arXiv:2405.16141v4 [cs.LG] 8 Oct 2024

AIGB: Generative Auto-bidding via Diffusion Modeling

Jiayan Guo*
guojiayan@pku.edu.cn
Peking University
Alibaba Group
Beijing, China

Yusen Huo
huoyusen.huoyusen@alibaba-inc.com
Alibaba Group
Beijing, China

Zhilin Zhang
zhangzhilin.pt@alibaba-inc.com
Alibaba Group
Beijing, China

Tianyu Wang
yves.wty@@alibaba-inc.com
Alibaba Group
Beijing, China

Chuan Yu
yuchuan.yc@alibaba-inc.com
Alibaba Group
Beijing, China

Jian Xu
xiyu.xj@alibaba-inc.com
Alibaba Group
Beijing, China

Yan Zhang
zhyzhy001@pku.edu.cn
Peking University
Beijing, China

Bo Zheng†
bozheng@alibaba-inc.com
Alibaba Group
Beijing, China

ABSTRACT

Auto-bidding plays a crucial role in facilitating online advertising by automatically providing bids for advertisers. Reinforcement learning (RL) has gained popularity for auto-bidding. However, most current RL auto-bidding methods are modeled through the Markovian Decision Process (MDP), which assumes the Markovian state transition. This assumption restricts the ability to perform in long horizon scenarios and makes the model unstable when dealing with highly random online advertising environments. To tackle this issue, this paper introduces AI-Generated Bidding (AIGB), a novel paradigm for auto-bidding through generative modeling. In this paradigm, we propose DiffBid, a conditional diffusion modeling approach for bid generation. DiffBid directly models the correlation between the return and the entire trajectory, effectively avoiding error propagation across time steps in long horizons. Additionally, DiffBid offers a versatile approach for generating trajectories that maximize given targets while adhering to specific constraints. Extensive experiments conducted on the real-world dataset and online A/B test on Alibaba advertising platform demonstrate the effectiveness of DiffBid, achieving 2.81% increase in GMV and 3.36% increase in ROI.

CCS CONCEPTS

• **Information systems → Computational advertising**.

KEYWORDS

Online Advertising, Auto-bidding, Generative Learning, Diffusion Modeling

ACM Reference Format:
Jiayan Guo, Yusen Huo, Zhilin Zhang, Tianyu Wang, Chuan Yu, Jian Xu, Yan Zhang, and Bo Zheng. 2024. AIGB: Generative Auto-bidding via Diffusion Modeling. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3637528.3671526

1 INTRODUCTION

The ever-increasing digitalization of commerce has exponentially expanded the scope and importance of online advertising platforms [13, 18]. These ad platforms have become indispensable for businesses to effectively target their audience and drive sales. Traditionally, advertisers need to manually adjust bid prices to optimize overall ad performance. However, this coarse bidding process becomes impractical when dealing with trillions of impression opportunities, requiring extensive domain knowledge [10] and comprehensive information about the advertising environments.

To alleviate the burden of bid optimization for advertisers, these ad platforms provide auto-bidding services [4, 5, 12, 39]. These services automate the determination of bids for each impression opportunity by employing well-designed bidding strategies. Such strategies consider a variety of factors about advertising environments and advertisers, such as the distribution of impression opportunities, budgets, and average cost constraints [31]. Considering the dynamic nature of advertising environments, it is essential to regularly optimize the bidding strategy, typically at intervals of a few minutes, in response to changing conditions. With advertising episodes typically extending beyond 24 hours, auto-bidding can be seen as a sequential decision-making process with a long planning horizon where the bidding strategy seeks to optimize performance throughout the entire episode.

Recently, reinforcement learning (RL) techniques have been employed to optimize auto-bidding strategies through the training of

*Work is done during the internship at Alibaba Group.
†Bo Zheng is the corresponding author.

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.
KDD '24, August 25–29, 2024, Barcelona, Spain
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0490-1/24/08…$15.00
https://doi.org/10.1145/3637528.3671526

arXiv:2405.16141v4 [cs.LG] 8 Oct 2024

AIGB：通过扩散建模实现生成式自动出价

Jiayan Guo*
guojiayan@pku.edu.cn
Peking University
Alibaba Group
Beijing, China

Yusen Huo
huoyusen.huoyusen@alibaba-inc.com
Alibaba Group
Beijing, China

Zhilin Zhang
zhangzhilin.pt@alibaba-inc.com
Alibaba Group
Beijing, China

Tianyu Wang
yves.wty@@alibaba-inc.com
Alibaba Group
Beijing, China

Chuan Yu
yuchuan.yc@alibaba-inc.com
Alibaba Group
Beijing, China

Jian Xu
xiyu.xj@alibaba-inc.com
Alibaba Group
Beijing, China

Yan Zhang
zhyzhy001@pku.edu.cn
Peking University
Beijing, China

Bo Zheng†
bozheng@alibaba-inc.com
Alibaba Group
Beijing, China

摘要

自动出价通过自动为广告主提供出价，在推动在线广告发展方面发挥着关键作用。强化学习（RL）在自动出价中日益普及。然而，目前大多数基于 RL 的自动出价方法都通过马尔可夫决策过程（MDP）建模，假设状态转移具有马尔可夫性。这一假设限制了模型在长时域场景中的表现，也使模型在处理高度随机的在线广告环境时不稳定。为解决这一问题，本文提出 AI 生成式出价（AIGB），这是一种通过生成式建模实现自动出价的新范式。在这一范式下，我们提出 DiffBid，一种用于出价生成的条件扩散建模方法。DiffBid 直接建模回报与整条轨迹之间的相关性，有效避免了长时域中误差跨时间步传播。此外，DiffBid 提供了一种通用的轨迹生成方法，能够在遵循特定约束的同时最大化给定目标。在真实数据集上开展的大量实验以及阿里巴巴广告平台上的在线 A/B 测试证明了 DiffBid 的有效性，其商品交易总额（GMV）提升了 2.81%，投资回报率（ROI）提升了 3.36%。

关键词

在线广告、自动出价、生成式学习、扩散建模

ACM Reference Format:
Jiayan Guo, Yusen Huo, Zhilin Zhang, Tianyu Wang, Chuan Yu, Jian Xu, Yan Zhang, and Bo Zheng. 2024. AIGB: Generative Auto-bidding via Diffusion Modeling. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3637528.3671526

1 引言

商业数字化的不断发展，使在线广告平台的规模和重要性呈指数级增长 [13, 18]。这些广告平台已成为企业有效触达目标受众并促进销售的不可或缺的工具。传统上，广告主需要手动调整出价，以优化整体广告效果。然而，在面对数万亿次曝光机会时，这种粗粒度的出价流程变得难以实行，因为它需要丰富的领域知识 [10] 以及有关广告环境的全面信息。

为减轻广告主优化出价的负担，这些广告平台提供了自动出价服务 [4, 5, 12, 39]。这些服务采用精心设计的出价策略，自动确定每次曝光机会的出价。这类策略会考虑广告环境和广告主的各种因素，例如曝光机会的分布、预算以及平均成本约束 [31]。鉴于广告环境的动态特性，有必要定期优化出价策略，通常每隔几分钟调整一次，以应对环境变化。广告回合通常持续 24 小时以上，因此，自动出价可被视为一个规划时域较长的序贯决策过程，出价策略旨在优化整个回合的表现。

近年来，强化学习（RL）技术已被用于优化自动出价策略，其方式是训练

CCS 概念

• 信息系统 →计算广告。

*Work is done during the internship at Alibaba Group.
†Bo Zheng is the corresponding author.

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获益，也不得为此而分发，且副本首页须包含本声明及完整引文。本作品中由作者以外的其他方拥有的组成部分，其版权必须得到尊重。允许在注明出处的情况下进行摘要摘录。其他形式的复制、再版、发布到服务器或重新分发至邮件列表，须事先获得特别许可和 / 或支付费用。请向 permissions@acm.org 申请许可。KDD '24，2024 年 8 月 25 – 29 日，西班牙巴塞罗那 © 2024 版权归所有者 / 作者所有。出版权已许可给 ACM。
ACM ISBN 979-8-4007-0490-1/24/08...$15.00 https://doi.org/10.1145/3637528.3671526

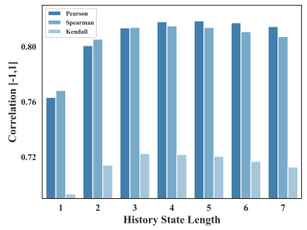


Figure 1: Correlation Coefficients between History and the Next State.

agents with bidding logs collected from online advertising environments [6, 21, 27, 35, 44, 48]. By leveraging historical realistic bidding information, these agents can learn patterns and trends to make informed bidding decisions. However, most existing RL auto-bidding methods are based on the Markovian decision process (MDP), where the next state only depends on the current state and action. In the online auto-bidding environment, this assumption may been challenged by our statistical analysis presented in Figure 1, which shows a significant increase in the correlation between the sequence lengths of history states and the next bidding state. This result indicates that solving auto-bidding considering only the last state will encounter several problems, including instability in the highly random online advertising environment. Additionally, the RL methods that rely on the Bellman equation often result in compound errors [15]. This issue is especially pronounced in the auto-bidding problem characterized by sparse return and limited data coverage. A detailed statistical analysis is provided in A.5. In this paper, instead of employing RL-based methods, we present a novel paradigm, AI Generated Bidding (AIGB), that regards auto-bidding as a generative sequential decision-making problem. AIGB directly capture the correlation between the return and the entire bidding trajectory that consists of a sequence of states or actions, thereby transforming the problem into learning to generate an optimal bidding trajectory. This approach enables us to overcome the limitations of RL when dealing with the highly random online advertising environment, sparse returns, and limited data coverage.

In the new paradigm, we propose ***Diff***usion auto-***bid***ding model DiffBid. It gradually corrupts the bidding trajectory by injecting scheduled Gaussian noises into the forward process. Then, it reconstructs trajectory from corrupted ones given returns and temporal conditions via a parameterized neural network. We further propose a non-Markovian inverse dynamics [17, 37, 49] to more accurately generate optimal bidding parameters. Taking one step further, DiffBid provides flexibility to closely align with the specific needs of advertisers by accommodating diverse constraints like cost-per-click (CPC) and incorporating human feedback. Notably, DiffBid serves as a unified model capable of mastering multiple tasks simultaneously, dynamically composing various bidding trajectory components to generate sequences that efficiently maximize diverse targets while adhering to a range of predefined constraints. To assess the effectiveness of DiffBid, we conducted extensive evaluations offline and online against baselines. Our results indicate that DiffBid surpasses RL methods for auto-bidding. In summary:

- We uncover that the Markov assumptions upon which common decision-making methods rely are not applicable to the auto-bidding problem. Therefore, we propose a novel bidding paradigm with non-Markovian properties based on generative learning. This paradigm represents a significant innovation in modeling methodology compared with existing RL methods commonly used in auto-bidding.
- Unlike common bidding methods, our approach captures the correlation between the return and the entire bidding trajectory. This design enables the method to address important challenges, such as sparse returns, and ensures stability in the highly random advertising environment. Finally, we prove that the proposed diffusion modeling is equivalent in terms of optimality to solving a non-Markovian decision problem.
- We demonstrate that the method can integrate capabilities to handle a variety of tasks within a unified solution, transcending the limitations of traditional task-specific methods. It shows that DiffBid outperforms conventional RL methods in auto-bidding, and achieves significant performance gain on a leading E-commerce ad platform through both offline and online evaluation. In specific, it achieves 2.81% increase in GMV and 3.36% in ROI.

2 PRELIMINARY

2.1 Problem Formulation

For simplicity, we consider auto-bidding with cost-related constraints. During a time period, suppose there are N impression opportunities arriving sequentially and indexed by i. In this setting, advertisers submit bids to compete for each impression opportunity. An advertiser will win the impression if its bid b_i is greater than others. Then it will incur a cost c_i for winning and getting the value.

During the period, the mission of an advertiser is to maximize the total received value $\sum_i o_i v_i$, where v_i is the value of impression i and o_i is whether the advertiser wins impression i. Besides, we have the budget and several constraints to control the performance of ad deliveries. Budget constraints are simply $\sum_i o_i c_i \le B$, where c_i is the cost of impression i and B is the budget. The other constraints are complex and according to [21] we have the unified formulation:

$$\frac{\sum_i c_{ij} o_i}{\sum_i p_{ij} o_i} \le C_j, \tag{1}$$

where C_j is the upper bound of j'th constraint. p_{ij} can be any performance indicator, e.g. return, or constant. c_{ij} is the cost of constraint j. Given J constraints, we have the Multi-constrained Bidding (MCB) as:

$$\begin{aligned} \underset{o_i}{\text{maximize}} \quad & \sum_i o_i v_i \\ \text{s.t.} \quad & \sum_i o_i c_i \le B \\ & \frac{\sum_i c_{ij} o_i}{\sum_i p_{ij} o_i} \le C_j, \quad \forall j \\ & o_i \in \{0,1\}, \ \forall i \end{aligned} \tag{2}$$

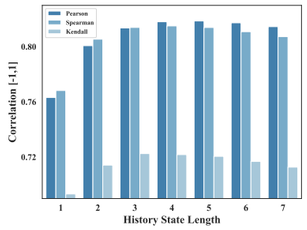


图 1：历史与下一状态之间的相关系数。

智能体，训练所用的出价日志收集自在线广告环境 [6, 21, 27, 35, 44, 48]。利用历史真实出价信息，这些智能体可以学习模式和趋势，从而作出有依据的出价决策。然而，现有的大多数 RL 自动出价方法都基于马尔可夫决策过程（MDP），其中下一状态仅取决于当前状态和动作。在在线自动出价环境中，图 1 所示的统计分析可能对这一假设构成挑战：随着历史状态序列长度的增加，历史状态与下一出价状态之间的相关性显著增强。这一结果表明，仅考虑最后一个状态来解决自动出价问题会遇到若干困难，包括在高度随机的在线广告环境中表现不稳定。此外，依赖贝尔曼方程的 RL 方法往往会产生累积误差 [15]。在回报稀疏且数据覆盖有限的自动出价问题中，这一问题尤为突出。详细的统计分析见 A.5。 本文没有采用基于 RL 的方法，而是提出了一种新范式——AI 生成式出价（AIGB），将自动出价视为生成式序贯决策问题。AIGB 直接捕捉回报与整条出价轨迹之间的相关性，该轨迹由一系列状态或动作构成，从而将问题转化为学习生成最优出价轨迹。这一方法使我们能够克服 RL 在应对高度随机的在线广告环境、稀疏回报和有限数据覆盖时的局限。

在这一新范式下，我们提出扩散自动出价模型 DiffBid。它在前向过程中按预定计划注入高斯噪声，逐步扰动出价轨迹。随后，在给定回报和时间条件的情况下，它通过参数化神经网络从受扰动的轨迹中重建轨迹。我们进一步提出非马尔可夫逆动力学 [17, 37, 49]，以更准确地生成最优出价参数。在此基础上，DiffBid 能够灵活适应单次点击成本（CPC）等多种约束，并融入人类反馈，从而紧密契合广告主的具体需求。值得注意的是，DiffBid 作为一个统一模型，能够同时掌握多项任务，动态组合各种出价轨迹成分，生成在遵循一系列预定义约束的同时高效最大化不同目标的序列。为评估 DiffBid 的有效性，我们在线下和线上开展了大量与基线方法的对比评估。结果表明，DiffBid 在自动出价中优于 RL 方法。概括而言：

- 我们发现，常见决策方法所依赖的马尔可夫假设并不适用于自动出价问题。因此，我们提出了一种基于生成式学习、具有非马尔可夫性质的新型出价范式。与自动出价中常用的现有强化学习方法相比，这一范式在建模方法上实现了重大创新。
- 与常见的出价方法不同，我们的方法捕捉回报与整个出价轨迹之间的相关性。这一设计使该方法能够应对回报稀疏等重要挑战，并确保其在高度随机的广告环境中保持稳定。最后，我们证明，所提出的扩散建模在最优性方面等价于求解一个非马尔可夫决策问题。
- 我们表明，该方法能够在统一解决方案中整合处理多种任务的能力，突破传统任务专用方法的局限。

结果表明，DiffBid 在自动出价中优于传统强化学习方法，并且在一家领先电商广告平台上的离线和在线评估中均取得了显著的性能提升。具体而言，商品交易总额（GMV）提高了 2.81%，投资回报率（ROI）提高了 3.36%。

2 预备知识

2.1 问题形式化

为简化起见，我们考虑具有成本相关约束的自动出价。在一个时间段内，假设有 N个曝光机会依次到达，并以 i为索引。在这一设定下，广告主提交出价以竞争每个曝光机会。如果某个广告主的出价 b_i高于其他广告主，就会赢得该次曝光。随后，它需要支付成本 c_i，以赢得曝光并获得价值。

在该时间段内，广告主的任务是最大化所获得的总价值 $\sum_i o_i v_i$，其中，v_i是曝光 i的价值，o_i表示广告主是否赢得曝光 i。此外，我们还通过预算和若干约束来控制广告投放的表现。预算约束可简单表示为 $\sum_i o_i c_i \le B$，其中，c_i是曝光 i的成本，B是预算。其他约束较为复杂，根据 [21]，可得到如下统一形式：

$$\frac{\sum_i c_{ij} o_i}{\sum_i p_{ij} o_i} \le C_j, \tag{1}$$

其中，C_j是第 j个约束的上界。 p_{ij}可以是任意性能指标，例如回报，也可以是常数。 c_{ij}是约束 j的成本。给定 J个约束，多约束出价（MCB）可表示为：

$$\begin{aligned} \underset{o_i}{\text{maximize}} \quad & \sum_i o_i v_i \\ \text{s.t.} \quad & \sum_i o_i c_i \le B \\ & \frac{\sum_i c_{ij} o_i}{\sum_i p_{ij} o_i} \le C_j, \quad \forall j \\ & o_i \in \{0,1\}, \ \forall i \end{aligned} \tag{2}$$

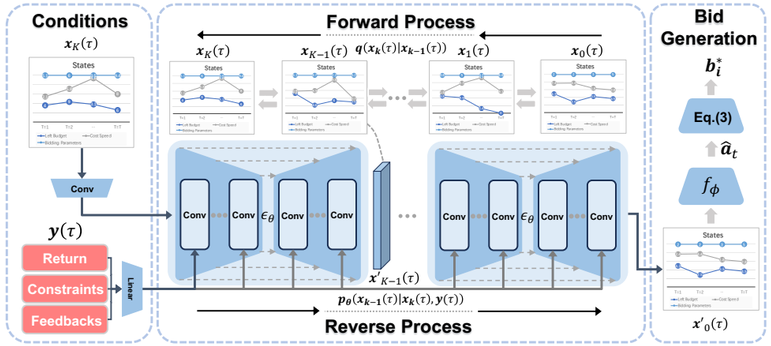


Figure 2: Overall Framework for Generative Auto-bidding.

A previous study [21] has already shown the optimal solution:

$$b_i^* = \lambda_0 v_i + C_i \sum_{j=1}^{J} \lambda_j p_{ij}, \tag{3}$$

where b_i^* is the predicted optimal bid for the impression i. $\lambda_j,\ j \in \{0, ..., J\}$ are the optimal bidding parameters. Specifically, when considering only the budget constraint, it is the Max Return bidding. However, when considering both the budget constraint and the CPC constraint, it is called Target-CPC bidding. From an alternative perspective, the optimal strategy involves arranging all impressions in order of their cost-effectiveness (CE) and then selecting every impression opportunity that surpasses the optimal CE ratio ce^*. This threshold enforces the constraint, and the optimal bidding parameters $\lambda_0 = 1/ce^*$.

2.2 Auto-Bidding as Decision-Making

Eq.(3) gives the formation of the optimal bid b_i^* with bidding parameters $\lambda_j,\ j \in \{0, ..., J\}$. However, in practice, the highly random and complex nature of the advertising environment prevents direct calculation of the bidding parameters. They must be carefully calibrated to adapt to the environment and dynamically adjusted as it evolves, This subsequently makes auto-bidding a sequential decision-making problem. To model it with decision-making, we introduce states $\boldsymbol{s}_t \in \mathcal{S}$ to describe the real-time advertising status and actions $\boldsymbol{a}_t \in \mathcal{A}$ to adjust the corresponding bidding parameters. The auto-bidding agent will take action $\boldsymbol{a}_t$ at the state $\boldsymbol{s}_t$ based on its policy π, and then the state will transit to the next state $\boldsymbol{s}_{t+1}$ and gain reward $\boldsymbol{r}_t \in \mathcal{R}$ according to the advertising environment dynamics $\mathcal{T}$. When $\mathcal{T} : \boldsymbol{s}_t \times \boldsymbol{a}_t \rightarrow \boldsymbol{s}_{t+1} \times \boldsymbol{r}_t$ satisfies, it is called the Markovian decision process (MDP). Otherwise, it is a non-Markovian decision process. We next describe the key items of the automated bidding agent in the industrial online advertising system:

- State $\boldsymbol{s}_t$ describes the real-time advertising status at time period t, which includes 1) remaining time of the advertiser; 2) remaining budget; 3) budget spend speed; 4) real-time cost-efficiency (CPC), 5) and average cost-efficiency (CPC).
- Action $\boldsymbol{a}_t$ indicates the adjustment to the bidding parameters at the time period t, which has the dimension of the number of bidding parameters $\lambda_j,\ j = 1, .., J$ and modeled as $(a_t^{\lambda_0}, ..., a_t^{\lambda_J})$.
- The reward $\boldsymbol{r}_t$ is the value contributed to the objective obtained within the time period t.
- A trajectory τ is the index of a sequence of states, actions, and rewards within an episode.

In the online advertising system, learning policy through direct interaction with the online environment is unfeasible due to safety concerns. Nonetheless, access to historical bidding logs, incorporating trajectories from a variety of bidding strategies, is attainable and provides a viable alternative. Prevalent auto-bidding methods predominantly leverage this offline data to craft effective policies. Our approach is aligned with this practice and will be elaborated in detail in the subsequent chapters.

3 AIGB PARADIGM FOR AUTO-BIDDING

To thoroughly investigate the auto-bidding problem, we conducted a series of statistical analyses of bidding trajectories, with detailed information available in appendix **??**. These analyses provide us with the insight that devising an effective bidding strategy is essentially equivalent to optimizing a state trajectory. Armed with this insight, we propose a hierarchical paradigm for auto-bidding that prioritizes the state trajectory optimization and subsequently generates actions aligned with the optimized trajectory.

For state trajectory optimization, we can employ a generative model to capture the joint distribution of the entire bidding trajectory and its associated returns, subsequently generating the trajectory distribution conditioned on the desired return. This approach enables us to address key auto-bidding challenges by employing SOTA generative algorithms. This paper presents an implementation that utilizes Denoising Diffusion Probabilistic Models (DDPM). For action generation, several off-the-shelf methods can be utilized to predict the proper action given the target state trajectory. In this paper, we apply a widely used inverse dynamics model. The hierarchical paradigm divides auto-bidding into two supervised

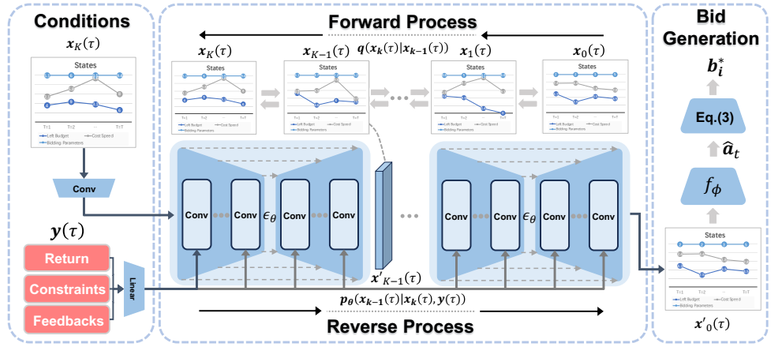


图 2：生成式自动出价的整体框架。

先前的一项研究 [21] 已给出了最优解：

$$b_i^* = \lambda_0 v_i + C_i \sum_{j=1}^{J} \lambda_j p_{ij}, \tag{3}$$

其中，b_i^*是曝光 i的预测最优出价。$\lambda_j,\ j \in \{0, ..., J\}$是最优出价参数。具体而言，仅考虑预算约束时，这称为最大回报出价；而同时考虑预算约束和 CPC 约束时，则称为目标 CPC 出价。从另一角度来看，最优策略是按成本效益（CE）对所有曝光进行排序，然后选择每个成本效益超过最优 CE 比率 ce^* 的曝光机会。这一阈值确保约束得到满足，且最优出价参数 $\lambda_0 = 1/ce^*$。

2.2 将自动出价视为决策问题

式（3）给出了最优出价 b_i^*由出价参数 $\lambda_j,\ j \in \{0, ..., J\}$构成的形式。然而，在实践中，广告环境高度随机且复杂，无法直接计算出价参数。必须仔细校准这些参数以适应环境，并随着环境的变化进行动态调整，这使自动出价成为一个序列决策问题。为了用决策方法对其建模，我们引入状态 $\boldsymbol{s}_t \in \mathcal{S}$来描述实时广告投放状况，并引入动作 $\boldsymbol{a}_t \in \mathcal{A}$来调整相应的出价参数。自动出价智能体会采取动作 $\boldsymbol{a}_t$，该动作是在状态 $\boldsymbol{s}_t$ 下依据其策略 π作出的；随后，状态会转移到下一个状态 $\boldsymbol{s}_{t+1}$并获得奖励 $\boldsymbol{r}_t \in \mathcal{R}$，这一过程遵循广告环境动力学 $\mathcal{T}$。当满足 $\mathcal{T} : \boldsymbol{s}_t \times \boldsymbol{a}_t \rightarrow \boldsymbol{s}_{t+1} \times \boldsymbol{r}_t$时，这被称为马尔可夫决策过程（MDP）；否则，就是非马尔可夫决策过程。下面，我们介绍工业级在线广告系统中自动出价智能体的关键要素：

- 状态 $\boldsymbol{s}_t$描述时间段 t的实时广告投放状况，包括：1）广告主的剩余时间；2）剩余预算；3）预算消耗速度；4）实时成本效率（CPC）；5）平均成本效率（CPC）。
- 动作 $\boldsymbol{a}_t$表示在时间段 t对出价参数的调整，其维度等于出价参数的数量 $\lambda_j,\ j = 1, .., J$，并建模为 $(a_t^{\lambda_0}, ..., a_t^{\lambda_J})$。
- 奖励 $\boldsymbol{r}_t$是在时间段 t内获得的、对目标有贡献的价值。
- 轨迹 τ是一个回合内一系列状态、动作和奖励的索引。

在在线广告系统中，出于安全性考虑，通过与线上环境直接交互来学习策略并不可行。不过，包含多种出价策略轨迹的历史出价日志是可以获取的，这提供了一种可行的替代方案。主流自动出价方法主要利用这些离线数据来构建有效策略。我们的方法也遵循这一实践，后续章节将对此作详细阐述。

3 用于自动出价的 AIGB 范式

为了深入研究自动出价问题，我们对出价轨迹进行了一系列统计分析，详细信息见附录 **??**。这些分析使我们认识到，设计有效的出价策略本质上等同于优化状态轨迹。基于这一认识，我们提出了一种自动出价的分层范式，首先优化状态轨迹，随后生成与优化后的轨迹相匹配的动作。

对于状态轨迹优化，我们可以使用生成式模型来捕捉整个出价轨迹及其对应回报的联合分布，随后生成以期望回报为条件的轨迹分布。该方法使我们能够运用当前最先进的生成算法来应对自动出价中的关键挑战。本文给出了一种使用去噪扩散概率模型（DDPM）的实现。对于动作生成，可以使用多种现成方法，根据目标状态轨迹预测适当的动作。本文采用一种广泛使用的逆动力学模型。这一分层范式将自动出价划分为两个监督

learning problems, offering several advantages that include enhanced interpretability and increased stability during the training process.

4 DIFFUSION AUTO-BIDDING MODEL

In this section, we give a detailed introduction of the proposed diffusion Auto-bidding Model (DiffBid). We will first give the modeling of Auto-bidding through diffusion models in Section 4.1.1. Then we give a detailed description of the forward process in Section 4.1.2, the reverse process in Section 4.1.3, and the training process in Section 4.2. Finally, we will give the complexity analysis in Section 4.4.

4.1 Diffusion Modeling of Auto-bidding

4.1.1 ***Overview.*** We model such sequential decision-making problem through conditional generative modeling [2, 9] by maximum likelihood estimation (MLE):

$$\max_{\theta} \mathbb{E}_{\tau\sim D}\left[\log p_\theta(\boldsymbol{x}_0(\tau)|\boldsymbol{y}(\tau))\right] \tag{4}$$

where τ is the trajectory index, $\boldsymbol{x}_0(\tau)$ is the original trajectory of states and $\boldsymbol{y}(\tau)$ is the corresponding property. The goal is to estimate the conditional data distribution with p_θ so that the future states of a trajectory $\boldsymbol{x}_0(\tau)$ from information $\boldsymbol{y}(\tau)$ can be generated. For example, in the context of online advertising, $\boldsymbol{y}(\tau)$ can be the constraints or the total value of the entire trajectory. Under such a setting, we can formalize the conditional diffusion modeling for auto-bidding:

$$q(\boldsymbol{x}_{k+1}(\tau)|\boldsymbol{x}_k(\tau)),\ \ p_\theta(\boldsymbol{x}_{k-1}(\tau)|\boldsymbol{x}_k(\tau),\boldsymbol{y}(\tau)), \tag{5}$$

where q represents the forward process in which noises are gradually added to the trajectory while p_θ is the reverse process where a model is used for denoising. The detailed introduction of diffusion modeling can be found in Appendix A.2. The overall framework is presented in Figure 2. We will make a detailed discussion about the two modeling processes in the following sections.

4.1.2 ***Forward Process via Diffusion over States.*** We model the forward process $q(\boldsymbol{x}_{k+1}(\tau)|\boldsymbol{x}_k(\tau))$ via diffusion over states, where:

$$\boldsymbol{x}_k(\tau) := (\boldsymbol{s}_1, ..., \boldsymbol{s}_t, ..., \boldsymbol{s}_T)_k\,, \tag{6}$$

where $\boldsymbol{s}_t$ is modeled as a one-dimensional vector. $\boldsymbol{x}_k(\boldsymbol{\tau})$ is a noise sequence of states and can be represented by a two-dimensional array where the first dimension is the time periods and the second dimension is the state values. Merely sampling states is not enough for an agent. Given $\boldsymbol{x}_k(\tau)$, we model the diffusion process as a Markov chain, where $\boldsymbol{x}_k(\tau)$ is only dependent on $\boldsymbol{x}_{k-1}(\tau)$:

$$q(\boldsymbol{x}_k(\tau)|\boldsymbol{x}_{k-1}(\tau)) = \mathcal{N}\left(\boldsymbol{x}_k(\tau); \sqrt{1-\beta_k}\boldsymbol{x}_{k-1}(\tau), \beta_k I\right), \tag{7}$$

when $k \to \infty$, $x_k(\tau)$ approaches a sequence of standard Gaussian distribution where we can make sampling through the reparameterization trick and then gradually denoise the trajectory to produce the final state sequence. For the design of $\beta_k,\ \ k = 1, ..., K$, we apply cosine schedule [38] to assign the corresponding values which smoothly increases diffusion noises using a cosine function to prevent sudden changes in the noise level. The details for noise schedule can be found in the appendix.

4.1.3 ***Reverse Process for Bid Generation.*** Following [2, 23] we use a classifier-free guidance strategy with low-temperature sampling to guide the generation of bidding, to extract high-likelihood trajectories in the dataset. During the training phase, we jointly train the unconditional model $\epsilon_\theta(\boldsymbol{x}_k(\tau), k)$ and conditional model $\epsilon_\theta(\boldsymbol{x}_k(\tau), \boldsymbol{y}(\tau), k)$ by randomly dropping out conditions. During generation, a linear combination of conditional and unconditional score estimates is used:

$$\hat{\epsilon}_k := \epsilon_\theta(\boldsymbol{x}_k(\tau), k) + \omega\left(\epsilon_\theta\left(\boldsymbol{x}_k(\tau), \boldsymbol{y}(\tau), k\right) - \epsilon_\theta\left(\boldsymbol{x}_k(\tau), k\right)\right), \tag{8}$$

where the scale ω is applied to extract the most suitable portion of the trajectory in the dataset that coappeared with $\boldsymbol{y}(\tau)$. After that, we can sample from DiffBid to produce bidding parameters through sampling from $p_\theta(\boldsymbol{x}_{k-1}(\tau)|\boldsymbol{x}_k(\tau), \boldsymbol{y}(\tau))$:

$$\boldsymbol{x}_{k-1}(\tau) \sim \mathcal{N}\left(\boldsymbol{x}_{k-1}(\tau)|\boldsymbol{\mu}_\theta\left(\boldsymbol{x}_k(\tau), \boldsymbol{y}(\tau), k\right), \Sigma_\theta\left(\boldsymbol{x}_k(\tau), k\right)\right) \tag{9}$$

where a widely used parameterization here is $\boldsymbol{\mu}_\theta(\boldsymbol{x}_k(\tau), \boldsymbol{y}(\tau), k) = \frac{1}{\sqrt{\alpha_k}}(\boldsymbol{x}_k(\tau) - \frac{\beta_k}{\sqrt{1-\bar{\alpha}_k}}\hat{\epsilon}_k)$ and $\Sigma_\theta(\cdot) = \beta_k$, in which $\alpha_k = 1 - \beta_k$ and $\overline{\alpha_k} = \prod_{i=1}^{k} \alpha_k$. When serving at time period t, the agent first sample a initial trajectory $x'_K(\tau) \sim \mathcal{N}(0, I)$ and assign the history states $\boldsymbol{s}_{0:t}$ into it. Then, we can sample predicted states with the reverse process recursively by

$$\boldsymbol{x}'_{k-1}(\tau) = \boldsymbol{\mu}_\theta(\boldsymbol{x}'_k(\tau), \boldsymbol{y}(\tau), k) + \sqrt{\beta_k}\boldsymbol{z} \tag{10}$$

where $\boldsymbol{z} \sim \mathcal{N}(0, I)$. Given $\boldsymbol{x}'_0(\tau)$, we can extract the next predicted state s'_{t+1}, and determine how much it should bid to achieve that state. In this setting, we apply widely used inverse dynamics [1, 40] with non-Markovian state sequence to determine current bidding parameters at time period t:

$$\hat{\boldsymbol{a}}_t = f_\phi(\boldsymbol{s}_{t-L:t}, \boldsymbol{s}'_{t+1}), \tag{11}$$

where $\hat{\boldsymbol{a}}_t \in \mathbb{R}^J$ contains predicted bidding parameters (i.e. $\lambda_i,\ i = 1, ..., n$) at time t. L is the length of history states. The inverse dynamic function f_ϕ can be trained with the same offline logs as the reverse process. This design disentangles the learning of states and actions, making it easier to learn the connection between states thus achieving better empirical performance. The overall procedure is summarized in Algorithm 1.

4.2 DiffBid Training

Following [22], we train DiffBid to approximate the given noise and the returns in a supervised manner. Given a bidding trajectory $\boldsymbol{x}_0(\tau)$, we have its corresponding returns e.g., values the advertiser received, the constraint the model should obey and the history states $s_l, l = 1, ..., t$ before time $t + 1$. Then we just train the reverse process model p_θ which is parameterized through the noise model ϵ_θ and the inverse dynamics f_ϕ through:

$$\begin{aligned}\mathcal{L}(\theta, \phi) = \mathbb{E}_{k,\tau\in\mathcal{D}}\left[||\epsilon - \epsilon_\theta(\boldsymbol{x}_k(\tau), \boldsymbol{y}(\tau), k)||^2\right] \\ + \mathbb{E}_{(\boldsymbol{s}_{t-L:t}, \boldsymbol{a}_t, \boldsymbol{s}'_{t+1})\in\mathcal{D}}\left[||\boldsymbol{a}_t - f_\phi(\boldsymbol{s}_{t-L:t}, \boldsymbol{s}'_{t+1})||^2\right],\end{aligned} \tag{12}$$

In the training process, we randomly sample a bidding trajectory τ and a time step k, then we construct a noise trajectory $\boldsymbol{x}_k(\tau)$ and predict the noise through Eq (8). Following [23], we randomly drop conditions $\boldsymbol{y}(\tau)$ with probability p to train DiffBid to enhance the robustness. The process is presented in Algorithm 2.

学习问题，带来了多项优势，包括增强可解释性，以及提高训练过程中的稳定性。

4 扩散自动出价模型

本节将详细介绍所提出的扩散自动出价模型（DiffBid）。我们首先在第 4.1.1 节介绍如何通过扩散模型对自动出价进行建模。随后，在第 4.1.2 节，详细描述前向过程，在第 4.1.3 节描述反向过程，并在第 4.2 节描述训练过程。最后，我们将在第 4.4 节给出复杂度分析。

4.1 自动出价的扩散建模

4.1.1 概述。我们通过条件生成式建模 [2, 9]，利用最大似然估计（MLE）对这一序列决策问题进行建模：

$$\max_{\theta} \mathbb{E}_{\tau\sim D}\left[\log p_\theta(\boldsymbol{x}_0(\tau)|\boldsymbol{y}(\tau))\right] \tag{4}$$

其中，τ是轨迹索引，$\boldsymbol{x}_0(\tau)$是原始状态轨迹，$\boldsymbol{y}(\tau)$是相应的属性。目标是利用 p_θ估计条件数据分布，使轨迹的未来状态 $\boldsymbol{x}_0(\tau)$可以根据信息 $\boldsymbol{y}(\tau)$生成。例如，在在线广告场景中，$\boldsymbol{y}(\tau)$可以是约束，也可以是整个轨迹的总价值。在这一设定下，我们可以将自动出价的条件扩散建模形式化为：

$$q(\boldsymbol{x}_{k+1}(\tau)|\boldsymbol{x}_k(\tau)),\ \ p_\theta(\boldsymbol{x}_{k-1}(\tau)|\boldsymbol{x}_k(\tau),\boldsymbol{y}(\tau)), \tag{5}$$

其中，q表示前向过程，在此过程中逐渐向轨迹添加噪声；p_θ则表示反向过程，在此过程中使用模型进行去噪。扩散建模的详细介绍见附录 A.2。整体框架如图 2 所示。我们将在后续各节详细讨论这两个建模过程。

4.1.2 通过状态扩散实现前向过程。我们通过状态扩散对前向过程 $q(\boldsymbol{x}_{k+1}(\tau)|\boldsymbol{x}_k(\tau))$进行建模，其中：

$$\boldsymbol{x}_k(\tau) := (\boldsymbol{s}_1, ..., \boldsymbol{s}_t, ..., \boldsymbol{s}_T)_k\,, \tag{6}$$

其中，$\boldsymbol{s}_t$被建模为一维向量。$\boldsymbol{x}_k(\boldsymbol{\tau})$是状态的噪声序列，可以表示为一个二维数组，其中第一维为时段，第二维为状态值。对智能体而言，仅对状态采样还不够。给定 $\boldsymbol{x}_k(\tau)$，我们将扩散过程建模为马尔可夫链，其中 $\boldsymbol{x}_k(\tau)$仅依赖于 $\boldsymbol{x}_{k-1}(\tau)$：

$$q(\boldsymbol{x}_k(\tau)|\boldsymbol{x}_{k-1}(\tau)) = \mathcal{N}\left(\boldsymbol{x}_k(\tau); \sqrt{1-\beta_k}\boldsymbol{x}_{k-1}(\tau), \beta_k I\right), \tag{7}$$

当 $k \to \infty$时，$x_k(\tau)$趋近于标准高斯分布序列，我们可以通过重参数化技巧进行采样，随后逐步对轨迹去噪，生成最终的状态序列。在 $\beta_k,\ \ k = 1, ..., K$的设计中，我们采用余弦调度[38]来分配相应的数值，利用余弦函数平滑地增加扩散噪声，避免噪声水平发生突变。噪声调度的详细信息见附录。

4.1.3 用于出价生成的逆向过程。遵循 [2, 23]，我们采用结合低温采样的无分类器引导策略来指导出价生成，以提取数据集中高似然的轨迹。在训练阶段，我们通过随机丢弃条件来联合训练无条件模型 $\epsilon_\theta(\boldsymbol{x}_k(\tau), k)$和条件模型$\epsilon_\theta(\boldsymbol{x}_k(\tau), \boldsymbol{y}(\tau), k)$。在生成阶段，我们使用条件得分估计与无条件得分估计的线性组合：

$$\hat{\epsilon}_k := \epsilon_\theta(\boldsymbol{x}_k(\tau), k) + \omega\left(\epsilon_\theta\left(\boldsymbol{x}_k(\tau), \boldsymbol{y}(\tau), k\right) - \epsilon_\theta\left(\boldsymbol{x}_k(\tau), k\right)\right), \tag{8}$$

其中，尺度 ω用于提取数据集中与 $\boldsymbol{y}(\tau)$共同出现的轨迹中最合适的部分。随后，我们可以通过从 $p_\theta(\boldsymbol{x}_{k-1}(\tau)|\boldsymbol{x}_k(\tau), \boldsymbol{y}(\tau))$采样，利用 DiffBid 生成出价参数：

$$\boldsymbol{x}_{k-1}(\tau) \sim \mathcal{N}\left(\boldsymbol{x}_{k-1}(\tau)|\boldsymbol{\mu}_\theta\left(\boldsymbol{x}_k(\tau), \boldsymbol{y}(\tau), k\right), \Sigma_\theta\left(\boldsymbol{x}_k(\tau), k\right)\right) \tag{9}$$

其中，这里一种广泛使用的参数化形式是 $\boldsymbol{\mu}_\theta(\boldsymbol{x}_k(\tau), \boldsymbol{y}(\tau), k) = \frac{1}{\sqrt{\alpha_k}}(\boldsymbol{x}_k(\tau) - \frac{\beta_k}{\sqrt{1-\bar{\alpha}_k}}\hat{\epsilon}_k)$和 $\Sigma_\theta(\cdot) = \beta_k$，其中 $\alpha_k = 1 - \beta_k$和 $\overline{\alpha_k} = \prod_{i=1}^{k} \alpha_k$。在时间段 t提供服务时，智能体首先采样一条初始轨迹 $x'_K(\tau) \sim \mathcal{N}(0, I)$，并将历史状态$\boldsymbol{s}_{0:t}$赋入其中。随后，我们可以通过以下方式递归地利用反向过程采样预测状态：

$$\boldsymbol{x}'_{k-1}(\tau) = \boldsymbol{\mu}_\theta(\boldsymbol{x}'_k(\tau), \boldsymbol{y}(\tau), k) + \sqrt{\beta_k}\boldsymbol{z} \tag{10}$$

其中，$\boldsymbol{z} \sim \mathcal{N}(0, I)$。给定 $\boldsymbol{x}'_0(\tau)$，我们可以提取下一个预测状态 s'_{t+1}，并确定应出价多少才能达到该状态。在这一设定下，我们采用广泛使用的逆动力学 [1, 40]，利用非马尔可夫状态序列来确定时间段 t的当前出价参数：

$$\hat{\boldsymbol{a}}_t = f_\phi(\boldsymbol{s}_{t-L:t}, \boldsymbol{s}'_{t+1}), \tag{11}$$

其中，$\hat{\boldsymbol{a}}_t \in \mathbb{R}^J$包含预测出价参数（即 $\lambda_i,\ i = 1, ..., n$），对应时刻为 t。L是历史状态的长度。逆动力学函数 f_ϕ可以使用与反向过程相同的离线日志进行训练。这一设计将状态学习与动作学习解耦，使状态之间的联系更易于学习，从而取得更好的实证表现。整体流程总结于算法 1。

4.2 DiffBid 训练

遵循[22]，我们以监督学习的方式训练 DiffBid，使其近似给定的噪声和回报。给定一条出价轨迹$\boldsymbol{x}_0(\tau)$，我们拥有其相应的回报，例如广告主获得的价值、模型应遵守的约束，以及历史状态 $s_l, l = 1, ..., t$，这些状态位于时刻 $t + 1$之前。随后，我们只需通过下式训练反向过程模型 p_θ（通过噪声模型ϵ_θ参数化）以及逆动力学 f_ϕ：

$$\begin{aligned}\mathcal{L}(\theta, \phi) = \mathbb{E}_{k,\tau\in\mathcal{D}}\left[||\epsilon - \epsilon_\theta(\boldsymbol{x}_k(\tau), \boldsymbol{y}(\tau), k)||^2\right] \\ + \mathbb{E}_{(\boldsymbol{s}_{t-L:t}, \boldsymbol{a}_t, \boldsymbol{s}'_{t+1})\in\mathcal{D}}\left[||\boldsymbol{a}_t - f_\phi(\boldsymbol{s}_{t-L:t}, \boldsymbol{s}'_{t+1})||^2\right],\end{aligned} \tag{12}$$

在训练过程中，我们随机采样一条出价轨迹 τ和一个时间步 k，随后构造噪声轨迹 $\boldsymbol{x}_k(\tau)$，并通过式（8）预测噪声。遵循[23]，我们随机丢弃条件 $\boldsymbol{y}(\tau)$，丢弃概率为 p，以此来训练 DiffBid，以增强鲁棒性。该过程如算法 2 所示。

4.3 Design of Conditions.

In this section, we present approaches transforming industrial metrics into conditions of DiffBid.

4.3.1 ***Generation with Returns.*** For each trajectory τ we have the total value the advertiser received as the the return $R(\tau) = \sum_{t=1}^{T} r_t$. We normalize the return by:

$$R = \frac{R(\tau) - R_{\min}}{R_{\max} - R_{\min}}, \tag{13}$$

where $R_{\min}$ and $R_{\max}$ are the smallest and the largest return in the dataset. Through Eq. 13 we normalize the return into $[0, 1]$ and merge it into $y(\tau)$. Subsequently, we train the model to generate trajectories conditioned on the normalized returns. It should be noted that trajectories with more values received have higher normalized returns. Thus $R = 1$ indicates the best trajectory with the highest values which will better fit the advertisers' needs. When generation, we just set $R = 1$ and generate the trajectory under the max return condition to the advertiser.

4.3.2 ***Generation with Constraints or Human Feedback.*** In MCB, the cumulative performance related to the constraints within a given episode should be controlled so as not to exceed the advertisers' expectations. In such a setting, we can design $\boldsymbol{y}(\tau)$ to control the generation process. For example, in the Target-CPC setting, we can maintain a binary variable E to indicate whether the final CPC exceeds the given constraint C:

$$E = \mathrm{I}_{x \le C}(x) \tag{14}$$

where $x = \frac{\sum_i c_i o_i}{\sum_i p_i o_i}$ is defined in Eq (2). We can then normalize x into $[0, 1]$ through min-max normalization for simplification. E can be used to indicate whether trajectory τ break the CPC constraint. We can also design $\boldsymbol{y}(\tau)$ to include $E = 1$ to make the model generate bids that do not break the CPC constraint. Sometimes it is also important to adjust the bidding parameters given real-time feedback provided by the advertiser to enable flexibility. Here we use two example indicators that reflect the experience of advertisers:

(1) **Smoothness**: an advertiser may expect the cost curve as smooth as possible to avoid sudden change. By defining $x = \frac{1}{T}\sum_t |cost_t - cost_{t-1}|$, we can model it as a binary variable S indicating whether the max cost change between adjacent time period exceeds a threshold as in Eq (14).
(2) **Early/Late Spend**: an advertiser may expect the budget to be cost in the morning or in the evening when there are promotions. Here we model the ratio of cost in the early half day through $x = \frac{\sum_{t=0}^{T/2} cost_i}{\sum_{t=0}^{T} cost_i}$, and use a binary variable to indicate whether the spend in the early half day exceeds a certain threshold C as in Eq (14).

We can also compose several constraints together to form $\boldsymbol{y}(\tau)$ to guide the model to generate bid parameters that adhere to different constraints. In this setting, $\boldsymbol{y}(\tau)$ will be a vector.

4.4 Complexity Analysis

The complexity analysis for training DiffBid consists of the training process and the inference process. For training, given the time complexity of the noise prediction model ϵ_θ is $O(T_1)$, the complexity for the inverse dynamic model f_ϕ is $O(T_2)$, the complexity for a training epoch is $O(|\mathcal{B}|(T_1 + T_2))$. It can be seen that the training complexity is linear with the input given T_1 and T_2 are relatively fixed. Thus the training of DiffBid is efficient. For generation, given the total diffusion step K, the trajectory length L, then the time complexity for inference is $O(KL(T_1+T_2))$. We can observe that the time complexity for inference is linearly scaled with the diffusion step K. In image generation, K is usually very large to ensure good generation quality, which brings the problem of non-efficiency. However, for bidding generation, we find K needs not to be very large. Relatively small K has already generated promising results. Moreover, in auto-bidding, a higher tolerance for latency is acceptable, enabling the use of relatively larger K.

5 THEORETICAL ANALYSIS

In this section, we theoretically analyze the property of DiffBid. In specific, we show that DiffBid that utilize MLE as the objective has a corresponding non-Markovian decision problem [16, 33, 36, 41]. The detailed proofs can be found in the Appendix A.7.

Lemma 5.1 (MLE as non-Markovian decision-making). *Assuming the Markovian transition $p_{\gamma^*}(s_{t+1}|s_t, a_t)$ is known, the ground-truth conditional state distribution $p^*(s_{t+1}|s_{0:t})$ for demonstration sequences is accessible, we can construct a non-Markovian sequential decision-making problem, based on a reward function $r_\alpha(s_{t+1}, s_{0:t}) := \log \int p_\alpha(a_t|s_{0:t}) p_{\gamma^*}(s_{t+1}|s_t, a_t) da_t$ for an arbitrary energy-based policy $p_\alpha(a_t|s_{0:t})$. Its objective is*

$$\sum_{t=0}^{T} \mathbb{E}_{p^*(s_{0:t})}\left[V^{p_\alpha}(s_{0:t})\right] = \mathbb{E}_{p^*(s_{0:T})}\left[\sum_{t=0}^{T}\sum_{k=t}^{T} r_\alpha(s_{k+1}; s_{0:k})\right] \tag{15}$$

$V^{p_\alpha}(s_{0:t}) := \mathbb{E}_{p^*(s_{t+1:T}|s_{0:t})}[\sum_{k=1}^{T} r_\alpha(s_{t+1}; s_{0:t})]$ is the value function of p_α. This objective yields the save optimal policy as the Maximum Likelihood Estimation $\mathbb{E}_{p^*(s_{0:T})}[\log p_\theta(s_{0:T})]$.

Remarks. This analysis shows that DiffBid utilizing MLE objective has its corresponding non-Markovian decision problem, and their optimal are equivalent. It means that DiffBid does not require the MDP assumption of problems and thus is more powerful in handling randomness and sparse return like in the advertising environment.

6 EXPERIMENTS

6.1 Experimental Setup

6.1.1 ***Experimental Environment.*** The simulated experimental environment is conducted in a manually built offline real advertising system (RAS) as in [35]. Specifically, the RAS is composed of two consecutive stages, where the auction mechanisms resemble those in the RAS. We consider the bidding process in a day, where the episode is divided into 96 time steps. Thus, the duration between any two adjacent time steps t and $t + 1$ is 15 minutes. The number of impression opportunities between time step t and $t+1$ fluctuates from 100 to 500. Detailed parameters in the RAS are shown in Table 5. We keep the parameters the same for all experiments.

6.1.2 ***Data Collection.*** We use the widely applied auto-bidding RL method USCB in the online environment to generate the bidding logs for offline RL training. This results in a total 5,000 trajectories for the based dataset and 50,000 for a larger one. To increase the

4.3 条件设计。

本节介绍将工业界指标转化为 DiffBid 条件的方法。

4.3.1 基于回报的生成。对于每条轨迹 τ，我们将广告主获得的总价值作为回报 $R(\tau) = \sum_{t=1}^{T} r_t$。我们通过下式对回报进行归一化：

$$R = \frac{R(\tau) - R_{\min}}{R_{\max} - R_{\min}}, \tag{13}$$

其中， $R_{\min}$和 $R_{\max}$分别是数据集中的最小回报和最大回报。通过式13，我们将回报归一化为 $[0, 1]$，并将其合并到 $y(\tau)$中。随后，我们训练模型，使其以归一化后的回报为条件生成轨迹。需要注意的是，获得价值越多的轨迹，其归一化回报越高。因此， $R = 1$表示价值最高的最佳轨迹，它将更好地满足广告主的需求。在生成阶段，我们只需设置 $R = 1$，并在最大回报条件下为广告主生成轨迹。

4.3.2 基于约束或人类反馈的生成。在 MCB 中，应控制一个给定回合内与约束相关的累积表现，使其不超出广告主的预期。在这种设定下，我们可以设计 $\boldsymbol{y}(\tau)$来控制生成过程。例如，在 Target-CPC 设定下，我们可以维护一个二元变量 E，以表示最终的单次点击成本是否超过给定约束 C：

$$E = \mathrm{I}_{x \le C}(x) \tag{14}$$

其中， $x = \frac{\sum_i c_i o_i}{\sum_i p_i o_i}$在式（2）中定义。为简化起见，我们随后可以通过最小—最大归一化将 x归一化为$[0, 1]$。 E可用于表示轨迹 τ是否违反 CPC 约束。我们还可以设计 $\boldsymbol{y}(\tau)$，使其包含 $E = 1$，从而让模型生成不违反 CPC 约束的出价。有时，根据广告主提供的实时反馈调整出价参数以实现灵活性也很重要。这里，我们使用两个反映广告主体验的示例指标：

（1）平滑性：广告主可能希望成本曲线尽可能平滑，以避免突变。通过定义 $x = \frac{1}{T}\sum_t |cost_t - cost_{t-1}|$，我们可以将其建模为二元变量$S$，表示相邻时间段之间的最大成本变化是否超过某个阈值，如式（14）所示。（2）早 / 晚花费：广告主可能希望预算

在早上或晚上的促销期间花费。在这里，我们将前半天的成本占比

通过 $x = \frac{\sum_{t=0}^{T/2} cost_i}{\sum_{t=0}^{T} cost_i}$进行建模，并使用一个二元变量表示前半天的花费是否超过某个阈值 C，如式（14）所示。

我们还可以将多个约束组合成 $\boldsymbol{y}(\tau)$，以引导模型生成符合不同约束的出价参数。在这种设定下， $\boldsymbol{y}(\tau)$将是一个向量。

4.4 复杂度分析

DiffBid 的复杂度分析包括训练过程和推理过程。对于训练，给定噪声预测模型 ϵ_θ的时间复杂度为 $O(T_1)$，逆动力学模型 f_ϕ的复杂度为 $O(T_2)$，则一个训练轮次的复杂度为 $O(|\mathcal{B}|(T_1 + T_2))$。可以看出，当 T_1和 T_2相对固定时，训练复杂度与输入呈线性关系。因此，DiffBid 的训练是高效的。对于生成，给定总扩散步数 K和轨迹长度 L，推理的时间复杂度为 $O(KL(T_1+T_2))$。我们可以观察到，推理的时间复杂度随扩散步数 K线性增长。在图像生成中， K通常非常大，以确保良好的生成质量，这会带来效率低下的问题。然而，对于出价生成，我们发现 K不必非常大。相对较小的 K就已产生了很有潜力的结果。此外，在自动出价中，可以接受更高的延迟容忍度，因此能够使用相对较大的 K。

5 理论分析

本节从理论上分析 DiffBid 的性质。具体而言，我们表明，以最大似然估计（MLE）为目标的 DiffBid 存在一个与之对应的非马尔可夫决策问题 [16, 33, 36, 41]。详细证明见附录 A.7。

引理 5.1（将 MLE 视为非马尔可夫决策）。假设已知马尔可夫状态转移$p_{\gamma^*}(s_{t+1}|s_t, a_t)$，且可获取示范序列的真实条件状态分布 $p^*(s_{t+1}|s_{0:t})$，则我们可以基于奖励函数 $r_\alpha(s_{t+1}, s_{0:t}) := \log \int p_\alpha(a_t|s_{0:t}) p_{\gamma^*}(s_{t+1}|s_t, a_t) da_t$，为任意基于能量的策略 $p_\alpha(a_t|s_{0:t})$构建一个非马尔可夫序贯决策问题。其目标为

$$\sum_{t=0}^{T} \mathbb{E}_{p^*(s_{0:t})}\left[V^{p_\alpha}(s_{0:t})\right] = \mathbb{E}_{p^*(s_{0:T})}\left[\sum_{t=0}^{T}\sum_{k=t}^{T} r_\alpha(s_{k+1}; s_{0:k})\right] \tag{15}$$

$V^{p_\alpha}(s_{0:t}) := \mathbb{E}_{p^*(s_{t+1:T}|s_{0:t})}[\sum_{k=1}^{T} r_\alpha(s_{t+1};\ s_{0:t})]$是 p_α的价值函数。这一目标产生的最优策略与最大似然估计 $\mathbb{E}_{p^*(s_{0:T})}[\log p_\theta(s_{0:T})]$相同。

说明。这一分析表明，采用 MLE 目标的 DiffBid 存在与之对应的非马尔可夫决策问题，且二者的最优解等价。这意味着 DiffBid 不要求问题满足 MDP 假设，因此在处理广告环境这类场景中的随机性和稀疏回报时更具能力。

6 实验

6.1 实验设置

6.1.1 实验环境。仿真实验在人工构建的离线真实广告系统（RAS）中进行，与 [35] 相同。具体而言，RAS 由两个连续阶段组成，其拍卖机制与真实广告系统中的机制相似。我们考虑一天内的出价过程，将该回合划分为 96 个时间步。因此，任意两个相邻时间步 t和 $t + 1$之间的时长为 15 分钟。时间步 t与 $t+1$之间的曝光机会数量在 100 到 500 之间波动。RAS 的详细参数见表 5。所有实验均使用相同的参数。

6.1.2 数据收集。我们在在线环境中使用广泛应用的自动出价 RL 方法 USCB，生成用于离线 RL 训练的出价日志。由此，基础数据集共获得5,000条轨迹，更大的数据集共获得50,000条轨迹。为增加

Table 1: Performance Comparison with baselines in different settings, including different data scales, and budgets in Max Return bidding. *improv* indicates the relative improvement of DiffBid against the most comparative baseline. The best results are bolded and the best second results are underlined.

Training Dataset	Budget	USCB	BCQ	CQL	IQL	DT	DiffBid	*improv*
USCB-5K	1.5K	454.25	454.72	461.82	456.80	477.39	**480.76**	0.71%
	2.0K	482.67	483.50	475.78	486.56	507.30	**511.17**	0.76%
	2.5K	497.66	498.77	481.37	518.27	527.88	**531.29**	0.65%
	3.0K	500.60	501.86	491.36	549.19	550.66	**556.32**	1.03%
USCBEx-5K	1.5K	454.25	453.74	358.43	464.69	378.64	**475.62**	2.35%
	2.0K	482.67	487.63	356.80	529.36	439.03	**544.38**	2.84%
	2.5K	497.66	510.75	356.41	613.67	505.43	**624.29**	1.73%
	3.0K	500.60	512.18	355.42	670.65	574.79	**678.73**	1.17%
USCBEx-50K	1.5K	454.25	458.64	435.06	446.23	396.24	**495.57**	8.05%
	2.0K	482.67	491.72	431.49	533.58	478.29	**551.73**	3.40%
	2.5K	497.66	513.23	428.39	592.32	554.48	**606.34**	2.37%
	3.0K	500.60	526.21	425.29	633.26	611.50	**644.88**	1.83%

diversity of the action space, we also randomly make explorations to generate a dataset with more noise. The above process results in three datasets: USCB-5k, USCBEx-5k, and USCBEx-50k, where USCBEx indicates USCB logs with random exploration data.

6.1.3 ***Baselines.*** We use the state-of-the-art auto-bidding method USCB as well as other 4 recently proposed offline RL methods as our baselines. The details of the baselines are as follows:

- **USCB** [21] an RL method designed for real-time bidding to dynamically adjust parameters to achieve the optimum. It has outperformed many RL baselines and is also the base policy that is used to collect the data for offline training.
- **BCQ** [14] a classic offline RL method without interaction with the environment.
- **CQL** [30] address the limitations of offline RL methods by learning a conservative Q-function such that the expected value of a policy under it lower-bounds its true value.
- **IQL** [29] an offline RL method that does not require evaluating actions outside of the dataset, yet it enables substantial improvement of the learned policy beyond the best behavior in the data through generalization
- **DT** [9] a prevalent generative method based on the transformer architecture for sequential decision-making.

6.1.4 ***Implementation Details.*** For the implementation of baselines, we use the default hyper-parameters suggested from their papers and also tune through our best effort. For DiffBid, the diffusion steps is searched within $\{5, 10, 20, 30, 50\}$. γ is set to 0.008. L is searched in $\{1, 2, 3\}$. ω for noise schedule is set to 0.2 empirically. The batch size is set to 2% of all training trajectories. Total training epochs is set to 500. For the implementation of p_θ, we adopt the most widely used model U-Net for diffusion modeling with hidden sizes of 128 and 256. We use Adam optimizer with a learning rate $1e{-}4$ to optimize the model. The condition dropout ratio is set to 0.2 during training. We update the model with momentum updates over a period of 4 steps.

6.1.5 ***Evaluation.*** For evaluation, we randomly initialize a multi-agent advertising environment with USCB as the base auto-bidding agents and use other methods to compete with these agents. We test the performance under 4 different budgets, 1500, 2000, 2500, and 3000, to test the generalization under different budget scales. We use the cumulative reward as the evaluation metric, which reflects the total gain received by the target agent. For each method, we randomly initialize 50 times and report the average of top-5 scores.

6.2 Performance Evaluation

The performance against baselines is shown in Table 1. In this table, we show the cumulative reward from different budgets of all the models. We have the following discoveries. One of the key takeaways from the performance comparison presented in Table 1 is that offline RL methods consistently outperform the state-of-the-art auto-bidding method, USCB. This finding underscores the advantages of leveraging historical bidding data to train RL agents. Offline RL methods, such as BCQ, IQL, and DT, exhibit superior performance in terms of cumulative rewards across various budget scenarios. The superiority of offline RL methods can be attributed to their ability to learn from past bidding experiences without interaction with a simulation environment. This mitigates the challenges associated with inconsistencies between the online bidding environment and the offline bidding environment, leading to policies that are better aligned with real-world scenarios. Notably, DiffBid stands out as the top-performing approach among all the methods evaluated. In all budget scenarios and training datasets, DiffBid consistently achieves the highest cumulative rewards. This remarkable performance highlights the efficacy of the DiffBid approach in optimizing bidding strategies by directly modeling the correlation with the returns and entire trajectories. By decoupling the computational complexity from horizon length, DiffBid achieves superior decision-making capabilities, outperforming traditional RL methods in both foresight and strategy. Another important observation

Table 1: Performance Comparison with baselines in different settings, including different data scales, and budgets in Max Return bidding. *improv* indicates the relative improvement of DiffBid against the most comparative baseline. The best results are bolded and the best second results are underlined.

Training Dataset	Budget	USCB	BCQ	CQL	IQL	DT	DiffBid	*improv*
USCB-5K	1.5K	454.25	454.72	461.82	456.80	477.39	**480.76**	0.71%
	2.0K	482.67	483.50	475.78	486.56	507.30	**511.17**	0.76%
	2.5K	497.66	498.77	481.37	518.27	527.88	**531.29**	0.65%
	3.0K	500.60	501.86	491.36	549.19	550.66	**556.32**	1.03%
USCBEx-5K	1.5K	454.25	453.74	358.43	464.69	378.64	**475.62**	2.35%
	2.0K	482.67	487.63	356.80	529.36	439.03	**544.38**	2.84%
	2.5K	497.66	510.75	356.41	613.67	505.43	**624.29**	1.73%
	3.0K	500.60	512.18	355.42	670.65	574.79	**678.73**	1.17%
USCBEx-50K	1.5K	454.25	458.64	435.06	446.23	396.24	**495.57**	8.05%
	2.0K	482.67	491.72	431.49	533.58	478.29	**551.73**	3.40%
	2.5K	497.66	513.23	428.39	592.32	554.48	**606.34**	2.37%
	3.0K	500.60	526.21	425.29	633.26	611.50	**644.88**	1.83%

动作空间的多样性，我们还进行了随机探索，以生成噪声更多的数据集。上述过程得到三个数据集：USCB-5k、USCBEx-5k 和 USCBEx-50k，其中 USCBEx 表示包含随机探索数据的 USCB 日志。

6.1.3 基线方法。我们采用最先进的自动出价方法 USCB，以及其他 4 种近期提出的离线 RL 方法作为基线。各基线方法的详情如下：

• **USCB** [21] 是一种面向实时出价的强化学习方法，通过动态调整参数来实现最优。它已优于许多强化学习基线，也是用于收集离线训练数据的基础策略。• **BCQ** [14] 是一种经典的离线强化学习方法，无需与环境交互。• **CQL** [30] 通过学习保守的 Q 函数来解决离线强化学习方法的局限，使得在该函数下策略的期望价值构成其真实价值的下界。• **IQL** [29] 是一种无需评估数据集之外动作的离线强化学习方法，但它能够通过泛化，使学得的策略显著优于数据中的最佳行为。• **DT** [9] 是一种基于 Transformer 架构、用于序列决策的主流生成式方法。

6.1.4 实现细节。对于基线的实现，我们采用其论文建议的默认超参数，并尽最大努力进行调优。对于 DiffBid，扩散步数在 $\{5, 10, 20, 30, 50\}$内搜索。 γ设为0.008. L在 $\{1, 2, 3\}$中搜索。根据经验，噪声调度的 ω设为 0.2。批大小设为全部训练轨迹数量的 2%。总训练轮数设为 500。在实现 p_θ时，我们采用扩散建模中应用最广泛的 U-Net 模型，隐藏层大小为 128 和 256。我们使用学习率为$1e{-}4$的 Adam 优化器来优化模型。训练期间，条件丢弃比例设为 0.2。我们以 4 步为周期，通过动量更新来更新模型。

6.1.5 评估。评估时，我们随机初始化一个多智能体广告环境，以 USCB 作为基础自动出价智能体，并使用其他方法与这些智能体竞争。我们在 1500、2000、2500 和 3000 这 4 种不同预算下测试性能，以检验在不同预算规模下的泛化能力。我们采用累积奖励作为评估指标，它反映目标智能体获得的总收益。对于每种方法，我们进行 50 次随机初始化，并报告得分最高的 5 次的平均值。

6.2 性能评估

与基线方法的性能对比如表1所示。该表展示了所有模型在不同预算下的累积奖励。我们有以下发现。表1中性能比较的一个关键结论是，离线强化学习方法始终优于最先进的自动出价方法 USCB。这一发现凸显了利用历史出价数据训练强化学习智能体的优势。BCQ、IQL 和 DT 等离线强化学习方法在各种预算情境下的累积奖励方面表现更优。离线强化学习方法的优势可归因于其能够从以往的出价经验中学习，而无需与仿真环境交互。这缓解了线上出价环境与离线出价环境不一致所带来的挑战，使策略更符合真实世界的情境。值得注意的是，在所有被评估的方法中，DiffBid 的性能最为突出。在所有预算情境和训练数据集上，DiffBid 都始终取得最高的累积奖励。这一出色表现表明，DiffBid 通过直接对回报与完整轨迹之间的相关性建模，能够有效优化出价策略。通过将计算复杂度与规划时域长度解耦，DiffBid 实现了更强的决策能力，在前瞻性和策略方面均优于传统强化学习方法。另一个重要观察

Table 2: Ablation Study

Model	USCBEx-5K	USCBEx-50K
DiffBid	2280.12	2395.60
DiffBid w/o cond	1812.64	1852.21
DiffBid w/o non-mkv	2254.78	2287.41

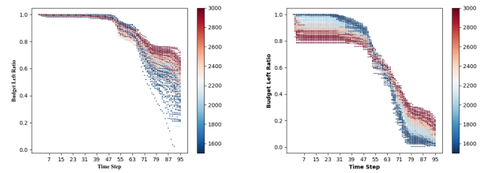
from the results is the impact of training dataset size on model performance. When comparing the "USCB-5K" and "USCBEx-50K" settings, it becomes evident that a larger training dataset consistently leads to improved cumulative rewards. This finding underscores the significance of data size in training RL models for automated bidding. A richer dataset allows the models to capture more diverse bidding scenarios and make more informed decisions, ultimately resulting in better performance. One intriguing aspect of DiffBid's performance is its resilience to noise. In real-world advertising environments, there can be inherent uncertainty and variability in the bidding process due to factors like market dynamics and competitor behavior. DiffBid appears to handle such noise more effectively than the RL baselines. This means that even in situations where bidding outcomes are less predictable, DiffBid manages to maintain competitive performance.

6.3 Ablation Study

To study different parts of the proposed DiffBid, we run the model without a certain module to see if the removed corresponding module will result in a performance drop. The result of the ablation study is shown in Table 2. Due to the space limitation, we only provide the results on USCBEx-5K and USCBEx-50K. w/o cond refers to the DiffBid with the condition set to 0.0 (rather than 1.0). w/o non-mkv refers to the situation where we only use the current state and the predicted next state to generate the bidding coefficient. From the table, we find both of the two parts contribute to the final result, and removing either of them will result in a performance drop. It verifies the effectiveness of the proposed methods in boosting DiffBid's performance for auto-bidding.

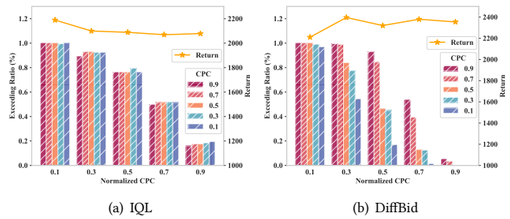
6.4 In-depth Analysis

6.4.1 ***Study of State Transition***. Here we compare the state transition of the baseline method USCB and our proposed method DiffBid. The result for grouped and non-grouped state transition during a day is shown in Figure 3. In this figure, we plot the budget left ratio with time steps in one day. From the figure, we can observe that under USCB, most of the advertisers' consumption does not exhaust their budget. This is attributed to the inconsistency between the offline virtual environment and the real online environment faced by USCB. On the contrary, the budget completion situation improves under DiffBid, where most of the advertisers spend more than 80% of their budgets. One possible reason is that DiffBid finds trajectories with a high budget completion ratio will also have a high cumulative reward, and thus tend to generate trajectories with a high budget completion ratio. Moreover, advertisers with small budgets undertend to spend money in the afternoon. This is because the impressions in the afternoon offer a higher cost-effectiveness, albeit with a limited quantity.



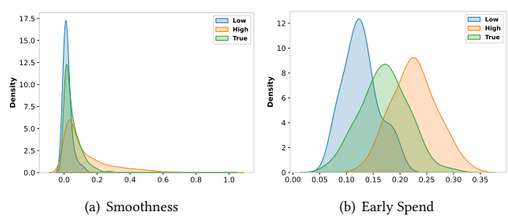
(a) USCB (b) DiffBid

Figure 3: State Transition in One Episode.



(a) IQL (b) DiffBid

Figure 4: Performance under CPC constraint.



(a) Smoothness (b) Early Spend

Figure 5: Performance of Human Feedback.

6.4.2 ***Performance under Constraints and Feedbacks.*** We additionally investigate DiffBid's multi-objective optimization capability under specific constraints, comparing its performance with Offline RL. Specifically, we choose CPC ratio and overall return as metrics and examine the ability of DiffBid and IQL to control the overall CPC exceeding ratio while maximizing the overall return. During training, we set different thresholds of CPC as in Eq (14). Then when testing, we make DiffBid generating trajectories under the expected CPC. In Figure 4, we show the exceeding ratio and overall return under different CPC constraints and training settings. From the figure, we find that DiffBid has the ability to control diverse levels of exceeding ratio while maintaining an intact return, surpassing IQL by a significant margin. Consequently, DiffBid holds a distinct advantage in effectively addressing MCB problems. We also study the performance under different advertiser

Table 2: Ablation Study

Model	USCBEx-5K	USCBEx-50K
DiffBid	2280.12	2395.60
DiffBid w/o cond	1812.64	1852.21
DiffBid w/o non-mkv	2254.78	2287.41

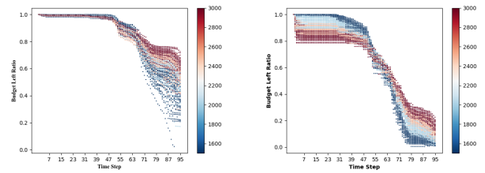
来自这些结果，即训练数据集规模对模型性能的影响。比较“USCB-5K”和“USCBEx-50K”设定时，可以明显看出，更大的训练数据集始终带来更高的累积奖励。这一发现凸显了数据规模对训练自动出价强化学习模型的重要性。更丰富的数据集使模型能够捕捉更多样的出价情境，并做出依据更充分的决策，最终取得更好的性能。DiffBid 性能中一个值得关注的方面是其抗噪能力。在真实广告环境中，由于市场动态和竞争对手行为等因素，出价过程可能存在固有的不确定性和变化。DiffBid 似乎比强化学习基线更能有效应对此类噪声。这意味着，即便在出价结果较难预测的情况下，DiffBid 仍能保持具有竞争力的性能。

6.3 消融研究

为研究所提出的 DiffBid 的各个部分，我们在移除某个模块的情况下运行模型，以观察移除相应模块是否会导致性能下降。消融研究结果如表2所示。由于篇幅限制，我们仅提供 USCBEx-5K 和 USCBEx-50K 上的结果。w/o cond 表示将条件设为 0.0（而非 1.0）的 DiffBid。w/o non-mkv 表示仅使用当前状态和预测的下一状态来生成出价系数的情况。从表中可以看出，这两个部分均对最终结果有所贡献，移除其中任何一个都会导致性能下降。这验证了所提出的方法能够有效提升 DiffBid 在自动出价中的性能。

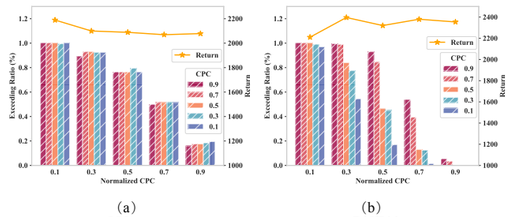
6.4 深入分析

6.4.1 状态转移研究。这里，我们比较基线方法 USCB 与所提出的 DiffBid 方法的状态转移。一天内分组和未分组的状态转移结果见图 3。图中绘制了一天内剩余预算比例随时间步的变化。由图可见，在 USCB 下，大多数广告主的消费没有用尽预算。这是由于 USCB 所面对的离线虚拟环境与真实在线环境存在不一致。相比之下，DiffBid 下的预算完成情况有所改善，大多数广告主花费了 80% 以上的预算。一种可能的原因是，DiffBid 发现预算完成率高的轨迹也具有较高的累积奖励，因此倾向于生成预算完成率较高的轨迹。此外，预算较少的广告主倾向于在下午花费预算。这是因为下午的曝光机会性价比更高，但数量有限。



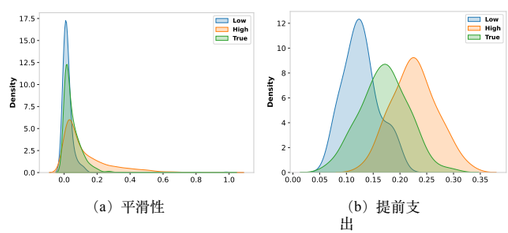
(a) USCB (b) DiffBid

图 3：一个回合内的状态转移。



(a) IQL (b) DiffBid

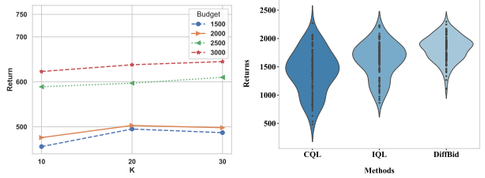
图 4：CPC 约束下的表现。



（a）平滑性 （b）提前支出

图 5：人类反馈的表现。

6.4.2 约束和反馈下的表现。我们还研究了 DiffBid 在特定约束下的多目标优化能力，并将其表现与离线 RL 进行比较。具体而言，我们选择 CPC 比率和总体回报作为指标，考察 DiffBid 和 IQL 在最大化总体回报的同时控制总体 CPC 超出比例的能力。在训练期间，我们按照式（14）设置不同的 CPC 阈值。随后，在测试时，让 DiffBid 在期望的 CPC 条件下生成轨迹。图 4 展示了不同 CPC 约束和训练设置下的超出比例及总体回报。由图可见，DiffBid 能够在保持回报不受损失的同时控制不同水平的超出比例，显著优于 IQL。因此，DiffBid 在有效解决 MCB 问题方面具有明显优势。我们还研究了不同广告主



(a) Impact of Diffusion Steps (b) Stability

Figure 6: In-depth Analysis.

Table 3: Online A/B Test Result.

Metrics	#Plan	Budget	Cost	Buycnt	GMV	ROI
Baseline	2068	886744	834426.104	23584.6836	1853823	2.221
DiffBid	2068	886744	829992.384	24078.6883	1905954	2.296
compare	-	-	-0.53%	+2.09%	+2.81%	+3.36%

feedbacks. During training we split the trajectories through thresholds of Eq. (14) into high and low levels, and learn the conditional distribution under different levels. During generation, we adjust the condition and generate corresponding samples and summarize the metrics. The results for the statistic distribution of metrics for low level, high level and the original trajectories are shown in Figure 5. We find that the trajectory obtained from deploying DiffBid is well controlled by the condition.

6.4.3 ***Impact of Diffusion Steps.*** We also study the overall performance under different diffusion steps, which is an important factor in influencing the efficiency and performance. The overall impact of diffusion steps with respect to different budgets is illustrated in Figure 6. From the figure, we have the following discoveries. First of all, we observe that diffusion steps have a larger impact on advertisers with small budgets (1500 yuan). Secondly, larger budgets are not sensitive to the diffusion steps, where we can get the best result in most situations within 30 diffusion steps.

6.4.4 ***Stability.*** In this study, we randomly initialized the parameters of three models - CQL, IQL, and DiffBid - and conducted thirty training trials for each to examine stability in performance. As depicted in Figure 3(b), the RL-based models, CQL and IQL, showed a tendency towards instability under varying random seeds. Notably, IQL demonstrated slightly better performance than CQL, which may be attributed to its design optimized for conservative regularization. Contrasting with these, the generative model DiffBid exhibited remarkable stability, with significantly fewer instances of failure compared to its RL counterparts.

6.5 Online A/B Test

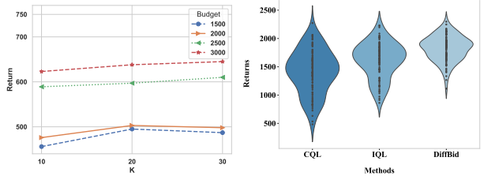
To further substantiate the effectiveness of DiffBid, we have deployed it on Alibaba advertising platform for comparison against the baseline IQL [29] method, which performs best among various auto-bidding methods. The online A/B test is conducted from February 01, 2024, to February 08, 2024. The results are shown in Table 3. It shows that DiffBid can significantly improve the Buycnt by 2.09%, the GMV by 2.81%, the ROI by 3.36%, showing its effectiveness in optimizing the overall performance. For efficiency, DiffBid takes 0.2s per request with GPU acceleration while the baseline is 0.07s, which means latency can be well guaranteed.

7 RELATED WORKS

Offline-Reinforcement Learning. Offline reinforcement learning is a research direction that has gained significant attention in recent years. The primary goal of offline RL is to learn effective policies from a fixed dataset without additional online interaction with the environment. This approach is particularly beneficial when online interaction is costly, risky, or otherwise not feasible.Notable works include Conservative Q-learning (CQL) by Kumar et al. [30], and Batch-Constrained deep Q-learning (BCQ) by Fujimoto et al. [14]. Both algorithms aim to tackle overestimation bias which tends to occur in offline RL settings. Kostrikov et al. [29] propose an implicit q-learning approach to address the training instability for CQL. Chen et al. [9] propose to use transformers for offline RL to increase the model capability. Hansen-Estruch et al. [19] proposes a diffusion-based approach with implicit Q-learning for offline RL.
Diffusion Models. They recently have shown the capability of high-quality generation [11], unconditional generation [3] and conditional generation [7, 25]. It has shown promising performance in decision-making. Hansen-Estruch et al. [19] proposes a diffusion-based approach with implicit q-learning for offline RL. Wang et al. [45] propose a expressive policy though diffusion modeling. Chen et al. [8] propose to use diffusion models for behavior modeling. Hu et al. [24] introduce temporal conditions for trajectory generation. Despite these preliminary explorations, no work has been payed for diffusion based auto-bidding which requires the model to adapt to the random advertising environment. Li et al. [32] utilize diffusion model in anti-money laundering.
Auto-bidding. Auto-bidding systems are widely used in programmatic advertising, where they are employed to automatically place bids on ad spaces. The main focus of such systems is to optimise a given key performance indicator (KPI), such as the number of clicks or conversions, while maintaining a certain budget [44]. Cai et al. [6] proposed an RL-based approach to the problem of auto-bidding for display advertising. They designed a bidding environment and applied a deep RL algorithm to learn the optimal bidding strategy. He et al. [21] propose a unified solution with RL to enable multiple constraints for auto-bidding. Jin et al. extend the RL to enable multi-agent competition [27]. Zhang et al. [35] also adopted the RL framework for auto-bidding and showed that their approach can outperform traditional bidding strategies. Wen et al. [46] propose a multi-agent-based approach for auto-bidding, which enables the modeling of multiple auto-bidding agents at the same time to include more information and also has been deployed online.

8 CONCLUSION

In this paper, we design a new paradigm for auto-bidding through the lens of generative modeling. To achieve this goal, we propose a decision-denoising diffusion approach to generate conditional bidding trajectories and at the same time control the generated samples under certain constraints. This new generative modeling approach enables integrating different kinds of industrial metrics,



（a）扩散步数的影响 （b）稳定性

图 6：深入分析。

Table 3: Online A/B Test Result.

Metrics	#Plan	Budget	Cost	Buycnt	GMV	ROI
Baseline	2068	886744	834426.104	23584.6836	1853823	2.221
DiffBid	2068	886744	829992.384	24078.6883	1905954	2.296
compare	-	-	-0.53%	+2.09%	+2.81%	+3.36%

反馈下的表现。在训练期间，我们通过式（14）中的阈值将轨迹划分为高、低两个水平，并学习不同水平下的条件分布。在生成期间，我们调整条件，生成相应的样本，并汇总指标。低水平、高水平以及原始轨迹的指标统计分布结果见图 5。我们发现，部署 DiffBid 所得到的轨迹能够受到条件的良好控制。

6.4.3 扩散步数的影响。我们还研究了不同扩散步数下的整体表现，扩散步数是影响效率和性能的重要因素。图 6 展示了扩散步数在不同预算下的总体影响。由图可得以下发现。首先，我们观察到，扩散步数对小预算（1500 元）广告主的影响更大。其次，较大预算对扩散步数不敏感，在大多数情况下，30 个扩散步以内即可获得最佳结果。

6.4.4 稳定性。在本研究中，我们随机初始化 CQL、IQL 和 DiffBid 三个模型的参数，并对每个模型开展三十次训练试验，以考察其性能稳定性。如图 3(b) 所示，基于 RL 的 CQL 和 IQL 模型在不同随机种子下表现出不稳定的倾向。值得注意的是，IQL 的表现略优于 CQL，这可能归因于其针对保守正则化所优化的设计。相比之下，生成式模型 DiffBid 展现了显著的稳定性，失败次数明显少于对应的 RL 模型。

6.5 在线 A/B 测试

为进一步验证 DiffBid 的有效性，我们已将其部署到阿里巴巴广告平台，与在各种自动出价方法中表现最佳的基线 IQL [29] 方法进行比较。在线 A/B 测试于 2024 年 2 月 01 日至 2024 年 2 月 08 日进行。结果见表 3。结果表明，DiffBid 能够显著提高购买次数（Buycnt）2.09%、商品交易总额（GMV）提升 2.81%、投资回报率（ROI）提升 3.36%，表明其在优化整体表现方面的有效性。在效率方面，使用 GPU 加速时，DiffBid 每次请求耗时 0.2 秒，而基线为 0.07 秒，这意味着延迟能够得到良好保障。

7 相关工作

离线强化学习。离线强化学习是近年来受到广泛关注的一个研究方向。离线强化学习的主要目标是从固定数据集中学习有效的策略，而无需额外与环境进行在线交互。当在线交互成本高、风险大或因其他原因不可行时，这种方法尤其有益。代表性工作包括 Kumar 等人提出的保守 Q 学习（CQL）[30]，以及 Fujimoto 等人提出的批次约束深度 Q 学习（BCQ）[14]。这两种算法都旨在解决离线强化学习场景中容易出现的过高估计偏差。Kostrikov 等人[29]提出了一种隐式 Q 学习方法，以解决 CQL 的训练不稳定问题。Chen 等人 [9] 提出在离线强化学习中使用 Transformer，以提升模型能力。Hansen-Estruch 等人[19]提出了一种结合隐式 Q 学习、基于扩散的离线强化学习方法。

扩散模型。扩散模型近来已展现出高质量生成[11]、无条件生成 [3] 和条件生成 [7, 25]的能力。在决策问题中，它也展现出很有前景的表现。Hansen-Estruch 等人[19]提出了一种结合隐式 Q 学习、基于扩散的离线强化学习方法。Wang 等人[45]通过扩散建模提出了一种具有强大表达能力的策略。Chen 等人 [8] 提出使用扩散模型进行行为建模。Hu 等人[24]为轨迹生成引入了时间条件。尽管已有这些初步探索，但尚无研究关注基于扩散的自动出价，而这一任务要求模型适应随机的广告环境。Li 等人[32]将扩散模型用于反洗钱。

自动出价。自动出价系统广泛应用于程序化广告，用于自动对广告位出价。此类系统的主要目标是在保持一定预算的同时，优化给定的关键绩效指标（KPI），例如点击次数或转化次数[44]。Cai 等人 [6] 针对展示广告中的自动出价问题，提出了一种基于强化学习的方法。他们设计了出价环境，并应用深度强化学习算法来学习最优出价策略。He 等人[21]提出了基于强化学习的统一解决方案，使自动出价能够支持多项约束。Jin 等人对强化学习进行了扩展，以支持多智能体竞争[27]。Zhang 等人[35]也采用强化学习框架实现自动出价，并表明其方法能够优于传统出价策略。Wen 等人[46]提出了一种基于多智能体的自动出价方法，能够同时对多个自动出价智能体进行建模，以纳入更多信息，并且已在线上部署。

8 结论

本文从生成式建模的角度设计了一种自动出价新范式。为实现这一目标，我们提出一种决策去噪扩散方法，用于生成条件出价轨迹，同时使生成样本满足一定的约束。这种新的生成式建模方法能够融合不同类型的工业指标，

which is the first unified model for bidding. Extensive experiments on real-world simulation environments demonstrate the effectiveness of the newly proposed approach. In the future, we will consider developing new methods to accelerate the generation process and new methods to ensure the robustness of DiffBid.

REFERENCES

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

[2] Anurag Ajay, Yilun Du, Abhi Gupta, Joshua B Tenenbaum, Tommi S Jaakkola, and Pulkit Agrawal. 2022. Is Conditional Generative Modeling all you need for Decision Making?. In *The Eleventh International Conference on Learning Representations*.

[3] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems* 34 (2021), 17981–17993.

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

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

[6] 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 *Proceedings of the tenth ACM international conference on web search and data mining*. 661–670.

[7] Chen-Hao Chao, Wei-Fang Sun, Bo-Wun Cheng, Yi-Chen Lo, Chia-Che Chang, Yu-Lun Liu, Yu-Lin Chang, Chia-Ping Chen, and Chun-Yi Lee. 2022. Denoising likelihood score matching for conditional score-based data generation. *arXiv preprint arXiv:2203.14206* (2022).

[8] Huayu Chen, Cheng Lu, Chengyang Ying, Hang Su, and Jun Zhu. 2022. Offline reinforcement learning via high-fidelity generative behavior modeling. *arXiv preprint arXiv:2209.14548* (2022).

[9] 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. *Advances in neural information processing systems* 34 (2021), 15084–15097.

[10] Harry L Chiesi, George J Spilich, and James F Voss. 1979. Acquisition of domain-related information in relation to high and low domain knowledge. *Journal of verbal learning and verbal behavior* 18, 3 (1979), 257–273.

[11] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. 2023. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

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

[13] David S Evans. 2009. The online advertising industry: Economics, evolution, and privacy. *Journal of economic perspectives* 23, 3 (2009), 37–60.

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

[15] Scott Fujimoto, David Meger, Doina Precup, Ofir Nachum, and Shixiang Shane Gu. 2022. Why Should I Trust You, Bellman? The Bellman Error is a Poor Replacement for Value Error. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.). PMLR, 6918–6943. https://proceedings.mlr.press/v162/fujimoto22a.html

[16] Maor Gaon and Ronen Brafman. 2020. Reinforcement learning with non-markovian rewards. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 3980–3987.

[17] Jiayan Guo, Yaming Yang, Xiangchen Song, Yuan Zhang, Yujing Wang, Jing Bai, and Yan Zhang. 2022. Learning Multi-granularity Consecutive User Intent Unit for Session-based Recommendation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining* (Virtual Event, AZ, USA) *(WSDM '22)*. Association for Computing Machinery, New York, NY, USA, 343–352. https://doi.org/10.1145/3488560.3498524

[18] Louisa Ha. 2008. Online advertising research in advertising journals: A review. *Journal of Current Issues & Research in Advertising* 30, 1 (2008), 31–48.

[19] Philippe Hansen-Estruch, Ilya Kostrikov, Michael Janner, Jakub Grudzien Kuba, and Sergey Levine. 2023. Idql: Implicit q-learning as an actor-critic method with diffusion policies. *arXiv preprint arXiv:2304.10573* (2023).

[20] Xiaotian Hao, Zhaoqing Peng, Yi Ma, Guan Wang, Junqi Jin, Jianye Hao, Shan Chen, Rongquan Bai, Mingzhou Xie, Miao Xu, Zhenzhe Zheng, Chuan Yu, Han Li, Jian Xu, and Kun Gai. 2020. Dynamic Knapsack Optimization Towards Efficient Multi-Channel Sequential Advertising. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 4060–4070. http://proceedings.mlr.press/v119/hao20b.html

[21] 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 *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2993–3001.

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

[23] Jonathan Ho and Tim Salimans. 2021. Classifier-Free Diffusion Guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*.

[24] Jifeng Hu, Yanchao Sun, Sili Huang, SiYuan Guo, Hechang Chen, Li Shen, Lichao Sun, Yi Chang, and Dacheng Tao. 2023. Instructed Diffuser with Temporal Condition Guidance for Offline Reinforcement Learning. *arXiv preprint arXiv:2306.04875* (2023).

[25] R Huang, MWY Lam, J Wang, D Su, D Yu, Y Ren, and Z Zhao. 2022. FastDiff: A Fast Conditional Diffusion Model for High-Quality Speech Synthesis. In *IJCAI International Joint Conference on Artificial Intelligence*. IJCAI: International Joint Conferences on Artificial Intelligence Organization, 4157–4163.

[26] Edwin T Jaynes. 1957. Information theory and statistical mechanics. *Physical review* 106, 4 (1957), 620.

[27] Junqi Jin, Chengru Song, Han Li, Kun Gai, Jun Wang, and Weinan Zhang. 2018. Real-time bidding with multi-agent reinforcement learning in display advertising. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 2193–2201.

[28] Durk P Kingma, Tim Salimans, and Max Welling. 2015. Variational dropout and the local reparameterization trick. *Advances in neural information processing systems* 28 (2015).

[29] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. 2021. Offline Reinforcement Learning with Implicit Q-Learning. In *International Conference on Learning Representations*.

[30] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. 2020. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems* 33 (2020), 1179–1191.

[31] Juncheng Li and Pingzhong Tang. 2022. Auto-bidding Equilibrium in ROI-Constrained Online Advertising Markets. *arXiv preprint arXiv:2210.06107* (2022).

[32] Xujia Li, Yuan Li, Xueying Mo, Hebing Xiao, Yanyan Shen, and Lei Chen. 2023. Diga: Guided diffusion model for graph recovery in anti-money laundering. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4404–4413.

[33] Sultan Javed Majeed and Marcus Hutter. 2018. On Q-learning Convergence for Non-Markov Decision Processes.. In *IJCAI*, Vol. 18. 2546–2552.

[34] Diganta Misra. 2019. Mish: A self regularized non-monotonic activation function. *arXiv preprint arXiv:1908.08681* (2019).

[35] Zhiyu Mou, Yusen Huo, Rongquan Bai, Mingzhou Xie, Chuan Yu, Jian Xu, and Bo Zheng. 2022. Sustainable Online Reinforcement Learning for Auto-bidding. *Advances in Neural Information Processing Systems* 35 (2022), 2651–2663.

[36] Mirco Mutti, Riccardo De Santi, and Marcello Restelli. 2022. The importance of non-markovianity in maximum state entropy exploration. In *International Conference on Machine Learning*. PMLR, 16223–16239.

[37] Duy Nguyen-Tuong, Jan Peters, Matthias Seeger, and Bernhard Schölkopf. 2008. Learning inverse dynamics: a comparison. In *European symposium on artificial neural networks*.

[38] Alexander Quinn Nichol and Prafulla Dhariwal. 2021. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*. PMLR, 8162–8171.

[39] Weitong Ou, Bo Chen, Yingxuan Yang, Xinyi Dai, Weiwen Liu, Weinan Zhang, Ruiming Tang, and Yong Yu. 2023. Deep landscape forecasting in multi-slot real-time bidding. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4685–4695.

[40] Deepak Pathak, Parsa Mahmoudieh, Guanghao Luo, Pulkit Agrawal, Dian Chen, Yide Shentu, Evan Shelhamer, Jitendra Malik, Alexei A Efros, and Trevor Darrell. 2018. Zero-shot visual imitation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2050–2053.

[41] Aoyang Qin, Feng Gao, Qing Li, Song-Chun Zhu, and Sirui Xie. 2023. Learning non-Markovian Decision-Making from State-only Sequences. In *Thirty-seventh Conference on Neural Information Processing Systems*.

[42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 234–241.

[43] Pascal Vincent. 2011. A connection between score matching and denoising autoencoders. *Neural computation* 23, 7 (2011), 1661–1674.

[44] Jun Wang, Weinan Zhang, Shuai Yuan, et al. 2017. Display advertising with real-time bidding (RTB) and behavioural targeting. *Foundations and Trends® in*

构成了首个用于出价的统一模型。在真实世界仿真环境中开展的大量实验证明了这一新方法的有效性。未来，我们将考虑开发加速生成过程的新方法，以及确保 DiffBid 鲁棒性的新方法。

REFERENCES

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

[2] Anurag Ajay, Yilun Du, Abhi Gupta, Joshua B Tenenbaum, Tommi S Jaakkola, and Pulkit Agrawal. 2022. Is Conditional Generative Modeling all you need for Decision Making?. In *The Eleventh International Conference on Learning Representations*.

[3] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems* 34 (2021), 17981–17993.

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

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

[6] 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 *Proceedings of the tenth ACM international conference on web search and data mining*. 661–670.

[7] Chen-Hao Chao, Wei-Fang Sun, Bo-Wun Cheng, Yi-Chen Lo, Chia-Che Chang, Yu-Lun Liu, Yu-Lin Chang, Chia-Ping Chen, and Chun-Yi Lee. 2022. Denoising likelihood score matching for conditional score-based data generation. *arXiv preprint arXiv:2203.14206* (2022).

[8] Huayu Chen, Cheng Lu, Chengyang Ying, Hang Su, and Jun Zhu. 2022. Offline reinforcement learning via high-fidelity generative behavior modeling. *arXiv preprint arXiv:2209.14548* (2022).

[9] 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. *Advances in neural information processing systems* 34 (2021), 15084–15097.

[10] Harry L Chiesi, George J Spilich, and James F Voss. 1979. Acquisition of domain-related information in relation to high and low domain knowledge. *Journal of verbal learning and verbal behavior* 18, 3 (1979), 257–273.

[11] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. 2023. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

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

[13] David S Evans. 2009. The online advertising industry: Economics, evolution, and privacy. *Journal of economic perspectives* 23, 3 (2009), 37–60.

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

[15] Scott Fujimoto, David Meger, Doina Precup, Ofir Nachum, and Shixiang Shane Gu. 2022. Why Should I Trust You, Bellman? The Bellman Error is a Poor Replacement for Value Error. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.). PMLR, 6918–6943. https://proceedings.mlr.press/v162/fujimoto22a.html

[16] Maor Gaon and Ronen Brafman. 2020. Reinforcement learning with non-markovian rewards. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 3980–3987.

[17] Jiayan Guo, Yaming Yang, Xiangchen Song, Yuan Zhang, Yujing Wang, Jing Bai, and Yan Zhang. 2022. Learning Multi-granularity Consecutive User Intent Unit for Session-based Recommendation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining* (Virtual Event, AZ, USA) *(WSDM '22)*. Association for Computing Machinery, New York, NY, USA, 343–352. https://doi.org/10.1145/3488560.3498524

[18] Louisa Ha. 2008. Online advertising research in advertising journals: A review. *Journal of Current Issues & Research in Advertising* 30, 1 (2008), 31–48.

[19] Philippe Hansen-Estruch, Ilya Kostrikov, Michael Janner, Jakub Grudzien Kuba, and Sergey Levine. 2023. Idql: Implicit q-learning as an actor-critic method with diffusion policies. *arXiv preprint arXiv:2304.10573* (2023).

[20] Xiaotian Hao, Zhaoqing Peng, Yi Ma, Guan Wang, Junqi Jin, Jianye Hao, Shan Chen, Rongquan Bai, Mingzhou Xie, Miao Xu, Zhenzhe Zheng, Chuan Yu, Han Li, Jian Xu, and Kun Gai. 2020. Dynamic Knapsack Optimization Towards Efficient Multi-Channel Sequential Advertising. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 4060–4070. http://proceedings.mlr.press/v119/hao20b.html

[21] 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 *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2993–3001.

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

[23] Jonathan Ho and Tim Salimans. 2021. Classifier-Free Diffusion Guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*.

[24] Jifeng Hu, Yanchao Sun, Sili Huang, SiYuan Guo, Hechang Chen, Li Shen, Lichao Sun, Yi Chang, and Dacheng Tao. 2023. Instructed Diffuser with Temporal Condition Guidance for Offline Reinforcement Learning. *arXiv preprint arXiv:2306.04875* (2023).

[25] R Huang, MWY Lam, J Wang, D Su, D Yu, Y Ren, and Z Zhao. 2022. FastDiff: A Fast Conditional Diffusion Model for High-Quality Speech Synthesis. In *IJCAI International Joint Conference on Artificial Intelligence*. IJCAI: International Joint Conferences on Artificial Intelligence Organization, 4157–4163.

[26] Edwin T Jaynes. 1957. Information theory and statistical mechanics. *Physical review* 106, 4 (1957), 620.

[27] Junqi Jin, Chengru Song, Han Li, Kun Gai, Jun Wang, and Weinan Zhang. 2018. Real-time bidding with multi-agent reinforcement learning in display advertising. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 2193–2201.

[28] Durk P Kingma, Tim Salimans, and Max Welling. 2015. Variational dropout and the local reparameterization trick. *Advances in neural information processing systems* 28 (2015).

[29] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. 2021. Offline Reinforcement Learning with Implicit Q-Learning. In *International Conference on Learning Representations*.

[30] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. 2020. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems* 33 (2020), 1179–1191.

[31] Juncheng Li and Pingzhong Tang. 2022. Auto-bidding Equilibrium in ROI-Constrained Online Advertising Markets. *arXiv preprint arXiv:2210.06107* (2022).

[32] Xujia Li, Yuan Li, Xueying Mo, Hebing Xiao, Yanyan Shen, and Lei Chen. 2023. Diga: Guided diffusion model for graph recovery in anti-money laundering. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4404–4413.

[33] Sultan Javed Majeed and Marcus Hutter. 2018. On Q-learning Convergence for Non-Markov Decision Processes.. In *IJCAI*, Vol. 18. 2546–2552.

[34] Diganta Misra. 2019. Mish: A self regularized non-monotonic activation function. *arXiv preprint arXiv:1908.08681* (2019).

[35] Zhiyu Mou, Yusen Huo, Rongquan Bai, Mingzhou Xie, Chuan Yu, Jian Xu, and Bo Zheng. 2022. Sustainable Online Reinforcement Learning for Auto-bidding. *Advances in Neural Information Processing Systems* 35 (2022), 2651–2663.

[36] Mirco Mutti, Riccardo De Santi, and Marcello Restelli. 2022. The importance of non-markovianity in maximum state entropy exploration. In *International Conference on Machine Learning*. PMLR, 16223–16239.

[37] Duy Nguyen-Tuong, Jan Peters, Matthias Seeger, and Bernhard Schölkopf. 2008. Learning inverse dynamics: a comparison. In *European symposium on artificial neural networks*.

[38] Alexander Quinn Nichol and Prafulla Dhariwal. 2021. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*. PMLR, 8162–8171.

[39] Weitong Ou, Bo Chen, Yingxuan Yang, Xinyi Dai, Weiwen Liu, Weinan Zhang, Ruiming Tang, and Yong Yu. 2023. Deep landscape forecasting in multi-slot real-time bidding. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4685–4695.

[40] Deepak Pathak, Parsa Mahmoudieh, Guanghao Luo, Pulkit Agrawal, Dian Chen, Yide Shentu, Evan Shelhamer, Jitendra Malik, Alexei A Efros, and Trevor Darrell. 2018. Zero-shot visual imitation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2050–2053.

[41] Aoyang Qin, Feng Gao, Qing Li, Song-Chun Zhu, and Sirui Xie. 2023. Learning non-Markovian Decision-Making from State-only Sequences. In *Thirty-seventh Conference on Neural Information Processing Systems*.

[42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 234–241.

[43] Pascal Vincent. 2011. A connection between score matching and denoising autoencoders. *Neural computation* 23, 7 (2011), 1661–1674.

[44] Jun Wang, Weinan Zhang, Shuai Yuan, et al. 2017. Display advertising with real-time bidding (RTB) and behavioural targeting. *Foundations and Trends® in*

Information Retrieval 11, 4-5 (2017), 297–435.
[45] Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. 2022. Diffusion Policies as an Expressive Policy Class for Offline Reinforcement Learning. In *The Eleventh International Conference on Learning Representations*.
[46] 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 *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 1129–1139.
[47] Yuxin Wu and Kaiming He. 2018. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*. 3–19.
[48] Haoqi Zhang, Lvyin Niu, Zhenzhe Zheng, Zhilin Zhang, Shan Gu, Fan Wu, Chuan Yu, Jian Xu, Guihai Chen, and Bo Zheng. 2023. A Personalized Automated Bidding Framework for Fairness-aware Online Advertising. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 5544–5553.
[49] Peiyan Zhang, Jiayan Guo, Chaozhuo Li, Yueqi Xie, Jae Boum Kim, Yan Zhang, Xing Xie, Haohan Wang, and Sunghun Kim. 2023. Efficiently Leveraging Multi-level User Intent for Session-based Recommendation via Atten-Mixer Network. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining* (, Singapore, Singapore,) *(WSDM '23)*. Association for Computing Machinery, New York, NY, USA, 168–176. https://doi.org/10.1145/3539597.3570445
[50] Brian D. Ziebart. 2010. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Ph. D. Dissertation. USA. Advisor(s) Bagnell, J. Andrew. AAI3438449.

A APPENDIX

A.1 Notations

Table 4: Definition of Notations.

Symbol	Definition
τ	The trajectory index of a serving policy.
$\boldsymbol{x}(\tau)_k$	Sequence of states of trajectory τ in diffusion step k.
$\boldsymbol{y}(\tau)$	Properties or conditions for τ.
R	Return of a trajectory.
E	Binary indicator variable.
B	The budget of the advertiser.
C_i	The i's constraint.
$\boldsymbol{o}_i$	Whether the advertiser wins impression i.
$\boldsymbol{v}_i$	The true value of the impression i.
$\boldsymbol{b}_i^*$	The optimal bidding price for the impression i.
$\boldsymbol{s}_t$	The state at time period t.
$\hat{\boldsymbol{a}}_t$	Predicted bidding parameters at time period t.
λ_i	The bidding parameters.
$\boldsymbol{\epsilon}_\theta$	The denoising model that predict the noise.
f_ϕ	The model that generate bids.
$\overline{\alpha}_k$	The cumulative product of $1-\beta_j, j=0,...,k$
β_k	Schedualing factors.
α_k	1-β_k.

A.2 Diffusion Modeling

As a kind of generative model, diffusion models [22, 43] use the diffusion process to gradually denoise latent samples to generate the new sample and have been widely used in generating pictures, videos, and audio. One of the widely used diffusion models, denoising diffusion probabilistic model (DDPM), consists of two processes:

Forward process. In the forward process, the noise is gradually added to the latent variable, which is parameterized by a Markov chain with the transition $q(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) = \mathcal{N}\left(x_k; \sqrt{1-\beta_k}\boldsymbol{x}_k, \beta_k I\right)$, where $k \in \{1, ...K\}$ refers to the diffusion step, and $\beta_k \in (0,1)$ is a pre-defined scale that controls the noise scale at step k. By defining $\overline{\alpha}_k = \prod_{i=1}^{k} \alpha_i = \prod_{i=1}^{k}(1-\beta_i)$, we can have the conditional distribution:

$$q(x_k|x_0) = \mathcal{N}\left(x_k; \sqrt{\overline{\alpha}_k}x_0, (1-\overline{\alpha}_k)I\right) \tag{16}$$

In this paper, we apply cosine noise schedule to control the noise by:

$$\overline{\alpha}_k = \frac{g(t)}{g(0)} = \frac{\cos\left(\frac{k/K+\gamma}{1+\gamma}\cdot\frac{\pi}{2}\right)}{\cos\left(\frac{\gamma}{1+\gamma}\cdot\frac{\pi}{2}\right)}, \tag{17}$$

where γ is a constant. When $K \to \infty$, $q(x_K)$ approaches to a standard Gaussian distribution [22]. Given the original trajectory x_0 and $\epsilon \sim \mathcal{N}(0, I)$, we have the noisy version at k by $x_k = \sqrt{\overline{\alpha}_k}x_0 + \epsilon\sqrt{1-\overline{\alpha}_k}$

Reverse process. In the reverse process, diffusion models plan to remove the added noise on x_k and recursively recover x_{k-1}. To achieve this goal, a Gaussian distribution parameterized by $p_\theta(\boldsymbol{x}_{k-1}|\boldsymbol{x}_k) = \mathcal{N}\left(\boldsymbol{x}_{k-1}|\boldsymbol{\mu}_\theta(\boldsymbol{x}_k,k), \Sigma_\theta(\boldsymbol{x}_k,k)\right)$ is learned, where $\boldsymbol{\mu}_\theta(\boldsymbol{x}_k,k)$ is the learned mean and $\Sigma_\theta(\boldsymbol{x}_k,k)$ is the learned covariance of the Gaussian distribution parameterized by a neural network with parameter θ. For generating new samples, we can simply use re-parameterization trick [28] to sample a noise $\boldsymbol{x}_K \sim \boldsymbol{\mu}_K + \epsilon\sigma_K$ and recursively denoise the sample by $p_\theta(\boldsymbol{x}_{k-1}|\boldsymbol{x}_k)$ for generation.

Optimization. DDPM optimizes the Evidence Lower BOund (ELBO) of generative models. In the context of the diffusion model, we can take the latent samples as hidden variables and rewrite ELBO in the following form:

$$\begin{aligned}
&\mathbb{E}_q\left[-\log p_\theta(\boldsymbol{x_0})\right] \\
\leq &\mathbb{E}_q\left[-\log\frac{p_\theta(\boldsymbol{x}_{0:K})}{q(\boldsymbol{x}_{1:K}|\boldsymbol{x}_0)}\right] \\
= &\mathbb{E}_q\left[D_{KL}(q(\boldsymbol{x}_K|\boldsymbol{x}_0)||p_\theta(\boldsymbol{x}_K))\right] - \mathbb{E}_q\left[\log p_\theta\left(\boldsymbol{x}_0|\boldsymbol{x}_1\right)\right] \\
+ &\mathbb{E}_q\left[\sum_{t>1} D_{KL}\left(q(\boldsymbol{x}_{k-1}|\boldsymbol{x}_k,\boldsymbol{x}_0)||p_\theta(\boldsymbol{x}_{k-1}|\boldsymbol{x}_k)\right)\right],
\end{aligned} \tag{18}$$

where the first term has no learned variable given variance β_k is fixed to constants, thus can be ignored during training. The second term is the reconstruction term where $p_\theta(\cdot)$ is trained to recover the original sample $\boldsymbol{x}_0$ from the noise sample $\boldsymbol{x}_1$. The last term is the denoising term where $p_\theta(\cdot)$ is trained to denoise $\boldsymbol{x}_k$ to get $\boldsymbol{x}_{k-1}$, thus we can recurrently denoise the latent samples. In the original paper [22] the author shows that the last term can be simplified to the noise prediction objective $\mathbb{E}_{k,\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\sqrt{\overline{\alpha}_k}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_k}\boldsymbol{\epsilon}, k\right)||\right]$, where $\overline{\alpha}_k = \prod_{i=1}^{k}\alpha_k = \prod_{i=1}^{k}(1-\beta_k)$.

A.3 Model Configuration

We parameterize the noise model ϵ_θ with a temporal U-Net [42], consisting of 3 repeated residual blocks. Each block is consisted of two temporal convolutions, followed by group normalization [47], and a final Mish activation function [34]. Timestamp and condition embeddings, both 128-dimensional vectors, are produced by separate 2-layered MLP (with 256 hidden units and Mish activation function) and are concatenated together before getting added to

KDD '24, August 25–29, 2024, Barcelona, Spain Jiayan and Yusen, et al.

Information Retrieval 11, 4-5 (2017), 297–435.
[45] Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. 2022. Diffusion Policies as an Expressive Policy Class for Offline Reinforcement Learning. In *The Eleventh International Conference on Learning Representations*.
[46] 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 *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 1129–1139.
[47] Yuxin Wu and Kaiming He. 2018. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*. 3–19.
[48] Haoqi Zhang, Lvyin Niu, Zhenzhe Zheng, Zhilin Zhang, Shan Gu, Fan Wu, Chuan Yu, Jian Xu, Guihai Chen, and Bo Zheng. 2023. A Personalized Automated Bidding Framework for Fairness-aware Online Advertising. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 5544–5553.
[49] Peiyan Zhang, Jiayan Guo, Chaozhuo Li, Yueqi Xie, Jae Boum Kim, Yan Zhang, Xing Xie, Haohan Wang, and Sunghun Kim. 2023. Efficiently Leveraging Multi-level User Intent for Session-based Recommendation via Atten-Mixer Network. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining* (, Singapore, Singapore,) *(WSDM '23)*. Association for Computing Machinery, New York, NY, USA, 168–176. https://doi.org/10.1145/3539597.3570445
[50] Brian D. Ziebart. 2010. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Ph. D. Dissertation. USA. Advisor(s) Bagnell, J. Andrew. AAI3438449.

A 附录 A.1
符号说明

Table 4: Definition of Notations.

Symbol	Definition
τ	The trajectory index of a serving policy.
$\boldsymbol{x}(\tau)_k$	Sequence of states of trajectory τ in diffusion step k.
$\boldsymbol{y}(\tau)$	Properties or conditions for τ.
R	Return of a trajectory.
E	Binary indicator variable.
B	The budget of the advertiser.
C_i	The i's constraint.
$\boldsymbol{o}_i$	Whether the advertiser wins impression i.
$\boldsymbol{v}_i$	The true value of the impression i.
$\boldsymbol{b}_i^*$	The optimal bidding price for the impression i.
$\boldsymbol{s}_t$	The state at time period t.
$\hat{\boldsymbol{a}}_t$	Predicted bidding parameters at time period t.
λ_i	The bidding parameters.
$\boldsymbol{\epsilon}_\theta$	The denoising model that predict the noise.
f_ϕ	The model that generate bids.
$\overline{\alpha}_k$	The cumulative product of $1-\beta_j, j=0,...,k$
β_k	Schedualing factors.
α_k	1-β_k.

A.2 扩散建模

扩散模型 [22, 43] 是一种生成式模型，利用扩散过程逐步对潜在样本去噪，以生成新样本，已被广泛用于生成图片、视频和音频。去噪扩散概率模型（DDPM）是一种广泛使用的扩散模型，包含两个过程：

前向过程。在前向过程中，噪声被逐渐加入潜变量，这一过程由具有如下转移的马尔可夫链进行参数化：

$$q(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) = \mathcal{N}\left(x_k\sqrt{1-\beta_k}\boldsymbol{x}_k, \beta_k I\right) \quad ; \quad ,$$

其中， $k \in \{1, ...K\}$表示扩散步， $\beta_k \in (0,1)$是预先定义的尺度，用于控制第 k步的噪声尺度。定义 $\overline{\alpha}_k = \prod_{i=1}^{k}\alpha_i = \prod_{i=1}^{k}(1-\beta_i)$后，可得条件分布：

$$q(x_k|x_0) = \mathcal{N}\left(x_k; \sqrt{\overline{\alpha}_k}x_0, (1-\overline{\alpha}_k)I\right) \tag{16}$$

本文采用余弦噪声调度来控制噪声，具体如下：

$$\overline{\alpha}_k = \frac{g(t)}{g(0)} = \frac{\cos\left(\frac{k/K+\gamma}{1+\gamma}\cdot\frac{\pi}{2}\right)}{\cos\left(\frac{\gamma}{1+\gamma}\cdot\frac{\pi}{2}\right)}, \tag{17}$$

其中， γ是一个常数。当 $K \to \infty$时， $q(x_K)$趋近于标准高斯分布 [22]。给定原始轨迹x_0和 $\epsilon \sim \mathcal{N}(0,I)$，其在 k处的含噪版本可通过 x_k =得到： $\sqrt{\overline{\alpha}_k}x_0 + \epsilon\sqrt{1-\overline{\alpha}_k}$

反向过程。在反向过程中，扩散模型旨在去除加到 x_k上的噪声，并递归恢复 x_{k-1}。为实现这一目标，需要学习一个由 $p_\theta(\boldsymbol{x}_{k-1}|\boldsymbol{x}_k) = \mathcal{N}\left(\boldsymbol{x}_{k-1}|\boldsymbol{\mu}_\theta(\boldsymbol{x}_k,k), \Sigma_\theta(\boldsymbol{x}_k,k)\right)$参数化的高斯分布，其中， $\boldsymbol{\mu}_\theta(\boldsymbol{x}_k,k)$是学习得到的均值， $\Sigma_\theta(\boldsymbol{x}_k,k)$是学习得到的协方差，该高斯分布由参数为 θ的神经网络参数化。为了生成新样本，只需使用重参数化技巧 [28] 采样噪声 $\boldsymbol{x}_K \sim \boldsymbol{\mu}_K + \epsilon\sigma_K$，再通过 $p_\theta(\boldsymbol{x}_{k-1}|\boldsymbol{x}_k)$递归地对样本去噪，即可完成生成。

优化。DDPM 优化生成模型的证据下界（ELBO）。在扩散模型的语境下，我们可以将潜在样本视为隐变量，并将 ELBO 改写为以下形式：

$$\begin{aligned}
&\mathbb{E}_q\left[-\log p_\theta(\boldsymbol{x_0})\right] \\
\leq &\mathbb{E}_q\left[-\log\frac{p_\theta(\boldsymbol{x}_{0:K})}{q(\boldsymbol{x}_{1:K}|\boldsymbol{x}_0)}\right] \\
= &\mathbb{E}_q\left[D_{KL}(q(\boldsymbol{x}_K|\boldsymbol{x}_0)||p_\theta(\boldsymbol{x}_K))\right] - \mathbb{E}_q\left[\log p_\theta\left(\boldsymbol{x}_0|\boldsymbol{x}_1\right)\right] \\
+ &\mathbb{E}_q\left[\sum_{t>1} D_{KL}\left(q(\boldsymbol{x}_{k-1}|\boldsymbol{x}_k,\boldsymbol{x}_0)||p_\theta(\boldsymbol{x}_{k-1}|\boldsymbol{x}_k)\right)\right],
\end{aligned} \tag{18}$$

其中，由于方差 β_k固定为常数，第一项不包含需要学习的变量，因此可以在训练时忽略。第二项是重建项，其中通过训练 $p_\theta(\cdot)$来恢复原始样本 $\boldsymbol{x}_0$，所依据的是含噪样本 $\boldsymbol{x}_1$。最后一项是去噪项，其中通过训练 $p_\theta(\cdot)$对 $\boldsymbol{x}_k$去噪，从而得到 $\boldsymbol{x}_{k-1}$，因此我们可以递归地对潜在样本去噪。在原始论文 [22] 中，作者表明，最后一项可以简化为

噪声预测目标 $\mathbb{E}_{k,\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\sqrt{\overline{\alpha}_k}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_k}\boldsymbol{\epsilon}, k\right)||\right]$ ，

其中， $\overline{\alpha}_k = \prod_{i=1}^{k}\alpha_k = \prod_{i=1}^{k}(1-\beta_k)$。

A.3 模型配置

我们使用时序 U-Net 对噪声模型 ϵ_θ进行参数化 [42]，该网络由 3 个重复的残差块组成。每个块包含两个时序卷积，后接组归一化 [47]，最后使用 Mish 激活函数 [34]。时间戳嵌入和条件嵌入均为 128 维向量，分别由独立的两层 MLP（含 256 个隐藏单元，使用 Mish 激活函数）生成，两者拼接后被加到

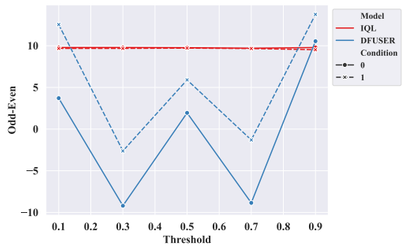


Figure 7: Ability of Action Control.

the activation functions of the first temporal convolution within each block. f_ϕ is parameterized with a 3-layer MLP.

A.4 Pseudo-code

The process of training and inference of DiffBid is shown in Algorithm 2 and Algorithm 1 respectively.

Algorithm 1 Bid Generation with DiffBid.

Input: noise model ϵ_θ, inverse dynamics f_ϕ, guidance scale ω, condition $\boldsymbol{y}$, max diffusion step K, scales $\beta_t,\ t = 1, ..., K$.
Output: Bidding parameters $\boldsymbol{a}_t$

Get history of states $\boldsymbol{s}_{0:t}$;
Sample $\boldsymbol{x}_K(\tau) \sim \mathcal{N}(0, \beta_K I)$;
for $k = K, ..., 0$ **do**
  $\boldsymbol{x}_k(\tau)[:t] \leftarrow \boldsymbol{s}_{0:t}$
  Estimating noise $\hat{\epsilon}$ through Eq. (8)
  $(\mu_{k-1}, \Sigma_{k-1}) \leftarrow \text{Denoise}(\boldsymbol{x}_k(\tau), \hat{\epsilon}_k)$
  $\boldsymbol{x}_{k-1} \sim \mathcal{N}(\mu_{k-1}, \alpha\Sigma_{k-1})$
end for
Extract $(\boldsymbol{s}_{t-L:t}, \boldsymbol{s}'_{t+1})$ from $\boldsymbol{x}_0(\tau)$
Generate $\hat{\boldsymbol{a}}_t = f_\phi(\boldsymbol{s}_{t-L:t}, \boldsymbol{s}'_{t+1})$;
return $\hat{\boldsymbol{a}}_t$.

Algorithm 2 Training of DiffBid.

Input: randomly initialized θ, ϕ, bidding trajectory set $\mathcal{D}$
Output: optimized θ, ϕ

while not converge **do**
  Sample a batch of trajectories $\mathcal{B} \in \mathcal{D}$;
  for all $\tau \in \mathcal{B}$ **do**
    Sample $k \sim \text{Uniform}(1, K)$, $\epsilon \sim \mathcal{N}(0, I)$;
    Compute $\boldsymbol{x}_k(\tau)$ via $q(\boldsymbol{x}_k(\tau)|\boldsymbol{x}_0(\tau))$ in Eq (7);
    Compute $\mathcal{L}(\theta, \phi)$ by Eq (12);
    Perform gradient descent to optimize θ and ϕ;
  end for
end while
return optimized θ, ϕ

A.5 Analytical Results for Action Control

We analyze the ability of different models in controlling actions. To achieve this goal, we re-define the return function to be the summation of actions in odd time steps minus the summation of actions in even time steps. The results are shown in Figure 7. We find DiffBid can better control the action than IQL. The main reason is that controlling of actions is difficult for RL in long horizons. Instead, DiffBid directly models the correlation of trajectories and returns, thus can well handle the long trajectory situation.

A.6 Statistical Analyses for Bidding Trajectory

The study by [20] indicates that CE follows a power-law decline as the number of winning impressions increases. Our statistical analysis confirms that this finding holds true at every discrete time step, with decay rates varying temporally due to the heterogeneous nature of the impressions. Figure 8(a) shows three steps sampled from the online advertising system, From which another key insight is that the optimal bidding strategy's $ce*$ is equivalent to selecting a specific number of winning impressions per time step. We denote the number at time step t as n_t.

Another finding illustrated in Figure 8(b) is that the costs of impressions remain relatively stable throughout the total episode, fluctuating by less than 5%. This stability allows us to approximate the cost of each impression c_i with the average cost $\bar{c} = \frac{1}{N}\sum_{i=0}^{N} c_i$, where N represents the number of winning impressions of the total episode. Therefore, the total cost at each time step $c_t = n_t \cdot \bar{c}$. In auto-bidding modeling, c_t can be calculated from the state trajectory by using the difference in the remaining budget between two consecutive steps. Consequently, we can conclude that the optimal strategy correlates to a specific state trajectory.

A.7 Theoretical Analysis

We first give the definition of several decision process and then show the theoretical analysis.

Definition A.1 (Markovian Decision Process (MDP)). *MDP is a stochastic mapping from a state-action pair to state-reward pairs. Formally, $\mathcal{T} : \mathcal{S}\times\mathcal{A} \to \mathcal{S}\times\mathcal{R}$, where $\mathcal{T}$ denotes a stochastic mapping.*

Definition A.2 (History-based Decision Process (HDP)). *HDP is a stochastic mapping from a history-action pair to observation-reward pairs. Formally, $\mathcal{P} : \mathcal{H}^* \times \mathcal{A} \to \mathcal{O}\times\mathcal{R}$, where $\mathcal{P}$ denotes a stochastic mapping.*

We show that a sequential decision-making problem can be constructed to maximize the same objective. The main results are given by [41] and we put the proofs here for completeness. To start, let the ground-truth distribution of demonstrations be $p^*(\boldsymbol{x}_0(\tau))$ and the learned marginal distributions of state sequences be $p_\theta(\boldsymbol{x}_0(\tau))$. Then Eq. (4) is an empirical estimation of

$$\mathbb{E}_{p^*(\boldsymbol{s}_0)}\left[\log p^*(\boldsymbol{s}_0) + \mathbb{E}_{p^*(\boldsymbol{s}_{1:T}|\boldsymbol{s}_0)}\left[\log p_\theta(\boldsymbol{s}_{1:T}|\boldsymbol{s}_0)\right]\right] \tag{19}$$

Suppose the MLE yields the maximum, we will have $p^*_\theta = p^*$. Then we define $V^*(s_0) := \mathbb{E}_{p^*(s_{1:T}|s_0)}[\log p^*(s_{1:T}|s_0)]$, and generalize it to have a V function:

$$V^*(s_{0:t}) = \mathbb{E}_{p^*(s_{t+1:T}|s_{0:t})}[\log p^*(s_{t+1:T}|s_{0:t})] \tag{20}$$

which comes with a Bellman optimal equation:

$$V^*(s_{0:t}) := \mathbb{E}_{p^*(s_{t+1}|s_{0:t})}[r(s_{t+1}, s_{0:t}) + V^*(s_{0:t+1})] \tag{21}$$

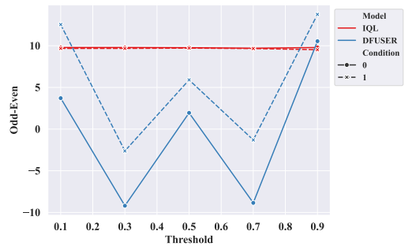


图 7：动作控制能力。

每个块中第一个时序卷积的激活函数上。f_ϕ使用一个三层 MLP 进行参数化。

A.4 伪代码

DiffBid 的训练和推理过程分别如算法 2 和算法 1 所示。

Algorithm 1 Bid Generation with DiffBid.

Input: noise model ϵ_θ, inverse dynamics f_ϕ, guidance scale ω, condition $\boldsymbol{y}$, max diffusion step K, scales $\beta_t,\ t = 1, ..., K$.
Output: Bidding parameters $\boldsymbol{a}_t$

Get history of states $\boldsymbol{s}_{0:t}$;
Sample $\boldsymbol{x}_K(\tau) \sim \mathcal{N}(0, \beta_K I)$;
for $k = K, ..., 0$ **do**
  $\boldsymbol{x}_k(\tau)[:t] \leftarrow \boldsymbol{s}_{0:t}$
  Estimating noise $\hat{\epsilon}$ through Eq. (8)
  $(\mu_{k-1}, \Sigma_{k-1}) \leftarrow \text{Denoise}(\boldsymbol{x}_k(\tau), \hat{\epsilon}_k)$
  $\boldsymbol{x}_{k-1} \sim \mathcal{N}(\mu_{k-1}, \alpha\Sigma_{k-1})$
end for
Extract $(\boldsymbol{s}_{t-L:t}, \boldsymbol{s}'_{t+1})$ from $\boldsymbol{x}_0(\tau)$
Generate $\hat{\boldsymbol{a}}_t = f_\phi(\boldsymbol{s}_{t-L:t}, \boldsymbol{s}'_{t+1})$;
return $\hat{\boldsymbol{a}}_t$.

Algorithm 2 Training of DiffBid.

Input: randomly initialized θ, ϕ, bidding trajectory set $\mathcal{D}$
Output: optimized θ, ϕ

while not converge **do**
  Sample a batch of trajectories $\mathcal{B} \in \mathcal{D}$;
  for all $\tau \in \mathcal{B}$ **do**
    Sample $k \sim \text{Uniform}(1, K)$, $\epsilon \sim \mathcal{N}(0, I)$;
    Compute $\boldsymbol{x}_k(\tau)$ via $q(\boldsymbol{x}_k(\tau)|\boldsymbol{x}_0(\tau))$ in Eq (7);
    Compute $\mathcal{L}(\theta, \phi)$ by Eq (12);
    Perform gradient descent to optimize θ and ϕ;
  end for
end while
return optimized θ, ϕ

A.5 动作控制的分析结果

我们分析了不同模型控制动作的能力。为实现这一目标，我们将回报函数重新定义为奇数时间步的动作之和减去偶数时间步的动作之和。结果如图 7 所示。我们发现，DiffBid 比 IQL 更能有效地控制动作。主要原因在于，强化学习在较长时间跨度内很难控制动作。相比之下，DiffBid 直接对轨迹与回报的相关性建模，因此能够很好地处理长轨迹情形。

A.6 出价轨迹的统计分析

研究 [20] 表明，随着赢得曝光的数量增加，成本效益（CE）呈幂律下降。我们的统计分析证实，该发现对每个离散时间步都成立，但由于曝光具有异质性，衰减率会随时间变化。图 8(a) 展示了从在线广告系统中采样的三个时间步，由此得到的另一个关键认识是，最优出价策略的 $ce*$等价于在每个时间步选择特定数量的获胜曝光。我们将时间步 t的这一数量记为 n_t。

图 8(b) 展示的另一项发现是，曝光成本在整个回合中保持相对稳定，波动小于 5%。这一稳定性使我们可以用平均成本来近似每次曝光的成本 c_i，该平均成本为$\bar{c} = \frac{1}{N}\sum_{i=0}^{N} c_i$，其中 N表示整个回合中获胜曝光的数量。因此，每个时间步的总成本为 $c_t = n_t \cdot \bar{c}$。在自动出价建模中，可以利用连续两个时间步之间剩余预算的差值，从状态轨迹中计算 c_t。因此，我们可以得出结论：最优策略与特定的状态轨迹相关联。

A.7 理论分析

我们首先给出几种决策过程的定义，随后进行理论分析。

定义 A.1（马尔可夫决策过程（MDP））。MDP 是从状态 – 动作对到状态 – 奖励对的随机映射。形式化地，
$\mathcal{T} : \mathcal{S}\times\mathcal{A} \to \mathcal{S}\times\mathcal{R}$，其中 $\mathcal{T}$ 表示随机映射。

定义 A.2（基于历史的决策过程（HDP））。HDP 是从历史 – 动作对到观测 – 奖励对的随机映射。形式化地，
$\mathcal{P} : \mathcal{H}^* \times \mathcal{A} \to \mathcal{O}\times\mathcal{R}$，其中 $\mathcal{P}$表示随机映射。

我们证明，可以构造一个序列决策问题，以最大化同一个目标。主要结果由[41]给出，为保证完整性，我们在此列出证明。首先，设示范数据的真实分布为 $p^*(\boldsymbol{x}_0(\tau))$，学习得到的状态序列边缘分布为 $p_\theta(\boldsymbol{x}_0(\tau))$。那么，式（4）是以下表达式的经验估计：

$$\mathbb{E}_{p^*(\boldsymbol{s}_0)}\left[\log p^*(\boldsymbol{s}_0) + \mathbb{E}_{p^*(\boldsymbol{s}_{1:T}|\boldsymbol{s}_0)}\left[\log p_\theta(\boldsymbol{s}_{1:T}|\boldsymbol{s}_0)\right]\right] \tag{19}$$

假设 MLE 达到最大值，则有 $p^*_\theta = p^*$。随后，我们定义 $V^*(s_0) := \mathbb{E}_{p^*(s_{1:T}|s_0)}[\log p^*(s_{1:T}|s_0)]$，并将其推广为一个 V函数：

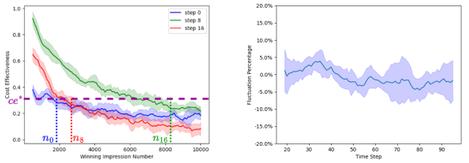
$$V^*(s_{0:t}) = \mathbb{E}_{p^*(s_{t+1:T}|s_{0:t})}[\log p^*(s_{t+1:T}|s_{0:t})] \tag{20}$$

其对应的贝尔曼最优方程为：

$$V^*(s_{0:t}) := \mathbb{E}_{p^*(s_{t+1}|s_{0:t})}[r(s_{t+1}, s_{0:t}) + V^*(s_{0:t+1})] \tag{21}$$

Table 5: Parameters of the Real Advertising System.

Parameters	Values
Number of advertisers	30
Time steps in an episode, T	96
Minimum number of impression opportunities $N_{\min}$	50
Maximum number of impression opportunities $N_{\max}$	300
Minimum budget	1000 Yuan
Maximum budget	4000 Yuan
Value of impression opportunities in stage 1, $v^1_{j,t}$	0∼ 1
Value of impression opportunities in stage 2, $v^2_{j,t}$	0∼ 1
Minimum bid price, $A_{\min}$	0 Yuan
Maximum bid price, $A_{\max}$	1000 Yuan
Maximum value of impression opportunity, v_M	1
Maximum market price, p_M	1000 Yuan



(a) Cost Effectiveness Curve Samples from Three Steps (b) Impression Cost Fluctuation at Different Time Steps

Figure 8: Statistical Results From Online Advertising System.

with $r(s_{t+1}, s_{0:t}) := \log p^*(s_{t+1}|s_{0:t}) = \log p^*_a(s_t|s_{0:t})p^*(s_{t+1}|s_t, a_t)dt$, $V^*(s_{0:T}) := 0$. It is worth noting that the r defined above involves the optimal policy, which may not be known a priori. We can resolve this by replacing it with r_α for an arbitrary policy $p_\alpha(a_t|s_{0:t})$. All Bellman identities and updates should still hold. The entailed Bellman update, *value iteration*, for arbitrary V and α is

$$V(s_{0:t}) = \mathbb{E}_{p^*(s_{t+1}|s_{0:t})}[r_\alpha(s_{0:t}, s_{t+1}) + V(s_{0:t+1})]. \quad (22)$$

We then define $r(s_{t+1}, a_t, s_{0:t}) := r(s_{t+1}, s_{0:t}) + \log p^*_a(a_t|s_{0:t})$ to construct a Q function:

$$Q^*(a_t; s_{0:t}) := \mathbb{E}_{p^*(s_{t+1}|s_{0:t})}[r(s_{t+1}, a_t, s_{0:t}) + V^*(s_{0:t+1})], \quad (23)$$

which entails a Bellman update, *Q backup*, for arbitrary α, Q and V

$$Q(a_t; s_{0:t}) = \mathbb{E}_{p^*(s_{t+1}|s_{0:t})}[r_\alpha(s_{0:t}, a_t, s_{t+1}) + V(s_{0:t+1})]. \quad (24)$$

Also note that the V and Q in identities Eq. (23) and Eq. (25) respectively are not necessarily associated with the policy $p_\alpha(a_t|s_{0:t})$. Slightly overloading the notations, we use Q_α, V_α to denote the expected returns from policy $p_\alpha(a_t|s_{0:t})$. By now, we finish the construction of atomic algebraic components and move on to check if the relations between them align with the algebraic structure of a sequential decision-making problem. We first prove the construction above is valid at optimality.

Lemma A.1. *When $f_\alpha(a_t; s_{0:t}) = Q^*(a_t; s_{0:t}) - V^*(s_{0:t}), p_\alpha(a_t|s_{0:t})$ is the optimal policy.*

Proof. Note that the construction gives us

$$\begin{aligned} & Q^*(a_t; s_{0:t}) \\ =& \mathbb{E}_{p^*(s_{t+1}|s_{0:t})}\left[r(s_{t+1}, s_{0:t}) + \log p^*_\alpha(a_t|s_{0:t}) + V^*(s_{0:t+1})\right] \\ =& \log p^*_\alpha(a_t|s_{0:t}) + \mathbb{E}_{p^*(s_{t+1}|s_{0:t})}\left[r(s_{t+1}, s_{0:t}) + V^*(s_{0:t+1})\right] \\ =& \log p^*_\alpha(a_t|s_{0:t}) + V^*(s_{0:t}) \end{aligned} \quad (25)$$

□

Obviously, $Q^*(a_t; s_{0:t})$ lies in the hypothesis space of $f_\alpha(a_t; s_{0:t})$. It indicates that we need to either parameterize $f_\alpha(a_t; s_{0:t})$ or $Q(a_t; s_{0:t})$. While Q^α and V^α are constructed from the optimality, the derived Q^α and V^α measure the performance of an interactive agent when it executes with the policy $p_\alpha(a_t|s_{0:t})$. They should be consistent.

Lemma A.2. *$V^\alpha(s_{0:t})$ and $\mathbb{E}_{p_\alpha(a_t|s_{0:t})}[Q^\alpha(a_t; s_{0:t})]$ yield the same optimal policy $p^*_\alpha(a_t|s_{0:t})$*

Proof.

$$\begin{aligned} & \mathbb{E}_{p_\alpha(a_t|s_{0:t})}[Q^\alpha(a_t; s_{0:t})] \\ :=& \mathbb{E}_{p_\alpha(a_t|s_{0:t})}\left[\mathbb{E}_{p^*(s_{t+1}|s_{0:t})}[r(s_{t+1}, a_t, s_{0:t}) + V^\alpha(s_{0:t+1})]\right] \\ =& \mathbb{E}_{p_\alpha(a_t|s_{0:t})}\mathbb{E}_{p^*(s_{t+1}|s_{0:t})}\left[\log p^*_\alpha(a_t|s_{0:t}) + r(s_{t+1}, s_{0:t}) + V^\alpha(s_{0:t+1})\right] \\ =& \mathbb{E}_{p^*(s_{t+1}|s_{0:t})}[r(s_{t+1}, s_{0:t}) - H_\alpha(a_t|s_{0:t}) + V^\alpha(s_{0:t+1})] \\ & - \sum_{k=t+1}^{T-1} \mathbb{E}_{p^*(s_{t+1:k}|s_{0:t})}[H_\alpha(a_k|s_{0:k})] \end{aligned}$$

where $\mathcal{H}(\cdot)$ is the entropy term. The last line is derived by recursively applying the Bellman equation in the line above until $s_{0:T}$. As an energy-based policy, $p_\alpha(a_t|s_{0:t})$'s entropy is inherently maximized [26]. Therefore, within the hypothesis space, $p^*_\alpha(a_t|s_{0:t})$ that optimizes $V^\alpha(s_{0:t})$ also leads to the optimal expected return $\mathbb{E}_{p_\alpha(a_t|s_{0:t})}[Q^\alpha(a_t; s_{0:t})]$. □

Given the convergence proof by Ziebart [50], we have:

Lemma A.3. *If $p^*(s_{t+1}|s_{0:t})$ is accessible and $p^*_\gamma(s_{t+1}|s_t, a_t)$ is known, soft policy iteration and soft Q learning both converge to $p^*_\alpha(a_t|s_{0:t}) = p^*_\alpha(a_t|s_{0:t}) \propto \exp(Q^*(a_t; s_{0:t}))$ under conditions.*

Lemma 3 means given $p^*(s_{t+1}|s_{0:t})$ and $p^*_\gamma(s_{t+1}|s_t, a_t)$, we can recover p^*_α through reinforcement learning methods, instead of the proposed MLE. So $p_\alpha(a_t|s_{0:t})$ is a viable policy space for the constructed sequential decision-making problem. Together, Lemma A.1, Lemma A.2 and Lemma A.3 provide proof for a valid sequential decision-making problem that maximizes the same objective of MLE, by Lemma A.4.

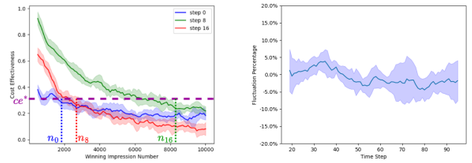
Lemma A.4 (MLE as non-Markovian decision-making process). *Assuming the Markovian transition $p_{\gamma^*}(s_{t+1}|s_t, a_t)$ is known, the ground-truth conditional state distribution $p^*(s_{t+1}|s_{0:t})$ for demonstration sequences is accessible, we can construct a non-Markovian sequential decision-making problem, based on a reward function $r_\alpha(s_{t+1}, s_{0:t}) := \log \int p_\alpha(a_t|s_{0:t})p_{\gamma^*}(s_{t+1}|s_t, a_t)da_t$ for an arbitrary energy-based policy $p_\alpha(a_t|s_{0:t})$. Its objective is*

$$\sum_{t=0}^{T} \mathbb{E}_{p^*(s_{0:t})}\left[V^{p_\alpha}(s_{0:t})\right] = \mathbb{E}_{p^*(s_{0:T})}\left[\sum_{t=0}^{T}\sum_{k=t}^{T} r_\alpha(s_{k+1}; s_{0:k})\right] \quad (26)$$

$V^{p_\alpha}(s_{0:t}) := \mathbb{E}_{p^*(s_{t+1:T}|s_{0:t})}[\sum_{k=1}^{T} r_\alpha(s_{k+1}; s_{0:k})]$ is the value function of p_α. This objective yields the save optimal policy as the Maximum Likelihood Estimation $\mathbb{E}_{p^*(s_{0:T})}[\log p_\theta(s_{0:T})]$.

Table 5: Parameters of the Real Advertising System.

Parameters	Values
Number of advertisers	30
Time steps in an episode, T	96
Minimum number of impression opportunities $N_{\min}$	50
Maximum number of impression opportunities $N_{\max}$	300
Minimum budget	1000 Yuan
Maximum budget	4000 Yuan
Value of impression opportunities in stage 1, $v^1_{j,t}$	0∼ 1
Value of impression opportunities in stage 2, $v^2_{j,t}$	0∼ 1
Minimum bid price, $A_{\min}$	0 Yuan
Maximum bid price, $A_{\max}$	1000 Yuan
Maximum value of impression opportunity, v_M	1
Maximum market price, p_M	1000 Yuan



（a）成本效益曲线样本，来自三个时间步 （b）曝光成本在不同时间步的波动

图 8：在线广告系统的统计结果。

其中，

$r(s_{t+1}, s_{0:t}) := \log p^*(s_{t+1}|s_{0:t}) = \log p^*_a(s_t|s_{0:t})p^*(s_{t+1}|s_t, a_t)dt$，

$V^*(s_{0:T}) := 0$。需要注意的是，上文定义的 r涉及最优策略，而该策略可能无法预先获知。我们可以将其替换为 r_α来解决这一问题，其中策略 $p_\alpha(a_t|s_{0:t})$是任意的。所有贝尔曼恒等式和更新仍应成立。对于任意 V和 α，由此得到的贝尔曼更新，即价值迭代，为

$$V(s_{0:t}) = \mathbb{E}_{p^*(s_{t+1}|s_{0:t})}[r_\alpha(s_{0:t}, s_{t+1}) + V(s_{0:t+1})]. \quad (22)$$

随后，我们定义 $r(s_{t+1}, a_t, s_{0:t}) := r(s_{t+1}, s_{0:t}) + \log p^*_a(a_t|s_{0:t})$，以构造一个 Q函数：

$$Q^*(a_t; s_{0:t}) := \mathbb{E}_{p^*(s_{t+1}|s_{0:t})}[r(s_{t+1}, a_t, s_{0:t}) + V^*(s_{0:t+1})], \quad (23)$$

对于任意 α、 Q和 V，这给出了一个贝尔曼更新，即 Q 回溯更新

$$Q(a_t; s_{0:t}) = \mathbb{E}_{p^*(s_{t+1}|s_{0:t})}[r_\alpha(s_{0:t}, a_t, s_{t+1}) + V(s_{0:t+1})]. \quad (24)$$

还需注意，恒等式（23）和式（25）中的 V和 Q分别不一定与策略 $p_\alpha(a_t|s_{0:t})$相关。这里对符号作轻微复用，我们用 Q_α, V_α表示策略 $p_\alpha(a_t|s_{0:t})$的期望回报。至此，我们完成了基本代数组件的构造，接下来检查它们之间的关系是否符合序列决策问题的代数结构。我们首先证明上述构造在最优情况下是有效的。

引理 A.1。当 $f_\alpha(a_t; s_{0:t}) = Q^*(a_t; s_{0:t}) - V^*(s_{0:t}), p_\alpha(a_t|s_{0:t})$为最优策略。

证明。注意，该构造给出

$$\begin{aligned} & Q^*(a_t; s_{0:t}) \\ =& \mathbb{E}_{p^*(s_{t+1}|s_{0:t})}\left[r(s_{t+1}, s_{0:t}) + \log p^*_\alpha(a_t|s_{0:t}) + V^*(s_{0:t+1})\right] \\ =& \log p^*_\alpha(a_t|s_{0:t}) + \mathbb{E}_{p^*(s_{t+1}|s_{0:t})}\left[r(s_{t+1}, s_{0:t}) + V^*(s_{0:t+1})\right] \\ =& \log p^*_\alpha(a_t|s_{0:t}) + V^*(s_{0:t}) \end{aligned} \quad (25)$$

□

显然， $Q^*(a_t; s_{0:t})$位于 $f_\alpha(a_t; s_{0:t})$的假设空间中。这表明，我们需要参数化 $f_\alpha(a_t; s_{0:t})$或$Q(a_t; s_{0:t})$。尽管 Q^α和 V^α是基于最优性构造的，但推导得到的Q^α和 V^α衡量的是交互式智能体按照策略 $p_\alpha(a_t|s_{0:t})$执行时的性能。它们应当保持一致。

引理 A.2. $V^\alpha(s_{0:t})$和$\mathbb{E}_{p_\alpha(a_t|s_{0:t})}[Q^\alpha(a_t; s_{0:t})]$产生相同的最优策略 $p^*_\alpha(a_t|s_{0:t})$

证明。

$$\begin{aligned} & \mathbb{E}_{p_\alpha(a_t|s_{0:t})}[Q^\alpha(a_t; s_{0:t})] \\ :=& \mathbb{E}_{p_\alpha(a_t|s_{0:t})}\left[\mathbb{E}_{p^*(s_{t+1}|s_{0:t})}[r(s_{t+1}, a_t, s_{0:t}) + V^\alpha(s_{0:t+1})]\right] \\ =& \mathbb{E}_{p_\alpha(a_t|s_{0:t})}\mathbb{E}_{p^*(s_{t+1}|s_{0:t})}\left[\log p^*_\alpha(a_t|s_{0:t}) + r(s_{t+1}, s_{0:t}) + V^\alpha(s_{0:t+1})\right] \\ =& \mathbb{E}_{p^*(s_{t+1}|s_{0:t})}[r(s_{t+1}, s_{0:t}) - H_\alpha(a_t|s_{0:t}) + V^\alpha(s_{0:t+1})] \\ & - \sum_{k=t+1}^{T-1} \mathbb{E}_{p^*(s_{t+1:k}|s_{0:t})}[H_\alpha(a_k|s_{0:k})] \end{aligned}$$

其中， $\mathcal{H}(\cdot)$是熵项。最后一行通过递归应用上一行的贝尔曼方程，直至 $s_{0:T}$而推导得到。作为基于能量的策略，$p_\alpha(a_t|s_{0:t})$的熵天然被最大化[26]。因此，在假设空间内，$p^*_\alpha(a_t|s_{0:t})$使 $V^\alpha(s_{0:t})$最优时，也会得到最优的期望回报 $\mathbb{E}_{p_\alpha(a_t|s_{0:t})}[Q^\alpha(a_t; s_{0:t})]$。□

根据 Ziebart[50]给出的收敛性证明，我们有：

引理 A.3。如果 $p^*(s_{t+1}|s_{0:t})$可获取，且 $p^*_\gamma(s_{t+1}|s_t, a_t)$已知，则软策略迭代和软 Q学习在一定条件下均收敛于 $p^*_\alpha(a_t|s_{0:t}) = p^*_\alpha(a_t|s_{0:t}) \propto \exp(Q^*(a_t; s_{0:t}))$。

引理 3 意味着，给定 $p^*(s_{t+1}|s_{0:t})$和 $p^*_\gamma(s_{t+1}|s_t, a_t)$，我们可以通过强化学习方法恢复 p^*_α，而不必使用所提出的 MLE。因此， $p_\alpha(a_t|s_{0:t})$是所构造的序列决策问题的一个可行策略空间。综合来看，根据引理 A.4，引理 A.1、引理 A.2 和引理 A.3 共同证明了一个有效序列决策问题的存在，该问题最大化与 MLE 相同的目标。

引理 A.4（作为非马尔可夫决策过程的 MLE）。假设马尔可夫转移 $p_{\gamma^*}(s_{t+1}|s_t, a_t)$已知，且示范序列的真实条件状态分布 $p^*(s_{t+1}|s_{0:t})$可获取，则对于任意基于能量的策略，我们可以基于奖励函数$r_\alpha(s_{t+1}, s_{0:t}) := \log \int p_\alpha(a_t|s_{0:t})p_{\gamma^*}(s_{t+1}|s_t, a_t)da_t$构造一个非马尔可夫序列决策问题，其中该策略为 $p_\alpha(a_t|s_{0:t})$。其目标为

$$\sum_{t=0}^{T} \mathbb{E}_{p^*(s_{0:t})}\left[V^{p_\alpha}(s_{0:t})\right] = \mathbb{E}_{p^*(s_{0:T})}\left[\sum_{t=0}^{T}\sum_{k=t}^{T} r_\alpha(s_{k+1}; s_{0:k})\right] \quad (26)$$

$V^{p_\alpha}(s_{0:t}) := \mathbb{E}_{p^*(s_{t+1:T}|s_{0:t})}[\sum_{k=1}^{T} r_\alpha(s_{k+1}; s_{0:k})]$是 p_α的价值函数。这一目标产生的最优策略与最大似然估计 $\mathbb{E}_{p^*(s_{0:T})}[\log p_\theta(s_{0:T})]$相同。