\theta(\cdot) = \beta_k$. Note that the initial noisy trajectory $\tau'_K \sim \mathcal{N}(0, I)$ is assigned with history states $s_{1:t}$ for the first t states to ensure history consistency. This is consistent with the training process. By recursively applying the reverse diffusion process using:

$$\tau'_{k-1} = \mu_\theta(\tau'_k, y(\tau), k) + \sqrt{\beta_k}z, \tag{20}$$

where $z \sim \mathcal{N}(0, I)$, we obtain the final denoised trajectory τ'_0, from which the next state $\hat{s}_{t+1}$ is derived. Then the action is generated through an inverse dynamics $\hat{a}_t = \hat{f}_\phi(s_t, \hat{s}_{t+1})$.

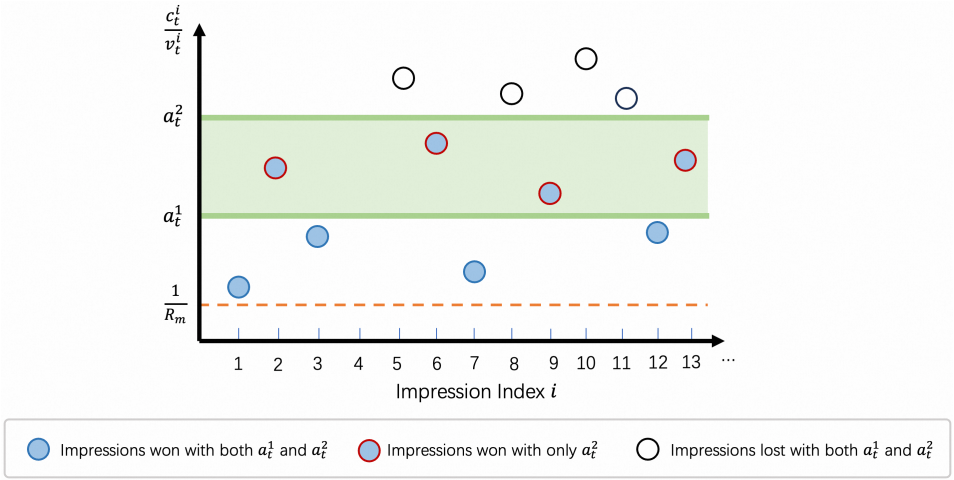

Figure 5: The impression opportunities within time step t and $t+1$, where p_t^i/v_t^i is the $1/\text{ROI}$ of impression i. Without loss of generality, consider two actions $a_{1,t}$ and $a_{2,t}$, and let $a_{2,t} \geq a_{1,t}$. The impressions within the shadow area are the impressions won by action $a_{2,t}$ but lost by action $a_{1,t}$.

C Theoretical Proofs

C.1 Proof of Theorem 1.

Theorem 1 (Lipschitz Continuous of $y(\tau)$.)**.** *The trajectory quality $y(\tau)$ is $\sqrt{T}R_m$-Lipschitz continuous with respect to the Frobenius norm.*

Proof. Recall from Section 2.1 that the cost c_t and reward r_t under action a_t between time step t and $t+1$ can be written as:

$$c_t = \sum_i \mathbb{1}\left\{a_t \geq \frac{p_t^i}{v_t^i}\right\} p_t^i \quad \text{and} \quad r_t = \sum_i \mathbb{1}\left\{a_t \geq \frac{p_t^i}{v_t^i}\right\} v_t^i, \tag{21}$$

where p_t^i and v_t^i denote the market price and the value of the i-th impression between time step t and $t+1$. Accordingly, the cost ratio $\bar{c}_t$ and the normalized reward $\bar{r}_t$ can be written as:

$$\bar{c}_t = \frac{1}{B}\sum_i \mathbb{1}\left\{a_t \geq \frac{p_t^i}{v_t^i}\right\} p_t^i \quad \text{and} \quad \bar{r}_t = \frac{1}{B}\sum_i \mathbb{1}\left\{a_t \geq \frac{p_t^i}{v_t^i}\right\} v_t^i, \tag{22}$$

Consider two different trajectories τ_1 and τ_2 with actions, cost ratios and normalized rewards sequences $\{a_{1,t}, \bar{c}_{1,t}, \bar{r}_{1,t}\}_{t=1}^T$ and $\{a_{2,t}, \bar{c}_{2,t}, \bar{r}_{2,t}\}_{t=1}^T$, respectively. The trajectory quality gap between τ_1 and τ_2 holds that:

$$|y(\tau_1) - y(\tau_2)| = \left|\sum_t \bar{r}_{1,t} - \sum_t \bar{r}_{2,t}\right| \leq \sum_t |\bar{r}_{1,t} - \bar{r}_{2,t}|. \tag{23}$$

Consider the reward gap between time step t and $t+1$, as shown in Fig.5. Without loss of generality, let $a_{2,t} \geq a_{1,t}$. We have:

$$\begin{aligned}
|\bar{r}_{1,t} - \bar{r}_{2,t}| &= \frac{1}{B}\sum_i \left[\mathbb{1}\left\{a_{2,t} \geq \frac{p_t^i}{v_t^i}\right\} - \mathbb{1}\left\{a_{1,t} \geq \frac{p_t^i}{v_t^i}\right\}\right] v_t^i \\
&= \frac{1}{B}\sum_i \mathbb{1}\left\{a_{2,t} \geq \frac{p_t^i}{v_t^i} \geq a_{1,t}\right\} v_t^i \\
&= \frac{1}{B}\sum_i \mathbb{1}\left\{a_{2,t} \geq \frac{p_t^i}{v_t^i} \geq a_{1,t}\right\} \frac{v_t^i}{p_t^i} p_t^i \\
&\leq \frac{R_m}{B}\sum_i \mathbb{1}\left\{a_{2,t} \geq \frac{p_t^i}{v_t^i} \geq a_{1,t}\right\} p_t^i.
\end{aligned} \tag{24}$$

Note that the cost ratio gap between time step t and $t+1$ can be written as:

$$|\bar{c}_{1,t} - \bar{c}_{2,t}| = \frac{1}{B}\sum_i \left[\mathbb{1}\left\{a_{2,t} \geq \frac{p_t^i}{v_t^i}\right\} - \mathbb{1}\left\{a_{1,t} \geq \frac{p_t^i}{v_t^i}\right\}\right] p_t^i = \frac{1}{B}\sum_i \mathbb{1}\left\{a_{2,t} \geq \frac{p_t^i}{v_t^i} \geq a_{1,t}\right\} p_t^i. \tag{25}$$

Therefore, combining Eq. 24 and Eq. 25, we have:

$$|\bar{r}_{1,t} - \bar{r}_{2,t}| \leq R_m |\bar{c}_{1,t} - \bar{c}_{2,t}|. \tag{26}$$

We examine the Frobenius norm of the gap between τ_1 and τ_2:

$$\begin{aligned}
\|\tau_1 - \tau_2\|_F &= \left\| \begin{bmatrix} 1 & \bar{c}_{1,0} & x \\ 2 & \bar{c}_{1,1} & x \\ \vdots & \vdots & \vdots \\ T & c_{1,T-1} & x \end{bmatrix} - \begin{bmatrix} 1 & \bar{c}_{2,0} & x \\ 2 & \bar{c}_{2,1} & x \\ \vdots & \vdots & \vdots \\ T & \bar{c}_{2,T-1} & x \end{bmatrix} \right\|_F \\
&= \sqrt{\sum_t (\bar{c}_{1,t} - \bar{c}_{2,t})^2} \\
&\geq \frac{1}{\sqrt{T}} \sum_t |\bar{c}_{1,t} - \bar{c}_{2,t}| \qquad \text{(Cauchy-Schwarz Inequality)}
\end{aligned} \tag{27}$$

Combining Eq. 23, Eq. 26 and Eq. 27, we can obtain that:

$$\begin{aligned}
|y(\tau_1) - y(\tau_2)| &\leq \sum_t |\bar{r}_{1,t} - \bar{r}_{2,t}| \\
&\leq R_m \sum_t |\bar{c}_{1,t} - \bar{c}_{2,t}| \\
&\leq \sqrt{T} R_m \frac{1}{\sqrt{T}} \sum_t |\bar{c}_{1,t} - \bar{c}_{2,t}| \\
&\leq \sqrt{T} R_m \|\tau_1 - \tau_2\|_F .
\end{aligned} \tag{28}$$

This concludes the proof. □

C.2 Proof of Theorem 2

Here, we list two lemmas used in the proof of Theorem 2.

Lemma 1 (Additivity of the Lipschitz)**.** *Let $f_1(x)$ and $f_2(x)$ be two Lipschitz continuous functions with Lipschitz constants $L_1 > 0$ and $L_2 > 0$, respectively. Then $|f_1(x) + f_2(x)|$ is also a Lipschitz continuous function, with Lipschitz constant at most $L_1 + L_2$.*

Proof. Recall the Reverse Triangle Inequality states that $\forall a, b$, we have $||a| - |b|| \leq |a - b|$. Then, $\forall x, y$, we have:

$$\begin{aligned}
||f_1(x) + f_2(x)| - |f_1(y) + f_2(y)|| &\leq |f_1(x) + f_2(x) - f_1(y) - f_2(y)| \\
&\leq |f_1(x) - f_1(y)| + |f_2(x) - f_2(y)| \\
&\leq (L_1 + L_2)|x - y|.
\end{aligned} \tag{29}$$

This concludes the proof. □

Lemma 2 (Kantorovich-Rubinstein Duality Theorem (Villani, 2021))**.** *Let (X, d) be a metric space, and let p and q be two probability distributions on X. Let $f : X \to \mathbb{R}$ be an L-Lipschitz function, and $W_1(p, q)$ denotes the 1-Wasserstein distance between p and q. Then we have:*

$$|\mathbb{E}_{x\sim p} f(x) - \mathbb{E}_{x\sim q} f(x)| \leq L \cdot W_1(p, q). \tag{30}$$

We next give the proof of Theorem 2.

Theorem 2 (Evaluator Bias in Planning Performance Bound)**.** *Let the upper bound of the evaluator's bias on its training dataset $\mathcal{D}$ be $\delta_D > 0$. The gap between the planner's score $L(\theta)$ and its true performance $J(\theta)$ can be bounded by:*

$$|J(\theta) - L(\theta)| \leq \delta_D + (1+k)\sqrt{T}R_m \mathbb{E}_{y\sim p_D(y)}\Bigg[\underbrace{W_1(p_\theta(\tau|y^*), p_\theta(\tau|y))}_{\text{Lipschitz sensitivity to } y} + \underbrace{W_1(p_\theta(\tau|y), p_D(\tau|y))}_{\text{imitation error on } \mathcal{D}}\Bigg],$$

where W_1 denotes the 1-Wasserstein distance.

Proof. The evaluator bias in the planner's performance can be written as:

$$|J(\theta) - L(\theta)| = |\mathbb{E}_{\tau\sim p_\theta(\tau|y^*)}[y(\tau) - \hat{y}_\phi(\tau)]| \leq \mathbb{E}_{\tau\sim p_\theta(\tau|y^*)} \underbrace{|y(\tau) - \hat{y}_\phi(\tau)|}_{\triangleq f(\tau)} \tag{31}$$

Let $f(\tau) \triangleq |y(\tau) - \hat{y}_\phi(\tau)|$ be the evaluator bias in trajectory τ. From Theorem 1 and Lemma 1, we know that $f(\tau)$ is a $(1+k)\sqrt{T}R_m$-Lipschitz continuous function. Then, we have:

$$\begin{aligned}
|J(\theta) - L(\theta)| &\leq \mathbb{E}_{\tau\sim p_\theta(\tau|y^*)} f(\tau) \\
&= \mathbb{E}_{y\sim p_D(y)}\Big[\mathbb{E}_{\tau\sim p_\theta(\tau|y^*)} f(\tau) - \mathbb{E}_{\tau\sim p_D(\tau|y)} f(\tau) + \mathbb{E}_{\tau\sim p_D(\tau|y)} f(\tau)\Big] \\
&= \underbrace{\mathbb{E}_{y\sim p_D(y)}\mathbb{E}_{\tau\sim p_D(\tau|y)} f(\tau)}_{\leq \delta_D} + \mathbb{E}_{y\sim p_D(y)} \underbrace{\Big[\mathbb{E}_{\tau\sim p_\theta(\tau|y^*)} f(\tau) - \mathbb{E}_{\tau\sim p_D(\tau|y)} f(\tau)\Big]}_{\leq (1+k)\sqrt{T}R_m W_1(p_\theta(\tau|y^*), p_D(\tau|y)),\ \text{(Lemma 2)}} \\
&\leq \delta_D + (1+k)\sqrt{T}R_m \mathbb{E}_{y\sim p_D(y)}[W_1(p_\theta(\tau|y^*), p_D(\tau|y))].
\end{aligned} \tag{32}$$

To further enhance physical intuition and simplify constraint computation, we decompose the Wasserstein distance term $W_1(p_\theta(\tau|y^*), p_D(\tau|y))$ via the triangle inequality:

$$W_1(p_\theta(\tau|y^*), p_D(\tau|y)) \leq W_1(p_\theta(\tau|y^*), p_\theta(\tau|y)) + W_1(p_\theta(\tau|y), p_D(\tau|y)). \tag{33}$$

Therefore, we have:

$$|J(\theta) - L(\theta)| \leq \delta_D + (1+k)\sqrt{T}R_m \mathbb{E}_{y\sim p_D(y)}\Big[W_1(p_\theta(\tau|y^*), p_\theta(\tau|y)) + W_1(p_\theta(\tau|y), p_D(\tau|y))\Big], \tag{34}$$

This concludes the proof. □

C.3 Proof of Eq. 6

We give the proof of Eq. 6 as follows. Denote $\text{Lip}_{W_1}(p_\theta(\tau|y))$ as the planner's Lipschitz constant with respect to y regarding the Wasserstein distance W_1, we have:

$$\begin{aligned}
\mathbb{E}_{y\sim p_D(y)}[W_1(p_\theta(\tau|y^*), p_\theta(\tau|y))] &\leq \text{Lip}_{W_1}(p_\theta(\tau|y))\mathbb{E}_{y\sim p_D(y)}[((1+\epsilon)y_m - y)] \\
&= \text{Lip}_{W_1}(p_\theta(\tau|y)) \int_0^{y_m} p_D(y)[(1+\epsilon)y_m - y]\mathrm{d}y \\
&\leq \text{Lip}_{W_1}(p_\theta(\tau|y)) \int_0^{y_m} p_D(y)[(1+\epsilon)y_m]\mathrm{d}y \\
&= (1+\epsilon)y_m \text{Lip}_{W_1}(p_\theta(\tau|y)),
\end{aligned} \tag{35}$$

where we leverage the non-negativity property of the condition $y \geq 0, \forall y \in \mathcal{D}$. This completes the proof.

C.4 Proof of Eq. 7

Lemma 3 (Pinsker's Inequality (Tsybakov, 2008))**.** *Let P and Q be two probability measures defined on the same measurable space, and assume that P is absolutely continuous with respect to Q, i.e., $P \ll Q$. Then the total variation distance between P and Q is bounded above by the KL divergence from P to Q as follows:*

$$\|P - Q\|_{TV} \leq \sqrt{\frac{1}{2} D_{KL}(P\|Q)}. \tag{36}$$

Lemma 4 (Wasserstein–Total Variation Inequality on Bounded Metric Spaces (Villani et al., 2008)). *Let $(\mathcal{Z}, d)$ be a metric space with diameter diam$(\mathcal{Z}) \triangleq \sup_{z_1,z_2\in\mathcal{Z}} d(z_1, z_2)$. Let P and Q be two probability measures on $\mathcal{Z}$. Then the 1-Wasserstein distance between P and Q satisfies:*

$$W_1(P, Q) \leq \textit{diam}(\mathcal{Z})\|P - Q\|_{TV}. \tag{37}$$

We give the proof of Eq. 7 as follows. Equipped with the above two lemmas, we have:

$$\begin{aligned} W_1(p_\theta(\tau|y), p_D(\tau|y)) &\leq \text{diam}(\mathcal{T})\|p_\theta(\tau|y) - p_D(\tau|y)\|_{\text{TV}} \\ &\leq \text{diam}(\mathcal{T})\sqrt{\frac{1}{2}D_{KL}(p_D(\tau|y)\|p_\theta(\tau|y))}, \end{aligned} \tag{38}$$

where $\mathcal{T}$ is the trajectory space. Note that due to the budget constraint $\sum_t c_t \leq B$ [6], we have the sum of the cost ratio satisfies $\sum_t \bar{c}_t \leq 1$. The trajectory space can be expressed as:

$$\mathcal{T} = \left\{ \left[[1, \bar{c}_0, x], [2, \bar{c}_1, x], \cdots, [T, \bar{c}_{T-1}, x] \right] \middle| \bar{c}_t \geq 0, \forall t, \text{and} \sum_t \bar{c}_t \leq 1 \right\} \tag{39}$$

We next prove that the diameter of the trajectory space, diam$(\mathcal{T})$, can be bounded by a constant. Specifically, the diameter only depends on the largest possible distance between the cost ratio sequences in two trajectories since:

$$\begin{aligned} \text{diam}(\mathcal{T}) &= \sup_{\tau_1,\tau_2\in\mathcal{T}} \|\tau_1 - \tau_2\|_F \\ &= \sup_{\tau_1,\tau_2\in\mathcal{T}} \left\| \begin{bmatrix} 1 & \bar{c}_{1,0} & x \\ 2 & \bar{c}_{1,1} & x \\ \vdots & \vdots & \vdots \\ T & c_{1,T-1} & x \end{bmatrix} - \begin{bmatrix} 1 & \bar{c}_{2,0} & x \\ 2 & \bar{c}_{2,1} & x \\ \vdots & \vdots & \vdots \\ T & \bar{c}_{2,T-1} & x \end{bmatrix} \right\|_F \\ &= \sup_{\tau_1,\tau_2\in\mathcal{T}} \sqrt{\sum_t (\bar{c}_{1,t} - \bar{c}_{2,t})^2}. \end{aligned} \tag{40}$$

For convenience, we let $\mathbf{c}_i \triangleq [\bar{c}_{i,0}, \bar{c}_{i,2}, \cdots, \bar{c}_{i,T-1}], i \in \{1, 2\}$. Then, the key part in the above result, $\sum_t (\bar{c}_{1,t} - \bar{c}_{2,t})^2$, can be written as:

$$\begin{aligned} \sum_t (\bar{c}_{1,t} - \bar{c}_{2,t})^2 &= \sum_t (\bar{c}_{1,t}^2 - 2\bar{c}_{1,t}\bar{c}_{2,t} + \bar{c}_{2,t}^2) \\ &= \|\mathbf{c}_1\|_2^2 + \|\mathbf{c}_2\|_2^2 - 2\langle \mathbf{c}_1, \mathbf{c}_2 \rangle \\ &\leq \|\mathbf{c}_1\|_2^2 + \|\mathbf{c}_2\|_2^2, \end{aligned} \tag{41}$$

where $\langle \mathbf{c}_1, \mathbf{c}_2 \rangle \geq 0$. As $\bar{c}_{i,t} \geq 0$ and $\sum_t \bar{c}_{i,t} \leq 1$, we have $0 \leq \bar{c}_{i,t} \leq 1$. Therefore, it holds that:

$$\|\mathbf{c}_i\|_2^2 = \sum_t \bar{c}_{i,t}^2 \leq \sum_t \bar{c}_{i,t} \leq 1. \tag{42}$$

Combining Eq. 41 and Eq. 42, we have:

$$\sqrt{\sum_t (\bar{c}_{1,t} - \bar{c}_{2,t})^2} \leq \sqrt{\|\mathbf{c}_1\|_2^2 + \|\mathbf{c}_2\|_2^2} \leq \sqrt{2}. \tag{43}$$

Therefore, we have diam$(\mathcal{T}) = \sqrt{2}$. According to Eq. 38, we have:

$$W_1(p_\theta(\tau|y), p_D(\tau|y)) \leq \sqrt{D_{KL}(p_D(\tau|y)\|p_\theta(\tau|y))}. \tag{44}$$

Recall that we impose the KL-constraint as:

$$\mathbb{E}_{y\sim p_D(y)}[D_{\text{KL}}(p_D(\tau|y)\|p_\theta(\tau|y))] \leq \delta_K, \tag{45}$$

[6] As explained in Footnote 1, the budget constraint is guaranteed to be satisfied in real-world advertising systems thanks to an automatic suspension mechanism that halts bidding once the budget is exhausted.

Taking the expectation over $y \sim p_D(y)$ on both sides of Eq. 44, we have:

$$\begin{aligned}\mathbb{E}_{y\sim p_D(y)}[W_1(p_\theta(\tau|y), p_D(\tau|y))] &\leq \mathbb{E}_{y\sim p_D(y)}\left[\sqrt{D_{KL}(p_D(\tau|y)\|p_\theta(\tau|y))}\right] \\ &\leq \sqrt{\mathbb{E}_{y\sim p_D(y)}[D_{KL}(p_D(\tau|y)\|p_\theta(\tau|y))]} \quad \text{(Jensen Inequality)} \\ &\leq \sqrt{\delta_K} \quad \text{(KL constraint Eq.45).}\end{aligned} \tag{46}$$

This completes the proof.

C.5 Proof of Theorem 3

Theorem 3 (Sub-optimality Gap Bound)**.** *Let $\delta_M \triangleq \mathbb{E}_{y\sim p_D(y)}[D_{KL}(p_D(\tau|y)\|p_{\theta^*}(\tau|y))]$ be the expected distance between the optimal trajectory distribution and the trajectory distribution of the offline dataset $\mathcal{D}$. The true performance gap between the optimal parameter θ^* and the solution $\hat{\theta}$ to Eq. 8 is bounded by:*

$$J(\theta^*) - J(\hat{\theta}) \leq 2\delta_D + (1+2k)\sqrt{T}R_m\left[\sqrt{\delta_M} + \sqrt{\delta_K} + (1+\epsilon)y_m L_p\right]. \tag{47}$$

Proof. The sub-optimality gap can be expressed as follows:

$$\begin{aligned}J(\theta^*) - J(\hat{\theta}) &= \big(J(\theta^*) - L(\theta^*)\big) + \big(L(\theta^*) - L(\hat{\theta})\big) + \big(L(\hat{\theta}) - J(\hat{\theta})\big) \\ &\leq \underbrace{|J(\theta^*) - L(\theta^*)|}_{\text{evaluator bias in } p_{\theta^*}} + \underbrace{|L(\theta^*) - L(\hat{\theta})|}_{\text{score gap}} + \underbrace{|L(\hat{\theta}) - J(\hat{\theta})|}_{\text{evaluator bias in } p_{\hat{\theta}}}.\end{aligned} \tag{48}$$

We examine the above three terms accordingly.

(1) Evaluator Bias in p_{θ^*}. Denote the evaluator bias on trajectory τ as $f(\tau) \triangleq |y(\tau) - \hat{y}_\phi(\tau)|$. Following the derivation process in Eq. 32, we have:

$$\begin{aligned}|J(\theta^*) - L(\theta^*)| &\leq \mathbb{E}_{y\sim p_D(y)}\mathbb{E}_{\tau\sim p_D(\tau|y)}f(\tau) + \mathbb{E}_{y\sim p_D(y)}\left[\mathbb{E}_{\tau\sim p_{\theta^*}(\tau|y^*)}f(\tau) - \mathbb{E}_{\tau\sim p_D(\tau|y)}f(\tau)\right] \\ &\leq \delta_D + (1+k)\sqrt{T}R_m\mathbb{E}_{y\sim p_D(y)}[W_1(p_{\theta^*}(\tau|y^*), p_D(\tau|y))],\end{aligned} \tag{49}$$

where $W_1(p_{\theta^*}(\tau|y^*), p_D(\tau|y))$ denotes the probability distribution distance between the optimal planner and the offline dataset. Based on the derivation in Appendix C.4, we have:

$$\mathbb{E}_{y\sim p_D(y)}[W_1(p_{\theta^*}(\tau|y^*), p_D(\tau|y))] \leq \sqrt{\mathbb{E}_{y\sim p_D(y)}[D_{KL}(p_D(\tau|y)\|p_{\theta^*}(\tau|y))]} \tag{50}$$

Let $\delta_M \triangleq \mathbb{E}_{y\sim p_D(y)}[D_{KL}(p_D(\tau|y)\|p_{\theta^*}(\tau|y))]$ be the distance between the optimal trajectory distribution and the offline dataset trajectory distribution. We have:

$$|J(\theta^*) - L(\theta^*)| \leq \delta_D + (1+k)\sqrt{T}R_m\sqrt{\delta_M}. \tag{51}$$

(2) Score Gap. Recall that the trained evaluator $\hat{y}_\phi(\tau)$ is a $k\sqrt{T}R_m$-Lipschitz continuous function with the Lipschitz constraint design. With Lemma 2, we have:

$$\begin{aligned}|L(\theta^*) - L(\hat{\theta})| &= |\mathbb{E}_{\tau\sim p_{\theta^*}(\tau|y^*)}\hat{y}_\phi(\tau) - \mathbb{E}_{\tau\sim p_{\hat{\theta}}(\tau|y^*)}\hat{y}_\phi(\tau)| \\ &\leq k\sqrt{T}R_m W_1(p_{\theta^*}(\tau|y^*), p_{\hat{\theta}}(\tau|y^*)) \\ &\leq k\sqrt{T}R_m\mathbb{E}_{y\sim p_D(y)}\left[W_1(p_{\theta^*}(\tau|y^*), p_D(\tau|y)) + W_1(p_D(\tau|y), p_{\hat{\theta}}(\tau|y^*))\right] \\ &\leq k\sqrt{T}R_m\left[\sqrt{\delta_M} + (1+\epsilon)y_m L_p + \sqrt{\delta_K}\right],\end{aligned} \tag{52}$$

where we leverage the fact that $\hat{\theta}$ is a solution to Eq. 8 which satisfies the KL and Lipschitz constraint, and we leverage the results in Eq. 6, Eq. 7 and Eq. 50.

(3) Evaluator Bias in $p_{\hat{\theta}}$. Since $\hat{\theta}$ satisfies the KL and Lipschitz constraint in Eq. 8, we can use the results in Theorem 2, Eq. 6 and Eq. 7 to obtain:

$$|J(\hat{\theta}) - L(\hat{\theta})| \leq \delta_D + (1+k)\sqrt{T}R_m[(1+\epsilon)y_m L_p + \sqrt{\delta_K}], \tag{53}$$

Overall, combining the results in Eq. 51, Eq. 52, and Eq. 53, we have:

$$J(\theta^*) - J(\hat{\theta}) \leq 2\delta_D + (1+2k)\sqrt{T}R_m\left[\sqrt{\delta_M} + \sqrt{\delta_K} + (1+\epsilon)y_m L_p\right]. \tag{54}$$

This concludes the proof.

□

C.6 Proof of Score Gradient

The probability of the trajectory generated by the Causal Transformer can be decomposed into:

$$p_\theta(s_{1:T}|y) = \Pi_t p_\theta(s_t|s_{1:t-1}, y). \tag{55}$$

Then, we have:

$$\begin{aligned}
\nabla_\theta L(\theta) &= \nabla_\theta \int_\tau p_\theta(\tau|y^*)\hat{y}_\phi(\tau)\mathrm{d}\tau \\
&= \nabla_\theta \int_{s_1,\cdots,s_T} p_\theta(s_{1:T}, \tau|y^*)\hat{y}_\phi(\tau)\mathrm{d}s_1\cdots\mathrm{d}s_T \\
&= \int_{s_{1:T}} p_\theta(s_{1:T}|y^*)\frac{\nabla_\theta p_\theta(s_{1:T}|y^*)}{p_\theta(s_{1:T}|y^*)}\hat{y}_\phi(\tau)\mathrm{d}s_1\cdots\mathrm{d}s_T \\
&= \mathbb{E}_{s_{1:T}\sim p_\theta(s_{1:T}|y^*)}\left[\nabla_\theta \log p_\theta(s_{1:T}|y^*)\hat{y}_\phi(\tau)\right] \\
&= \mathbb{E}_{s_{1:T}\sim p_\theta(s_{1:T}|y^*)}\left[\nabla_\theta \log \Pi_t p_\theta(s_t|s_{1:t-1}, y^*)\hat{y}_\phi(\tau)\right] \\
&= \mathbb{E}_{s_{1:T}\sim p_\theta(s_{1:T}|y^*)}\left[\sum_t \nabla_\theta \log p_\theta(s_t|s_{1:t-1}, y^*)\hat{y}_\phi(\tau)\right].
\end{aligned} \tag{56}$$

D AIGB-Pearl Algorithm Summary

Algorithm 1 summarizes the training process of AIGB-Pearl. The development of AIGB-Pearl is supported by ROLL (Wang et al., 2025).

D.1 Additional Designs For Evaluator Accuracy Enhancement

To further enhance the reliability of the trajectory evaluator, we design two specific techniques. Specifically, as described in the following, we (i) integrate an LLM into its model architecture for better representational capacity; and (ii) mix pair-wise and point-wise losses for better score estimation accuracy.

E Additional Experiments

E.1 Simulated Experiment Settings

We include the detailed simulated experiment settings in Table 6. Specifically, we consider the bidding process in a day, where the bidding episode is divided into 96 time steps. Thus, the duration between two adjacent time steps t and $t+1$ is 15 minutes. The number of impression opportunities between time steps t and $t+1$ fluctuates from 100 to 500. The minimum and maximum budgets of advertisers are 1000 Yuan and 4000 Yuan, respectively. The upper bound of the bid price is 10 Yuan, and the values of impressions are positive.

Hardware Resource. The simulated experiments are conducted based on an NVIDIA T4 Tensor Core GPU. We use 10 CPUs and 200G memory.

Algorithm 1: AIGB-Pearl (Planning with EvaluAtor via RL)

Input : Offline dataset $\mathcal{D}$, desired condition y^*, hyper-parameters $\beta_1, \beta_2, \beta_3$.
Output : Optimized θ and ϕ
Initialization: randomly initialized planner parameter θ, trajectory evaluator parameters ϕ
// Determining the Lipschitz Value
Calculate the Lipschitz value of $y(\tau)$ and $p_D(\tau|y)$ using the offline dataset $\mathcal{D}$.
Set the Lipschitz constraint value L_e for the evaluator and L_p for the planner to be bigger than the Lipschitz value of $y(\tau)$ and $p_D(\tau|y)$, respectively.
// Training the trajectory evaluator
while *not converged* **do**
  Update ϕ by minimizing Eq. 11;
end
// Training the generative planner
Warm start with pretrained planner p_θ;
while *not converged* **do**
  Generate bidding trajectories $\tau \sim p_\theta(\tau|y^*)$;
  Score generated trajectories with frozen ϕ: $\hat{y}_\phi(\tau)$;
  Update θ by maximizing Eq. 12.
end

Table 6: Settings of the simulated experiments.

Parameters	Values
Number of advertisers	30
Time steps in an episode, T	96
Minimum number of impressions within a time step	50
Maximum number of impressions within a time step	300
Minimum budget	1000 Yuan
Maximum budget	4000 Yuan
Value of impressions	> 0
Minimum bid price, $\min\{av_i\}$	0 Yuan
Maximum bid price, $\max\{av_i\}$	10 Yuan
Maximum market price, p_M	10 Yuan

E.2 Real-world Experiment Settings

We include the detailed real-world experiment settings in Table 7. Specifically, we consider the bidding process in a day, where the bidding episode is divided into 96 time steps. Thus, the duration between two adjacent time steps t and $t+1$ is 15 minutes. The number of impression opportunities between time steps t and $t+1$ fluctuates from 100 to 2,500. The minimum and maximum budgets of advertisers are 50 Yuan and 10,000 Yuan, respectively. The upper bound of the bid price is 25 Yuan, and the values of impressions are positive.

Hardware Resource. The training process in the real-world experiments is conducted using 10 NVIDIA T4 Tensor Core GPUs in a distributed manner. For each distributional worker, we use 10 CPUs and 200 GB of memory.

E.3 Real-world Experiments on TargetROAS Bidding Problem

In addition to the budget-constrained auto-bidding problem, we also apply the proposed AIGB-Pearl algorithm to a more challenging type of auto-bidding problem, named TargetROAS, with an extra ROI (Return on Investment) constraint. We evaluate our method in a real-world experiment on TaoBao involving 300k advertisers over 22 days. The offline dataset comprises 16 million trajectories of 800k advertisers. The results are given in Table 8. AIGB-Pearl achieves a **+5.1%** improvement in GMV compared to the SOTA auto-bidding method, DiffBid, demonstrating its effectiveness in managing more complex and realistic constraints.

Table 7: Settings of the real-world experiments.

Parameters	Values
Number of advertisers	6,000
Time steps in an episode, T	96
Minimum number of impressions within a time step	100
Maximum number of impressions within a time step	2,500
Minimum budget	50 Yuan
Maximum budget	10,000 Yuan
Value of impressions	> 0
Minimum bid price, $\min\{av_i\}$	0 Yuan
Maximum bid price, $\max\{av_i\}$	25 Yuan
Maximum market price, p_M	25 Yuan

Table 8: Overall performance of TargetROAS in real-world A/B test, involving 300k advertisers over 22 days.

Methods	GMV	BuyCnt	ROI	Cost
DiffBid	779,642,891	11,519,082	4.68	166,544,918
AIGB-Pearl (ours)	819,550,812	11,886,501	4.70	174,234,673
Δ	**+5.1%**	**+3.2%**	+0.5%	**+4.6%**

E.4 Pathological Trajectory Behavior Explanation

In industrial practice, stable and effective metrics have been developed to evaluate pathological behaviors. For the case of the budget-constrained auto-bidding problem with bidding cycles structured as 24-hour episodes (T = 96 time steps), the following three key metrics are commonly used to identify pathological behaviors:

- **Excessive budget consumption**: there exists a time step t such that the cost between time step t and $t+1$ exceeds 10% of the budget B;
- **Forward- (or Backward-) trending pacing**: the cost between time step 1 and 24 (or between time step $T-24$ and T) exceeds 40% (or 40%) of the budget B;
- **Underutilization of available budgets**: the total cost over the bidding episode is lower than 90% of the budget B.

E.5 Training Stability

We present additional comparisons between the training curves of the offline RL with bootstrapping and those of AIGB-Pearl in Fig. 6, Fig. 7, Fig. 8, and Fig. 9 concerning:

- **Cumulative Rewards**: the main performance index of the considered auto-bidding problem;
- **Online Rate**: the ratio between the bidding period before the budget runs out and the total bidding period. A larger Online Rate indicates a better performance.
- **Bad Case Rate**: the ratio between the number of "bad" trajectories and the total number of generated trajectories. A lower Bad Case Rate indicates a better performance.
- **Cost Rate**: the ratio between the cost and the budget. A larger cost rate indicates a better performance.

We can see that the offline RL method tends to suffer from significant instability throughout training, showing high variance across different seeds. In contrast, AIGB-Pearl achieves much smoother and more consistent learning progress, demonstrating the improved training stability.

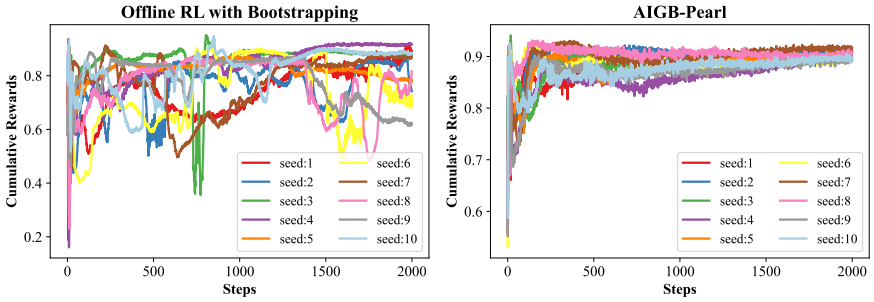

Figure 6: Learning curves of cumulative rewards between offline RL with bootstrapping method and AIGB-Peral under 10 seeds.

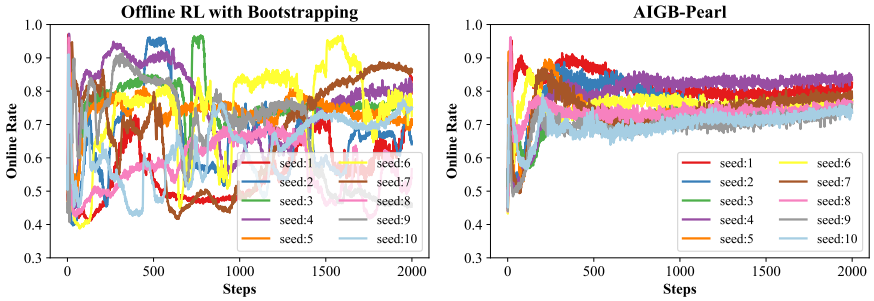

Figure 7: Learning curves of online rate between offline RL with bootstrapping method and AIGB-Peral under 10 seeds.

E.6 Empirical performance with general offline data distributions

Note that in many real-world auto-bidding systems, including the one considered in the paper, due to operational safety constraints, the online-deployed bidding policy is typically a single fixed model, and the offline dataset is collected exclusively from this single policy over multiple days, where an advertiser contributes a single trajectory per day. For example, in the considered auto-bidding system, the online-deployed baseline policy is a conditional generative model that generates identical trajectory plans for a given advertiser under identical conditions each day. The variation across different trajectories of the same advertiser in the offline dataset is solely due to stochastic environmental factors (e.g., traffic fluctuations). Since these exogenous perturbations are typically the sum of many independent impression-level sources of noise, the resulting trajectory deviations can be reasonably approximated as a Gaussian distribution.

To demonstrate the broad applicability of the proposed algorithm, we evaluate its performance in settings where multiple policies are used for data collection. Specifically, we collect trajectories in the simulated environment using nine distinct bidding policies, thereby constructing an offline dataset with a multi-modal distribution that violates the Gaussian distribution. The empirical results are presented in Table 9.

Table 9: Empirical performance with multiple data-collection policies.

Methods	GMV	ROI	Cost
DiffBid	548.5	5.00	109.4
AIGB-Pearl (ours)	575.7	5.04	114.6
Δ	**+4.9%**	**+0.7%**	**+4.2%**

We observe that AIGB-Pearl still outperforms AIGB by +4.9%, indicating that its performance is robust to the specific distribution of the offline dataset.

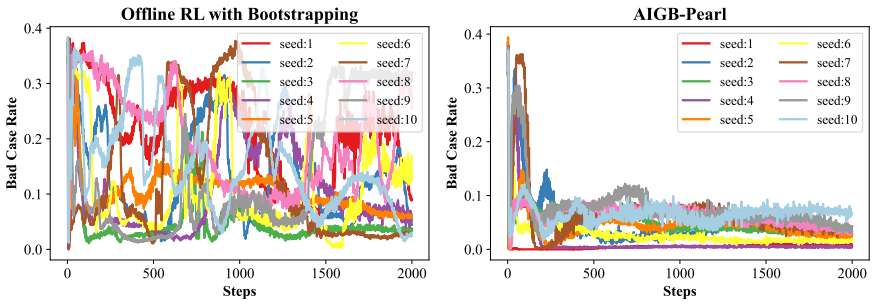

Figure 8: Learning curves of bad case rate between offline RL with bootstrapping method and AIGB-Peral under 10 seeds.

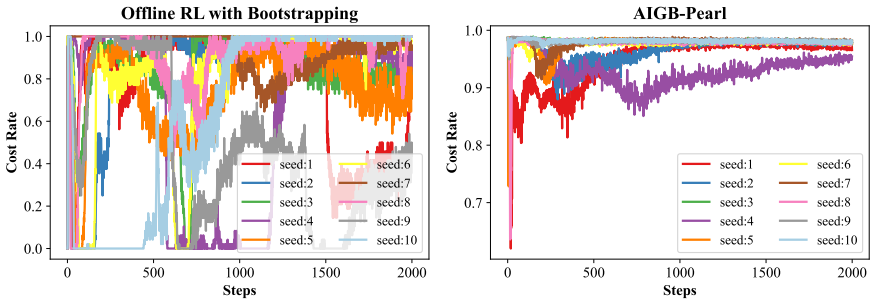

Figure 9: Learning curves of cost rate between offline RL with bootstrapping method and AIGB-Pearl under 10 seeds.

F Extending AIGB-Pearl to first-price auctions

We note that the proposed method remains effective in first-price auctions with a proper adaptation. Specifically, unlike second-price auctions where the optimal bid for impression i takes the form $\text{bid}_i = av_i$, in first-price auctions, the optimal bid for impression i is given by $\text{bid}_i = \min(av_i, p_i)$, which typically involves an extra bid shading method to predict the winning price (Gligorijevic et al., 2020; Wu et al., 2015). Equipped with an off-the-shelf bid shading method (whose design is beyond the scope of this work), the auto-bidding problem in a first-price auction remains an offline sequential decision problem, i.e., making decisions over an a-sequence, to which the proposed method applies directly.

To validate the effectiveness of the proposed method under first-price auctions, we additionally conduct a simulated first-price auction experiment against the state-of-the-art AIGB method, where all methods are equipped with the same bid shading method. The results are presented in Table 10, demonstrating the effectiveness of our proposed method.

G Extending AIGB-Pearl to Online Setting Discussion

We note that AIGB-Pearl can be extended to online settings when equipped with a safe online exploration policy. Specifically, due to safety constraints, the auto-bidding policy during training cannot interact directly with the live advertising system; only safe exploration policies are permitted to collect data online (Mou et al., 2022). Consequently, an online auto-bidding framework typically involves two parts:

- a safe online exploration policy, which is a well-established component in existing work (Mou et al., 2022) and beyond the scope of this paper;
- an offline policy training method that leverages the data collected online.

Table 10: Empirical performance in the simulated experiments with a first-price auction.

Methods	GMV	ROI	Cost
DiffBid	1,546	5.13	301
AIGB-Pearl (ours)	1,611	5.18	311
Δ	**+4.2%**	**+1.0%**	**+3.3%**

AIGB-Pearl can be directly applied as the offline policy training method within the online framework without modification. In practice, due to the safety and stability concerns, many industrial auto-bidding systems adopt an offline optimization paradigm. For this practical reason, we focus on the offline setting in this work.

H LLM USAGE

The authors have used Large Language Models (LLMs) exclusively for grammar checking and lexical refinement during the writing process. No LLM-generated content, data analysis, or substantive contributions to the research methodology, results, or conclusions are involved in this work.