arXiv:2504.14587v2 [cs.LG] 25 Apr 2025

Generative Auto-Bidding with Value-Guided Explorations

Jingtong Gao
City University of Hong Kong
Hong Kong, China
jt.g@my.cityu.edu.hk

Yewen Li
Nanyang Technological University
Singapore, Singapore
yewen001@e.ntu.edu.sg

Shuai Mao
The Chinese University of Hong Kong
Hong Kong, China
smao@mae.cuhk.edu.hk

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

Nan Jiang
Kuaishou Technology
Beijing, China
jiangnan07@kuaishou.com

Yejing Wang
City University of Hong Kong
Hong Kong, China
yejing.wang@my.cityu.edu.hk

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

Fei Pan
Kuaishou Technology
Beijing, China
panfei05@kuaishou.com

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

Kun Gai
Unaffiliated
Beijing, China
gai.kun@qq.com

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

Xiangyu Zhao*
City University of Hong Kong
Hong Kong, China
xianzhao@cityu.edu.hk

Abstract

Auto-bidding, with its strong capability to optimize bidding decisions within dynamic and competitive online environments, has become a pivotal strategy for advertising platforms. Existing approaches typically employ rule-based strategies or Reinforcement Learning (RL) techniques. However, rule-based strategies lack the flexibility to adapt to time-varying market conditions, and RL-based methods struggle to capture essential historical dependencies and observations within Markov Decision Process (MDP) frameworks. Furthermore, these approaches often face challenges in ensuring strategy adaptability across diverse advertising objectives. Additionally, as offline training methods are increasingly adopted to facilitate the deployment and maintenance of stable online strategies, the issues of documented behavioral patterns and behavioral collapse resulting from training on fixed offline datasets become increasingly significant. To address these limitations, this paper introduces a novel offline **G**enerative **A**uto-bidding framework with **V**alue-Guided **E**xplorations (**GAVE**). GAVE accommodates various advertising objectives through a score-based Return-To-Go (RTG) module. Moreover, GAVE integrates an action exploration mechanism with an RTG-based evaluation method to explore novel actions while ensuring stability-preserving updates. A learnable value function is also designed to guide the direction of action exploration and mitigate Out-of-Distribution (OOD) problems. Experimental results on two offline datasets and real-world deployments demonstrate that GAVE outperforms state-of-the-art baselines in both offline evaluations and online A/B tests. By applying the core methods of this framework, we proudly secured first place in the NeurIPS 2024 competition, 'AIGB Track: Learning Auto-Bidding Agents with Generative Models' [1]. The implementation code is publicly available to facilitate reproducibility and further research [2].

CCS Concepts

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

Keywords

Auto-bidding, Generative Model, Decision Transformer

ACM Reference Format:
Jingtong Gao, Yewen Li, Shuai Mao, Peng Jiang, Nan Jiang, Yejing Wang, Qingpeng Cai, Fei Pan, Peng Jiang, Kun Gai, Bo An, and Xiangyu Zhao. 2025. Generative Auto-Bidding with Value-Guided Explorations. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25), July 13–18, 2025, Padua, Italy*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3726302.3729987

1 Introduction

Bidding remains a fundamental component of modern online advertising platforms [11, 14, 58, 62], enabling businesses to engage target audiences and increase sales. As digital advertising evolves, traditional manual bid management methods have become increasingly inadequate and cost-ineffective, failing to address the demands of today's dynamic market [18, 34]. The complexity of modern advertising systems, characterized by fluctuating market conditions and diverse user behaviors [1], requires automated solutions that can

*Corresponding authors.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGIR '25, July 13–18, 2025, Padua, Italy
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-1592-1/2025/07
https://doi.org/10.1145/3726302.3729987

[1] https://tianchi.aliyun.com/competition/entrance/532236/rankingList
[2] https://github.com/Applied-Machine-Learning-Lab/GAVE

arXiv:2504.14587v2 [cs.LG] 25 Apr 2025

价值引导探索的生成式自动出价

Jingtong Gao
City University of Hong Kong
Hong Kong, China
jt.g@my.cityu.edu.hk

Yewen Li
Nanyang Technological University
Singapore, Singapore
yewen001@e.ntu.edu.sg

Shuai Mao
The Chinese University of Hong Kong
Hong Kong, China
smao@mae.cuhk.edu.hk

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

Nan Jiang
Kuaishou Technology
Beijing, China
jiangnan07@kuaishou.com

Yejing Wang
City University of Hong Kong
Hong Kong, China
yejing.wang@my.cityu.edu.hk

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

Fei Pan
Kuaishou Technology
Beijing, China
panfei05@kuaishou.com

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

Kun Gai
Unaffiliated
Beijing, China
gai.kun@qq.com

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

Xiangyu Zhao*
City University of Hong Kong
Hong Kong, China
xianzhao@cityu.edu.hk

摘要

自动出价能够在动态且竞争激烈的在线环境中有效优化出价决策，已成为广告平台的一项关键策略。现有方法通常采用基于规则的策略或强化学习（Reinforcement Learning, RL）技术。然而，基于规则的策略缺乏适应时变市场条件的灵活性，而基于 RL 的方法难以在马尔可夫决策过程（Markov Decision Process, MDP）框架内捕捉关键的历史依赖关系和观测信息。此外，这些方法往往难以确保策略能够适应多样化的广告目标。同时，随着离线训练方法日益广泛地用于促进稳定在线策略的部署和维护，在固定离线数据集上训练所带来的局限于已记录行为模式以及行为坍塌问题愈发突出。为克服这些局限，本文提出一种新的离线框架：价值引导探索的生成式自动出价（Generative Auto-bidding with Value-Guided Explorations, GAVE）。GAVE 通过基于评分的回报到当前（Return-To-Go, RTG）模块适配不同的广告目标。此外，GAVE 将动作探索机制与基于 RTG 的评估方法相结合，在探索新动作的同时确保更新保持稳定。我们还设计了一个可学习的价值函数，用于引导动作探索的方向并缓解分布外（Out-of-Distribution, OOD）问题。在两个离线数据集及真实部署中的实验结果表明，GAVE 在离线评估和在线 A/B 测试中均优于最先进的基线方法。通过应用该框架的核心方法，我们荣获 NeurIPS 2024 竞赛"AIGB 赛道：使用生成式模型学习自动出价智能体"的第一名 [1]。实现代码已公开，以便复现和进一步研究 [2]。

CCS 概念

• 信息系统 →计算广告。

关键词

自动出价，生成式模型，决策变换器（Decision Transformer, DT）

ACM 引用格式： Jingtong Gao, Yewen Li, Shuai Mao, Peng Jiang, Nan Jiang, Yejing Wang, Qingpeng Cai, Fei Pan, Peng Jiang, Kun Gai, Bo An, and Xiangyu Zhao. 2025. 价值引导探索的生成式自动出价。载于第 48 届国际 ACM SIGIR 信息检索研究与开发会议论文集（SIGIR '25），2025 年 7 月 13 – 18 日，意大利帕多瓦。ACM，美国纽约州纽约市，11 页。https://doi.org/10.1145/3726302.3729987

1 引言

出价始终是现代在线广告平台的一项基础组成部分 [11, 14, 58, 62]，使企业能够触达目标受众并提高销售额。随着数字广告的发展，传统的人工出价管理方法日益难以满足需求，且成本效益不佳，无法应对当今动态市场的要求 [18, 34]。现代广告系统具有市场条件波动和用户行为多样化等特征 [1]，其复杂性要求采用能够

*Corresponding authors.

允许免费制作本作品全部或部分内容的数字副本或纸质副本，供个人或课堂使用，前提是这些副本不得用于营利或商业利益而制作或分发，并且在副本首页保留本声明及完整引用信息。本作品中由作者以外的其他方拥有的各组成部分，其版权必须得到尊重。允许在注明来源的情况下编写摘要。其他复制、再版、发布到服务器或重新分发至邮件列表的行为，须事先获得特别许可和 / 或支付费用。请通过 permissions@acm.org 申请许可。SIGIR '25，2025 年 7 月 13 – 18 日，意大利帕多瓦。© 2025 版权归所有者 / 作者所有。出版权已授权给 ACM。
ACM ISBN 979-8-4007-1592-1/2025/07 https://doi.org/10.1145/3726302.3729987

[1] https://tianchi.aliyun.com/competition/entrance/532236/rankingList [2] https://github.com/Applied-Machine-Learning-Lab/GAVE

adapt to these variations while aligning with advertisers' diverse objectives [4, 48]. This need is further intensified by the massive volume of ad impressions requiring real-time processing, where human intervention becomes impractical and often suboptimal for achieving advertising goals [4, 16].

To meet these requirements, existing solutions have evolved into two primary categories: predefined rule-based strategies [7, 55] and Reinforcement Learning (RL)-based methods [5, 13, 54, 59]. While rule-based strategies are computationally lightweight and straightforward to deploy, their static nature makes them ill-suited for dynamic markets and unable to accommodate advertisers' diverse demands [5, 15, 24, 30]. RL-based approaches, though employing Markov Decision Processes (MDPs) [35, 54, 60, 61] to adapt to environmental changes and obtain better performance, face a critical structural constraint: the MDP framework's state-independence assumption inherently disregards temporal dependencies and observations within bidding sequences [6, 25, 27]. This limitation obstructs the identification of evolving behavioral patterns and market fluctuations, substantially undermining RL's real-world applicability in highly volatile real-time bidding environments.

Recently, Decision Transformer (DT) [6, 10, 12] has emerged as a powerful framework, effectively capturing temporal dependencies and historical context. Therefore, applying DT to offline bidding modeling offers a promising direction for improving strategies. Specifically, by adopting an offline training paradigm, DT circumvents the risks and implementation challenges of online training, ensuring broader applicability across diverse scenarios [9, 26, 31, 49]. The generative modeling foundation of DT further enables explicit capture of temporal dependencies and historical bidding context, allowing adaptive decision-making that aligns with the dynamic nature of real-world advertising environments.

However, several critical challenges emerge when implementing a DT-based auto-bidding approach in real-world scenarios. First, practical deployment requires accommodating complex advertising objectives where evaluation metrics extend beyond elementary indicators like total clicks or conversions. These objectives typically involve sophisticated functions with different preferences on interdependent parameters—such as Cost Per Action (CPA) thresholds and Cost Per Click (CPC) ceilings [16, 27]—requiring DT modeling with adaptive optimization objectives to align with diverse operational criteria. Second, directly training DT models in offline environments may restrict to documented behavioral patterns [21, 29] and suffer from behavioral collapse [9], which requires amplified action explorations with stable updates.

To tackle these challenges, we propose a unified framework that enhances DT for offline **G**enerative **A**uto-bidding through **V**alue-guided **E**xplorations (GAVE). First, to accommodate complex advertising objectives [16, 17], we design **a score-based Return-To-Go (RTG)** module with customizable score functions, enabling adaptive modeling of various objective requirements like CPA constraints through differentiable programming. Second, **an action exploration mechanism** is proposed alongside an RTG-based evaluation method to explore and evaluate actions outside the fixed dataset while ensuring stability-preserving updates between explored and original actions. However, it is rather challenging to learn a beneficial strategy through random explorations and avoid Out-of-Distribution (OOD) risks in such a sensitive bidding environment with large action space [20, 32, 53]. Thus, we introduce **a learnable value function** [21, 46] to guide the action exploration process, directing exploration toward potentially optimal actions. This mechanism anchors explorations within plausible regions while enabling controlled extrapolation, thereby facilitating strategy improvement and further mitigating OOD issues.

Our contributions are summarized as follows:

- We introduce an innovative framework GAVE that leverages DT to optimize auto-bidding strategies, which is designed for seamless adaptability to various real-world scenarios.
- This paper presents three technical innovations: (1) A score-based RTG module with customizable functions for various advertising objectives through differentiable programming; (2) An action exploration mechanism with RTG-based evaluation to ensure stability-preserving updates; (3) A learnable value function to anchor exploration to plausible regions, thus mitigating OOD risks and enabling controlled extrapolation for strategy improvement.
- Experiments on two public datasets, along with results from online deployments, demonstrate the effectiveness of GAVE compared to various state-of-the-art offline bidding baselines. Additionally, by applying the core methods of this framework, we proudly secured first place in the NeurIPS 2024 competition, 'AIGB Track: Learning Auto-Bidding Agents with Generative Models' hosted by Alimama [1].

2 Preliminary

In this section, we first illustrate the auto-bidding problem, and then introduce the DT-based decision-making process for modeling.

2.1 Auto-Bidding Problem

Consider a sequence of I impression opportunities arriving over a discrete time period $i = 1, ..., I$. Advertisers engage in real-time competition by submitting bids $\{b_i\}_{i=1}^{I}$ for these impressions.

The auction mechanism operates under the following rules: An advertiser wins impression i if his bid b_i exceeds b_i^-, the highest competing bid from other participants. The winning advertiser then incurs a cost c_i, which is determined by the auction mechanism. Following standard industry practice [2, 8], we adopt the generalized second-price auction mechanism where the winning cost equals the second-highest bid. The advertiser's objective is to maximize the total acquired value through won impressions during the period. This optimization problem can be formally expressed as:

$$\max \sum_{i=1}^{I} x_i v_i \tag{1}$$

where $v_i \in \mathbb{R}^+$ represents the advertiser's private valuation for impression i such as conversion or click-through rate, and $x_i \in \{0, 1\}$ denotes the binary decision variable indicating auction outcome:

$$x_i = \begin{cases} 1 & \text{if } b_i > b_i^- \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

适应这些变化、同时契合广告主多样化目标的自动化解决方案 [4, 48]。大量需要实时处理的广告展示进一步强化了这一需求；在这种情形下，人工干预既不切实际，也往往无法以最优方式实现广告目标 [4, 16]。

为满足这些要求，现有解决方案已发展为两大类：预定义的基于规则的策略 [7, 55]，以及基于 RL 的方法 [5, 13, 54, 59]。尽管基于规则的策略计算开销小且易于部署，但其静态特性使其不适合动态市场，也无法满足广告主多样化的需求 [5, 15, 24, 30]。基于 RL 的方法虽然利用 MDP [35, 54, 60, 61] 适应环境变化并获得更好的性能，但面临一项关键的结构性约束：MDP 框架的状态独立性假设从根本上忽略了出价序列中的时间依赖关系和观测信息 [6, 25, 27]。这一局限阻碍了对不断演化的行为模式和市场波动的识别，严重削弱了 RL 在高度波动的实时出价环境中的实际适用性。

近年来，DT [6, 10, 12] 已成为一种强大的框架，能够有效捕捉时间依赖关系和历史上下文。因此，将 DT 应用于离线出价建模，为改进策略提供了一个前景可期的方向。具体而言，通过采用离线训练范式，DT 避开了在线训练的风险和实施挑战，从而确保其能够更广泛地应用于多样化场景 [9, 26, 31, 49]。DT 的生成式建模基础还使其能够显式捕捉时间依赖关系和历史出价上下文，实现与真实广告环境的动态特性相适应的自适应决策。

然而，在真实场景中实现基于 DT 的自动出价方法时，会出现若干关键挑战。首先，实际部署需要适配复杂的广告目标，其评估指标超越了总点击数或总转化数等基础指标。这些目标通常涉及复杂函数，对相互依赖的参数具有不同偏好，例如单次行动成本（Cost Per Action, CPA）阈值和单次点击成本（Cost Per Click, CPC）上限 [16, 27]；这要求 DT 建模采用自适应的优化目标，以契合多样化的运营标准。其次，在离线环境中直接训练 DT 模型可能使其局限于已记录的行为模式 [21, 29]，并出现行为坍塌 [9]，因此需要加强动作探索，同时保持更新稳定。

为应对这些挑战，我们提出了一个统一框架，通过价值引导探索来增强 DT，以实现离线生成式自动出价，即 GAVE。首先，为适配复杂的广告目标 [16, 17]，我们设计了一个具有可定制评分函数的基于评分的 RTG 模块，以便

通过可微分编程，对 CPA 约束等各种目标要求进行自适应建模。其次，我们提出一种动作探索机制，并结合基于回报到当前（Return-To-Go, RTG）的评估方法，探索和评估固定数据集之外的动作，同时确保在探索动作与原始动作之间进行保持稳定性的更新。然而，相当大的挑战在于通过随机探索学习有益策略，并避免在如此敏感且动作空间庞大的竞价环境中的分布外（Out-of-Distribution, OOD）风险 [20, 32, 53]。因此，我们引入一种可学习的价值函数 [21, 46]，以引导动作探索过程，使探索朝向可能最优的动作。该机制将探索锚定在合理区域内，同时允许受控外推，从而促进策略改进并进一步缓解 OOD 问题。

我们的贡献概括如下：

• 我们提出了创新框架 GAVE，利用决策变换器（Decision Transformer, DT）优化自动出价策略，其设计旨在无缝适应各种现实场景。• 本文提出了三项技术创新：(1) 基于评分的 RTG 模块，通过可微分编程为各种广告目标定制函数；(2) 结合基于 RTG 的评估的动作探索机制，以确保更新过程保持稳定；(3) 可学习的价值函数，将探索锚定在合理区域，从而缓解 OOD 风险，并通过受控外推改进策略。• 在两个公开数据集上的实验以及线上部署结果表明，与各种最先进的离线出价基线相比，GAVE 具有有效性。此外，通过应用该框架的核心方法，我们自豪地夺得了 NeurIPS 2024 竞赛的第一名，该竞赛为"AIGB 赛道：利用生成式

模型学习自动出价智能体"，由阿里妈妈主办 [1]。

2 预备知识

本节首先阐述自动出价问题，然后介绍用于建模的基于 DT 的决策过程。

2.1 自动出价问题 考虑一系列共 I次广告展示机会，它们在离散时间段 $i = 1, ..., I$内到达。广告主通过为这些广告展示提交出价 $\{b_i\}_{i=1}^{I}$，参与实时竞争。拍卖机制遵循以下规则：当广告主对广告展示 i的出价 b_i超过其他参与者的最高竞争出价 b_i^-时，该广告主赢得此次广告展示。获胜的广告主随后承担成本 c_i，其数额由拍卖机制决定。遵循行业标准惯例 [2, 8]，我们采用广义第二价格拍卖机制，其中获胜成本等于第二高的出价。广告主的目标是最大化在该时间段内通过竞得广告展示所获得的总价值。该优化问题可形式化表示为：

$$\max \sum_{i=1}^{I} x_i v_i \tag{1}$$

其中，$v_i \in \mathbb{R}^+$表示广告主对广告展示 i的私有估值，例如转化率或点击率；$x_i \in \{0, 1\}$表示指示拍卖结果的二元决策变量：

$$x_i = \begin{cases} 1 & \text{if } b_i > b_i^- \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Simultaneously, advertisers must satisfy multiple constraints to ensure efficient campaign management. The fundamental constraint is the total budget limitation:

$$\sum_{i=1}^{I} x_i c_i \leq B \tag{3}$$

where $B \in \mathbb{R}^+$ represents the advertiser's total budget. Other Key Performance Indicator (KPI) constraints, exemplified by Cost Per Acquisition (CPA), can be formulated as follows:

$$\frac{\sum_{i=1}^{I} x_i c_i}{\sum_{i=1}^{I} x_i v_i} \leq C \tag{4}$$

where $C \in \mathbb{R}^+$ denotes the maximum allowable CPA. This ratio quantifies the efficiency of advertising expenditure relative to value creation. Since most other KPI constraints can be modeled similarly, we consider only the CPA constraint for simplicity in this paper. However, unlike budget constraints, which are directly managed by the auction platform, these KPI constraints are generally not strict in practical scenarios. This is because calculating these constraints requires the advertiser's v_i for all bidding impressions, making it possible to determine the true CPA only after the entire bidding process concludes. Nevertheless, we still hope to use them as soft constraints in modeling.

Therefore, the whole bidding process could be expressed as:

$$\begin{aligned} \max_{b_1,\cdots,b_I} & \sum_i x_i v_i \\ \text{s.t.} & \sum_i x_i c_i \leq B \\ & \frac{\sum_i x_i c_i}{\sum_i x_i v_i} \leq C \end{aligned} \tag{5}$$

Solving this optimization problem presents inherent challenges stemming from both the high cardinality of impressions and the fundamental uncertainty about future auction performance. Previous research [17] reformulates this problem as a Linear Programming problem to yield a simplified optimal bidding strategy:

$$b_i^* = \lambda_0^* v_i - \Sigma_j \lambda_j^* \left(\mathbb{q}_{ij} \left(1 - \mathbb{1}_{CR_j} \right) - \mathbb{k}_j \mathbb{p}_{ij} \right) \tag{6}$$

where b_i^* denotes the theoretically optimal bid for impression i, $\mathbb{q}_{ij}$ can be any performance indicator or constant and $\mathbb{1}_{CR_j}$ is the indicator function of whether constraint j is cost-related. $\mathbb{p}_{ij}$ and $\mathbb{k}_j$ can be treated as expanded expressions of v_i and C in Equation (5) under multiple KPI consditions. This reformulation transforms the auto-bidding problem into the identification of the optimal λ_0^* and λ_j^* that satisfy all constraints. By substituting Equation (6) into Equation (5) with $j = 1$, $\mathbb{1}_{CR_j} = 1$, $\mathbb{p}_{ij} = v_i$, $\mathbb{k}_j = C$ and $\mathbb{q}_{ij}$ being any performance indicator or constant, we can obtain:

$$b_i^* = (\lambda_0^* + \lambda_1^* C) v_i = \lambda^* v_i \tag{7}$$

where $\lambda^* = \lambda_0^* + \lambda_1^* C$ serves as the unified bidding parameter. Therefore, many recent studies have sought to address the bidding problem by iteratively identifying the optimal λ^* within the bidding process [17, 27, 44]. Additionally, it is worth noting that when solving a bidding problem according to Equation (7), the first condition, i.e., s.t. $\sum_i x_i c_i \leq B$, is always satisfied. This is because the bidding platform will automatically control x_i when the advertiser's budget is insufficient to ensure that the advertiser does not owe money. However, the second condition is not always satisfied since there's a gap between our predicted λ and the optimal λ^*. A simple solution to this problem is to add a penalty term about the CPA condition to the objective function in Equation (7) in the evaluation stage for model selection [44], which will be discussed later in Section 3.2.

2.2 DT-based Auto-bidding

To solve the auto-bidding problem, existing approaches employ either rule-based policies [3, 42] or RL methods [51, 56] for optimization. However, rule-based policies often fail to adapt to the highly dynamic nature of real-world bidding environments [17], and RL approaches [47] rely on state transitions defined by $s_{t+1} = f(s_t, a_t)$, which complicates the modeling of essential temporal dependencies and historical observations inherent in auction ecosystems [57].

Recent advancements in transformer architectures [23, 28] have led to the emergence of DT [6, 52, 63], positioning them as state-of-the-art for sequential decision-making. DTs excel in capturing long-range dependencies, making them ideal for bidding environments where auction outcomes display significant temporal correlations. Building on this framework, we approach auto-bidding as a sequence modeling task [27, 44] under DT settings. The bidding period is divided into discrete time steps, with each step configured under specific environmental settings:

- **state** s_t: The state vector s_t encompasses a collection of features that characterizes the bidding conditions at timestep t. For advertising scenarios, these features could be the remaining time, unused budget, historical bidding statistics, etc.
- **action** a_t: The action a_t denotes the bidding variables that could be iteratively adjusted through the whole bidding period. In this paper, according to Equation (7), the optimal action is $a = \lambda^*$. Therefore, we denote the real action at time step t as:

$$a_t = \lambda_t \tag{8}$$

- **reward** rw_t: Suppose there are N_t candidate impressions coming between t and $t+1$. The reward rw_t could then be defined as:

$$rw_t = \sum_{n=0}^{N_t} x_{n_t} v_{n_t} \tag{9}$$

where x_{n_t} and v_{n_t} are the binary indicator and value of the n^{th} impression at time step t.

- **Return-To-Go (RTG)** r_t: The RTG value indicates the total amount of rewards to be obtained in the future time steps:

$$r_t = \sum_{t'=t}^{T} rw_{t'} \tag{10}$$

where T is the final time step.

These settings result in the following trajectory representation, which is well-suited for autoregressive training and inference:

$$\tau = (r_1, s_1, a_1, r_2, s_2, a_2, \ldots, r_T, s_T, a_T) \tag{11}$$

3 Framework

Here, we will detail the GAVE's overview and key components.

同时，广告主必须满足多项约束，以确保高效管理广告活动。最基本的约束是总预算限制：

$$\sum_{i=1}^{I} x_i c_i \leq B \tag{3}$$

其中，$B \in \mathbb{R}^+$表示广告主的总预算。其他关键绩效指标（Key Performance Indicator, KPI）约束，例如获客成本（Cost Per Acquisition, CPA），可表述如下：

$$\frac{\sum_{i=1}^{I} x_i c_i}{\sum_{i=1}^{I} x_i v_i} \leq C \tag{4}$$

其中，$C \in \mathbb{R}^+$表示允许的 CPA 上限。该比值衡量广告支出相对于价值创造的效率。由于大多数其他 KPI 约束也可以采用类似方式建模，为简化起见，本文仅考虑 CPA 约束。不过，与拍卖平台直接管理的预算约束不同，这些 KPI 约束在实际场景中通常并不严格。这是因为计算这些约束需要广告主针对所有参与竞价的广告展示的 v_i，因此只有在整个竞价过程结束后才能确定真实的 CPA。尽管如此，我们仍希望在建模中将其用作软约束。

因此，整个竞价过程可表示为：

$$\begin{aligned} \max_{b_1,\cdots,b_I} & \sum_i x_i v_i \\ \text{s.t.} & \sum_i x_i c_i \leq B \\ & \frac{\sum_i x_i c_i}{\sum_i x_i v_i} \leq C \end{aligned} \tag{5}$$

求解这一优化问题存在固有挑战，既源于广告展示数量庞大，也源于未来拍卖表现的根本不确定性。先前的研究 [17] 将该问题重新表述为线性规划问题，从而得到一种简化的最优出价策略：

$$b_i^* = \lambda_0^* v_i - \Sigma_j \lambda_j^* \left(\mathbb{q}_{ij} \left(1 - \mathbb{1}_{CR_j} \right) - \mathbb{k}_j \mathbb{p}_{ij} \right) \tag{6}$$

其中，b_i^*表示广告展示 i的理论最优出价，$\mathbb{q}_{ij}$可以是任意绩效指标或常数，$\mathbb{1}_{CR_j}$是指示约束 j是否与成本相关的指示函数。$\mathbb{p}_{ij}$和 $\mathbb{k}_j$可视为 v_i和 C在公式（5）中的表达式在多 KPI 条件下的扩展形式。这种重新表述将自动出价问题转化为寻找满足所有约束的最优 λ_0^*和λ_j^*。将公式（6）代入公式（5），并令 $j = 1$、$\mathbb{1}_{CR_j} = 1$、$\mathbb{p}_{ij} = v_i$、$\mathbb{k}_j = C$，且 $\mathbb{q}_{ij}$为任意绩效指标或常数，可得：

$$b_i^* = (\lambda_0^* + \lambda_1^* C) v_i = \lambda^* v_i \tag{7}$$

其中，$\lambda^* = \lambda_0^* + \lambda_1^* C$作为统一的出价参数。因此，近年来许多研究尝试通过在竞价过程中迭代地确定最优的 λ^*来解决竞价问题 [17, 27, 44]。此外，在根据公式（7）求解竞价问题时，第一个条件，即约束条件 $\sum_i x_i c_i \leq B$，始终成立。这是因为竞价平台会在广告主预算不足时自动控制 x_i，以确保广告主不会欠款。然而，第二个条件并不总是成立，因为我们预测的 λ与最优的 λ^*之间存在差距。对此，一个简单的解决办法是在用于模型选择的评估阶段，将有关 CPA 条件的惩罚项加入公式（7）中的目标函数 [44]，后文第 3.2 节将对此展开讨论。

2.2 基于 DT 的自动出价

为解决自动出价问题，现有方法采用基于规则的策略 [3, 42] 或强化学习（Reinforcement Learning, RL）方法 [51, 56] 进行优化。然而，基于规则的策略往往难以适应现实竞价环境高度动态的特性 [17]，而 RL 方法 [47] 依赖由 $s_{t+1} = f(s_t, a_t)$定义的状态转移，这使得对拍卖生态系统中固有且至关重要的时间依赖关系和历史观测进行建模变得复杂 [57]。

Transformer 架构的近期进展 [23, 28] 催生了决策变换器（Decision Transformer, DT）[6, 52, 63]，使其成为序列决策领域最先进的方法。DT 擅长捕捉长程依赖关系，因此非常适合拍卖结果呈现显著时间相关性的出价环境。基于这一框架，我们在 DT 设置下将自动出价视为序列建模任务 [27, 44]。出价周期被划分为离散时间步，每一步都在特定的环境设置下进行配置：

• 状态 s_t：状态向量 s_t包含一组用于描述时间步 t的出价条件的特征。在广告场景中，这些特征可以是剩余时间、未使用的预算、历史出价统计信息等。• 动作 a_t：动作 a_t表示可在整个出价周期内迭代调整的出价变量。本文根据式（7），得到最优动作为 $a = \lambda^*$。因此，我们将时间步 t的实际动作记为：

$$a_t = \lambda_t \tag{8}$$

• 奖励 rw_t：假设有 N_t个候选广告展示在 t与 $t+1$之间到来，则奖励 rw_t可定义为：

$$rw_t = \sum_{n=0}^{N_t} x_{n_t} v_{n_t} \tag{9}$$

其中，x_{n_t}和 v_{n_t}分别表示二元指示变量和值，对应的是 n^{th}广告展示，其时间步为 t。• 回报到当前（Return-To-Go, RTG）r_t：RTG 值表示未来时间步中将获得的奖励总量：

$$r_t = \sum_{t'=t}^{T} rw_{t'} \tag{10}$$

其中，T为最后一个时间步。

这些设置得到以下轨迹表示，该表示非常适合自回归训练与推理：

$$\tau = (r_1, s_1, a_1, r_2, s_2, a_2, \ldots, r_T, s_T, a_T) \tag{11}$$

3 框架

下面将详细介绍 GAVE 的总体框架及关键组成部分。

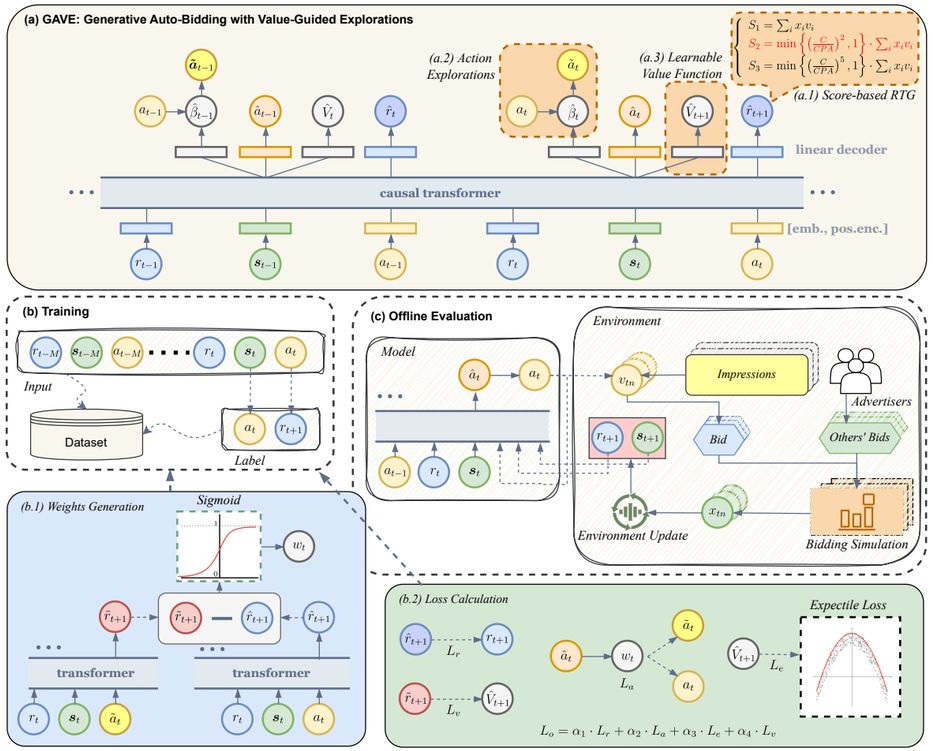


Figure 1: Overall structure of GAVE.

3.1 GAVE Overview

As shown in Figure 1, GAVE adopts a DT architecture, where the pair of RTGs, states, and actions form the input sequence, i.e., (r_t, s_t, a_t) for the timestamp t. Unlike conventional DTs, GAVE introduces several key innovations to achieve adaptive optimization, enhance stability, and facilitate strategy improvement. These include a **score-based RTG** (Figure 1 (a.1)) for aligning with diverse advertising objectives, an **action exploration module** (Figure 1 (a.2)) equipped with an RTG-based evaluation mechanism for discovering and evaluating new actions and stabilizing updates, and a **learnable value function** (Figure 1 (a.3)) to steer exploration for strategy improvement while mitigating Out-of-Distribution (OOD) risks. The training of GAVE follows an offline paradigm (Figure 1 (b)), using sequence samples as input to generate predicted labels. For evaluation, a simulated bidding environment (Figure 1 (c)) is employed, where the test model interacts with fixed-policy agents.

Specifically, GAVE employs an adaptive score-based RTG function that can align optimization objectives with varying advertising objectives. During action exploration at time step t, in addition to predicting the action $\hat{a}_t$, GAVE predicts a coefficient $\hat{\beta}_t$, a $\hat{V}_{t+1}$ for estimating the learnable value function V_{t+1}, and an RTG value $\hat{r}_{t+1}$. The process is formally expressed as follows:

$$\begin{cases} (\hat{\beta}_t, \hat{a}_t, \hat{V}_{t+1}) = GAVE(r_{t-M}, s_{t-M}, a_{t-M}, \ldots, r_t, s_t) \\ \hat{r}_{t+1} = GAVE(r_{t-M}, s_{t-M}, a_{t-M}, \ldots, r_t, s_t, a_t) \\ \tilde{a}_t = \hat{\beta}_t a_t \end{cases} \tag{12}$$

where M is a hyper-parameter, indicating a sequence with $M+1$ input time steps.

By evaluating the explored action $\tilde{a}_t$ and the action label a_t with an RTG-based evaluation method, GAVE applies a balanced update strategy to reconcile $\tilde{a}_t$ and a_t. This ensures a stability-preserving update process. Additionally, the learnable value function V_{t+1} is introduced to direct the model toward potentially optimal strategies while further reducing OOD risks. These innovations collectively enable GAVE to achieve improved performance and robustness.

3.2 Score-based RTG

As illustrated in Section 2.1, directly optimizing the cumulative value of won impressions may cause the CPA constraint to significantly exceed its limit. To address this issue, objective functions incorporating penalty terms can be established as evaluation metrics, allowing the degree of emphasis on CPA restrictions to be tailored to specific advertising objectives. This approach facilitates

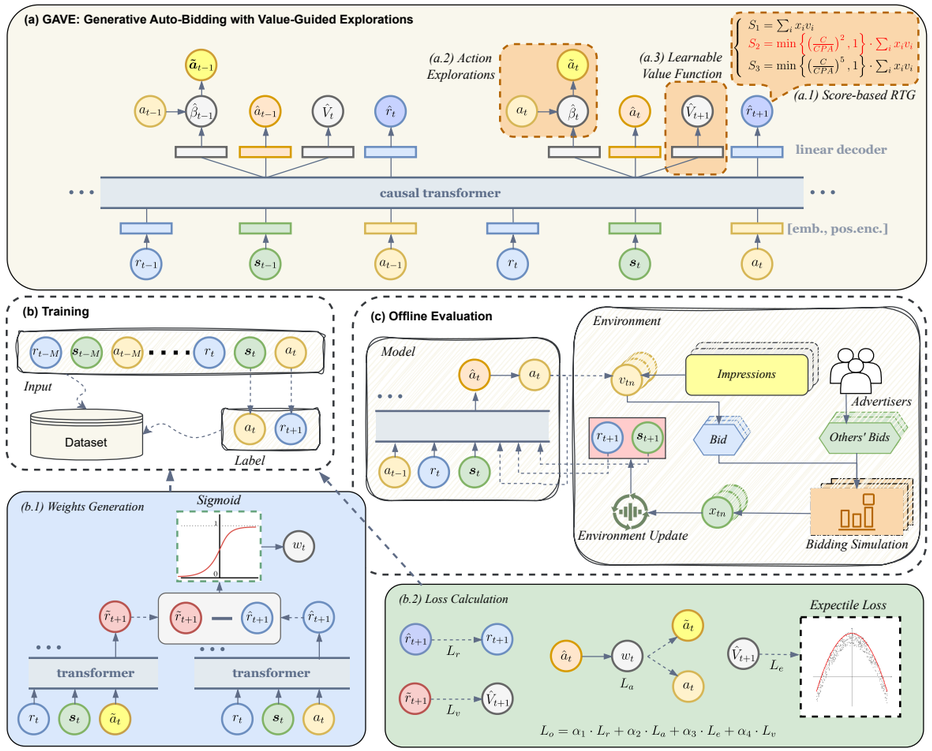


图 1：GAVE 的总体结构。

3.1 GAVE 概述

如图 1 所示，GAVE 采用 DT 架构，将回报到当前（Return-To-Go, RTG）、状态与动作配对组成输入序列，即(r_t, s_t, a_t)，对应时间戳 t。与传统 DT 不同，GAVE 引入了几项关键创新，以实现自适应优化、增强稳定性并促进策略改进。这些创新包括：基于评分的 RTG（图 1（a.1））、用于与多种广告目标对齐；配备基于 RTG 的评估机制的动作探索模块（图 1（a.2））、用于发现和评估新动作，并稳定更新过程；以及可学习的价值函数（图 1（a.3））、用于引导探索以改进策略，同时缓解分布外（Out-of-Distribution, OOD）风险。GAVE 的训练遵循离线范式（图 1（b））、将序列样本作为输入以生成预测标签。评估时采用模拟出价环境（图 1（c））、待测模型在其中与采用固定策略的智能体交互。

具体而言，GAVE 采用自适应的基于评分的 RTG 函数，能够将优化目标与不同广告目标对齐。在时间步 t进行动作探索时，除了预测动作$\hat{a}_t$，GAVE 还预测一个系数$\hat{\beta}_t$、一个$\hat{V}_{t+1}$（用于估计可学习的价值函数V_{t+1}），以及一个 RTG 值$\hat{r}_{t+1}$。这一过程形式化表示如下：

$$\begin{cases} (\hat{\beta}_t, \hat{a}_t, \hat{V}_{t+1}) = GAVE(r_{t-M}, s_{t-M}, a_{t-M}, \ldots, r_t, s_t) \\ \hat{r}_{t+1} = GAVE(r_{t-M}, s_{t-M}, a_{t-M}, \ldots, r_t, s_t, a_t) \\ \tilde{a}_t = \hat{\beta}_t a_t \end{cases} \tag{12}$$

其中， M是一个超参数，表示序列包含 $M+1$个输入时间步。

GAVE 使用基于 RTG 的评估方法评估所探索的动作$\tilde{a}_t$和动作标签 a_t，并采用平衡的更新策略来协调$\tilde{a}_t$与 a_t。这确保了一个保持稳定性的

更新过程。此外，引入可学习的价值函数 V_{t+1}，以引导模型朝向潜在最优策略，同时进一步降低 OOD 风险。这些创新共同使 GAVE 获得更好的性能与鲁棒性。

3.2 基于评分的 RTG

如第 2.1 节所述，直接优化赢得的广告展示的累计价值，可能导致 CPA 显著超出其约束上限。为解决这一问题，可以建立包含惩罚项的目标函数作为评估指标，从而根据具体广告目标调整对 CPA 限制的重视程度。这种方法有助于

the evaluation and selection of the optimal model. For example, previous work [44] proposes using a score S to assess the model's actual performance during the testing phase, thereby enabling the selection of higher-performing models. This score integrates a penalty term for CPA constraints to evaluate the overall performance of the bidding model throughout the entire bidding period with $\gamma = 2$:

$$\begin{cases} CPA = \frac{\sum_i x_i c_i}{\sum_i x_i v_i} \\ \mathbb{P}(CPA; C) = \min\left\{\left(\frac{C}{CPA}\right)^\gamma, 1\right\} \\ S = \mathbb{P}(CPA; C) \cdot \sum_i x_i v_i \end{cases} \tag{13}$$

In this paper, we integrate the constraints directly into the training stage, moving beyond reliance on pre-trained model selection to achieve improved evaluation scores. To align with evaluation metrics for various advertising objectives, we propose employing a constrained score function instead of the unconstrained $\sum_{i=1}^{I} x_i v_i$ for RTG modeling in GAVE, as illustrated in Figure 1 (a.1). For example, based on the evaluation metric defined in Equation (13), the following score-based RTG function can be utilized to synchronize training with evaluation:

$$\begin{cases} CPA_t = \frac{\sum_i^{I_t} x_i c_i}{\sum_i^{I_t} x_i v_i} \\ \mathbb{P}(CPA_t; C) = \min\left\{\left(\frac{C}{CPA_t}\right)^\gamma, 1\right\} \\ S_t = \mathbb{P}(CPA_t; C) \cdot \sum_i^{I_t} x_i v_i \\ r_t = S_T - S_{t-1} \end{cases} \tag{14}$$

Here, I_t denotes the number of impressions from time step 0 to time step t, S_t represents the generalized score function at time step t, and T signifies the final time step in a bidding period. By generalizing the score calculation to each time step, the RTG r_t is derived to represent the future score yet to be obtained, guiding the optimization direction of GAVE.

Furthermore, in practical applications, different advertising objectives may exhibit varying degrees of dependence on CPA constraints, resulting in different evaluation metrics. Nonetheless, training and evaluation can remain aligned by generalizing S to each time step (i.e., S_t) in a similar way, leading to:

$$r_t = S_T - S_{t-1} \tag{15}$$

This score-based RTG function enhances the flexibility of GAVE, ensuring its applicability across diverse advertising objectives.

3.3 Action Explorations

The primary objective of this section is to explore novel actions during training to discover strategies potentially absent from the offline dataset, thereby enabling better model optimization. However, in an offline setting where environment interaction is impossible, learning merely from the fixed dataset may lead to documented behavioral patterns. Conversely, exploring actions beyond the dataset can introduce inherent distribution shifts, potentially leading to behavioral collapse [6, 21]. Moreover, compared to real action labels, the impact of explored actions on model performance can be either beneficial or detrimental, presenting significant challenges in developing a stability-preserving update procedure.

To address these challenges, GAVE introduces a novel action exploration mechanism in conjunction with an RTG-based evaluation method as illustrated in Figure 1 (a.2). This enables GAVE to adaptively adjust both the exploration and update directions of actions by identifying their significance, thereby achieving stability-preserving updates. Specifically, at time step t, GAVE predicts a coefficient $\hat{\beta}_t$ with the same dimensionality as a_t to generate a new action $\tilde{a}_t$. This process is formally expressed as:

$$\begin{cases} \hat{\beta}_t = \sigma(FC_\beta(DT(r_{t-M}, \mathbf{s}_{t-M}, a_{t-M}, \ldots, r_t, \mathbf{s}_t))) \\ \tilde{a}_t = \hat{\beta}_t a_t \end{cases} \tag{16}$$

where $DT()$ represents the DT backbone, $FC_\beta()$ denotes a fully-connected layer, and σ is scaling function. To mitigate OOD issues, the scaling function is defined as:

$$\sigma(x) = Sigmoid(x) + 0.5 \tag{17}$$

This formulation constrains $\hat{\beta}_t$ to the interval $(0.5, 1.5)$, ensuring the explored action $\tilde{a}_t$ remains in proximity to the action label a_t.

To minimize distribution shift and obtain stability-preserving updates during training, rather than directly utilizing $\tilde{a}_t$ for generating new samples, we employ it as an additional label to balance action updates in conjunction with the original label a_t. This approach necessitates estimating the relative significance of $\tilde{a}_t$ and a_t to determine an optimal update direction for the predicted action $\hat{a}_t$. Following reinforcement learning conventions [21, 39, 41], we define the action-value of a_t as r_{t+1} (the RTG at time step $t+1$), as it represents the cumulative future returns after executing action a_t. This enables the design of w_t as illustrated in Figure 1 (b.1) to balance the update direction:

$$\begin{cases} \tilde{r}_{t+1} = GAVE(r_{t-M}, \mathbf{s}_{t-M}, a_{t-M}, \ldots, r_t, \mathbf{s}_t, \tilde{a}_t))) \\ w_t = Sigmoid(\alpha_r \cdot (\tilde{r}_{t+1} - \hat{r}_{t+1})) \end{cases} \tag{18}$$

where $\tilde{r}_{t+1}$ and $\hat{r}_{t+1}$ represent the estimated RTG for $\tilde{a}_t$ and a_t respectively. The corresponding loss function for action explorations is defined as:

$$\begin{cases} L_r = \frac{1}{M+1} \sum_{t-M}^{t} (\hat{r}_{t+1} - r_{t+1})^2 \\ L_a = \frac{1}{M+1} \sum_{t-M}^{t} ((1 - w'_t) \cdot (\hat{a}_t - a_t)^2 + w'_t \cdot (\hat{a}_t - \tilde{a}'_t)^2) \end{cases} \tag{19}$$

where w' and $\tilde{a}'_t$ denote w and $\tilde{a}_t$ with frozen gradients. Through L_r, GAVE ensures accurate RTG prediction, enabling reliable estimation of the RTG for both $\tilde{a}_t$ and a_t. Through L_a, GAVE maintains a balanced and stability-preserving updating process between $\tilde{a}_t$ and a_t, directing updates toward $\tilde{a}_t$ when it proves superior ($w_t > 0.5$), and toward a_t otherwise to mitigate OOD issues and potential negative impacts from exploration.

3.4 Learnable Value Function

While the action exploration mechanism ensures explorations outside the dataset and a stability-preserving update process, randomly generated $\tilde{a}_t$ cannot guarantee improved model performance. To address this limitation, we propose a learnable value function that

评估和选择最优模型。例如，先前工作 [44] 提出在测试阶段使用评分S评估模型的实际表现，从而能够选择性能更好的模型。该评分结合了针对 CPA 约束的惩罚项，在 $\gamma = 2$的情况下，评估出价模型在整个出价周期内的整体性能：

$$\begin{cases} CPA = \frac{\sum_i x_i c_i}{\sum_i x_i v_i} \\ \mathbb{P}(CPA; C) = \min\left\{\left(\frac{C}{CPA}\right)^\gamma, 1\right\} \\ S = \mathbb{P}(CPA; C) \cdot \sum_i x_i v_i \end{cases} \tag{13}$$

本文将约束直接融入训练阶段，以超越仅依赖预训练模型选择的方式，获得更高的评估评分。为与不同广告目标的评估指标对齐，我们提出在 GAVE 的 RTG 建模中使用带约束的评分函数，替代无约束的 $\sum_{i=1}^{I} x_i v_i$，如图 1（a.1）所示。例如，基于式（13）定义的评估指标，可以使用以下基于评分的 RTG 函数，使训练与评估同步：

$$\begin{cases} CPA_t = \frac{\sum_i^{I_t} x_i c_i}{\sum_i^{I_t} x_i v_i} \\ \mathbb{P}(CPA_t; C) = \min\left\{\left(\frac{C}{CPA_t}\right)^\gamma, 1\right\} \\ S_t = \mathbb{P}(CPA_t; C) \cdot \sum_i^{I_t} x_i v_i \\ r_t = S_T - S_{t-1} \end{cases} \tag{14}$$

这里，I_t表示从时间步 0 到时间步 t的广告展示数量，S_t表示时间步 t的广义评分函数，T表示一个出价周期的最后一个时间步。通过将评分计算推广到每个时间步，可以得到 RTG r_t，用于表示未来尚未获得的评分，并引导 GAVE 的优化方向。

此外，在实际应用中，不同广告目标对 CPA 约束的依赖程度可能有所不同，因而产生不同的评估指标。尽管如此，仍可通过类似方式将 S推广到每个时间步（即 S_t），使训练与评估保持一致，从而得到：

$$r_t = S_T - S_{t-1} \tag{15}$$

这一基于评分的 RTG 函数增强了 GAVE 的灵活性，确保其能够适用于多种广告目标。

3.3 动作探索

本节的主要目标是在训练期间探索新动作，以发现离线数据集中可能不存在的策略，从而实现更好的模型优化。然而，在无法与环境交互的离线设置中，仅从固定数据集学习可能会使模型遵循已有记录中的行为模式。相反，探索数据集之外的动作会引入固有的分布偏移，可能导致行为坍塌 [6, 21]。此外，与实际动作标签相比，所探索的动作对模型性能的影响可能有利，也可能有害，这给设计保持稳定性的更新过程带来了重大挑战。

为应对这些挑战，GAVE 引入了一种新颖的动作探索机制，并结合基于 RTG 的评估方法，如图 1（a.2）所示。这使 GAVE 能够通过识别动作的重要性，自适应地调整动作的探索方向与更新方向，从而实现保持稳定性的更新。具体而言，在时间步 t、GAVE 预测一个系数$\hat{\beta}_t$，其维度与 a_t相同，用于生成一个新动作$\tilde{a}_t$。这一过程形式化表示如下：

$$\begin{cases} \hat{\beta}_t = \sigma(FC_\beta(DT(r_{t-M}, \mathbf{s}_{t-M}, a_{t-M}, \ldots, r_t, \mathbf{s}_t))) \\ \tilde{a}_t = \hat{\beta}_t a_t \end{cases} \tag{16}$$

其中， $DT()$ 表示 DT 主干网络， $FC_\beta()$ 表示全连接层， σ 为缩放函数。为缓解 OOD 问题，缩放函数定义为：

$$\sigma(x) = Sigmoid(x) + 0.5 \tag{17}$$

这一表述将$\hat{\beta}_t$约束在区间 $(0.5, 1.5)$内，确保探索得到的动作$\tilde{a}_t$保持在动作标签 a_t附近。

为尽量减小分布偏移并在训练期间实现保持稳定的更新，我们不直接利用$\tilde{a}_t$生成新样本，而是将其用作一个额外标签，与原始标签 a_t共同平衡动作更新。这种方法需要估计$\hat{a}_t$与 a_t的相对重要性，以确定预测动作的最优更新方向，该预测动作为

$\hat{a}_t$。遵循强化学习（Reinforcement Learning, RL）的惯例 [21, 39, 41]，我们将 a_t的动作价值定义为 r_{t+1}（即时间步 $t+1$的回报到当前（Return-To-Go, RTG））， 因为它表示执行动作a_t之后的未来累计回报。据此，我们可以设计 w_t，如图 1（b.1）所示，以平衡更新方向：

$$\begin{cases} \tilde{r}_{t+1} = GAVE(r_{t-M}, \mathbf{s}_{t-M}, a_{t-M}, \ldots, r_t, \mathbf{s}_t, \tilde{a}_t))) \\ w_t = Sigmoid(\alpha_r \cdot (\tilde{r}_{t+1} - \hat{r}_{t+1})) \end{cases} \tag{18}$$

其中，$\tilde{r}_{t+1}$和$\hat{r}_{t+1}$分别表示对$\tilde{a}_t$和 a_t的 RTG 估计值。动作探索对应的损失函数定义为：

$$\begin{cases} L_r = \frac{1}{M+1} \sum_{t-M}^{t} (\hat{r}_{t+1} - r_{t+1})^2 \\ L_a = \frac{1}{M+1} \sum_{t-M}^{t} ((1 - w'_t) \cdot (\hat{a}_t - a_t)^2 + w'_t \cdot (\hat{a}_t - \tilde{a}'_t)^2) \end{cases} \tag{19}$$

其中，w'和$\tilde{a}'_t$分别表示停止梯度的w和$\tilde{a}_t$。通过 L_r，GAVE 确保 RTG 预测准确，从而可靠估计$\tilde{a}_t$和 a_t的 RTG。通过 L_a，GAVE 在$\tilde{a}_t$与a_t之间保持平衡、稳定的更新：当$\tilde{a}_t$表现更优（$w_t > 0.5$）时朝该动作更新，否则朝 a_t更新，以缓解 OOD 问题及探索可能带来的负面影响。

3.4 可学习的价值函数

尽管动作探索机制确保了对数据集之外的探索以及保持稳定性的更新过程，但随机生成的$\tilde{a}_t$无法保证改善模型性能。为解决这一局限，我们提出一个可学习的价值函数，它

Algorithm 1 Optimization algorithm of GAVE

Input: A training dataset $\mathcal{D} = \left\{\left(\boldsymbol{z}_j, \boldsymbol{y}_j\right)\right\}_{j=1}^{|\mathcal{D}|}$ with $|\mathcal{D}|$ samples sampled from a bidding environment. $\boldsymbol{z}_j$ is a sequence with $M+1$ time steps and $\boldsymbol{y}_j$ are the label set.
$\boldsymbol{z}_j = \{r_{t-M}, \boldsymbol{s}_{t-M}, a_{t-M}, \ldots, r_t, \boldsymbol{s}_t, a_t\}$; $\boldsymbol{y}_j = \{a_t, r_{t+1}\}$
Output: A well-trained model f with parameters $\boldsymbol{\Phi}$

```
Randomly initialize parameters Φ of the model f
for Step 1,..., Max Step do
    Sample a training batch B from D
    Obtain β̂_t, â_t, V̂_{t+1}, r̂_{t+1} and ã_t with f(B) via Equation (12)
    Obtain r̃_{t+1} and w_t via Equation (18)
    Calculate loss L_r, L_a, L_e and L_v based on Equation (19), (21) and (22)
    Calculate the overall loss L_o based on Equation (23)
    Update Φ via minimizing the loss L_o
end for
return f
```

facilitates the discovery of superior actions for strategy improvement, as illustrated in Figure 1 (a.3). Specifically, drawing inspiration from reinforcement learning conventions [21, 39, 41], we propose a sequence-value function V_{t+1} analogous to the optimal state-value function in RL, which represents the upper bound of r_{t+1} as follows:

$$V_{t+1} = \arg\max_{a_t \in \mathbb{A}} r_{t+1} \tag{20}$$

where $\mathbb{A}$ denotes the available action space. Due to the extensive action space and limited real actions within offline datasets, the direct statistical computation of V_{t+1} is infeasible. However, we can learn this value through an expectile regression process with r_{t+1}:

$$\begin{aligned} L_e &= \frac{1}{M+1}\sum_{t-M}^{t}\left(L_2^{\tau}(r_{t+1} - \hat{V}_{t+1})\right) \\ &= \frac{1}{M+1}\sum_{t-M}^{t}\left(|\tau - \mathbb{1}((r_{t+1} - \hat{V}_{t+1}) < 0)|(r_{t+1} - \hat{V}_{t+1})^2\right) \end{aligned} \tag{21}$$

where $L_2^{\tau}(y - m(x))$ represents the loss function for predicting the expectile $\tau \in (0,1)$ of labels y with model $m(x)$ [21]. $\hat{V}_{t+1}$ represents the predicted value of V_{t+1}. Following Equation (20), we set $\tau = 0.99$ to learn the upper bound of r_{t+1}, effectively estimating V_{t+1}.

By estimating V_{t+1} with $\hat{V}_{t+1}$ and utilizing it to guide the updating directions of $\tilde{r}_{t+1}$, GAVE implicitly steers the update direction of the explored $\tilde{a}_t$ toward potentially optimal actions. This process is illustrated in Figure 1 (b.2) and can be formalized as:

$$L_v = \frac{1}{M+1}\sum_{t-M}^{t}(\tilde{r}_{t+1} - \hat{V}'_{t+1})^2 \tag{22}$$

where $\hat{V}'_{t+1}$ represents the gradient-frozen version of $\hat{V}_{t+1}$. Through the application of L_v, GAVE implicitly guides the update direction of $\tilde{a}_t$ toward the optimal action by anchoring their RTGs near $\hat{V}_{t+1}$. This approach mitigates OOD risks and enables controlled extrapolation for strategy improvement.

Table 1: Data statistics.

Params	AuctionNet	AuctionNet-Sparse
Trajectories	479,376	479,376
Delivery Periods	9,987	9,987
Time steps in a trajectory	48	48
State dimension	16	16
Action dimension	1	1
Return-To-Go Dimension	1	1
Action range	[0, 493]	[0, 589]
Impression's value range	[0, 1]	[0, 1]
CPA range	[6, 12]	[60, 130]
Total conversion range	[0, 1512]	[0, 57]

3.5 Optimization Algorithm

Through the mechanisms described above, GAVE implements an offline generative auto-bidding framework incorporating value-guided explorations to enhance strategy learning. The comprehensive loss function is formulated as a weighted combination of the components defined in Equations (19), (21), and (22):

$$L_o = \alpha_1 \cdot L_r + \alpha_2 \cdot L_a + \alpha_3 \cdot L_e + \alpha_4 \cdot L_v \tag{23}$$

where $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ are hyperparameters controlling the relative contribution of each loss component. The complete optimization procedure of GAVE is detailed in Algorithm 1, with the training process visualized in Figure 1 (b).

During inference, as illustrated in Figure 1 (c), GAVE processes each input sequence to predict $\hat{a}_t = \lambda_t$, which serves as the bid parameter at time step t. The bidding price for the n-th impression at time step t is then computed according to Equation (7) as $b_{tn} = \lambda_t v_{tn}$, enabling real-time bidding simulation.

4 Offline Experiments

In this section, we conduct experiments on two public datasets to investigate the following questions:

- **RQ1:** How does GAVE perform compared to state-of-the-art auto-bidding baselines?
- **RQ2:** Can GAVE adapt to diverse advertising objectives?
- **RQ3:** How effective is the proposed learnable value function in facilitating action exploration?
- **RQ4:** How do the proposed components in GAVE contribute to the final bidding performance?

In the following subsections, we begin by outlining the evaluation settings. Then, we address the relevant questions by concisely analyzing our experimental findings.

4.1 Experimental Setup

4.1.1 ***Dataset.*** Previous auto-bidding research has predominantly relied on proprietary bidding logs for evaluation, with problem formulations often specific to particular scenarios. This heterogeneity in evaluation methodologies has hindered fair and systematic comparisons across different approaches. Recently, Alimama introduced AuctionNet [3] [44], the industry's first standardized large-scale simulated bidding benchmark, enabling comprehensive model evaluation under consistent conditions.

In this study, we utilize two datasets from the AuctionNet framework: (i) **AuctionNet**: The primary dataset containing comprehensive bidding trajectories, and (ii) **AuctionNet-Sparse**: A sparse variant of AuctionNet featuring reduced conversion rates. Both

[3]https://github.com/alimama-tech/AuctionNet

Algorithm 1 Optimization algorithm of GAVE

Input: A training dataset $\mathcal{D} = \left\{\left(\boldsymbol{z}_j, \boldsymbol{y}_j\right)\right\}_{j=1}^{|\mathcal{D}|}$ with $|\mathcal{D}|$ samples sampled from a bidding environment. $\boldsymbol{z}_j$ is a sequence with $M+1$ time steps and $\boldsymbol{y}_j$ are the label set.
$\boldsymbol{z}_j = \{r_{t-M}, \boldsymbol{s}_{t-M}, a_{t-M}, \ldots, r_t, \boldsymbol{s}_t, a_t\}$; $\boldsymbol{y}_j = \{a_t, r_{t+1}\}$
Output: A well-trained model f with parameters $\boldsymbol{\Phi}$

```
Randomly initialize parameters Φ of the model f
for Step 1,..., Max Step do
    Sample a training batch B from D
    Obtain β̂_t, â_t, V̂_{t+1}, r̂_{t+1} and ã_t with f(B) via Equation (12)
    Obtain r̃_{t+1} and w_t via Equation (18)
    Calculate loss L_r, L_a, L_e and L_v based on Equation (19), (21) and (22)
    Calculate the overall loss L_o based on Equation (23)
    Update Φ via minimizing the loss L_o
end for
return f
```

有助于发现更优的动作以改进策略，如图 1（a.3）所示。具体而言，受 RL 惯例 [21, 39, 41] 的启发，我们提出了一种序列价值函数 V_{t+1}，它类似于 RL 中的最优状态价值函数，表示 r_{t+1}的上界，具体如下：

$$V_{t+1} = \arg\max_{a_t \in \mathbb{A}} r_{t+1} \tag{20}$$

其中，$\mathbb{A}$表示可用的动作空间。由于动作空间庞大，而离线数据集中的实际动作有限，直接通过统计计算得到 V_{t+1}是不可行的。不过，我们可以通过期望分位数（expectile）回归过程，结合 r_{t+1}: •来学习这一数值。

$$\begin{aligned} L_e &= \frac{1}{M+1}\sum_{t-M}^{t}\left(L_2^{\tau}(r_{t+1} - \hat{V}_{t+1})\right) \\ &= \frac{1}{M+1}\sum_{t-M}^{t}\left(|\tau - \mathbb{1}((r_{t+1} - \hat{V}_{t+1}) < 0)|(r_{t+1} - \hat{V}_{t+1})^2\right) \end{aligned} \tag{21}$$

其中，$L_2^{\tau}(y - m(x))$表示用于预测期望分位数$\tau \in (0,1)$的损失函数，所用标签为u，模型为$m(x)$ [21]。$\hat{V}_{t+1}$表示V_{t+1}的预测值。根据公式（20），我们设置 $\tau = 0.99$以学习 r_{t+1}的上界，从而有效估计 V_{t+1}。通过估计V_{t+1}（使用$\hat{V}_{t+1}$），并利用该估计值引导$\tilde{r}_{t+1}$的更新方向，GAVE 隐式地将探索得到的$\tilde{a}_t$的更新方向引向潜在最优动作。该过程如图 1（b.2）所示，可形式化表示为：

$$L_v = \frac{1}{M+1}\sum_{t-M}^{t}(\tilde{r}_{t+1} - \hat{V}'_{t+1})^2 \tag{22}$$

其中，$\hat{V}'_{t+1}$表示冻结梯度后的$\hat{V}_{t+1}$。通过应用 L_v，GAVE 隐式地引导$\tilde{a}_t$朝最优动作的方向更新，方法是将其回报到当前（Return-To-Go, RTG）锚定在$\hat{V}_{t+1}$附近。该方法降低了分布外（Out-of-Distribution, OOD）风险，并能够通过受控外推改进策略。

Table 1: Data statistics.

Params	AuctionNet	AuctionNet-Sparse
Trajectories	479,376	479,376
Delivery Periods	9,987	9,987
Time steps in a trajectory	48	48
State dimension	16	16
Action dimension	1	1
Return-To-Go Dimension	1	1
Action range	[0, 493]	[0, 589]
Impression's value range	[0, 1]	[0, 1]
CPA range	[6, 12]	[60, 130]
Total conversion range	[0, 1512]	[0, 57]

3.5 Optimization Algorithm

通过上述机制，GAVE 实现了一个融合价值引导探索的离线生成式自动出价框架，以增强策略学习。完整的损失函数由公式（19）、(21) 和 (22) 定义的各项损失加权组合而成：

$$L_o = \alpha_1 \cdot L_r + \alpha_2 \cdot L_a + \alpha_3 \cdot L_e + \alpha_4 \cdot L_v \tag{23}$$

其中，$\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$为超参数，用于控制各损失项的相对贡献。GAVE 的完整优化过程详见算法 1，训练过程如图 1 (b) 所示。

在推理期间，如图 1 (c) 所示，GAVE 处理每个输入序列，预测$\hat{a}_t = \lambda_t$，将其作为时间步 t的出价参数。对于第 n次广告展示，在时间步 t的出价随后根据公式（7）计算为 $b_{tn} = \lambda_t v_{tn}$，从而实现实时竞价模拟。

4 离线实验

本节在两个公开数据集上开展实验，研究以下问题：

RQ1：与最先进的自动出价基线方法相比，GAVE 的表现如何？• RQ2：GAVE 能否适应多样化的广告目标？• RQ3：所提出的可学习价值函数在促进动作探索方面有多大效果？• RQ4：GAVE 中提出的各个组件对最终出价性能有何贡献？

在接下来的各小节中，我们首先概述评估设置，然后通过简要分析实验结果来回答相关问题。

4.1 实验设置

4.1.1 数据集。以往的自动出价研究主要依赖专有竞价日志进行评估，其问题形式化表述也往往针对特定场景。评估方法的这种异质性阻碍了对不同方法进行公平而系统的比较。近期，阿里妈妈推出了 AuctionNet [3] [44]，这是业界首个标准化的大规模模拟竞价基准，可在一致条件下对模型进行全面评估。

本研究使用 AuctionNet 框架中的两个数据集：(i) AuctionNet：包含完整竞价轨迹的主要数据集；(ii) AuctionNet-Sparse：AuctionNet 的稀疏变体，其转化率有所降低。两者

[3]https://github.com/alimama-tech/AuctionNet

Table 2: Performance comparison. The boldface denotes the highest score. The underline indicates the best result of baselines. "*" indicates the statistically significant improvements (i.e., two-sided t-test with $p < 0.05$) over the best baseline.

Dataset	Budget	DiffBid	USCB	CQL	IQL	BCQ	DT	CDT	GAS	GAVE	*Improve*
AuctionNet	50%	54	86	113	164	190	191	174	193	**201***	4.15%
	75%	100	135	139	232	259	265	242	287	**296***	3.14%
	100%	152	157	171	281	321	329	326	359	**376***	4.74%
	125%	193	220	201	355	379	396	378	409	**421***	2.93%
	150%	234	281	238	401	429	450	433	461	**467***	1.30%
AuctionNet-Sparse	50%	9.9	11.5	12.8	16.5	17.7	14.8	11.2	18.4	**19.6***	6.52%
	75%	15.4	14.9	16.7	22.1	24.6	22.9	18.0	27.5	**28.3***	2.91%
	100%	19.5	17.5	22.2	30.0	31.1	29.6	31.2	36.1	**37.2***	3.05%
	125%	25.3	26.7	28.6	37.1	34.2	34.3	31.7	40.0	**42.7***	6.75%
	150%	30.8	31.3	35.8	43.1	37.9	44.5	39.1	46.5	**47.4***	1.94%

datasets comprise approximately 500k bidding trajectories collected across 10k distinct delivery periods, each consisting of 48 time steps and interactions derived from millions of impression opportunities. Detailed statistics are presented in Table 1.

4.1.2 ***Evaluation Protocol.*** Our evaluation methodology follows the AuctionNet benchmark [44] and employs a simulated environment [27] that emulates real-world advertising systems, as illustrated in Figure 1 (c). The evaluation spans a 24-hour delivery period, discretized into 48 uniform time steps, during which the predicted action is utilized for bidding ($\hat{a}_t = a_t$). Within this simulated environment, 48 bidding agents with distinct strategies compete for incoming impression opportunities, with performance measured using Equation (13) with $\gamma = 2$.

To ensure comprehensive evaluation, we employ a round-robin testing strategy: the test model sequentially replaces each of the 48 agents, competing against the remaining agents in each round. The final performance is computed as the average score across all evaluations, providing a robust measure of the model's effectiveness.

4.1.3 ***Baselines.*** To assess the effectiveness of GAVE, we perform a thorough comparison with multiple baseline approaches:

- **DiffBid** [16]: applies the diffusion frameworks to simulate bidding trajectories and model bidding sequences.
- **USCB** [17]: dynamically adjust bidding parameters for optimal bidding performance in an online RL bidding environment.
- **CQL** [22]: learns a conservative value function to mitigate the overestimation problems in offline RL.
- **IQL** [21]: applies an expectile regression method to enable policy improvement without evaluating the out-of-scope actions.
- **BCQ** [9]: applies a restriction on the action space for a typical offline RL learning process.
- **DT** [6]: employs a transformer architecture for sequential decision-making modeling and utilizes a behavior cloning method to learn the average strategy from the dataset.
- **CDT** [33]: tries to train a constraint satisfaction policy in the offline settings for a balance of safety and task performance.
- **GAS** [27]: tries to model a DT-based offline bidding framework with post-training search by applying Monte Carlo Tree Search (MCTS) in modeling.

4.1.4 ***Implementation Details.*** Following previous studies [27, 44], we conduct evaluations using varying budget ratios from the original dataset. Performance is measured using the scoring metric:

$$S = \mathbb{P}(CPA; C) \cdot \sum_i x_i v_i \tag{24}$$

as defined in Equation (13) with $\gamma = 2$.

All experiments are conducted on NVIDIA H100 GPUs, utilizing a fixed batch size 128 for a maximum of 400k training steps. The GAVE implementation employs a causal transformer architecture with 8 layers and 16 attention heads. Model parameters are optimized using the AdamW optimizer [36] with a learning rate of $1e^{-5}$. Additional hyper-parameters are determined through a comprehensive grid search to maximize performance. To ensure statistical significance, we conduct 10 independent runs using the optimal parameter configuration and report the average performance metrics.

4.2 Overall Performance (RQ1)

We present a comprehensive comparison between GAVE and various baseline approaches across different budget settings, with results summarized in Table 2. Our experimental analysis reveals several key findings:

- GAVE demonstrates superior performance across all budget and dataset configurations, consistently outperforming existing methods. This superiority can be attributed to our novel action exploration method with the value function's guidance, which enables the discovery of novel, potentially optimal actions beyond the offline dataset while maintaining robust training through a stability-preserving update process balancing the exploration benefits and risks.
- Among all baselines, DT-based methods (GAS, DT, and CDT) exhibit superior performance, highlighting the effectiveness of DT structure in capturing temporal dependencies and facilitating sequential decision-making in bidding scenarios. Notably, GAS achieves better results compared to DT and CDT, validating the efficacy of its MCTS implementation in strategy optimization.
- DiffBid does not perform well on the datasets, probably due to the long sequence and highly dynamic environment posing extra challenges for DiffBid in accurately predicting trajectories and learning from its reverse process.

4.3 Alignment Analysis (RQ2)

As discussed in Section 3.2, advertising objectives may necessitate different evaluation metrics. To address this, GAVE employs an adaptive score-based RTG modeling that accommodates various optimization objectives, thereby aligning training objectives with

Table 2: Performance comparison. The boldface denotes the highest score. The underline indicates the best result of baselines. "*" indicates the statistically significant improvements (i.e., two-sided t-test with $p < 0.05$) over the best baseline.

Dataset	Budget	DiffBid	USCB	CQL	IQL	BCQ	DT	CDT	GAS	GAVE	*Improve*
AuctionNet	50%	54	86	113	164	190	191	174	193	**201***	4.15%
	75%	100	135	139	232	259	265	242	287	**296***	3.14%
	100%	152	157	171	281	321	329	326	359	**376***	4.74%
	125%	193	220	201	355	379	396	378	409	**421***	2.93%
	150%	234	281	238	401	429	450	433	461	**467***	1.30%
AuctionNet-Sparse	50%	9.9	11.5	12.8	16.5	17.7	14.8	11.2	18.4	**19.6***	6.52%
	75%	15.4	14.9	16.7	22.1	24.6	22.9	18.0	27.5	**28.3***	2.91%
	100%	19.5	17.5	22.2	30.0	31.1	29.6	31.2	36.1	**37.2***	3.05%
	125%	25.3	26.7	28.6	37.1	34.2	34.3	31.7	40.0	**42.7***	6.75%
	150%	30.8	31.3	35.8	43.1	37.9	44.5	39.1	46.5	**47.4***	1.94%

数据集包含约 50 万条出价轨迹，采集自 1 万个不同的投放周期；每个周期包含 48 个时间步，以及由数百万次广告展示机会产生的交互。详细统计信息见表 1。

4.1.2 评估方案。我们的评估方法遵循 AuctionNet 基准 [44]，并采用模拟真实广告系统的仿真环境 [27]，如图 1（c）所示。评估覆盖一个 24 小时的投放周期，并将其离散为 48 个等长时间步，在此期间使用预测动作进行出价（$\hat{a}_t = a_t$）。在该仿真环境中，48 个采用不同策略的出价智能体竞争到来的广告展示机会，并使用式（13）衡量性能，其中 $\gamma = 2$。

为确保评估全面，我们采用轮换测试策略：待测模型依次替换这 48 个智能体中的每一个，并在每一轮中与其余智能体竞争。最终性能取所有评估评分的平均值，从而稳健地衡量模型的有效性。

4.1.3 基线方法。为评估 GAVE 的有效性，我们将其与多种基线方法进行全面比较：

• DiffBid [16]：应用扩散框架模拟出价轨迹，并对出价序列建模。• USCB [17]：在在线强化学习（Reinforcement Learning, RL）出价环境中动态调整出价参数，以获得最优出价表现。• CQL [22]：学习保守的价值函数，以缓解离线 RL 中的过估计问题。• IQL [21]：应用期望分位数（expectile）回归方法，无须评估范围之外的动作即可改进策略。• BCQ [9]：在典型的离线 RL 学习过程中限制动作空间。• 决策变换器（Decision Transformer, DT）[6]：采用 Transformer 架构对序列决策进行建模，并使用行为克隆方法从数据集中学习平均策略。• CDT [33]：尝试在离线设置下训练满足约束的策略，以平衡安全性与任务表现。• GAS [27]：尝试通过在建模中应用蒙特卡洛树搜索（Monte Carlo Tree Search, MCTS），构建带有训练后搜索的、基于 DT 的离线出价框架。

4.1.4 实现细节。遵循先前研究 [27, 44]，我们采用相对于原始数据集的不同预算比例进行评估。性能使用以下评分指标衡量：

$$S = \mathbb{P}(CPA; C) \cdot \sum_i x_i v_i \tag{24}$$

as defined in Equation (13) with $\gamma = 2$.

所有实验均在 NVIDIA H100 GPU 上进行，批次大小固定为 128，最多训练 40 万步。GAVE 采用包含 8 层、16 个注意力头的因果 Transformer 架构。模型参数使用 AdamW 优化器 [36] 优化，学习率为 $1e^{-5}$。其他超参数通过全面的网格搜索确定，以获得最佳性能。为确保统计显著性，我们使用最优参数配置进行 10 次独立运行，并报告性能指标的平均值。

4.2 整体性能（RQ1）

我们全面比较了 GAVE 与各种基线方法在不同预算设置下的表现，结果汇总于表 2。实验分析揭示了以下关键发现：

• GAVE 在所有预算和数据集配置下均表现优异，持续优于现有方法。这一优势可归因于我们提出的价值函数引导的动作探索方法：它能够在离线数据集之外发现新的、潜在最优的动作，同时通过平衡探索收益与风险、保持稳定的更新过程维持稳健训练。• 在所有基线中，基于 DT 的方法（GAS、DT 和 CDT）表现更好，表明 DT 结构能够有效捕捉时间依赖关系，促进出价场景中的序列决策。值得注意的是，GAS 的结果优于 DT 和 CDT，验证了其 MCTS 实现在策略优化方面的有效性。• DiffBid 在这些数据集上表现不佳，可能是因为长序列和高度动态的环境为 DiffBid 准确预测轨迹、从逆向过程中学习带来了额外挑战。

4.3 对齐分析（RQ2）

如第 3.2 节所述，广告目标可能需要不同的评估指标。为此，GAVE 采用自适应的、基于评分的 RTG 建模，以适应各种优化目标，从而使训练目标与

Table 3: Alignment analysis on AuctionNet-Sparse with 100% budget. "Train" denotes applying the score to RTG modeling, while "Eval" denotes its utilization as the evaluation metric.

Train \ Eval	S_1	S_2	S_3
S_1	41.4	33.0	23.6
S_2	39.9	37.2	33.3
S_3	39.1	36.8	33.5

evaluation metrics as illustrated in Equation (15). In this section, we explore the performance of GAVE under different RTG and evaluation metric configurations to answer RQ2. Specifically, we consider three evaluation metrics defined as follows:

$$\begin{cases} S_1 = \sum_i x_i v_i, \\ S_2 = \min\left\{\left(\frac{C}{CPA}\right)^2, 1\right\} \cdot \sum_i x_i v_i, \\ S_3 = \min\left\{\left(\frac{C}{CPA}\right)^5, 1\right\} \cdot \sum_i x_i v_i, \end{cases} \tag{25}$$

where S_1 only considers the total impression values obtained and represents the business scenarios with elastic restrictions on CPA conditions. S_2 is the optimization goal and evaluation score of this paper. It adds a punishment on CPA constraints. S_3 further enhances the penalty coefficient of CPA to represent the business scenarios with strict restrictions on CPA conditions. These metrics are utilized either for RTG modeling during training or as evaluation criteria during testing. The results are presented in Table 3.

From Table 3, we observe that GAVE consistently achieves the highest performance when the training RTG corresponds to the same function used as the evaluation metric. This finding underscores the importance of aligning training objectives with specific evaluation metrics through our score-based RTG approach.

4.4 Parameter Analysis (RQ3)

To address RQ3, we conduct a parameter analysis of the weight w_t, as depicted in Figure 1 (b.1), to elucidate the distinction between $\tilde{a}_t$ and a_t during the training process. Specifically, Figure 2 visualizes the average overall loss L_o and the weight w_t across training steps, allowing us to monitor the disparity between $\tilde{r}_{t+1}$ and $\hat{r}_{t+1}$. A larger w_t signifies a greater influence of $\tilde{r}_{t+1}$ over $\hat{r}_{t+1}$, thereby demonstrating that $\tilde{a}_t$ is superior to a_t. This result underscores the effectiveness of the value function in guiding action exploration.

From Figure 2, it is evident that as training progresses, the parameter w_t increases from approximately 0.5 to a stable position above 0.5. The stable position is influenced by both the dataset distribution and the model's hyper-parameters. This trend confirms the efficacy of the learnable value function in directing action exploration. With the guidance of the value function, the model consistently explores actions $\tilde{a}$ with higher RTG values $\tilde{r}_{t+1}$ near the estimated optimal value $\hat{V}_{t+1}$. This approach facilitates learning potential optimal strategies while mitigating OOD issues.

4.5 Ablation Study (RQ4)

To further elucidate the contribution of each module within GAVE for answering RQ4, we conduct an ablation study by evaluating the following modified versions of GAVE:

- **GAVE-V**: excludes the learnable value function described in Section 3.4. In this configuration, the loss functions L_v and L_e are replaced with the following update rule to ensure that the explored actions generally surpass the original labels by enhancing their RTG values $\tilde{r}_{t+1}$:

$$L_w = 1 - \text{Sigmoid}(\alpha_r \cdot (\tilde{r}_{t+1} - \hat{r}'_{t+1})) \tag{26}$$

where $\hat{r}'_{t+1}$ is a gradient-frozen version of $\hat{r}_{t+1}$. However, without the value function, the update direction of $\tilde{r}_{t+1}$ becomes unbounded, leading to OOD issues and suboptimal performance.
- **GAVE-VA**: omits both the value function from Section 3.4 and the action exploration mechanism detailed in Section 3.3.
- **DT**: removes all designed modules related to GAVE, including those described in Sections 3.4, 3.3, and Section 3.2. Consequently, this configuration aligns with the pure DT framework [6] with $S = \sum_i x_i v_i$ for RTG modeling.

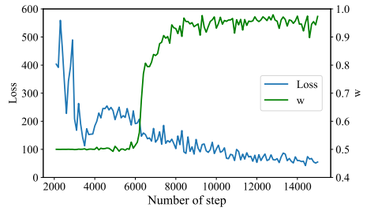
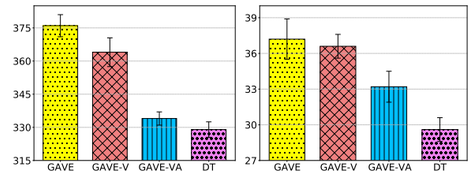


Figure 2: Parameter analysis of w on AuctionNet.



(a) AuctionNet. (b) AuctionNet-Sparse.

Figure 3: Ablation study with 100% budget.

Figure 3 presents the evaluation results. The findings indicate that: (i) Aligning the optimization objectives with the evaluation metrics using the score-based RTG modeling allows GAVE-VA to outperform DT, demonstrating the importance of objective alignment in training; (ii) Incorporating the action exploration mechanism and RTG-based evaluation in GAVE-V enables the model to discover potential strategies beyond the offline dataset and evaluate their significance for a stability-preserving update process, thereby achieving better performance than GAVE-VA; and (iii) Fully integrating the value function to guide action explorations within GAVE leverages potential optimal strategies, further alleviating OOD issues and enhancing overall performance.

5 Online Application

We evaluate GAVE's effectiveness through A/B tests in two industrial live bidding scenarios: Nobid [4] (maximizing conversions within daily budget) and Costcap (maximizing conversions with CPA/ROI limits). The experimental settings are detailed below.

- **State**: 20-step sequence with features including budget, CPA limit, predictions, traffic/cost speeds, time-phased budget, remaining time, and window-averaged bid coefficient.

[4]https://support.google.com/google-ads/answer/7381968?hl=en

Table 3: Alignment analysis on AuctionNet-Sparse with 100% budget. "Train" denotes applying the score to RTG modeling, while "Eval" denotes its utilization as the evaluation metric.

Train \ Eval	S_1	S_2	S_3
S_1	41.4	33.0	23.6
S_2	39.9	37.2	33.3
S_3	39.1	36.8	33.5

评估指标保持一致，如公式（15）所示。本节考察 GAVE 在不同 RTG 和评估指标配置下的表现，以回答 RQ2。具体而言，我们考虑以下三种评估指标：

$$\begin{cases} S_1 = \sum_i x_i v_i, \\ S_2 = \min\left\{\left(\frac{C}{CPA}\right)^2, 1\right\} \cdot \sum_i x_i v_i, \\ S_3 = \min\left\{\left(\frac{C}{CPA}\right)^5, 1\right\} \cdot \sum_i x_i v_i, \end{cases} \tag{25}$$

其中，S_1仅考虑所获得的广告展示总价值，表示对 CPA 条件的限制具有弹性的业务场景。S_2是本文的优化目标和评估评分，它对 CPA 约束添加了惩罚。S_3进一步增大了 CPA 的惩罚系数，以表示对 CPA 条件有严格限制的业务场景。这些指标既可用于训练期间的回报到当前（Return-To-Go, RTG）建模，也可作为测试期间的评估标准。结果见表 3。

从表 3 可以看出，当训练所用的 RTG 与评估指标采用同一函数时，GAVE 始终取得最高性能。这一发现凸显了通过我们基于评分的 RTG 方法，使训练目标与特定评估指标保持一致的重要性。

4.4 参数分析（RQ3）

为回答 RQ3，我们对权重 w_t进行参数分析，如图 1（b.1）所示，以阐明
$\tilde{a}_t$与 a_t在训练过程中的差异。具体而言，图 2 展示了平均总损失 L_o和权重 w_t随训练步数的变化，使我们能够监测$\tilde{r}_{t+1}$与$\hat{r}_{t+1}$之间的差距。更大的 w_t意味着$\tilde{r}_{t+1}$比$\hat{r}_{t+1}$具有更大的影响力，从而表明$\tilde{a}_t$优于 a_t。这一结果凸显了价值函数在引导动作探索方面的有效性。

从图 2 可以明显看出，随着训练推进，参数 w_t从约 0.5 上升，并稳定在高于 0.5 的水平。其稳定水平同时受到数据集分布和模型超参数的影响。这一趋势证实了可学习的价值函数在引导动作探索方面的有效性。在价值函数的引导下，模型持续探索动作$\tilde{a}$，这些动作具有更高的 RTG 值$\tilde{r}_{t+1}$，且接近估计的最优值$\hat{v}_{t+1}$。这种方法有助于学习潜在最优策略，同时缓解分布外（Out-of-Distribution, OOD）问题。

4.5 消融研究（RQ4）

为进一步阐明 GAVE 中各模块的贡献并回答 RQ4，我们开展了消融研究，评估以下 GAVE 修改版本：

• GAVE-V：移除第 3.4. 节所述的可学习的价值函数。在此配置中，损失函数 L_v和 L_e被

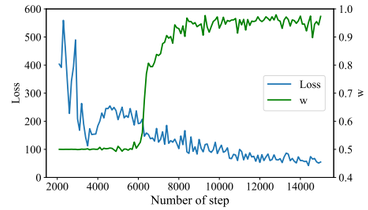
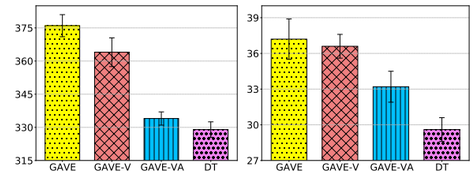


Figure 2: Parameter analysis of w on AuctionNet.



（b）AuctionNet-Sparse。
AuctionNet。

图 3：使用 100% 预

鉮的消融研究。gradient-frozen version of $\hat{r}_{t+1}$. However, without the value function, the update direction of $\tilde{r}_{t+1}$ becomes unbounded, leading to OOD issues and suboptimal performance.
- **GAVE-VA**: omits both the value function from Section 3.4 and the action exploration mechanism detailed in Section 3.3.
- **DT**: removes all designed modules related to GAVE, including those described in Sections 3.4, 3.3, and Section 3.2. Consequently, this configuration aligns with the pure DT framework [6] with $S = \sum_i x_i v_i$ for RTG modeling.

Figure 3 presents the evaluation results. The findings indicate that: (i) Aligning the optimization objectives with the evaluation metrics using the score-based RTG modeling allows GAVE-VA to outperform DT, demonstrating the importance of objective alignment in training; (ii) Incorporating the action exploration mechanism and RTG-based evaluation in GAVE-V enables the model to discover potential strategies beyond the offline dataset and evaluate their significance for a stability-preserving update process, thereby achieving better performance than GAVE-VA; and (iii) Fully integrating the value function to guide action explorations within GAVE leverages potential optimal strategies, further alleviating OOD issues and enhancing overall performance.

5 在线应用

我们在两个工业界实际出价场景中通过 A/B 测试评估 GAVE 的有效性：Nobid [4]（在每日预算内最大化转化数）和 Costcap（在 CPA/ROI 限制下最大化转化数）。实验设置详述如下。

• 状态：长度为 20 步的序列，特征包括预算、CPA 上限、预测值、流量 / 成本速率、分时预算、剩余时间以及窗口平均出价系数。

[4]https://support.google.com/google-ads/answer/7381968?hl=en

Table 4: Online A/B test.

	Cost	Conversion	Target cost	CPA valid ratio
Nobid	+0.8%	+8.0%	+3.2%	/
Costcap	+2.0%	+3.6%	+2.2%	+1.9%

- **Action**: To stabilize bidding results, the bid coefficient λ is determined based on a windowed-average of the preceding two hours containing E time steps, $\lambda_t = a_t + \frac{1}{|E|}\sum_{t'=t-E}^{t-1}\lambda_{t'}$, where a_t is GAVE's output action at time step t.
- **Return-To-Go (RTG)**: Given the sparsity of real conversions, we utilize the expected total conversions, $\sum_i pcvr_i$ during training, where $pcvr_i$ is the predicted conversion rate for winning traffic i. During inference, the RTG for the entire sequence is set to the campaign's total expected conversions from the previous day.

We compare GAVE with the offline reinforcement learning algorithm IQL [21], currently in production. Evaluation metrics include cost, conversions, target cost, and CPA valid ratio, with bidding strategies focused on maximizing conversions within budget and CPA constraints. To account for varying campaign targets, target cost serves as a value-weighted conversion measure. For Costcap campaigns, the conversion value equals the CPA limit, while Nobid campaigns use the average real CPA from total traffic. A Costcap campaign is CPA valid if its CPA remains below the limit, assessed solely for Costcap campaigns. Our five-day online A/B testing allocates 25% of each campaign's budget and traffic to the baseline bidding model and GAVE, with results summarized in Table 4.

For both Nobid and Costcap, GAVE improves cost and conversions. In Nobid campaigns, GAVE achieves a 0.8% increase in cost, 8.0% in conversions, and 3.2% in target cost. For Costcap campaigns, advertising revenue and advertiser value rise alongside a significant improvement in CPA validity, with +2.0% cost, +3.6% conversions, +2.2% target cost, and +1.9% valid CPA ratio.

6 Related Works

This section provides a brief review of relevant research topics, i.e., offline reinforcement learning, and auto-bidding.

6.1 Offline Reinforcement Learning and Decision Transformers

Reinforcement Learning (RL) trains decision-making agents through environment interactions, evolving from foundational works [37, 38] to advanced methods like policy gradient [45], deep Q-learning [39], and deterministic policy optimization [43]. While effective, their reliance on frequent online interactions poses risks and costs in real-world applications [19]. Offline RL addresses this by learning policies from static datasets, with methods like BCQ [9], CQL [22], and IQL [21] offering robust solutions for stable continuous control, mitigating value overestimation, and reducing distributional shifts. However, their dependence on Markov Decision Processes (MDPs) limits access to prior observations and modeling long-range dependencies, critical in sequential tasks with strong temporal patterns. Decision Transformers (DT) [6] overcome these limitations by reframing RL as sequential modeling, leveraging transformer architectures to capture historical patterns and long-term dependencies, achieving state-of-the-art offline RL performance. Extensions like CDT [33] further enable zero-shot constraint adaptation, balancing safety and performance without online fine-tuning. Building on DT, we propose an auto-bidding framework that aligns with DT's strengths by modeling bid adjustments as trajectory-based sequence generation, effectively capturing intricate temporal correlations in high-stakes, data-sensitive environments.

6.2 Auto-Bidding at Online Advertising Platforms

Auto-bidding plays a vital role in managing large-scale ad auctions by automatically optimizing bids per impression to meet advertisers' goals. Early methods like PID [7] and OnlineLP [55] focus on rule-based approaches, using feedback loops and stochastic programming to address budget pacing and bid optimization, though they depend on simplified assumptions. To tackle the complexity of modern advertising ecosystems, RL-based solutions such as RLB [5], USCB [17], MAAB [50], and SORL [40] enable adaptive decision-making with capabilities for handling high-dimensional states and multi-agent coordination. However, due to the risks of real-time bidding, offline RL methods like BCQ [9], CQL [22], and IQL [21] have gained prominence for leveraging historical data. These approaches, while effective under MDP frameworks, struggle with modeling long-range temporal dependencies crucial for sequential bid optimization. Generative sequential modeling methods such as DiffBid [16] and GAS [27] address these challenges using diffusion models and transformers with MCTS for improved bid trajectory generation. In this work, we propose a score-based RTG, action exploration mechanisms, and a learnable value function framework to align optimization objectives with evaluation metrics, enhance action exploration, and learn optimal strategies for improving bidding performance with Decision Transformer.

7 Conclusion

In this work, we propose GAVE to enhance DTs for offline generative auto-bidding through value-guided explorations. To accommodate complex advertising objectives, we design a customizable score-based RTG mechanism, enabling adaptive modeling of diverse optimization objectives to align with different evaluation metrics. Moreover, we integrate an action exploration mechanism with an RTG-based evaluation method to explore actions outside the offline dataset while ensuring a stability-preserving update process. To further guide the exploration and mitigate OOD risks, we employ a learnable value function to anchor RTG updates to distributionally plausible regions while allowing controlled extrapolation for strategy improvement. Extensive experiments, online deployments and NeurIPS competition [1] demonstrate the effectiveness of our GAVE framework in enhancing the adaptability and performance of auto-bidding strategies, providing a versatile solution for optimizing digital advertising campaigns in dynamic environments.

Acknowledgments

This research was partially supported by Kuaishou, Research Impact Fund (No.R1015-23), and Collaborative Research Fund (No.C1043-24GF).

Table 4: Online A/B test.

	Cost	Conversion	Target cost	CPA valid ratio
Nobid	+0.8%	+8.0%	+3.2%	/
Costcap	+2.0%	+3.6%	+2.2%	+1.9%

• 动作：为稳定竞价结果，出价系数 λ取决于

此前两小时的窗口平均值。

该窗口包含 E个时间步， $\lambda_t = a_t + \frac{1}{|E|}\sum_{t'=t-E}^{t-1}\lambda_{t'}$，其中， a_t 为 GAVE 在时间步 t输出的动作。• 回报到当前（Return-To-Go，RTG）：鉴于实际转化的稀疏性，我们在训练期间采用预期总转化量 $\sum_i pcvr_i$，其中， $pcvr_i$是竞得流量 i的预测转化率。在推理期间，整个序列的 RTG 被设定为该广告活动前一天的预期总转化量。

我们将 GAVE 与目前已在生产环境中运行的离线强化学习（Reinforcement Learning, RL）算法 IQL [21] 进行比较。评估指标包括成本、转化量、目标成本和 CPA 达标率，出价策略旨在预算和 CPA 约束内最大化转化量。为考虑不同广告活动的目标差异，目标成本被用作按价值加权的转化量指标。对于 Costcap 广告活动，转化价值等于 CPA 上限；而 Nobid 广告活动则采用总流量的平均实际 CPA。若某个 Costcap 广告活动的 CPA 保持在上限以下，则视为 CPA 达标，且仅对 Costcap 广告活动进行此项评估。我们的线上 A/B 测试持续五天，为基线出价模型和 GAVE 分配每个广告活动 25% 的预算和流量，结果汇总于表 4。

对于 Nobid 和 Costcap 两类广告活动，GAVE 均提高了成本和转化量。在 Nobid 广告活动中，GAVE 使成本增加了 0.8%，转化量增加了 8.0%，目标成本增加了 3.2%。对于 Costcap 广告活动，广告收入和广告主价值均有所增长，同时 CPA 达标情况显著改善：成本 +2.0%，转化量 +3.6%，目标成本 +2.2%，CPA 达标率 +1.9%。

6 相关工作

本节简要回顾相关研究主题，即离线 RL 和自动出价。

6.1 离线强化学习与决策变换器（Decision Transformer, DT）

RL 通过与环境交互来训练决策智能体，其发展从基础性研究 [37, 38]，逐步演进到策略梯度 [45]、深度 Q 学习 [39] 和确定性策略优化 [43] 等先进方法。尽管这些方法有效，但其对频繁在线交互的依赖会在实际应用中带来风险和成本 [19]。离线 RL 通过从静态数据集中学习策略来应对这一问题，BCQ [9]、CQL [22] 和 IQL [21] 等方法为实现稳定的连续控制、缓解价值高估和减少分布偏移提供了稳健的解决方案。然而，这些方法依赖马尔可夫决策过程（Markov Decision Process, MDP），因而限制了对先前观测的利用以及对长程依赖的建模，而这两点对于具有明显时间规律的序列任务至关重要。DT [6] 将 RL 重新表述为序列建模问题，利用变换器架构捕捉历史模式和长期依赖，从而克服这些局限，并取得了最先进的离线 RL 性能。CDT [33] 等扩展方法进一步实现了零样本约束适应，并在以下方面取得平衡：安全性与性能，而无须在线微调。在决策变换器（Decision Transformer, DT）的基础上，我们提出一种自动出价框架，将出价调整建模为基于轨迹的序列生成，以发挥 DT 的优势，有效捕捉高风险、数据敏感环境中的复杂时间相关性。

6.2 在线广告平台的自动出价

自动出价通过自动优化每次广告展示的出价来满足广告主目标，在管理大规模广告拍卖方面发挥着重要作用。PID [7] 和 OnlineLP [55] 等早期方法侧重于基于规则的方式，使用反馈环路和随机规划处理预算节奏控制与出价优化，但它们依赖简化假设。为应对现代广告生态系统的复杂性，基于强化学习（Reinforcement Learning, RL）的方案，如 RLB [5]、USCB [17]、MAAB [50] 和 SORL [40]，支持自适应决策，并具备处理高维状态与多智能体协调的能力。然而，由于实时出价存在风险，BCQ [9]、CQL [22] 和 IQL [21] 等利用历史数据的离线 RL 方法受到越来越多的重视。这些方法虽然在马尔可夫决策过程（Markov Decision Process, MDP）框架下有效，但难以建模对序列出价优化至关重要的长程时间依赖关系。DiffBid [16] 和 GAS [27] 等生成式序列建模方法通过使用扩散模型和结合 MCTS 的 Transformer，应对这些挑战并改善出价轨迹生成。在本工作中，我们提出基于评分的回报到当前（Return-To-Go, RTG）、动作探索机制，以及可学习的价值函数框架，以将优化目标与评估指标对齐、增强动作探索，并学习最优策略，从而利用 DT 改善出价性能。

7 结论

在本工作中，我们提出 GAVE，通过价值引导探索增强用于离线生成式自动出价的 DT。为适应复杂的广告目标，我们设计了可定制的基于评分的 RTG 机制，支持对多种优化目标进行自适应建模，以与不同评估指标对齐。此外，我们将动作探索机制与基于 RTG 的评估方法相结合，在探索离线数据集之外的动作时，确保更新过程保持稳定。为进一步引导探索并缓解分布外（Out-of-Distribution, OOD）风险，我们采用可学习的价值函数，将 RTG 更新锚定于在分布上合理的区域，同时允许受控外推以改进策略。大量实验、在线部署以及 NeurIPS 竞赛 [1] 证明了 GAVE 框架在增强自动出价策略适应性与性能方面的有效性，为在动态环境中优化数字广告活动提供了一种通用解决方案。

致谢

本研究获得快手、研究影响基金（编号 R1015-23）以及协作研究基金（编号 C1043-24GF）的部分资助。

References

[1] Gagan Aggarwal, Ashwinkumar Badanidiyuru, Santiago R Balseiro, Kshipra Bhawalkar, Yuan Deng, Zhe Feng, Gagan Goel, Christopher Liaw, Haihao Lu, Mohammad Mahdian, et al. 2024. Auto-bidding and auctions in online advertising: A survey. *ACM SIGecom Exchanges* (2024), 159–183.

[2] Gagan Aggarwal, Shan Muthukrishnan, Dávid Pál, and Martin Pál. 2009. General auction mechanism for search advertising. In *Proc. of WWW*. 241–250.

[3] Raju Balakrishnan and Rushi P Bhatt. 2014. Real-time bid optimization for group-buying ads. *ACM Transactions on Intelligent Systems and Technology (TIST)* (2014), 1–21.

[4] Nikolay Borissov, Dirk Neumann, and Christof Weinhardt. 2010. Automated bidding in computational markets: an application in market-based allocation of computing services. *Autonomous Agents and Multi-Agent Systems* (2010), 115–142.

[5] Han Cai, Kan Ren, Weinan Zhang, Kleanthis Malialis, Jun Wang, Yong Yu, and Defeng Guo. 2017. Real-time bidding by reinforcement learning in display advertising. In *Proc. of WSDM*. 661–670.

[6] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. 2021. Decision transformer: Reinforcement learning via sequence modeling. *Proc. of NeurIPS* (2021), 15084–15097.

[7] Ye Chen, Pavel Berkhin, Bo Anderson, and Nikhil R Devanur. 2011. Real-time bidding algorithms for performance-based display ad allocation. In *Proc. of KDD*. 1307–1315.

[8] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. 2007. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American economic review* (2007), 242–259.

[9] Scott Fujimoto, David Meger, and Doina Precup. 2019. Off-policy deep reinforcement learning without exploration. In *Proc. of ICML*. 2052–2062.

[10] Jingtong Gao, Bo Chen, Xiangyu Zhao, Weiwen Liu, Xiangyang Li, Yichao Wang, Wanyu Wang, Huifeng Guo, and Ruiming Tang. [n. d.]. LLM4Rerank: LLM-based Auto-Reranking Framework for Recommendations. In *Proc. of WWW*.

[11] Jingtong Gao, Bo Chen, Menghui Zhu, Xiangyu Zhao, Xiaopeng Li, Yuhao Wang, Yichao Wang, Huifeng Guo, and Ruiming Tang. 2024. HierRec: Scenario-Aware Hierarchical Modeling for Multi-scenario Recommendations. In *Proc. of CIKM*. 653–662.

[12] Jingtong Gao, Zhaocheng Du, Xiaopeng Li, Yichao Wang, Xiangyang Li, Huifeng Guo, Ruiming Tang, and Xiangyu Zhao. 2025. SampleLLM: Optimizing Tabular Data Synthesis in Recommendations. *arXiv preprint arXiv:2501.16125* (2025).

[13] Jingtong Gao, Xiangyu Zhao, Bo Chen, Fan Yan, Huifeng Guo, and Ruiming Tang. 2023. AutoTransfer: Instance transfer for cross-domain recommendations. In *Proc. of SIGIR*. 1478–1487.

[14] Jingtong Gao, Xiangyu Zhao, Muyang Li, Minghao Zhao, Runze Wu, Ruocheng Guo, Yiding Liu, and Dawei Yin. 2024. SMLP4Rec: an Efficient all-MLP architecture for sequential recommendations. *ACM Transactions on Information Systems* (2024), 1–23.

[15] Chaojie Guo, Russell G Thompson, Greg Foliente, and Xiaoshuai Peng. 2021. Reinforcement learning enabled dynamic bidding strategy for instant delivery trading. *Computers & Industrial Engineering* (2021), 107596.

[16] Jiayan Guo, Yusen Huo, Zhilin Zhang, Tianyu Wang, Chuan Yu, Jian Xu, Bo Zheng, and Yan Zhang. 2024. AIGB: Generative Auto-bidding via Conditional Diffusion Modeling. In *Proc. of KDD*. 5038–5049.

[17] Yue He, Xiujun Chen, Di Wu, Junwei Pan, Qing Tan, Chuan Yu, Jian Xu, and Xiaoqiang Zhu. 2021. A unified solution to constrained bidding in online display advertising. In *Proc. of KDD*. 2993–3001.

[18] Arti Jha, Harshit Jain, Parikshit Sharma, Yashvardhan Sharma, and Kamlesh Tiwari. 2024. Optimizing Real-Time Bidding Strategies: An Experimental Analysis of Reinforcement Learning and Machine Learning Techniques. *Procedia Computer Science* (2024), 2017–2026.

[19] Haruka Kiyohara, Kosuke Kawakami, and Yuta Saito. 2021. Accelerating offline reinforcement learning application in real-time bidding and recommendation: Potential use of simulation. *arXiv preprint arXiv:2109.08331* (2021).

[20] Ilya Kostrikov, Rob Fergus, Jonathan Tompson, and Ofir Nachum. 2021. Offline reinforcement learning with fisher divergence critic regularization. In *Proc. of ICML*. 5774–5783.

[21] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. [n. d.]. Offline Reinforcement Learning with Implicit Q-Learning. In *Proc. of ICLR*.

[22] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. 2020. Conservative q-learning for offline reinforcement learning. *Proc. of NeurIPS* (2020), 1179–1191.

[23] Pranjal Kumar. 2024. Large language models (LLMs): survey, technical frameworks, and future challenges. *Artificial Intelligence Review* (2024), 260.

[24] Chengxi Li, Yejing Wang, Qidong Liu, Xiangyu Zhao, Wanyu Wang, Yiqi Wang, Lixin Zou, Wenqi Fan, and Qing Li. 2023. STRec: Sparse transformer for sequential recommendations. In *Proc. of RecSys*. 101–111.

[25] Xinhang Li, Zhaopeng Qiu, Xiangyu Zhao, Zihao Wang, Yong Zhang, Chunxiao Xing, and Xian Wu. 2022. Gromov-wasserstein guided representation learning for cross-domain recommendation. In *Proc. of CIKM*. 1199–1208.

[26] Xiaopeng Li, Fan Yan, Xiangyu Zhao, Yichao Wang, Bo Chen, Huifeng Guo, and Ruiming Tang. 2023. Hamur: Hyper adapter for multi-domain recommendation. In *Proc. of CIKM*. 1268–1277.

[27] Yewen Li, Shuai Mao, Jingtong Gao, Nan Jiang, Yunjian Xu, Qingpeng Cai, Fei Pan, Peng Jiang, and Bo An. 2024. GAS: Generative Auto-bidding with Post-training Search. *arXiv preprint arXiv:2412.17018* (2024).

[28] Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. 2022. A survey of transformers. *AI open* (2022), 111–132.

[29] Weilin Lin, Xiangyu Zhao, Yejing Wang, Yuanshao Zhu, and Wanyu Wang. 2023. Autodenoise: Automatic data instance denoising for recommendations. In *Proc. of WWW*. 1003–1011.

[30] Haochen Liu, Da Tang, Ji Yang, Xiangyu Zhao, Hui Liu, Jiliang Tang, and Youlong Cheng. 2022. Rating distribution calibration for selection bias mitigation in recommendations. In *Proc. of WWW*. 2048–2057.

[31] Haochen Liu, Xiangyu Zhao, Chong Wang, Xiaobing Liu, and Jiliang Tang. 2020. Automated embedding size search in deep recommender systems. In *Proc. of SIGIR*. 2307–2316.

[32] Shuchang Liu, Qingpeng Cai, Bowen Sun, Yuhao Wang, Ji Jiang, Dong Zheng, Peng Jiang, Kun Gai, Xiangyu Zhao, and Yongfeng Zhang. 2023. Exploration and regularization of the latent action space in recommendation. In *Proc. of WWW*. 833–844.

[33] Zuxin Liu, Zijian Guo, Yihang Yao, Zhepeng Cen, Wenhao Yu, Tingnan Zhang, and Ding Zhao. 2023. Constrained decision transformer for offline safe reinforcement learning. In *Proc. of ICML*. 21611–21630.

[34] Ziru Liu, Shuchang Liu, Zijian Zhang, Qingpeng Cai, Xiangyu Zhao, Kesen Zhao, Lantao Hu, Peng Jiang, and Kun Gai. 2024. Sequential recommendation for optimizing both immediate feedback and long-term retention. In *Proc. of SIGIR*. 1872–1882.

[35] Ziru Liu, Jiejie Tian, Qingpeng Cai, Xiangyu Zhao, Jingtong Gao, Shuchang Liu, Dayou Chen, Tonghao He, Dong Zheng, Peng Jiang, et al. 2023. Multi-task recommendations with reinforcement learning. In *Proc. of WWW*. 1273–1282.

[36] I Loshchilov. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).

[37] Donald Michie and Roger A Chambers. 1968. BOXES: An experiment in adaptive control. *Machine intelligence* (1968), 137–152.

[38] Marvin Minsky. 1961. Steps toward artificial intelligence. *Proceedings of the IRE* (1961), 8–30.

[39] Volodymyr Mnih. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).

[40] Zhiyu Mou, Yusen Huo, Rongquan Bai, Mingzhou Xie, Chuan Yu, Jian Xu, and Bo Zheng. 2022. Sustainable online reinforcement learning for auto-bidding. *Proc. of NeurIPS* (2022), 2651–2663.

[41] Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. 2017. Bridging the gap between value and policy based reinforcement learning. *Proc. of NeurIPS* (2017).

[42] Amin Sayedi. 2018. Real-time bidding in online display advertising. *Marketing Science* (2018), 553–568.

[43] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. 2014. Deterministic policy gradient algorithms. In *Proc. of ICML*. 387–395.

[44] Kefan Su, Yusen Huo, Zhilin Zhang, Shuai Dou, Chuan Yu, Jian Xu, Zongqing Lu, and Bo Zheng. 2024. AuctionNet: A Novel Benchmark for Decision-Making in Large-Scale Games. In *Proc. of NeurIPS*.

[45] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. *Proc. of NeurIPS* (1999).

[46] Csaba Szepesvári and Michael L Littman. 1999. A unified analysis of value-function-based reinforcement-learning algorithms. *Neural computation* (1999), 2017–2060.

[47] X Wang, S Wang, X Liang, D Zhao, J Huang, X Xu, B Dai, and Q Miao. 2022. Deep Reinforcement Learning: A Survey. *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[48] Yuhao Wang, Ha Tsz Lam, Yi Wong, Ziru Liu, Xiangyu Zhao, Yichao Wang, Bo Chen, Huifeng Guo, and Ruiming Tang. 2023. Multi-task deep recommender systems: A survey. *arXiv preprint arXiv:2302.03525* (2023).

[49] Yuhao Wang, Xiangyu Zhao, Bo Chen, Qidong Liu, Huifeng Guo, Huanshuo Liu, Yichao Wang, Rui Zhang, and Ruiming Tang. 2023. PLATE: A prompt-enhanced paradigm for multi-scenario recommendations. In *Proc. of SIGIR*. 1498–1507.

[50] Chao Wen, Miao Xu, Zhilin Zhang, Zhenzhe Zheng, Yuhui Wang, Xiangyu Liu, Yu Rong, Dong Xie, Xiaoyang Tan, Chuan Yu, et al. 2022. A cooperative-competitive multi-agent framework for auto-bidding in online advertising. In *Proc. of WSDM*. 1129–1139.

[51] Di Wu, Xiujun Chen, Xun Yang, Hao Wang, Qing Tan, Xiaoxun Zhang, Jian Xu, and Kun Gai. 2018. Budget constrained bidding by model-free reinforcement learning in display advertising. In *Proc. of CIKM*. 1443–1451.

[52] Yueh-Hua Wu, Xiaolong Wang, and Masashi Hamaya. 2024. Elastic decision transformer. *Proc. of NeurIPS* (2024).

SIGIR '25，2025 年 7 月 13 – 18 日，意大利帕多瓦 Gao 等

References

[1] Gagan Aggarwal, Ashwinkumar Badanidiyuru, Santiago R Balseiro, Kshipra Bhawalkar, Yuan Deng, Zhe Feng, Gagan Goel, Christopher Liaw, Haihao Lu, Mohammad Mahdian, et al. 2024. Auto-bidding and auctions in online advertising: A survey. *ACM SIGecom Exchanges* (2024), 159–183.

[2] Gagan Aggarwal, Shan Muthukrishnan, Dávid Pál, and Martin Pál. 2009. General auction mechanism for search advertising. In *Proc. of WWW*. 241–250.

[3] Raju Balakrishnan and Rushi P Bhatt. 2014. Real-time bid optimization for group-buying ads. *ACM Transactions on Intelligent Systems and Technology (TIST)* (2014), 1–21.

[4] Nikolay Borissov, Dirk Neumann, and Christof Weinhardt. 2010. Automated bidding in computational markets: an application in market-based allocation of computing services. *Autonomous Agents and Multi-Agent Systems* (2010), 115–142.

[5] Han Cai, Kan Ren, Weinan Zhang, Kleanthis Malialis, Jun Wang, Yong Yu, and Defeng Guo. 2017. Real-time bidding by reinforcement learning in display advertising. In *Proc. of WSDM*. 661–670.

[6] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. 2021. Decision transformer: Reinforcement learning via sequence modeling. *Proc. of NeurIPS* (2021), 15084–15097.

[7] Ye Chen, Pavel Berkhin, Bo Anderson, and Nikhil R Devanur. 2011. Real-time bidding algorithms for performance-based display ad allocation. In *Proc. of KDD*. 1307–1315.

[8] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. 2007. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American economic review* (2007), 242–259.

[9] Scott Fujimoto, David Meger, and Doina Precup. 2019. Off-policy deep reinforcement learning without exploration. In *Proc. of ICML*. 2052–2062.

[10] Jingtong Gao, Bo Chen, Xiangyu Zhao, Weiwen Liu, Xiangyang Li, Yichao Wang, Wanyu Wang, Huifeng Guo, and Ruiming Tang. [n. d.]. LLM4Rerank: LLM-based Auto-Reranking Framework for Recommendations. In *Proc. of WWW*.

[11] Jingtong Gao, Bo Chen, Menghui Zhu, Xiangyu Zhao, Xiaopeng Li, Yuhao Wang, Yichao Wang, Huifeng Guo, and Ruiming Tang. 2024. HierRec: Scenario-Aware Hierarchical Modeling for Multi-scenario Recommendations. In *Proc. of CIKM*. 653–662.

[12] Jingtong Gao, Zhaocheng Du, Xiaopeng Li, Yichao Wang, Xiangyang Li, Huifeng Guo, Ruiming Tang, and Xiangyu Zhao. 2025. SampleLLM: Optimizing Tabular Data Synthesis in Recommendations. *arXiv preprint arXiv:2501.16125* (2025).

[13] Jingtong Gao, Xiangyu Zhao, Bo Chen, Fan Yan, Huifeng Guo, and Ruiming Tang. 2023. AutoTransfer: Instance transfer for cross-domain recommendations. In *Proc. of SIGIR*. 1478–1487.

[14] Jingtong Gao, Xiangyu Zhao, Muyang Li, Minghao Zhao, Runze Wu, Ruocheng Guo, Yiding Liu, and Dawei Yin. 2024. SMLP4Rec: an Efficient all-MLP architecture for sequential recommendations. *ACM Transactions on Information Systems* (2024), 1–23.

[15] Chaojie Guo, Russell G Thompson, Greg Foliente, and Xiaoshuai Peng. 2021. Reinforcement learning enabled dynamic bidding strategy for instant delivery trading. *Computers & Industrial Engineering* (2021), 107596.

[16] Jiayan Guo, Yusen Huo, Zhilin Zhang, Tianyu Wang, Chuan Yu, Jian Xu, Bo Zheng, and Yan Zhang. 2024. AIGB: Generative Auto-bidding via Conditional Diffusion Modeling. In *Proc. of KDD*. 5038–5049.

[17] Yue He, Xiujun Chen, Di Wu, Junwei Pan, Qing Tan, Chuan Yu, Jian Xu, and Xiaoqiang Zhu. 2021. A unified solution to constrained bidding in online display advertising. In *Proc. of KDD*. 2993–3001.

[18] Arti Jha, Harshit Jain, Parikshit Sharma, Yashvardhan Sharma, and Kamlesh Tiwari. 2024. Optimizing Real-Time Bidding Strategies: An Experimental Analysis of Reinforcement Learning and Machine Learning Techniques. *Procedia Computer Science* (2024), 2017–2026.

[19] Haruka Kiyohara, Kosuke Kawakami, and Yuta Saito. 2021. Accelerating offline reinforcement learning application in real-time bidding and recommendation: Potential use of simulation. *arXiv preprint arXiv:2109.08331* (2021).

[20] Ilya Kostrikov, Rob Fergus, Jonathan Tompson, and Ofir Nachum. 2021. Offline reinforcement learning with fisher divergence critic regularization. In *Proc. of ICML*. 5774–5783.

[21] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. [n. d.]. Offline Reinforcement Learning with Implicit Q-Learning. In *Proc. of ICLR*.

[22] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. 2020. Conservative q-learning for offline reinforcement learning. *Proc. of NeurIPS* (2020), 1179–1191.

[23] Pranjal Kumar. 2024. Large language models (LLMs): survey, technical frameworks, and future challenges. *Artificial Intelligence Review* (2024), 260.

[24] Chengxi Li, Yejing Wang, Qidong Liu, Xiangyu Zhao, Wanyu Wang, Yiqi Wang, Lixin Zou, Wenqi Fan, and Qing Li. 2023. STRec: Sparse transformer for sequential recommendations. In *Proc. of RecSys*. 101–111.

[25] Xinhang Li, Zhaopeng Qiu, Xiangyu Zhao, Zihao Wang, Yong Zhang, Chunxiao Xing, and Xian Wu. 2022. Gromov-wasserstein guided representation learning for cross-domain recommendation. In *Proc. of CIKM*. 1199–1208.

[26] Xiaopeng Li, Fan Yan, Xiangyu Zhao, Yichao Wang, Bo Chen, Huifeng Guo, and Ruiming Tang. 2023. Hamur: Hyper adapter for multi-domain recommendation. In *Proc. of CIKM*. 1268–1277.

[27] Yewen Li, Shuai Mao, Jingtong Gao, Nan Jiang, Yunjian Xu, Qingpeng Cai, Fei Pan, Peng Jiang, and Bo An. 2024. GAS: Generative Auto-bidding with Post-training Search. *arXiv preprint arXiv:2412.17018* (2024).

[28] Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. 2022. A survey of transformers. *AI open* (2022), 111–132.

[29] Weilin Lin, Xiangyu Zhao, Yejing Wang, Yuanshao Zhu, and Wanyu Wang. 2023. Autodenoise: Automatic data instance denoising for recommendations. In *Proc. of WWW*. 1003–1011.

[30] Haochen Liu, Da Tang, Ji Yang, Xiangyu Zhao, Hui Liu, Jiliang Tang, and Youlong Cheng. 2022. Rating distribution calibration for selection bias mitigation in recommendations. In *Proc. of WWW*. 2048–2057.

[31] Haochen Liu, Xiangyu Zhao, Chong Wang, Xiaobing Liu, and Jiliang Tang. 2020. Automated embedding size search in deep recommender systems. In *Proc. of SIGIR*. 2307–2316.

[32] Shuchang Liu, Qingpeng Cai, Bowen Sun, Yuhao Wang, Ji Jiang, Dong Zheng, Peng Jiang, Kun Gai, Xiangyu Zhao, and Yongfeng Zhang. 2023. Exploration and regularization of the latent action space in recommendation. In *Proc. of WWW*. 833–844.

[33] Zuxin Liu, Zijian Guo, Yihang Yao, Zhepeng Cen, Wenhao Yu, Tingnan Zhang, and Ding Zhao. 2023. Constrained decision transformer for offline safe reinforcement learning. In *Proc. of ICML*. 21611–21630.

[34] Ziru Liu, Shuchang Liu, Zijian Zhang, Qingpeng Cai, Xiangyu Zhao, Kesen Zhao, Lantao Hu, Peng Jiang, and Kun Gai. 2024. Sequential recommendation for optimizing both immediate feedback and long-term retention. In *Proc. of SIGIR*. 1872–1882.

[35] Ziru Liu, Jiejie Tian, Qingpeng Cai, Xiangyu Zhao, Jingtong Gao, Shuchang Liu, Dayou Chen, Tonghao He, Dong Zheng, Peng Jiang, et al. 2023. Multi-task recommendations with reinforcement learning. In *Proc. of WWW*. 1273–1282.

[36] I Loshchilov. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).

[37] Donald Michie and Roger A Chambers. 1968. BOXES: An experiment in adaptive control. *Machine intelligence* (1968), 137–152.

[38] Marvin Minsky. 1961. Steps toward artificial intelligence. *Proceedings of the IRE* (1961), 8–30.

[39] Volodymyr Mnih. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).

[40] Zhiyu Mou, Yusen Huo, Rongquan Bai, Mingzhou Xie, Chuan Yu, Jian Xu, and Bo Zheng. 2022. Sustainable online reinforcement learning for auto-bidding. *Proc. of NeurIPS* (2022), 2651–2663.

[41] Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. 2017. Bridging the gap between value and policy based reinforcement learning. *Proc. of NeurIPS* (2017).

[42] Amin Sayedi. 2018. Real-time bidding in online display advertising. *Marketing Science* (2018), 553–568.

[43] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. 2014. Deterministic policy gradient algorithms. In *Proc. of ICML*. 387–395.

[44] Kefan Su, Yusen Huo, Zhilin Zhang, Shuai Dou, Chuan Yu, Jian Xu, Zongqing Lu, and Bo Zheng. 2024. AuctionNet: A Novel Benchmark for Decision-Making in Large-Scale Games. In *Proc. of NeurIPS*.

[45] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. *Proc. of NeurIPS* (1999).

[46] Csaba Szepesvári and Michael L Littman. 1999. A unified analysis of value-function-based reinforcement-learning algorithms. *Neural computation* (1999), 2017–2060.

[47] X Wang, S Wang, X Liang, D Zhao, J Huang, X Xu, B Dai, and Q Miao. 2022. Deep Reinforcement Learning: A Survey. *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[48] Yuhao Wang, Ha Tsz Lam, Yi Wong, Ziru Liu, Xiangyu Zhao, Yichao Wang, Bo Chen, Huifeng Guo, and Ruiming Tang. 2023. Multi-task deep recommender systems: A survey. *arXiv preprint arXiv:2302.03525* (2023).

[49] Yuhao Wang, Xiangyu Zhao, Bo Chen, Qidong Liu, Huifeng Guo, Huanshuo Liu, Yichao Wang, Rui Zhang, and Ruiming Tang. 2023. PLATE: A prompt-enhanced paradigm for multi-scenario recommendations. In *Proc. of SIGIR*. 1498–1507.

[50] Chao Wen, Miao Xu, Zhilin Zhang, Zhenzhe Zheng, Yuhui Wang, Xiangyu Liu, Yu Rong, Dong Xie, Xiaoyang Tan, Chuan Yu, et al. 2022. A cooperative-competitive multi-agent framework for auto-bidding in online advertising. In *Proc. of WSDM*. 1129–1139.

[51] Di Wu, Xiujun Chen, Xun Yang, Hao Wang, Qing Tan, Xiaoxun Zhang, Jian Xu, and Kun Gai. 2018. Budget constrained bidding by model-free reinforcement learning in display advertising. In *Proc. of CIKM*. 1443–1451.

[52] Yueh-Hua Wu, Xiaolong Wang, and Masashi Hamaya. 2024. Elastic decision transformer. *Proc. of NeurIPS* (2024).

[53] Haoran Xu, Li Jiang, Jianxiong Li, Zhuoran Yang, Zhaoran Wang, Victor Wai Kin Chan, and Xianyuan Zhan. [n. d.]. Offline RL with No OOD Actions: In-Sample Learning via Implicit Value Regularization. In *Proc. of ICLR*.
[54] Yujian Ye, Dawei Qiu, Mingyang Sun, Dimitrios Papadaskalopoulos, and Goran Strbac. 2019. Deep reinforcement learning for strategic bidding in electricity markets. *IEEE Transactions on Smart Grid* (2019), 1343–1355.
[55] Hao Yu, Michael Neely, and Xiaohan Wei. 2017. Online convex optimization with stochastic constraints. *Proc. of NeurIPS* (2017).
[56] Congde Yuan, Mengzhuo Guo, Chaoneng Xiang, Shuangyang Wang, Guoqing Song, and Qingpeng Zhang. 2022. An actor-critic reinforcement learning model for optimal bidding in online display advertising. In *Proc. of CIKM*. 3604–3613.
[57] Shuai Yuan, Jun Wang, and Xiaoxue Zhao. 2013. Real-time bidding for online advertising: measurement and analysis. In *Proceedings of the seventh international workshop on data mining for online advertising*. 1–8.
[58] Jun Zhao, Guang Qiu, Ziyu Guan, Wei Zhao, and Xiaofei He. 2018. Deep reinforcement learning for sponsored search real-time bidding. In *Proc. of KDD*. 1021–1030.
[59] Xiangyu Zhao, Changsheng Gu, Haoshenglun Zhang, Xiwang Yang, Xiaobing Liu, Jiliang Tang, and Hui Liu. 2021. Dear: Deep reinforcement learning for online advertising impression in recommender systems. In *Proc. of AAAI*. 750–758.
[60] Xiangyu Zhao, Long Xia, Jiliang Tang, and Dawei Yin. 2019. " Deep reinforcement learning for search, recommendation, and online advertising: a survey" by Xiangyu Zhao, Long Xia, Jiliang Tang, and Dawei Yin with Martin Vesely as coordinator. *ACM sigweb newsletter* (2019), 1–15.
[61] Xiangyu Zhao, Liang Zhang, Zhuoye Ding, Long Xia, Jiliang Tang, and Dawei Yin. 2018. Recommendations with negative feedback via pairwise deep reinforcement learning. In *Proc. of KDD*. 1040–1048.
[62] Xiangyu Zhao, Xudong Zheng, Xiwang Yang, Xiaobing Liu, and Jiliang Tang. 2020. Jointly learning to recommend and advertise. In *Proc. of KDD*. 3319–3327.
[63] Qinqing Zheng, Amy Zhang, and Aditya Grover. 2022. Online decision transformer. In *Proc. of ICML*. 27042–27059.

Generative Auto-Bidding with Value-Guided Explorations SIGIR '25, July 13–18, 2025, Padua, Italy

[53] Haoran Xu, Li Jiang, Jianxiong Li, Zhuoran Yang, Zhaoran Wang, Victor Wai Kin Chan, and Xianyuan Zhan. [n. d.]. Offline RL with No OOD Actions: In-Sample Learning via Implicit Value Regularization. In *Proc. of ICLR*.
[54] Yujian Ye, Dawei Qiu, Mingyang Sun, Dimitrios Papadaskalopoulos, and Goran Strbac. 2019. Deep reinforcement learning for strategic bidding in electricity markets. *IEEE Transactions on Smart Grid* (2019), 1343–1355.
[55] Hao Yu, Michael Neely, and Xiaohan Wei. 2017. Online convex optimization with stochastic constraints. *Proc. of NeurIPS* (2017).
[56] Congde Yuan, Mengzhuo Guo, Chaoneng Xiang, Shuangyang Wang, Guoqing Song, and Qingpeng Zhang. 2022. An actor-critic reinforcement learning model for optimal bidding in online display advertising. In *Proc. of CIKM*. 3604–3613.
[57] Shuai Yuan, Jun Wang, and Xiaoxue Zhao. 2013. Real-time bidding for online advertising: measurement and analysis. In *Proceedings of the seventh international workshop on data mining for online advertising*. 1–8.
[58] Jun Zhao, Guang Qiu, Ziyu Guan, Wei Zhao, and Xiaofei He. 2018. Deep reinforcement learning for sponsored search real-time bidding. In *Proc. of KDD*. 1021–1030.
[59] Xiangyu Zhao, Changsheng Gu, Haoshenglun Zhang, Xiwang Yang, Xiaobing Liu, Jiliang Tang, and Hui Liu. 2021. Dear: Deep reinforcement learning for online advertising impression in recommender systems. In *Proc. of AAAI*. 750–758.
[60] Xiangyu Zhao, Long Xia, Jiliang Tang, and Dawei Yin. 2019. " Deep reinforcement learning for search, recommendation, and online advertising: a survey" by Xiangyu Zhao, Long Xia, Jiliang Tang, and Dawei Yin with Martin Vesely as coordinator. *ACM sigweb newsletter* (2019), 1–15.
[61] Xiangyu Zhao, Liang Zhang, Zhuoye Ding, Long Xia, Jiliang Tang, and Dawei Yin. 2018. Recommendations with negative feedback via pairwise deep reinforcement learning. In *Proc. of KDD*. 1040–1048.
[62] Xiangyu Zhao, Xudong Zheng, Xiwang Yang, Xiaobing Liu, and Jiliang Tang. 2020. Jointly learning to recommend and advertise. In *Proc. of KDD*. 3319–3327.
[63] Qinqing Zheng, Amy Zhang, and Aditya Grover. 2022. Online decision transformer. In *Proc. of ICML*. 27042–27059.