AMINO: GENERATIVE AUTO-BIDDING WITH MULTI-MODAL TRAJECTORY MODELING AND INITIAL NOISE OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

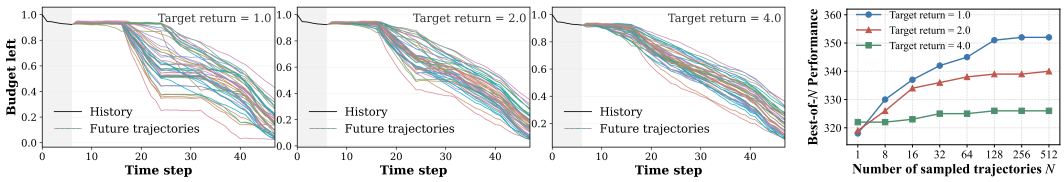
Auto-bidding is a critical technique in online advertising for achieving advertisers' commercial goals under constraints. Recent generative auto-bidding methods have shown strong potential by learning bidding policies directly from large-scale offline data. In this setting, auto-bidding trajectories are inherently multimodal, making conditional diffusion particularly suitable for trajectory modeling. However, our theoretical analysis reveals that conditional diffusion can suffer from mode concentration under the high target returns commonly used to steer generation toward high-value trajectories, potentially resulting in suboptimal bidding trajectories. Moreover, existing diffusion-based methods lack an explicit mechanism to exploit task feedback and improve beyond static offline data. We therefore propose AMINO (generative **A**uto-bidding with **M**ultimodal trajectory modeling and **I**nitial **N**oise **O**ptimization), a novel framework that decouples performance optimization from multimodal trajectory modeling. Rather than modifying the diffusion planner, AMINO keeps it fixed and learns an initial noise policy via reinforcement learning for performance optimization. To address the challenges of delayed feedback and offline optimization in auto-bidding, we develop a tailored noise optimization pipeline, enabling reliable trajectory-level feedback and stable exploration in the continuous noise space. Extensive experiments on a large-scale real-world dataset demonstrate the state-of-the-art performance of AMINO.

1 INTRODUCTION

The rapid growth of online advertising has made manual bid adjustment increasingly difficult in large-scale ad auctions (Evans, 2009; Wang & Yuan, 2015; Muthukrishnan, 2009). Auto-bidding has therefore become a fundamental component of modern advertising platforms, automatically optimizing bidding strategies to maximize impression value under budget and economic constraints such as cost-per-action (CPA) (He et al., 2021; Ren et al., 2017; Guo et al., 2024a). Due to operational safety concerns, auto-bidding is typically formulated as an offline sequential decision-making problem (Mou et al., 2022). While offline reinforcement learning (RL) has been widely studied for this setting, bootstrapped value estimation may introduce estimation errors and training instability (Kumar et al., 2020; Mao et al., 2024; Peng et al., 2024). Recent generative methods provide an alternative by directly modeling bidding trajectories from large-scale offline data.

Existing generative auto-bidding methods mainly follow two paradigms: Decision Transformer (DT) (Chen et al., 2021; Liu et al., 2023) and conditional diffusion (Guo et al., 2024a; Li et al., 2026a). DT-based methods autoregressively generate bid actions from historical sequences and have been extended to various advertising scenarios (Li et al., 2025; Gao et al., 2025). In contrast, conditional diffusion models the distribution of multi-step trajectories under given targets and histories. We observe that similar bidding histories and conditions can correspond to substantially different yet feasible future budget-consumption patterns, such as aggressive early spending, smooth allocation, and delayed spending, indicating that auto-bidding is inherently multimodal. Further evidence is provided in Appendix D.2. While regression-based DT methods tend to average across different behavioral modes, conditional diffusion can explicitly model their trajectory distribution.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) (b)

Figure 1: (a) Generated budget-consumption trajectories under target returns of 1.0, 2.0, and 4.0. The trajectories become progressively more concentrated as the target return increases. (b) Best-of-N performance as the number of sampled trajectories increases under different target return settings.

However, this multimodal modeling capability is not fully exploited by existing diffusion-based bidding methods. They typically use large target returns as conditioning signals to steer generation toward high-value trajectories, often substantially beyond the offline data range (Guo et al., 2024a). Such extrapolation receives limited supervision and may degrade generation quality. More importantly, our theoretical and empirical analyses reveal that excessive return conditioning progressively concentrates the generated trajectory distribution, which we term *mode concentration*. As shown in Figure 1(a), independently sampled futures become increasingly similar as the target return increases. This loss of diversity directly limits trajectory selection: Figure 1(b) shows that Best-of-N benefits more from increasing N when trajectories remain diverse, whereas mode concentration suppresses alternative behavioral modes and limits the opportunity to discover better trajectories.

Nevertheless, preserving multimodality alone does not guarantee high-quality trajectories. Conditional diffusion is trained to imitate the offline trajectory distribution and lacks an explicit mechanism to improve beyond the observed behaviors. Moreover, multiple trajectories may satisfy the same target return, making generation quality highly dependent on the sampled initial noise (Guo et al., 2024b; Tang et al., 2025; Ahn et al., 2026). One possible solution is to fine-tune the entire diffusion model with reinforcement learning (Fan et al., 2023; Uehara et al., 2024; Ren et al., 2025; Zhao et al., 2025). However, trajectory-level rewards are available only after complete trajectories are generated, making credit assignment across multiple denoising steps computationally expensive and unstable (Black et al., 2024; Clark et al., 2024), while potentially impairing the multimodal generation capability learned during pretraining.

To address these critical challenges, we propose **AMINO** (generative **A**uto-bidding with **M**ultimodal trajectory modeling and **I**nitial **N**oise **O**ptimization). Instead of modifying the pretrained diffusion planner, AMINO optimizes the sampling process through the initial noise space while keeping the planner fixed. We introduce Group-relative Noise Policy Optimization (GNPO), which samples groups of initial noises, maps them deterministically to candidate trajectories through the frozen planner, and updates the noise policy based on their relative quality. A trajectory evaluator provides reliable feedback by distinguishing observed histories from generated futures through type embeddings and evaluating only the candidate future. Finally, AMINO constrains noise exploration and reshapes overly optimistic evaluator rewards to reduce unreliable out-of-distribution optimization and stabilize policy learning.

To summarize, our contributions are threefold: (i) we theoretically and empirically reveal that excessively high return conditioning can induce severe *mode concentration* in conditional diffusion, potentially leading to suboptimal bidding trajectories; (ii) we propose **AMINO** to decouple performance optimization from multimodal trajectory modeling by optimizing an initial noise policy while freezing the diffusion planner. We further develop a tailored optimization pipeline combining GNPO, trajectory-level evaluation, and conservative optimization; (iii) extensive experiments on AuctionNet (Su et al., 2024) demonstrate the effectiveness and robustness of AMINO across diverse settings, with particularly strong performance under sparse-feedback and tight-budget conditions.

2 Related Work

Generative Auto-Bidding. Recently, generative models has emerged as a new paradigm for auto-bidding. Decision Transformer (DT) (Chen et al., 2021) formulates bidding as autoregressive sequence modeling. Building upon this paradigm, GAS (Li et al., 2025) performs post-training search with MCTS, and GAVE (Gao et al., 2025) employs value-guided exploration. BidFlow (Yang et al.,

2026) combines flow matching with Q-guided one-step distillation, while LLM-based approaches have also emerged, including RTBAgent (Cai et al., 2025), which directly generates bidding decisions through prompting, and LBM (Li et al., 2026b), which introduces a hierarchical Think–Act architecture. Despite these advances, these methods largely overlook the explicit modeling of multimodal future trajectories. DRIVE (Cui et al., 2026) explicitly models multimodal bidding behaviors with a Gaussian Mixture Model (GMM), although its expressiveness is tied to the predefined number of mixture components. In contrast, diffusion-based methods are naturally suited to modeling multimodal trajectory distributions: DiffBid (Guo et al., 2024a) adopts conditional diffusion for bidding trajectory generation, while CBD (Li et al., 2026a) further introduces a diffusion completer–aligner for trajectory completion. Yet, how aggressive return conditioning reshapes their underlying mode structure remains largely unexplored.

Initial Noise Optimization. Initial noise optimization improves pretrained diffusion models by optimizing the noise that initializes denoising while keeping the diffusion model fixed. DOODL (Wallace et al., 2023) introduced end-to-end latent optimization for diffusion guidance. Subsequent methods include INITNO (Guo et al., 2024b), which uses attention-based objectives with Gaussian regularization; DNO (Tang et al., 2025), which performs reward-based direct noise optimization; and ReNO (Eyring et al., 2024), which extends reward-based noise optimization to one-step text-to-image models. More recently, DSRL (Wagenmaker et al., 2025) learns a latent-noise policy using actor–critic reinforcement learning. However, auto-bidding poses additional challenges: feedback is delayed until a complete future trajectory is generated, while optimization must account for economic constraints and unreliable out-of-distribution evaluations from offline data. These characteristics make existing noise-optimization methods difficult to directly apply. AMINO therefore develops a tailored noise optimization pipeline for constraint-aware auto-bidding, combining trajectory-level evaluation, group-relative policy optimization, and conservative optimization.

3 PRELIMINARIES

Consider an auto-bidding scenario where impression opportunities arrive sequentially, each indexed by i, with $v_i > 0$ denoting the estimated value of impression i. An advertiser secures an impression if its bid b_i exceeds the competing bids, incurring a cost $c_i > 0$. The advertiser aims to maximize the cumulative acquired value while satisfying the total budget and cost-per-action (CPA) constraints(He et al., 2021).

Due to the sequential nature of the bidding process, auto-bidding is naturally formulated as a sequential decision-making problem. Specifically, we formulate it as a Markov Decision Process (MDP) $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P} \rangle$. At each decision step t, the agent observes a state $s_t \in \mathcal{S}$ containing the remaining time, unused budget, historical bidding statistics, and other relevant information, and selects an action $a_t \in \mathcal{A}$. The reward $r_t \geq 0$ measures the value of the impressions obtained during the current interval, while $\mathcal{P}$ denotes the underlying state transition dynamics. The objective is formulated as

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

where $c_t \geq 0$ denotes the incurred cost during interval t, $B > 0$ is the total budget, and $C \in \mathbb{R}_+$ denotes the maximum allowable CPA.

Prior work (He et al., 2021; Gao et al., 2025) has shown that the optimal impression-level bid takes a value-proportional form, $b_i^* = \lambda v_i$, where $\lambda > 0$ is the bidding scaling factor. Since market conditions vary over time, auto-bidding systems typically partition the bidding period into T fixed-length decision intervals (e.g., 30 minutes each) and recalibrate the scaling factor accordingly (Guo et al., 2024a; He et al., 2021; Su et al., 2024). Thus, the action a_t represents the scaling factor at interval t, yielding the impression-level bid $b_i = a_t v_i$.

Conditional Diffusion for Auto-Bidding Conditional diffusion formulates sequential decision making as conditional trajectory generation (Guo et al., 2024a). Let $\tau = [s_1, \ldots, s_T]$ denote a state trajectory over a bidding period and $y = \sum_{t=1}^{T} r_t$ denote its cumulative returns, which serves as the conditioning signal to steer generation toward high-value trajectories. Given an offline dataset $\mathcal{D}$, the conditional trajectory distribution is learned by $\max_\theta \mathbb{E}_{(\tau,y)\sim\mathcal{D}} [\log p_\theta(\tau \mid y)]$.

At inference time, diffusion-based auto-bidding methods do not directly generate bidding actions, but typically follow a *planning-and-control* paradigm (Guo et al., 2024a). Given the observed bidding history $s_{1:t}$ and a target return y, the diffusion planner first generates a future state trajectory $\hat{s}_{t+1:T}$. The next planned state $\hat{s}_{t+1}$, together with the current state s_t, is then fed into a pretrained inverse dynamics model to produce the executable bidding action, $a_t = f_\omega(s_t, \hat{s}_{t+1})$. See Appendix B for more details. Our method focuses on optimizing the trajectory planning stage while retaining the standard inverse dynamics controller for action execution.

4 Theoretical Analysis

In this section, we theoretically analyze the mode concentration introduced in Section 1.

Let $y_{\max}$ denote the maximum trajectory return in the offline dataset $\mathcal{D}$. During inference, diffusion-based auto-bidding methods often set an extrapolated target return $y^\star = (1+\epsilon)y_{\max}$, $\epsilon > 0$, to favor higher-return trajectories. However, an excessively large $y^\star$ requires extrapolation beyond the training support and can suppress the multimodal generation capability of the diffusion planner. We next analyze how such aggressive return conditioning induces mode concentration.

Under the data distribution $p_\mathcal{D}$ induced by the offline dataset $\mathcal{D}$, given bidding history h_t, suppose the future trajectory distribution contains M distinct behavioral modes $\{m_1, \ldots, m_M\}$. The quantity $p_\mathcal{D}(m_i \mid h_t)$ represents the prior probability of mode m_i under the observed history. Although trajectories within the same mode share a similar behavioral pattern, their realized future returns y_t can vary. To obtain an interpretable analysis, we assume that the mode conditional return density can be approximated by a Gaussian distribution, $p_\mathcal{D}(y_t \mid m_i, h_t) = \mathcal{N}(y_t; \mu_i, \sigma_i^2)$, where μ_i and σ_i^2 denote the mean and variance of the realized returns within mode m_i, respectively. We omit their dependence on the fixed history h_t for notational brevity. For these Gaussian components, dominance in the extreme high-return region is determined by the asymptotic decay of their right-tail densities, which depends jointly on both the mean and variance. A larger variance yields a slower asymptotic right-tail decay; among components with equal variance, the one with the larger mean dominates.

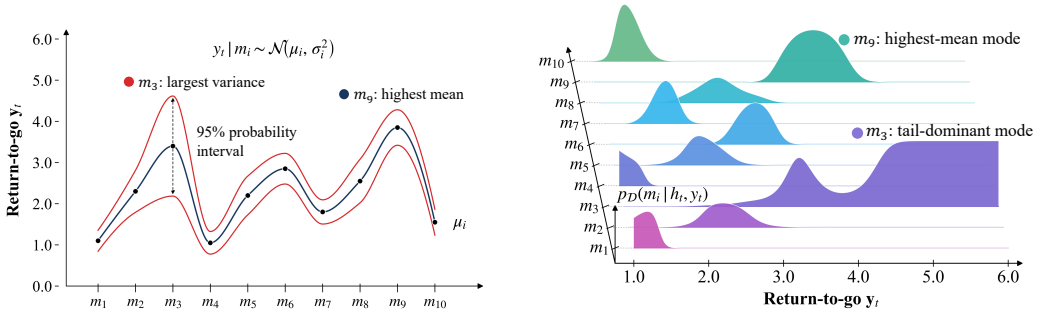


Figure 2: Illustration of *mode concentration*. For simplicity, we fix the bidding history h_t and consider ten modes with equal priors $p_\mathcal{D}(m_i \mid h_t)$. (*Left*) mode conditional return distributions $p_\mathcal{D}(y_t \mid m_i, h_t)$, where the blue curve denotes the mean and the red curves indicate the 95% probability intervals; m_9 has the highest mean, while m_3 has the largest variance. (*Right*) posterior mode probabilities $p_\mathcal{D}(m_i \mid h_t, y_t)$ obtained by Bayes' rule. As y_t increases, the distribution becomes dominated by m_3 despite m_9 being the highest mean mode.

When a target return $y^\star$ is imposed, the probability of each trajectory mode is reweighted according to how likely that return is under the corresponding mode. By Bayes' rule,

$$p_\mathcal{D}(m_i \mid h_t, y^\star) = \frac{p_\mathcal{D}(m_i \mid h_t) p_\mathcal{D}(y^\star \mid m_i, h_t)}{\sum_{j=1}^{M} p_\mathcal{D}(m_j \mid h_t) p_\mathcal{D}(y^\star \mid m_j, h_t)}. \tag{2}$$

Proposition 1 (Extreme-return-induced mode concentration). Given the Gaussian assumption above, suppose the extreme right tail admits a unique dominant mode $m^\dagger$. Then

$$\lim_{y^\star \to +\infty} p_\mathcal{D}(m^\dagger \mid h_t, y^\star) = 1, \qquad \lim_{y^\star \to +\infty} H(M \mid h_t, y^\star) = 0, \tag{3}$$

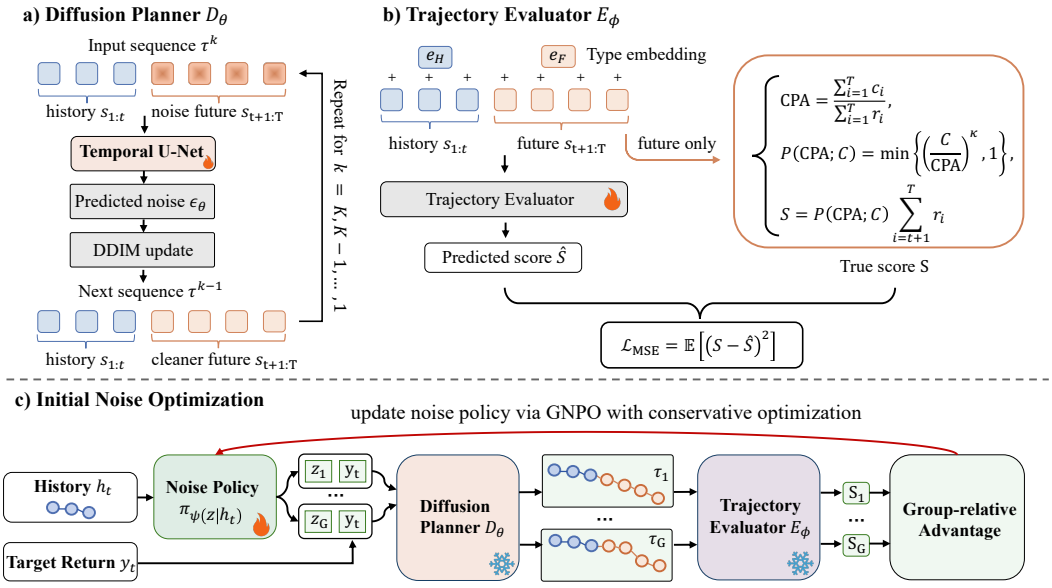


Figure 3: Overview of AMINO. (a) The diffusion planner generates future trajectories through K iterative DDIM denoising steps while preserving the observed history. (b) The trajectory evaluator uses type embeddings and future-only scoring to assess candidate trajectories. (c) Initial noise optimization samples G noise candidates, maps them to trajectories through the frozen diffusion planner, and updates the noise policy via GNPO using feedback from the trajectory evaluator.

where $H(M \mid h_t, y^\star) = -\sum_{i=1}^{M} p_{\mathcal{D}}(m_i \mid h_t, y^\star) \log p_{\mathcal{D}}(m_i \mid h_t, y^\star)$ is the conditional mode entropy. The proof is provided in Appendix C.

Proposition 1 shows that an excessively large target return can drive the conditional trajectory distribution toward a single mode, with its mode entropy approaching zero, thereby undermining the multimodal modeling advantage of the diffusion planner. Importantly, the resulting dominant mode is determined by the extrapolated likelihood behavior under the extreme return condition and is not guaranteed to be the most desirable mode for the current bidding state. Figure 2 provides an intuitive illustration of this phenomenon. This finding motivates using a moderate target return to preserve trajectory diversity. Performance improvement can then be pursued through initial noise optimization rather than increasing the target return.

5 Method

We propose AMINO, a diffusion-based auto-bidding framework that decouples performance optimization from multimodal trajectory modeling. Given the bidding history, a noise policy first generates suitable initial noises; a conditional diffusion planner then maps these noises into diverse future trajectories; and a trajectory evaluator assesses their quality and provides trajectory-level feedback to optimize the noise policy. Figure 3 illustrates the overall framework.

5.1 Diffusion Planner

Building on the conditional diffusion formulation in Section 3, AMINO employs a diffusion planner D_θ to model the distribution of feasible bidding trajectories. To maintain consistency with inference, where future trajectories are generated based on the observed history, we randomly sample a decision step t during training, treating $s_{1:t}$ as the observed history h_t and $s_{t+1:T}$ as the future segment to be generated. Accordingly, we define the future return condition as $y_t = \sum_{i=t+1}^{T} r_i$. The forward diffusion process gradually corrupts the future segment: $\tau_{t+1:T}^k = \sqrt{\bar{\alpha}_k}\tau_{t+1:T}^0 + \sqrt{1-\bar{\alpha}_k}\epsilon, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, where $\bar{\alpha}_k = \prod_{j=1}^{k}(1-\beta_j)$. A temporal U-Net ϵ_θ is trained to predict the injected noise with $\mathcal{L}_{\text{diff}} = \mathbb{E}\left[\|\epsilon - \epsilon_\theta(\tau^k, k, y_t)\|_{t+1:T}^2\right]$. For both training and inference, the observed prefix is

clamped to the ground-truth history at every diffusion step, i.e., $\tau^k_{1:t} \leftarrow h_t$, such that diffusion is performed only over the unknown future.

Since DDIM introduces no additional stochasticity during denoising, each initial noise deterministically corresponds to a generated trajectory under fixed history and condition, making it suitable for subsequent optimization in the initial noise space. Therefore, at inference time, instead of stochastic DDPM sampling (Ho et al., 2020), we adopt DDIM sampling (Song et al., 2021). At denoising step k, the future trajectory is updated as: $\tau^{k-1}_{t+1:T} = a_k \left(\tau^k_{t+1:T} - b_k \widetilde{\epsilon}_\theta(\tau^k, k, y_t)\right)$, where a_k and b_k are determined by the diffusion schedule. We employ classifier-free guidance, enabled by randomly dropping the return condition during training, to combine conditional and unconditional noise predictions: $\widetilde{\epsilon}_\theta = \epsilon_\theta(\tau^k, k) + w[\epsilon_\theta(\tau^k, k, y_t) - \epsilon_\theta(\tau^k, k)]$, where w denotes the guidance strength.

5.2 Trajectory Evaluator

To provide trajectory-level feedback for noise policy optimization, we train an evaluator E_ϕ to assess the quality of candidate futures under the observed bidding history. Specifically, E_ϕ jointly encodes the ground-truth history $h_t = s_{1:t}$ and a candidate future trajectory $\hat{\tau}_{t+1:T}$ using a bidirectional Transformer. A learnable type embedding is added to each state representation to explicitly distinguish observed historical states from generated future states, enabling more accurate discrimination among candidate futures that share the same history.

Since the observed history is fixed across candidate trajectories, the evaluator focuses on the utility of the generated future. Its supervision objective can be flexibly defined according to different advertiser preferences. In this work, we consider the widely used CPA-constrained setting (Gao et al., 2025), where optimizing future return alone[1] may favor trajectories with excessive cost and poor cost efficiency. We therefore combine the future return with a CPA-dependent penalty to construct the trajectory score:

$$\mathrm{CPA} = \frac{\sum_{i=1}^{T} c_i}{\sum_{i=1}^{T} r_i}, \qquad P(\mathrm{CPA}; C) = \min\left\{\left(\frac{C}{\mathrm{CPA}}\right)^{\kappa}, 1\right\}, \qquad S = P(\mathrm{CPA}; C) \sum_{i=t+1}^{T} r_i, \tag{4}$$

where $\kappa = 2$ controls the penalty steepness. The evaluator is trained by minimizing

$$\mathcal{L}_E = \mathbb{E}\Big[\left(E_\phi(h_t, \tau_{t+1:T}) - S\right)^2\Big], \tag{5}$$

Other advertiser preferences can be accommodated by modifying the supervision objective without changing the pretrained diffusion planner.

5.3 Group-Relative Noise Policy Optimization

Rather than directly fine-tuning the diffusion planner, we keep the pretrained planner D_θ fixed and formulate its initial noise as a continuous latent variable. We introduce a history-dependent noise policy $\pi_\psi(\cdot \mid h_t)$ parameterized by a Transformer. Given the history h_t, the Transformer produces a contextual representation, which is fed into two prediction heads to parameterize the mean $\mu_\psi(h_t)$ and log-standard deviation $\log \sigma_\psi(h_t)$ of a diagonal Gaussian distribution. To stabilize noise policy learning, AMINO incorporates parameter clipping, entropy regularization, and reward shaping.

Specifically, *parameter clipping* constrains the policy within a stable latent region by bounding the predicted distribution parameters:

$$\mu'_\psi(h_t) = \mathrm{clip}\left(\mu_\psi(h_t), -c_\mu, c_\mu\right), \qquad \log \sigma'_\psi(h_t) = \mathrm{clip}\left(\log \sigma_\psi(h_t), l_{\min}, l_{\max}\right). \tag{6}$$

We obtain $\sigma'_\psi(h_t) = \exp(\log \sigma'_\psi(h_t))$ and, for each history, sample G initial noise candidates as

$$z_i = \mu'_\psi(h_t) + \sigma'_\psi(h_t) \odot \xi_i, \qquad \xi_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \qquad i = 1, \ldots, G. \tag{7}$$

where $z_i \in \mathbb{R}^{d_t}$ denotes an initial noise candidate for the unknown future trajectory. Each candidate is deterministically mapped by the diffusion planner to a future trajectory $\hat{\tau}_i = D_\theta(h_t, y_t, z_i)$ and

[1] As in real-world advertising systems, bidding is automatically suspended once the advertiser's budget is exhausted. Therefore, an executable trajectory does not violate the budget constraint, making cumulative reward a natural measure of trajectory value (Mou et al., 2026).

evaluated by the trajectory evaluator to obtain $S_i = E_\phi(h_t, \hat{\tau}_i)$. To assign relative credit to different noises generated under the same history, we normalize their trajectory scores within each group:

$$A_i = \frac{S_i - \frac{1}{G}\sum_{j=1}^{G} S_j}{\mathrm{Std}_{j=1}^{G}(S_j) + \epsilon_A}. \tag{8}$$

The noise policy is optimized by maximizing

$$J_{\mathrm{GNPO}}(\psi) = \mathbb{E}_{h_t}\left[\frac{1}{Gd_t}\sum_{i=1}^{G}\sum_{j=1}^{d_t}\log \pi_\psi(z_{i,j} \mid h_t)\, A_i + \alpha\bar{\mathcal{H}}(\pi_\psi)\right], \tag{9}$$

where d_t is the dimensionality of the initial noise and $\bar{\mathcal{H}}(\pi_\psi) = \frac{1}{d_t}\sum_{j=1}^{d_t}\mathcal{H}[\pi_\psi(z_j \mid h_t)]$ denotes the average entropy across noise dimensions. The *entropy regularization* term prevents premature variance collapse and preserves exploration in the continuous initial noise space.

Conservative Noise Policy Optimization. Since the evaluator is trained solely on offline data, unconstrained policy optimization may exploit spuriously high predictions on out-of-distribution trajectories. In addition to parameter clipping, we introduce *reward shaping* to suppress abnormally high evaluator predictions. For each history h_t, we first sample N standard Gaussian initial noises $\xi_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and obtain a best-of-N reference score, which is averaged over histories to construct a global reference:

$$S^{\mathrm{ref}} = \mathbb{E}_{h_t}\left[\max_{1\le n\le N} E_\phi(h_t, D_\theta(h_t, y_t, \xi_n))\right], \qquad S^{\mathrm{th}} = (1+\delta)S^{\mathrm{ref}}. \tag{10}$$

Scores exceeding this reliability threshold are penalized as $S'_i = S_i - \lambda_s\left[S_i - S^{\mathrm{th}}\right]_+$. During conservative optimization, S'_i replaces S_i when computing the group-relative advantage in Eq. 8. Together, parameter clipping and reward shaping reduce the exploitation of unreliable out-of-distribution evaluations and improve optimization stability.

6 Experiments

6.1 Experimental Setup

Dataset. We evaluate all methods on AuctionNet (Su et al., 2024), a large-scale public real-world advertising auction benchmark released by Alibaba, together with its more challenging sparse variant, AuctionNet-Sparse. Both datasets contain 21 advertising delivery periods collected from a real-world advertising platform, with each period comprising approximately 500K impression opportunities divided into 48 decision intervals. More details are provided in Appendix D.1.

Evaluation Metrics. We adopt the following metrics (Yang et al., 2026; Li et al., 2026b) to evaluate the performance. Results are reported as mean $\pm$ standard deviation over 5 random seeds.

- **Conversions**: the total received impression value during a bidding period, defined as $\mathrm{Conversions} = \sum_i r_i$.
- **CPA Ratio**: the ratio between the realized CPA and the target CPA constraint, defined as $\mathrm{CPA\ Ratio} = \frac{C_{\mathrm{real}}}{C} = \frac{\sum_i c_i / \sum_i r_i}{C}$, where C_{real} denotes the realized CPA and C is the target CPA constraint.
- **Score**: a comprehensive metric that jointly measures conversion performance and CPA constraint satisfaction, defined as $\mathrm{Score} = \mathrm{Conversions} \cdot \min\left\{\left(\frac{C}{C_{\mathrm{real}}}\right)^2, 1\right\}$.

Baselines. We compare AMINO with a broad range of state-of-the-art baselines. For RL-based methods, we include the online RL method USCB (He et al., 2021) and representative offline RL methods BCQ (Fujimoto et al., 2019), CQL (Kumar et al., 2020), and IQL (Kostrikov et al., 2022). For generative methods, we compare with diffusion-based and DT-based methods. Diffusion-based

Table 1: Score comparison under different budget ratios. Higher is better. The best and second-best available results are shown in **bold** and underlined, respectively. "–" denotes unavailable results.

AuctionNet	Budget	USCB	CQL	IQL	BCQ	DiffBid	DT	DT-S	LBM	GAS	CBD	AMINO
Dense	50%	86	113	164	190	54	191	178	–	193	182	**214±3.02**
	75%	135	139	232	259	100	265	268	–	287	258	**298±2.14**
	100%	157	171	281	270	152	329	334	348	359	317	**374±2.73**
	125%	220	201	355	379	193	396	395	–	409	389	**425±2.98**
	150%	281	238	401	429	234	450	441	–	461	451	**475±3.87**
Sparse	50%	11.5	12.8	16.5	17.7	9.9	14.8	17.7	–	18.4	18.1	**21.3±0.21**
	75%	14.9	16.7	22.1	24.6	15.4	22.9	25.9	–	27.5	26.8	**30.6±0.47**
	100%	17.5	22.2	30.0	31.1	19.5	29.6	31.1	33.4	36.1	36.5	**38.3±0.32**
	125%	26.7	28.6	37.1	34.2	25.3	34.3	39.6	–	40.0	40.6	**44.1±0.51**
	150%	31.3	35.8	43.1	37.9	30.8	44.5	44.7	–	46.5	46.8	**49.2±0.46**

baselines include DiffBid (Guo et al., 2024a) and CBD (Li et al., 2026a), which introduces a diffusion completer–aligner framework. DT-based baselines include DT (Chen et al., 2021) and DT-S (Li et al., 2025), where DT-S leverages a reward function that jointly represents the winning value and KPI constraints. We further include GAS (Li et al., 2025), which performs post-training search via MCTS. For LLM-based methods, we compare with LBM (Li et al., 2026b), which designs a hierarchical Think-Act architecture.

Detailed implementation details are provided in Appendix A.

6.2 Main Results

As shown in Table 1, AMINO consistently achieves the best performance across all evaluated settings, demonstrating its effectiveness and robustness under varying bidding conditions. The advantage becomes more evident in the more challenging sparse-feedback and tight-budget settings: sparse feedback provides weaker supervision for learning high-value bidding behaviors, while limited budgets make temporal resource allocation more sensitive to suboptimal decisions. By preserving multimodal trajectory modeling and exploiting trajectory-level feedback to optimize the initial noise policy, AMINO can more effectively explore and select high-quality future trajectories, thereby improving decision quality under limited supervision and tighter resource constraints.

6.3 Ablation Study

Table 2 presents the component-wise ablation results of AMINO, with Best-of-N included as a sampling-based baseline. Compared with the diffusion planner, AMINO improves Score and Conversions by 17.6% and 13.9%, respectively, while further reducing the CPA Ratio, demonstrating the benefit of noise policy optimization. AMINO also clearly outperforms Best-of-N and avoids the repeated candidate sampling and evaluation required by random search. Removing parameter clipping causes the most severe degradation, as unconstrained Gaussian parameters can push the policy into unreliable OOD regions. Reward shaping is also important for preventing exploitation of spuriously high evaluator predictions, while entropy regularization prevents premature variance collapse and preserves noise diversity. Further comparisons with noise optimization based on the Cross-Entropy Method (CEM) and GMM-based multimodal modeling are provided in Appendix E.

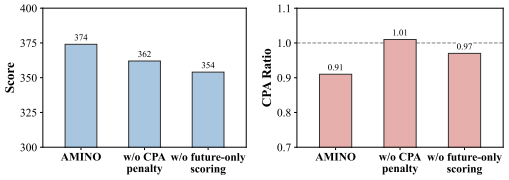


Figure 4: Ablation of trajectory evaluator

The trajectory evaluator jointly encodes the observed history prefix and generated future trajectory with type embeddings to distinguish them, and predicts a trajectory score that combines future return with a CPA penalty. To examine these design choices, we consider two variants: *w/o future-only scoring* directly scores the full trajectory without distinguishing the observed history from the generated future, while *w/o CPA penalty* removes the CPA penalty from the supervision target. The resulting evaluators are then used to train the noise policy under the same setting. As shown in

Table 2: Ablation of core components.

Method	Score ↑	Conversions ↑	CPA Ratio ↓
Diffusion Only	318	359	0.99
Best-of-N ($N = 64$)	335	384	1.02
AMINO w/o parameter clipping	67	138	1.92
AMINO w/o reward shaping	344	381	0.97
AMINO w/o entropy regularization	355	398	0.93
AMINO	**374**(↑ 17.6%)	**409**(↑ 13.9%)	**0.91**(↓ 8.1%)

Table 3: Evaluator accuracy.

Metric	Value
RMSE ↓	0.004
R^2 ↑	0.998
Spearman ρ ↑	0.997
AUC ↑	96.8%

Figure 4, both variants degrade downstream bidding performance, demonstrating the importance of these designs. We further evaluate the prediction reliability of the full evaluator by randomly splitting AuctionNet into training and test sets with a ratio of 8:2. As shown in Table 3, the evaluator achieves low prediction error and strong ranking consistency on the held-out test set.

6.4 Empirical Validation of Mode Concentration

We empirically validate mode concentration in Figure 1. We fix the same bidding history and, for each target return in $\{1.0, 2.0, 4.0\}$, sample 48 initial noises and generate 48 future trajectories with the pretrained diffusion planner. As shown in Figure 1(a), trajectories become increasingly concentrated as the target return increases. We further evaluate Best-of-N in the simulated environment (Gao et al., 2025), where N candidate trajectories are generated and the one with the highest evaluator score is selected at each decision step. Figure 1(b) reports the resulting bidding Score as N increases. Increasing N yields larger improvements in Score when trajectories remain diverse, whereas the improvement diminishes under larger target returns, confirming that mode concentration limits the benefit of trajectory selection.

6.5 Evolution of the Noise Policy

To visualize how GNPO reshapes the noise policy, we fix five bidding histories and record their noise representations at steps $\{0, 300, 600, 900, 1200, 1500\}$. Representations from all checkpoints are jointly projected into a shared t-SNE space, where each curve tracks the evolution of one case and marker size reflects the rescaled evaluator score.

As shown in Figure 5, the representations are initially concentrated within a small region, but gradually move toward distinct, history-dependent regions as training proceeds. Meanwhile, marker sizes generally increase, indicating that GNPO progressively shifts the noise policy toward higher-scoring, history-dependent solutions rather than collapsing to a common noise pattern. Corresponding trajectory and score evolution is further visualized in Appendix E.3.

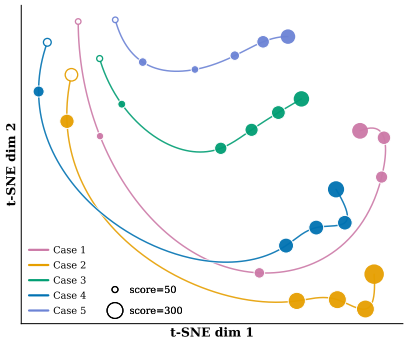


Figure 5: Evolution of noise representations during GNPO training.

7 Conclusion

This paper presents AMINO, a diffusion-based framework for generative auto-bidding. We identify the conditional multimodality of auto-bidding trajectories and reveal that excessively high target return can induce *mode concentration*, limiting the multimodal generation capability of the diffusion planner. AMINO addresses this issue by decoupling performance optimization from multimodal trajectory modeling and optimizing an initial noise policy with trajectory-level feedback while keeping the pretrained diffusion planner fixed. Extensive experiments across diverse bidding settings demonstrate consistent improvements over existing methods.

Reproducibility Statement

We provide detailed information to facilitate the reproducibility of our work in the appendix. Appendix A describes the complete training procedure, network architectures, and hyperparameter settings of AMINO. Appendix D provides detailed descriptions of the datasets and evaluation protocol used in our experiments. For the theoretical results, a complete proof of Proposition 1 is provided in Appendix C.

AI Use Statement

In this work, we used generative AI tools for language polishing, including grammar checking and wording refinement. We did not use generative AI tools to generate synthetic datasets, design or refine research methodology or experiments, develop or improve the proposed methods or research contributions, or assist with theoretical proofs. All AI-assisted outputs were reviewed by the authors, who take full responsibility for the final content of this work.

References

Donghoon Ahn, Jiwon Kang, Sanghyun Lee, Jaewon Min, Minjae Kim, Wooseok Jang, Hyoungwon Cho, Sayak Paul, SeonHwa Kim, Eunju Cha, Kyong Hwan Jin, and Seungryong Kim. A noise is worth diffusion guidance. In *The Fourteenth International Conference on Learning Representations*, 2026.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Leng Cai, Junxuan He, Yikai Li, Junjie Liang, Yuanping Lin, Ziming Quan, Yawen Zeng, and Jin Xu. RTBAgent: A LLM-based agent system for real-time bidding. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 104–113, 2025. doi: 10.1145/3701716.3715259.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In *Advances in Neural Information Processing Systems*, volume 34, pp. 15084–15097, 2021.

Kevin Clark, Paul Vicol, Kevin Swersky, and David J. Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *The Twelfth International Conference on Learning Representations*, 2024.

Miduo Cui, Haochen Wang, Shangqin Mao, Xun Yang, Qianlong Xie, Xingxing Wang, Xuri Ge, Ying Zhou, and Zhiwei Xu. DRIVE: Distributional and retrieval-augmented bidding with value evaluation. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

David S. Evans. The online advertising industry: Economics, evolution, and privacy. *Journal of Economic Perspectives*, 23(3):37–60, 2009.

Luca Eyring, Shyamgopal Karthik, Karsten Roth, Alexey Dosovitskiy, and Zeynep Akata. ReNO: Enhancing one-step text-to-image models through reward-based noise optimization. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. DPOK: Reinforcement learning for fine-tuning text-to-image diffusion models. In *Advances in Neural Information Processing Systems*, 2023.

Jiaheng Feng, Mingxiao Feng, Haolin Song, Wengang Zhou, and Houqiang Li. SUF: Stabilized unconstrained fine-tuning for offline-to-online reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11961–11969, 2024.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2052–2062. PMLR, 2019.

Jingtong Gao, Yewen Li, Shuai Mao, Peng Jiang, Nan Jiang, Yejing Wang, Qingpeng Cai, Fei Pan, Peng Jiang, Kun Gai, Bo An, and Xiangyu Zhao. Generative auto-bidding with value-guided explorations. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 244–254, 2025. doi: 10.1145/3726302.3729987.

Jiayan Guo, Yusen Huo, Zhilin Zhang, Tianyu Wang, Chuan Yu, Jian Xu, Bo Zheng, and Yan Zhang. Generative auto-bidding via conditional diffusion modeling. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5038–5049, 2024a. doi: 10.1145/3637528.3671526.

Xiefan Guo, Jinlin Liu, Miaomiao Cui, Jiankai Li, Hongyu Yang, and Di Huang. InitNO: Boosting text-to-image diffusion models via initial noise optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9380–9389, 2024b.

Yue He, Xiujun Chen, Di Wu, Junwei Pan, Qing Tan, Chuan Yu, Jian Xu, and Xiaoqiang Zhu. A unified solution to constrained bidding in online display advertising. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 2993–3001, 2021. doi: 10.1145/3447548.3467199.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

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

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1179–1191, 2020.

Seunghyun Lee, Younggyo Seo, Kimin Lee, Pieter Abbeel, and Jinwoo Shin. Offline-to-online reinforcement learning via balanced replay and pessimistic q-ensemble. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 1702–1712. PMLR, 2022.

Yewen Li, Shuai Mao, Jingtong Gao, Nan Jiang, Yunjian Xu, Qingpeng Cai, Fei Pan, Peng Jiang, and Bo An. GAS: Generative auto-bidding with post-training search. In *Companion Proceedings of the ACM Web Conference 2025*, 2025. doi: 10.1145/3701716.3715226.

Yewen Li, Jingtong Gao, Peng Jiang, Ruyi An, Xiangyu Zhao, Bo An, Fei Pan, Qingpeng Cai, Peng Jiang, and Kun Gai. Generative auto-bidding in large-scale auctions via diffusion completer-aligner. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2026a.

Yewen Li, Zhiyi Lyu, Peng Jiang, Qingpeng Cai, Fei Pan, Bo An, and Peng Jiang. LBM: Hierarchical large auto-bidding model via reasoning and acting. In *Proceedings of the ACM Web Conference 2026*, pp. 75–86, 2026b. doi: 10.1145/3774904.3792202.

Zuxin Liu, Zijian Guo, Yihang Yao, Zhepeng Cen, Wenhao Yu, Tingnan Zhang, and Ding Zhao. Constrained decision transformer for offline safe reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 21611–21630. PMLR, 2023.

Yixiu Mao, Qi Wang, Yun Qu, Yuhang Jiang, and Xiangyang Ji. Doubly mild generalization for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 51436–51473, 2024.

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

Zhiyu Mou, Yiqin Lv, Miao Xu, Qi Wang, Yixiu Mao, Jinghao Chen, Qichen Ye, Chao Li, Rongquan Bai, Chuan Yu, Jian Xu, and Bo Zheng. Enhancing generative auto-bidding with offline reward evaluation and policy search. In *The Fourteenth International Conference on Learning Representations*, 2026.

S. Muthukrishnan. Ad exchanges: Research issues. In *Internet and Network Economics*, volume 5929 of *Lecture Notes in Computer Science*, pp. 1–12. Springer, 2009.

Mitsuhiko Nakamoto, Yuexiang Zhai, Anikait Singh, Max Sobol Mark, Yi Ma, Chelsea Finn, Aviral Kumar, and Sergey Levine. Cal-QL: Calibrated offline rl pre-training for efficient online fine-tuning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

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

Allen Z. Ren, Justin Lidard, Lars Lien Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. In *The Thirteenth International Conference on Learning Representations*, 2025.

Kan Ren, Weinan Zhang, Ke Chang, Yifei Rong, Yong Yu, and Jun Wang. Bidding machine: Learning to bid for directly optimizing profits in display advertising. *IEEE Transactions on Knowledge and Data Engineering*, 30(4):645–659, 2017.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

Kefan Su, Yusen Huo, Zhilin Zhang, Shuai Dou, Chuan Yu, Jian Xu, Zongqing Lu, and Bo Zheng. AuctionNet: A novel benchmark for decision-making in large-scale games. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

Zhiwei Tang, Jiangweizhi Peng, Jiasheng Tang, Mingyi Hong, Fan Wang, and Tsung-Hui Chang. Inference-time alignment of diffusion models with direct noise optimization. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 58905–58930. PMLR, 2025.

Masatoshi Uehara, Yulai Zhao, Kevin Black, Ehsan Hajiramezanali, Gabriele Scalia, Nathaniel Lee Diamant, Alex M. Tseng, Sergey Levine, and Tommaso Biancalani. Feedback efficient online fine-tuning of diffusion models. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 48892–48918, 2024.

Andrew Wagenmaker, Yunchu Zhang, Mitsuhiko Nakamoto, Seohong Park, Waleed Yagoub, Anusha Nagabandi, Abhishek Gupta, and Sergey Levine. Steering your diffusion policy with latent space reinforcement learning. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 258–282. PMLR, 2025.

Bram Wallace, Akash Gokul, Stefano Ermon, and Nikhil Naik. End-to-end diffusion latent optimization improves classifier guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7280–7290, 2023.

Jun Wang and Shuai Yuan. Real-time bidding: A new frontier of computational advertising research. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pp. 415–416, 2015.

Shengtian Yang, Yewen Li, Peng Jiang, Zhiyi Lyu, Bo An, Peng Jiang, Qingpeng Cai, and Lei Feng. PlatformBid: An auto-bidding benchmark from a unified advertising platform's perspective. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2026.

Yulai Zhao, Masatoshi Uehara, Gabriele Scalia, Sunyuan Kung, Tommaso Biancalani, Sergey Levine, and Ehsan Hajiramezanali. Adding conditional control to diffusion models with reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

APPENDIX

A IMPLEMENTATION DETAILS

In this section, we provide detailed implementation information of AMINO to facilitate reproducibility. We first present the complete training procedure, including the pretraining of the diffusion planner, inverse dynamics model and trajectory evaluator, as well as the subsequent optimization of the initial noise policy. We then describe the network architectures and key hyperparameter configurations of the diffusion planner, trajectory evaluator, noise policy.

A.1 ALGORITHM

Algorithm 1 summarizes the complete training procedure of AMINO, which consists of three stages. First, the conditional diffusion planner and inverse dynamics controller are trained from the offline bidding data. Second, the trajectory evaluator is trained to provide trajectory-level feedback using the constraint-aware score defined in Eq. 4. Finally, the pretrained diffusion planner and trajectory evaluator are frozen, and the initial noise policy is optimized through GNPO with Gaussian parameter clipping and conservative reward shaping.

Algorithm 1 Training Procedure of AMINO

Require: Offline dataset $\mathcal{D}$; diffusion planner D_θ; trajectory evaluator E_ϕ; noise policy π_ψ; inverse dynamics model f_ω; group size G

// Phase 1: Diffusion Planner and Controller Training

while diffusion planner not converged **do**
  Sample a batch of trajectories $\tau \sim \mathcal{D}$
  Randomly sample decision step t and construct history $h_t = s_{1:t}$ and future $\tau_{t+1:T}$
  Sample diffusion step k and noise $\epsilon \sim \mathcal{N}(0, I)$
  Corrupt the future trajectory to obtain $\tau^k_{t+1:T}$
  Update D_θ by minimizing $\mathcal{L}_{\text{diff}} = \mathbb{E}\left[\left\|\epsilon - \epsilon_\theta(\tau^k, k, y_t)\right\|^2_{t+1:T}\right]$
end while
while inverse dynamics model not converged **do**
  Sample transitions $(s_t, a_t, s_{t+1}) \sim \mathcal{D}$
  Predict $\hat{a}_t = f_\omega(s_t, s_{t+1})$
  Update f_ω by minimizing $\mathcal{L}_{\text{inv}} = \mathbb{E}\left[\|a_t - f_\omega(s_t, s_{t+1})\|_2^2\right]$
end while

// Phase 2: Trajectory Evaluator Training

while trajectory evaluator not converged **do**
  Sample a batch of trajectories $\tau \sim \mathcal{D}$
  Randomly sample decision step t and construct $(h_t, \tau_{t+1:T})$
  Compute the ground-truth trajectory score S using Eq. 4
  Add history–future type embeddings and predict $\hat{S} = E_\phi(h_t, \tau_{t+1:T})$
  Update E_ϕ by minimizing $\mathcal{L}_E = \mathbb{E}\left[(E_\phi(h_t, \tau_{t+1:T}) - S)^2\right]$
end while

// Phase 3: Noise Policy Optimization

Freeze the pretrained diffusion planner D_θ and trajectory evaluator E_ϕ
for each noise policy optimization iteration **do**
  Sample a batch of bidding histories $h_t \sim \mathcal{D}$
  for each history h_t **do**
    Predict Gaussian parameters $\mu_\psi(h_t)$ and $\log \sigma_\psi(h_t)$
    Obtain clipped parameters $\mu'_\psi(h_t)$ and $\sigma'_\psi(h_t)$ according to Eq. 6
    Sample G initial noises $z_i = \mu'_\psi(h_t) + \sigma'_\psi(h_t) \odot \xi_i$, $\xi_i \sim \mathcal{N}(0, I)$
    Generate candidate futures $\hat{\tau}_i = D_\theta(h_t, y_t, z_i)$
    Evaluate candidate trajectories $S_i = E_\phi(h_t, \hat{\tau}_i)$
    Apply conservative reward shaping to obtain S'_i
    Compute group-relative advantages A_i using Eq. 8
  end for
  Update π_ψ by maximizing the GNPO objective in Eq. 9
end for
return D_θ, f_ω, E_ϕ, and π_ψ

A.2 HYPERPARAMETER SETTINGS

We provide the implementation details and hyperparameter settings of the three main components of AMINO: the diffusion planner, trajectory evaluator, and noise policy. The same configurations are used across different evaluation settings unless otherwise specified.

Diffusion Planner. The diffusion planner is implemented with a temporal U-Net to model state trajectories with a planning horizon of 48 and a state dimension of 16. We use $K = 20$ diffusion steps and deterministic DDIM sampling at inference. Classifier-free guidance is enabled by randomly dropping the return condition during training. The detailed hyperparameters are summarized in Table 4.

Table 4: Hyperparameters of the diffusion planner.

Parameter	Value	Description
Planning horizon H	48	Number of decision steps in a trajectory
State dimension D	16	Dimension of each bidding state
Diffusion steps K	20	Number of diffusion and denoising steps
Target return	1.0	Return condition used to guide trajectory generation
Return normalization scale	2000	Scaling factor used to normalize return conditions
CFG scale w	0.5	Strength of return-conditioned guidance
Condition dropout	0.25	Probability of dropping the return condition during CFG training
Base channels	128	Base width of the temporal U-Net
Channel multipliers	$(1, 2, 4)$	Channel scaling factors across U-Net stages
Convolution kernel size	5	Kernel size of temporal convolutions
Sampler	DDIM	Deterministic sampling for noise-to-trajectory mapping
Optimizer	AdamW	Optimizer used for diffusion training
Learning rate	2×10^{-4}	Initial learning rate
Weight decay	10^{-4}	AdamW weight-decay coefficient
Batch size	128	Number of trajectories per training batch
Training steps	4×10^{5}	Total number of gradient updates
Gradient clipping	1.0	Maximum global gradient norm

Trajectory Evaluator. The trajectory evaluator is implemented as a bidirectional Transformer that jointly encodes the observed history and candidate future trajectory. History–future type embeddings are introduced to distinguish the two segments, and the resulting representation is mapped to a scalar trajectory score. The evaluator is trained using the MSE objective described in Section 5.2. Its hyperparameters are reported in Table 5.

Noise Policy. The noise policy is parameterized by a Transformer that encodes the observed bidding history and predicts the parameters of a diagonal Gaussian distribution over the initial noise. The predicted mean and scale are bounded to prevent the policy distribution from deviating excessively from the pretrained diffusion prior. For each history, G noise candidates are sampled and optimized using GNPO with entropy regularization and conservative reward shaping. The detailed hyperparameters are provided in Table 6.

B CONDITIONAL DIFFUSION FOR AUTO-BIDDING

Conditional diffusion-based auto-bidding methods typically decompose the sequential decision-making problem into two stages: *planning* and *control*. Rather than directly predicting bidding actions, the planner models the correlation between the desired return and the entire bidding trajectory, thereby transforming auto-bidding into a conditional trajectory generation problem. Once a desirable state trajectory has been planned, a lightweight inverse dynamics model can subsequently translate the planned state transitions into executable bidding actions. In this paradigm, the main modeling capacity is therefore devoted to trajectory planning, while the control stage can be implemented with a relatively simple supervised model.

Table 5: Hyperparameters of the trajectory evaluator.

Parameter	Value	Description
Architecture	Bidirectional Transformer	Full-attention encoder over history and future states
Transformer layers	6	Number of Transformer encoder layers
Attention heads	8	Number of attention heads per layer
Hidden dimension	512	Dimension of token representations
Dropout	0.1	Dropout probability in the Transformer
Pooling	Last valid token	Sequence representation used for score prediction
Score head	$512 \to 512 \to 1$	MLP that maps the sequence representation to a scalar score
Activation	GELU	Nonlinear activation in the score head
Loss	MSE	Regression loss for trajectory-score prediction
Optimizer	AdamW	Optimizer used for evaluator training
Learning rate	10^{-5}	Initial learning rate
Weight decay	10^{-4}	AdamW weight-decay coefficient
Warmup steps	10^4	Linear learning-rate warmup steps
Batch size	128	Number of trajectories per training batch
Training steps	4×10^5	Total number of gradient updates
Gradient clipping	0.25	Maximum global gradient norm

Table 6: Hyperparameters of the noise policy.

Parameter	Value	Description
Architecture	Transformer	Encodes the observed bidding history
Transformer layers	6	Number of Transformer encoder layers
Attention heads	8	Number of attention heads per layer
Hidden dimension	512	Dimension of history representations
Dropout	0	Dropout probability in the noise policy
Mean clipping	$[-0.5, 0.5]$	Bounds the predicted Gaussian mean μ_ψ
Log-standard-deviation clipping	$[-5, 0]$	Bounds $\log \sigma_\psi$ for stable stochastic exploration
History batch size	256	Number of bidding histories sampled per update
Group size G	32	Number of initial noise samples for each history
Entropy coefficient α	10^{-2}	Weight of dimension-averaged entropy regularization
Reward threshold coefficient δ	0.8	Controls the reliability threshold for reward shaping
reward shaping coefficient λ_s	1	Penalty strength for scores above the reliability threshold
Optimizer	AdamW	Optimizer used for noise policy training
Learning rate	10^{-5}	Initial learning rate
Weight decay	10^{-4}	AdamW weight-decay coefficient
Warmup steps	500	Learning-rate warmup steps
Training steps	3000	Total number of noise policy updates
Gradient clipping	1.0	Maximum global gradient norm

Let

$$\tau_0 = [s_1, s_2, \ldots, s_T] \tag{11}$$

denote a clean bidding trajectory, and let y denote the return condition. Conditional diffusion models trajectory generation through a forward noising process and a reverse denoising process:

$$q(\tau_k \mid \tau_{k-1}), \qquad p_\theta(\tau_{k-1} \mid \tau_k, y), \tag{12}$$

where $k \in \{1, \ldots, K\}$ denotes the diffusion step.

Forward Process. The forward process gradually adds Gaussian noise to the trajectory according to a predefined variance schedule $\{\beta_k\}_{k=1}^K$:

$$q(\tau_k \mid \tau_{k-1}) = \mathcal{N}\left(\tau_k; \sqrt{1-\beta_k}\tau_{k-1}, \beta_k I\right). \tag{13}$$

Let

$$\alpha_k = 1 - \beta_k, \qquad \bar{\alpha}_k = \prod_{i=1}^{k} \alpha_i. \tag{14}$$

A useful property of the forward process is that the noisy trajectory at an arbitrary diffusion step k can be sampled directly from the clean trajectory:

$$q(\tau_k \mid \tau_0) = \mathcal{N}\left(\tau_k; \sqrt{\bar{\alpha}_k}\tau_0, (1 - \bar{\alpha}_k)I\right). \tag{15}$$

Equivalently,

$$\tau_k = \sqrt{\bar{\alpha}_k}\tau_0 + \sqrt{1 - \bar{\alpha}_k}\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{16}$$

As k increases, the trajectory becomes progressively corrupted and eventually approaches a standard Gaussian distribution.

Reverse Process. The reverse process aims to recover the clean bidding trajectory by gradually removing the noise introduced during the forward process. It is modeled as

$$p_\theta(\tau_{k-1} \mid \tau_k, y) = \mathcal{N}\left(\tau_{k-1}; \mu_\theta(\tau_k, y, k), \Sigma_\theta(\tau_k, y, k)\right). \tag{17}$$

In practice, instead of directly predicting the reverse mean, diffusion models typically parameterize the reverse process through a noise prediction network $\epsilon_\theta(\tau_k, y, k)$. A temporal U-Net is commonly adopted to model the sequential structure of bidding trajectories.

Training. During training, the reverse process does not need to be recursively executed from step K to step 0. For each clean trajectory τ_0, a diffusion step k is randomly sampled, together with Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$. Using the closed-form forward process, the corresponding noisy trajectory can be obtained directly as

$$\tau_k = \sqrt{\bar{\alpha}_k}\tau_0 + \sqrt{1 - \bar{\alpha}_k}\epsilon. \tag{18}$$

The denoising network then takes (τ_k, y, k) as input and predicts the injected noise. The diffusion model is trained by minimizing

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{\tau_0, y, k, \epsilon}\left[\|\epsilon - \epsilon_\theta(\tau_k, y, k)\|_2^2\right]. \tag{19}$$

Therefore, although the learned model represents a multi-step reverse generative process, each training sample only requires noise prediction at a randomly selected diffusion step.

The inverse dynamics controller can be trained simultaneously or separately from offline transitions. Given two consecutive states (s_t, s_{t+1}), it predicts the action that induces the corresponding state transition:

$$\hat{a}_t = f_\omega(s_t, s_{t+1}), \tag{20}$$

with the supervised objective

$$\mathcal{L}_{\text{inv}} = \mathbb{E}_{(s_t, a_t, s_{t+1}) \sim \mathcal{D}}\left[\|a_t - f_\omega(s_t, s_{t+1})\|_2^2\right]. \tag{21}$$

Inference. Inference differs fundamentally from training. Since the clean trajectory is unknown, trajectory generation starts from Gaussian noise

$$\tau_K \sim \mathcal{N}(0, I), \tag{22}$$

and the learned reverse process is recursively applied for $k = K, K-1, \ldots, 1$.

Given τ_k, the noise prediction $\epsilon_\theta(\tau_k, y, k)$ is first used to compute the mean of the reverse distribution. Under the standard DDPM parameterization,

$$\mu_\theta(\tau_k, y, k) = \frac{1}{\sqrt{\alpha_k}}\left(\tau_k - \frac{\beta_k}{\sqrt{1 - \bar{\alpha}_k}}\epsilon_\theta(\tau_k, y, k)\right). \tag{23}$$

The trajectory at the next reverse step is then sampled as

$$\tau_{k-1} = \mu_\theta(\tau_k, y, k) + \sigma_k z, \qquad z \sim \mathcal{N}(0, I), \tag{24}$$

where σ_k is determined by the diffusion variance schedule. By repeatedly applying this denoising procedure, the model gradually transforms the initial Gaussian noise into a generated bidding trajectory τ_0.

Conditional diffusion-based auto-bidding methods may further employ classifier-free guidance to strengthen the influence of the return condition. The conditional and unconditional noise predictions are combined as

$$\tilde{\epsilon}_k = \epsilon_\theta(\tau_k, k) + w\left[\epsilon_\theta(\tau_k, y, k) - \epsilon_\theta(\tau_k, k)\right], \tag{25}$$

where w denotes the guidance strength. The guided prediction $\tilde{\epsilon}_k$ is then used in place of $\epsilon_\theta(\tau_k, y, k)$ during the reverse denoising process.

After obtaining the generated trajectory τ_0, the planned next state $\hat{s}_{t+1}$ is converted into an executable bidding action through the inverse dynamics controller:

$$\hat{a}_t = f_\phi(s_t, \hat{s}_{t+1}). \tag{26}$$

Thus, conditional diffusion is responsible for high-level trajectory planning, while the inverse dynamics model performs low-level bidding control.

C Proof of Proposition 1

Let $p_{\mathcal{D}}$ denote the data distribution induced by the offline dataset $\mathcal{D}$. Accordingly, $p_{\mathcal{D}}(m_i \mid h_t)$ denotes the probability of trajectory mode m_i given history h_t, while $p_{\mathcal{D}}(y_t \mid m_i, h_t)$ denotes the conditional density of trajectory return y_t within mode m_i.

Proposition 1 (Extreme-return-induced mode concentration). Given history h_t, suppose $p_{\mathcal{D}}(y_t \mid m_i, h_t) = \mathcal{N}(y_t; \mu_i, \sigma_i^2)$, and the extreme right tail admits a unique dominant mode $m^\dagger$. Then

$$\lim_{y^\star \to +\infty} p_{\mathcal{D}}(m^\dagger \mid h_t, y^\star) = 1, \qquad \lim_{y^\star \to +\infty} H(M \mid h_t, y^\star) = 0. \tag{27}$$

Proof. For each trajectory mode m_i, the Gaussian return density is

$$p_{\mathcal{D}}(y_t \mid m_i, h_t) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(y_t - \mu_i)^2}{2\sigma_i^2}\right). \tag{28}$$

We first characterize the asymptotic behavior of different modes in the extreme high-return region. For two modes m_i and m_j,

$$\begin{aligned}\log \frac{p_{\mathcal{D}}(y_t \mid m_j, h_t)}{p_{\mathcal{D}}(y_t \mid m_i, h_t)} &= \log \frac{\sigma_i}{\sigma_j} - \frac{(y_t - \mu_j)^2}{2\sigma_j^2} + \frac{(y_t - \mu_i)^2}{2\sigma_i^2} \\ &= \left(\frac{1}{2\sigma_i^2} - \frac{1}{2\sigma_j^2}\right) y_t^2 + \left(\frac{\mu_j}{\sigma_j^2} - \frac{\mu_i}{\sigma_i^2}\right) y_t + C_{ij},\end{aligned} \tag{29}$$

where $C_{ij} = \log(\sigma_i/\sigma_j) - \mu_j^2/(2\sigma_j^2) + \mu_i^2/(2\sigma_i^2)$ is independent of y_t.

Eq. 29 shows that the quadratic term determines the asymptotic ordering when the variances differ. Specifically, if $\sigma_i^2 > \sigma_j^2$, then $1/(2\sigma_i^2) - 1/(2\sigma_j^2) < 0$. Therefore, as $y_t \to +\infty$,

$$\log \frac{p_{\mathcal{D}}(y_t \mid m_j, h_t)}{p_{\mathcal{D}}(y_t \mid m_i, h_t)} \to -\infty, \tag{30}$$

which implies

$$\frac{p_{\mathcal{D}}(y_t \mid m_j, h_t)}{p_{\mathcal{D}}(y_t \mid m_i, h_t)} \to 0. \tag{31}$$

If instead $\sigma_i^2 = \sigma_j^2 = \sigma^2$, the quadratic terms cancel. When $\mu_i > \mu_j$,

$$\log \frac{p_{\mathcal{D}}(y_t \mid m_j, h_t)}{p_{\mathcal{D}}(y_t \mid m_i, h_t)} = -\frac{(\mu_i - \mu_j)(2y_t - \mu_i - \mu_j)}{2\sigma^2} \to -\infty \qquad \text{as } y_t \to +\infty, \tag{32}$$

and consequently

$$\frac{p_{\mathcal{D}}(y_t \mid m_j, h_t)}{p_{\mathcal{D}}(y_t \mid m_i, h_t)} \to 0. \tag{33}$$

Thus, among Gaussian components, the extreme right tail is first dominated by the mode with the largest variance; if several modes share the same largest variance, the one with the largest mean dominates. Under the uniqueness condition in Proposition 1, this identifies a unique dominant mode $m^\dagger$, and for every $m_j \neq m^\dagger$,

$$\lim_{y_t \to +\infty} \frac{p_{\mathcal{D}}(y_t \mid m_j, h_t)}{p_{\mathcal{D}}(y_t \mid m^\dagger, h_t)} = 0. \tag{34}$$

We now consider the trajectory-mode probabilities conditioned on the target return $y^\star$. By Bayes' rule,

$$p_{\mathcal{D}}(m_i \mid h_t, y^\star) = \frac{p_{\mathcal{D}}(m_i \mid h_t) p_{\mathcal{D}}(y^\star \mid m_i, h_t)}{\sum_{l=1}^{M} p_{\mathcal{D}}(m_l \mid h_t) p_{\mathcal{D}}(y^\star \mid m_l, h_t)}. \tag{35}$$

For any $m_j \neq m^\dagger$, dividing its probability by that of $m^\dagger$ gives

$$\frac{p_{\mathcal{D}}(m_j \mid h_t, y^\star)}{p_{\mathcal{D}}(m^\dagger \mid h_t, y^\star)} = \frac{p_{\mathcal{D}}(m_j \mid h_t)}{p_{\mathcal{D}}(m^\dagger \mid h_t)} \frac{p_{\mathcal{D}}(y^\star \mid m_j, h_t)}{p_{\mathcal{D}}(y^\star \mid m^\dagger, h_t)}. \tag{36}$$

The first ratio on the right-hand side is independent of $y^\star$, while the second ratio converges to zero by Eq. 34. Therefore,

$$\lim_{y^\star \to +\infty} \frac{p_{\mathcal{D}}(m_j \mid h_t, y^\star)}{p_{\mathcal{D}}(m^\dagger \mid h_t, y^\star)} = 0, \qquad \forall m_j \neq m^\dagger. \tag{37}$$

Since the probabilities over the finite set of modes sum to one,

$$p_{\mathcal{D}}(m^\dagger \mid h_t, y^\star) = \frac{1}{1 + \sum_{j \neq \dagger} \frac{p_{\mathcal{D}}(m_j \mid h_t, y^\star)}{p_{\mathcal{D}}(m^\dagger \mid h_t, y^\star)}}. \tag{38}$$

Using Eq. 37, we obtain

$$\lim_{y^\star \to +\infty} p_{\mathcal{D}}(m^\dagger \mid h_t, y^\star) = 1. \tag{39}$$

Consequently, $p_{\mathcal{D}}(m_j \mid h_t, y^\star) \to 0$ for every $m_j \neq m^\dagger$.

Finally,

$$H(M \mid h_t, y^\star) = -\sum_{i=1}^{M} p_{\mathcal{D}}(m_i \mid h_t, y^\star) \log p_{\mathcal{D}}(m_i \mid h_t, y^\star). \tag{40}$$

As $y^\star \to +\infty$, the conditional mode distribution converges to a point mass on $m^\dagger$. Since $\lim_{x \to 0^+} x \log x = 0$ and $\lim_{x \to 1} x \log x = 0$, it follows that

$$\lim_{y^\star \to +\infty} H(M \mid h_t, y^\star) = 0. \tag{41}$$

This concludes the proof. □

D Experimental Details

D.1 Dataset and Evaluation Protocol

AuctionNet. We conduct experiments on AuctionNet (Su et al., 2024), a large-scale public benchmark released by Alibaba for sequential decision making in real-world advertising auctions. AuctionNet simulates the major components of an industrial advertising system, including impression opportunity generation, auto-bidding, and auction execution, thereby providing a realistic environment for evaluating bidding policies.

The benchmark contains two datasets, **AuctionNet** and **AuctionNet-Sparse**. AuctionNet-Sparse follows the same state representation and interaction protocol as AuctionNet, but contains substantially sparser conversion feedback. Both datasets include 21 advertising delivery periods, with each period containing approximately 500,000 impression opportunities. Each trajectory consists of 48 decision steps, with a 16-dimensional state and a one-dimensional bidding action. The detailed statistics of the two datasets are summarized in Table 7.

Table 7: Statistics of AuctionNet and AuctionNet-Sparse.

Parameter	AuctionNet	AuctionNet-Sparse
Trajectories	479,376	479,376
Time steps in a trajectory	48	48
State dimension	16	16
Action dimension	1	1
Return-to-Go dimension	1	1
Action range	$[0, 493]$	$[0, 589]$
Impression's value range	$[0, 1]$	$[0, 1]$
CPA range	$[6, 12]$	$[60, 130]$
Total conversion range	$[0, 1512]$	$[0, 57]$

State Representation. At each decision step, the environment provides a 16-dimensional state vector that summarizes the remaining resources, historical bidding statistics, short-term auction statistics, and current traffic information. Specifically, the state consists of the following features:

- time_left: the number of remaining decision intervals in the current advertising delivery period.
- budget_left: the remaining budget available to the advertiser.
- historical_bid_mean: the average bid value over previous decision intervals.
- last_three_bid_mean: the average bid value over the most recent three decision intervals.
- historical_LeastWinningCost_mean: the historical average of the least winning cost, i.e., the minimum cost required to win an impression.
- historical_pValues_mean: the average historical p-value over previous decision intervals.
- historical_conversion_mean: the historical average number of conversions.
- historical_xi_mean: the historical average winning indicator, where 1 represents winning an impression opportunity and 0 otherwise.
- last_three_LeastWinningCost_mean: the average least winning cost over the most recent three decision intervals.
- last_three_pValues_mean: the average p-value over the most recent three decision intervals.
- last_three_conversion_mean: the average number of conversions over the most recent three decision intervals.
- last_three_xi_mean: the average winning indicator over the most recent three decision intervals.
- current_pValues_mean: the mean p-value of impression opportunities in the current decision interval.
- current_pv_num: the number of impression opportunities in the current decision interval.
- last_three_pv_num_total: the total number of impression opportunities over the most recent three decision intervals.
- historical_pv_num_total: the total number of impression opportunities observed over previous decision intervals.

The state representation jointly captures long-term bidding statistics, short-term dynamics over the most recent three intervals, and information about the current traffic, providing the policy with a comprehensive description of the ongoing advertising delivery process.

Evaluation Details. Following AuctionNet (Su et al., 2024) and recent auto-bidding studies (Li et al., 2025; 2026b; Gao et al., 2025), we evaluate all methods in Alibaba's simulated advertising environment. Each episode represents a 24-hour delivery period with 48 decision intervals and approximately 500,000 sequential impression opportunities, where 48 advertisers compete under heterogeneous bidding strategies, budgets, and CPA constraints.

We adopt the round-robin evaluation protocol of AuctionNet, where the evaluated model sequentially replaces each of the 48 bidding agents and competes with the remaining 47 agents. Performance is averaged over all replacement rounds, different delivery periods and advertiser configurations, and models trained with 5 random seeds.

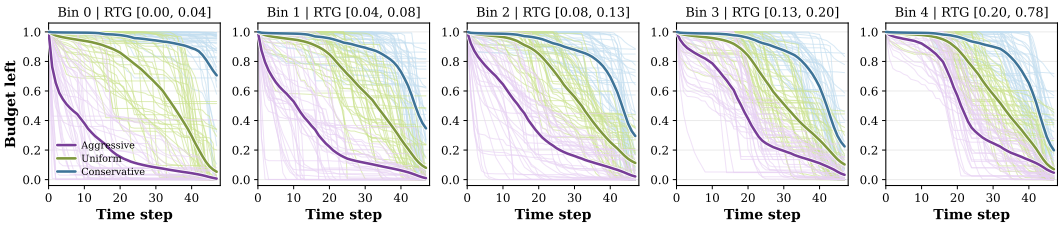


Figure 6: Budget-consumption patterns across five RTG bins. Thin curves denote individual normalized trajectories and thick curves denote the cluster centers of aggressive, uniform, and conservative patterns.

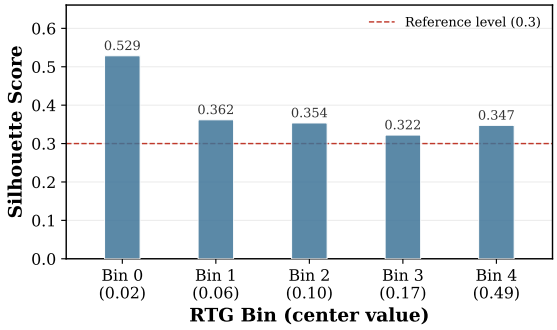


Figure 7: Silhouette Scores across RTG bins, with 0.3 shown as a reference level.

D.2 Multimodal Structure of Budget-Consumption Trajectories

To examine whether multimodal budget-consumption behaviors exist under similar return conditions, we analyze the offline trajectories in AuctionNet.

Trajectory Grouping and Clustering. We characterize each trajectory by its return-to-go (RTG) at $t = 0$ and divide the dataset into five quantile-based RTG bins. Within each bin, the normalized budget_left trajectories are clustered into three representative patterns: *aggressive*, *uniform*, and *conservative*.

To align cluster labels across RTG bins, we order the cluster centers according to their temporal center of budget consumption. Let b_t denote the normalized remaining budget and $\Delta c_t = b_{t-1} - b_t$. The temporal center is

$$c = \frac{\sum_{t=1}^{T} \frac{t}{T} \Delta c_t}{\sum_{t=1}^{T} \Delta c_t}. \tag{42}$$

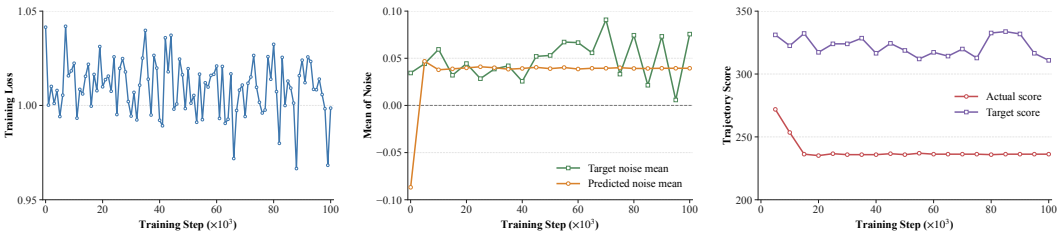
Smaller c corresponds to earlier spending, while larger c indicates later spending. The three clusters are therefore consistently mapped to aggressive, uniform, and conservative patterns.

Figure 6 shows that all five RTG bins contain clearly different consumption patterns despite having comparable return levels, providing direct evidence of conditional trajectory diversity.

Cluster Separation. We further quantify the separation between trajectory patterns using the Silhouette Score. For trajectory i, let $a(i)$ denote its average intra-cluster distance and $b(i)$ the minimum average distance to another cluster:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}. \tag{43}$$

The score ranges from -1 to 1, with larger values indicating more compact and better-separated clusters.



(a) (b) (c)

Figure 8: Training behavior of the CEM-based noise policy. (a) Clipped-MSE training loss. (b) Mean of the selected target noises and predicted noises. (c) Rescaled trajectory scores obtained from the deterministic predicted noise and the Best-of-64 target.

Table 8: Comparison between GNPO and CEM-based noise optimization.

Metric	AMINO	Diffusion+CEM
Score ↑	**374**	257
Conversions ↑	**409**	345
CPA Ratio ↓	**0.91**	1.33

As shown in Figure 7, the five bins achieve Silhouette Scores of 0.529, 0.362, 0.354, 0.322, and 0.347, respectively, all above the 0.3 reference level. Together with Figure 6, these results show that trajectories with similar returns can exhibit distinct and separable budget-consumption patterns, supporting the need to model the full trajectory distribution rather than a single deterministic prediction.

E Additional Experimental Results

E.1 Comparison with CEM-Based Noise Optimization

We further compare GNPO with a CEM-based alternative for optimizing the initial noise. We consider a CEM-based variant with an elite-set size of 1, i.e., only the highest-scoring noise is retained as the optimization target. Given a bidding history h_t, we sample $N = 64$ initial noises from a standard Gaussian distribution,

$$z_i \sim \mathcal{N}(0, I), \qquad i = 1, \ldots, N. \tag{44}$$

Each noise is combined with the current ground-truth return condition y_t and mapped by the frozen diffusion planner through DDIM to a candidate future trajectory, $\hat{\tau}_i = D_\theta(h_t, y_t, z_i)$. The frozen trajectory evaluator assigns each candidate a score $S_i = E_\phi(h_t, \hat{\tau}_i)$, and the highest-scoring noise is selected as the supervision target,

$$i^\star = \arg\max_{i \in \{1,\ldots,N\}} S_i, \qquad z^\star = z_{i^\star}. \tag{45}$$

Instead of repeating this Best-of-N search online, a noise policy f_ψ is trained to directly predict the selected noise from the bidding history. Specifically, its raw prediction is first clipped element-wise, $\hat{z}_\psi = \mathrm{clip}\left(f_\psi(h_t), -c, c\right)$, and optimized by minimizing the mean-squared error to the selected target,

$$\mathcal{L}_{\mathrm{CEM}} = \frac{1}{d_z} \sum_{j=1}^{d_z} \left(\hat{z}_{\psi,j} - z_j^\star\right)^2, \tag{46}$$

where d_z denotes the noise dimension and c is the clipping bound. At inference time, the clipped prediction $\hat{z}_\psi$ is directly fed into the frozen diffusion planner, eliminating the need for Best-of-N search.

Figure 8 illustrates the limitation of directly regressing the selected highest-scoring noise. The training loss in Figure 8(a) remains around 1.0, while Figure 8(b) shows that the predicted noise mean

Table 9: Conversions performance under different budget settings. Results of DT+GMM and DRIVE are taken from DRIVE (Cui et al., 2026).

Method	50%	75%	100%	125%	150%
DT+GMM	196	290	372	450	520
DRIVE	212	297	399	475	**551**
AMINO	**220**	**312**	**409**	**478**	525

quickly converges to an almost constant value near the overall target mean rather than tracking individual targets. This suggests that point-wise MSE averages heterogeneous high-quality noise targets, which explains the persistent regression error and can produce predictions that do not correspond to any desirable mode. Consequently, the deterministic prediction in Figure 8(c) remains substantially below the Best-of-N target and eventually stabilizes at a score even lower than its initial value.

GNPO instead optimizes a history-dependent noise policy through group-relative trajectory-level feedback. By comparing candidate trajectories within each group, it increases the probability of noises generating higher-quality trajectories and suppresses inferior ones, without imitating a single selected target. This shifts probability mass toward high-quality regions of the noise space while avoiding the averaging problem of point-wise regression.

E.2 Comparison with GMM-Based Multimodal Modeling

We further compare AMINO with GMM-based multimodal modeling using the results reported in DRIVE (Cui et al., 2026). Specifically, we consider the *Actor Only (Dominant Mean)* variant in DRIVE, denoted as DT+GMM here, which retains the DT backbone and GMM action head without the retrieval or critic modules. DRIVE models the conditional action distribution using M Gaussian components,

$$p(a_t \mid h_t) = \sum_{m=1}^{M} \alpha_m \mathcal{N}\left(a_t; \mu_m, \sigma_m^2\right), \tag{47}$$

where α_m, μ_m, and σ_m^2 denote the mixture weight, mean, and variance of the m-th component, respectively. This provides an explicit way to represent multiple action modes and alleviates the averaging effect of deterministic regression.

As shown in Table 9, AMINO consistently outperforms DT+GMM across all five budget settings. Although GMM explicitly models multimodal actions, its representational capacity remains tied to a fixed number M of parametric Gaussian components. Complex bidding behaviors may therefore be insufficiently represented, and selecting the dominant component can still lead to suboptimal trajectories. In contrast, AMINO models the distribution of complete future trajectories with a diffusion model and further optimizes the initial noise distribution toward high-quality trajectory regions, without restricting multimodality to a fixed number of Gaussian action components.

Full DRIVE alleviates the limitation of the standalone GMM policy by introducing retrieval-based candidate augmentation and critic-based selection, substantially improving over DT+GMM. Nevertheless, AMINO still achieves higher conversions in four of the five budget settings, while DRIVE performs better at the 150% budget. These results suggest that trajectory-level multimodal modeling combined with noise-space optimization provides an effective alternative to explicitly parameterizing multimodality at the action level.

E.3 Trajectory and Score Evolution Cases during GNPO Training

To complement the noise space visualization in Section 6.5, we further examine the decoded trajectories and evaluator scores of the same five bidding histories during GNPO training. For each checkpoint in $\{0, 300, 600, 900, 1200, 1500\}$, the corresponding noise is decoded by the frozen diffusion planner, and the resulting trajectory is de-normalized for visualization. The evaluator scores are rescaled for easier comparison.

As shown in Figures 9 and 10, the generated trajectories progressively change with noise policy optimization, accompanied by an overall improvement in evaluator scores. This provides direct

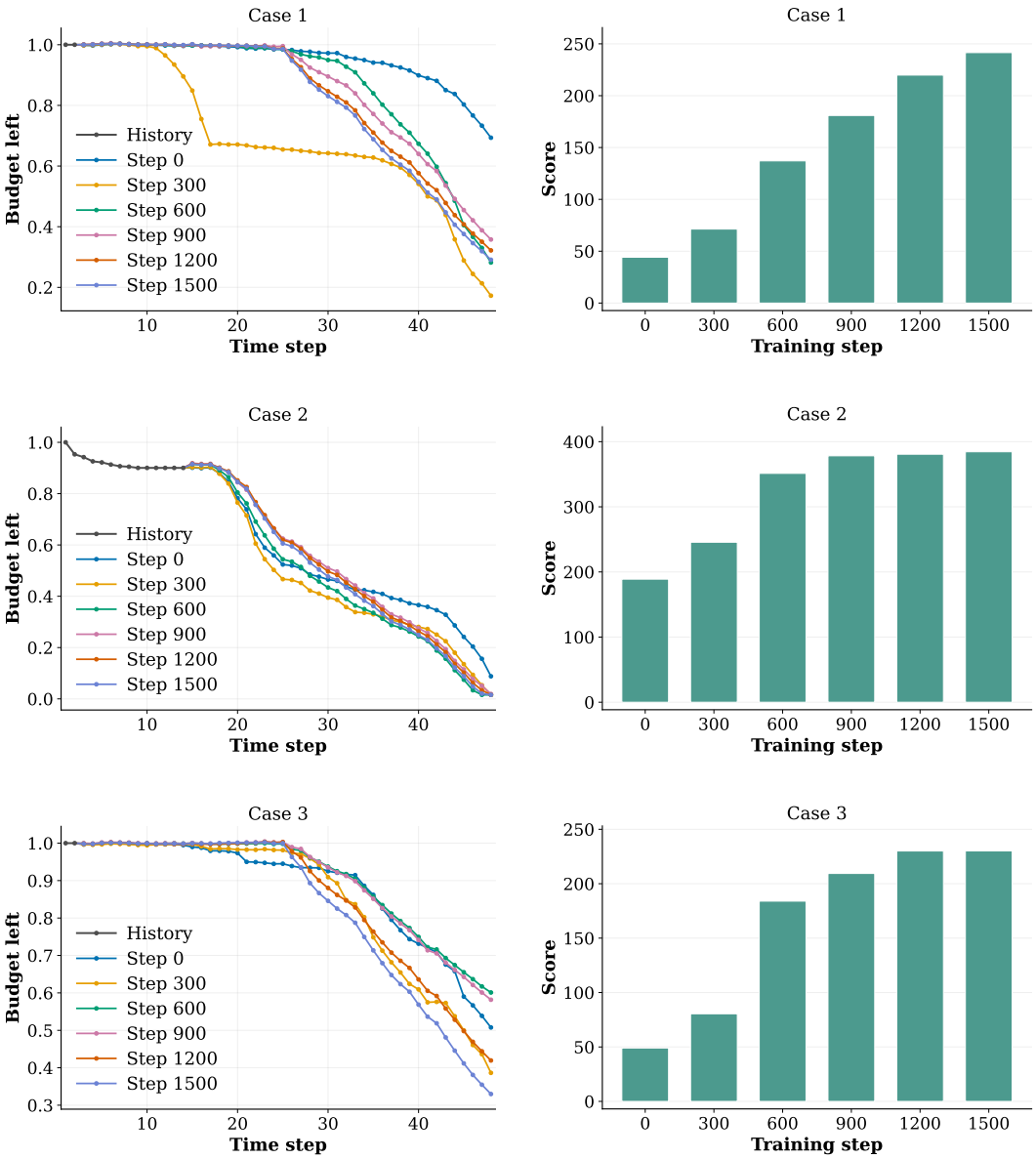


Figure 9: Trajectory and score evolution for Cases 1–3 during GNPO training. For each case, the left panel shows the decoded trajectories at different training checkpoints, and the right panel shows the corresponding rescaled evaluator scores.

trajectory-level evidence that the evolution observed in the noise space translates into higher-quality future trajectories.

F LIMITATION

AMINO is primarily trained on offline data and does not address adaptation after deployment. A key challenge is the offline-to-online transition, where direct online fine-tuning may cause severe performance degradation due to distribution shift and unstable policy updates (Lee et al., 2022; Nakamoto et al., 2023; Feng et al., 2024). Developing safe online adaptation that exploits newly collected interactions while preserving the pretrained policy's initial performance remains an important direction for future work.

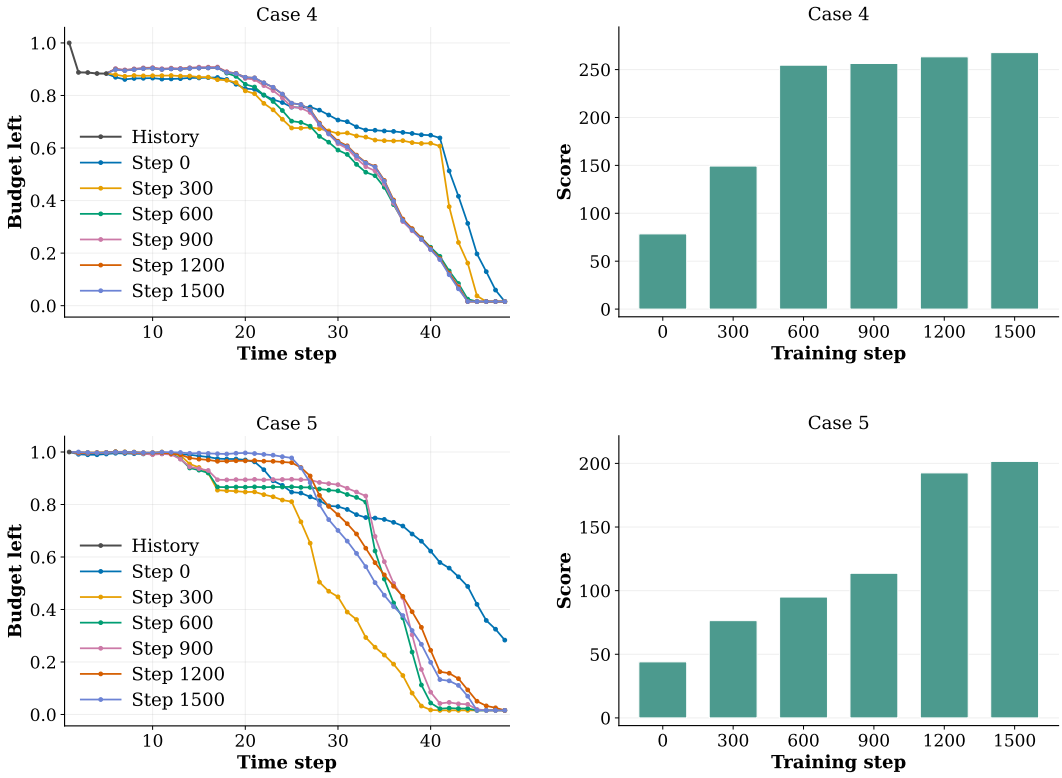


Figure 10: Trajectory and score evolution for Cases 4–5 during GNPO training.