Iterative Scarcity-Guided Exploration: Bootstrapping Generative Auto-bidding from Narrow Support

Qingmao Yao[1], **Guangzheng Hu**[2], **Yusen Huo**[4*], **Zhilin Zhang**[4*]
Chuan Yu[2], **Jian Xu**[2], **Xu Chu**[1†], **Bo Zheng**[2†], **Xiaotie Deng**[1,3†]
[1]Center on Frontiers of Computing Studies, School of Computer Science, Peking University
[2]Alimama Tech, Taobao & Tmall Group of Alibaba
[3]Department of Computer Science, City University of Hong Kong
[4]Independent Researcher
qmyao25@stu.pku.edu.cn, huguangzheng.hgz@taobao.com, huoyusen1@126.com
421525432@qq.com, {yuchuan.yc,xiyu.xj}@taobao.com
bozheng@alibaba-inc.com, {chu_xu,xiaotie}@pku.edu.cn

Abstract

Auto-bidding is a core algorithmic component in online advertising auctions. In practical cold-start settings, iterative training is commonly used to compensate for low-quality, narrowly supported historical data. Unfortunately, in iterative training, representative diffusion-based AI-Generated Bidding (AIGB) methods fail to sustain extrapolation beyond the narrow data support and stagnate at a suboptimal level. In this paper, we theoretically attribute this stagnation to Signal-to-Noise Ratio (SNR) collapse: the weak radial return signal is overwhelmed by a curvature-induced penalty. To break this stagnation, we propose **Iterative Scarcity-Guided Exploration (ISGE)**, which introduces a guidance handoff: as the return signal collapses, scarcity subsequently takes over as an exploratory signal to elevate the SNR above a critical threshold. Specifically, ISGE iteratively alternates between a *Judger* that assigns scarcity scores to trajectories and an *Explorer* that performs guided generation of high-scarcity and high-return trajectories, thereby bootstrapping from the narrow support with such self-generated trajectories. Extensive experiments on the industrial AuctionNet benchmark demonstrate that ISGE, starting from cold-start datasets, effectively surpasses the performance of the Full-Dataset baseline within three iterations. Code is available at https://github.com/yqpqry/ISGE.

1 Introduction

Auto-bidding in online advertising is a complex sequential decision-making problem [1–3]. Recently, diffusion-based AIGB methods have emerged as a powerful paradigm by formulating bidding as trajectory generation via conditional diffusion planners [4–6]. However, these models heavily rely on broad-coverage expert data. In practical cold-start settings for new advertisers or emerging business sectors, historical data is often generated by conservative strategies, yielding a narrow, low-quality distribution. Bootstrapping beyond this poor initialization requires iterative training, yet existing diffusion planners lack efficient mechanisms to explore high-return out-of-distribution (OOD) trajectories. This raises a central question:

Can diffusion-based AIGB methods perform efficient exploration to bootstrap a near-optimal policy solely from low-quality, narrowly supported data in just a few iterations?

*This work was completed while the authors were at Alibaba.
†Corresponding authors

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

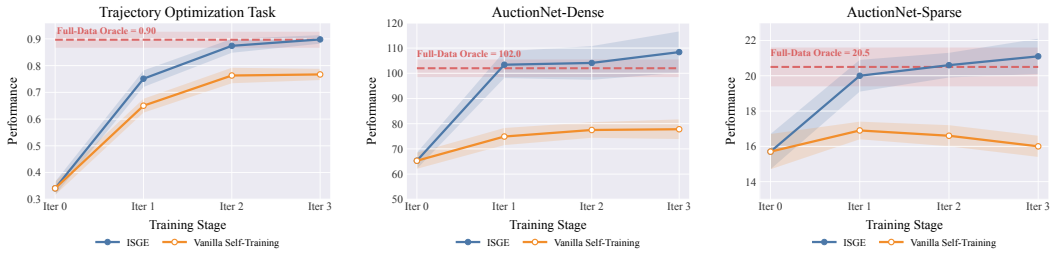


Figure 1: Iterative Experiments on Trajectory Optimization Task (left) and AuctionNet Benchmarks (middle, right). Vanilla Self-Training (orange) inevitably stagnates well below the optimal level during iterative training. We term this self-locking. In contrast, our proposed ISGE (blue) successfully breaks self-locking and consistently reaches or surpasses the Full-Data Oracle (red dashed line) within just three iterations. Full results in Section 3.1 and Section 5.

Investigating this, we observe that vanilla self-training (naively iterating diffusion planners) suffers from a bottleneck we term *self-locking*. As illustrated in Figure 1, despite initial extrapolation capabilities, performance inevitably stagnates well below the optimum. We theoretically attribute this to an inherent Signal-to-Noise Ratio (SNR) collapse at the boundary of the training support. As generated trajectories reach this suboptimal frontier, the return signal becomes sparse and misaligned, failing to overcome the high-dimensional curvature penalty and trapping the planner.

To break self-locking, we propose Iterative Scarcity-Guided Exploration (ISGE). ISGE relies on the core insight of a *guidance handoff*, utilizing scarcity (quantified via denoising autoencoder [7] reconstruction error) as a robust auxiliary signal. Geometrically, scarcity acts as an outward manifold repulsion field [8]. During reverse denoising, ISGE executes a guidance handoff: at early steps, return guidance navigates the known landscape to push trajectories to the suboptimal boundary. When the return signal inevitably collapses, scarcity subsequently takes over. This provides a stable outward impulse that reliably pushes the trajectory deeper into unexplored OOD space, explicitly preventing regression into the low-quality interior. By leveraging the local continuity of the return landscape, this handoff forms a highly effective exploration heuristic.

Practically, ISGE alternates between a Judger scoring trajectory scarcity and an Explorer using dual-condition guidance to generate high-return, high-scarcity "gift samples". Extensive experiments on the industrial AuctionNet [9] benchmark demonstrate that ISGE effectively bridges the data quality gap, surpassing the Full-Dataset baseline *within just three iterations* from narrowly supported cold-start data. To summarize, our main contributions are threefold:

1. We theoretically attribute return-guided self-locking to SNR collapse at the suboptimal boundary, and establish a guidance handoff mechanism where scarcity subsequently takes over to elevate the SNR and break the bottleneck.

2. We propose ISGE, an iterative framework that leverages return and scarcity guidance to generate "gift samples" beyond the training support.

3. We demonstrate that ISGE bootstraps near-optimal policies from cold-start datasets, surpassing the Full-Dataset baseline on the industrial AuctionNet benchmark within three iterations.

2 Preliminaries

Generative Auto-Bidding Auto-bidding is a sequential decision problem with long horizons, sparse rewards, and non-Markovian dynamics. Following standard settings (details in Appendix A), an agent allocates a global budget to maximize total return while satisfying strict KPI constraints over T steps. Diffusion-based AIGB [4] formulates this as conditional trajectory generation $p(\tau|y(\tau))$, where $\tau = (s_0, \dots, s_T)$ is the state trajectory and $y(\tau)$ is the target return (KPI constraints are typically absorbed into the return via reward shaping). This conditional distribution acts as the *policy*, implicitly determining a sequence of actions via an inverse dynamics model $a_t = \phi(s_t, s_{t+1})$. The planner effectively satisfies bidding objectives by generating state trajectories τ consistent with the specified condition y.

Conditional Diffusion Models and Classifier-Free Guidance Conditional diffusion models transform decision-making to conditional trajectory generation. Letting $\boldsymbol{x}_0 = \tau$ denote the clean trajectory, the forward process gradually corrupts it into pure noise $\boldsymbol{x}_T \sim \mathcal{N}(0, \mathbf{I})$. The reverse process learns to iteratively denoise $\boldsymbol{x}_t$ by predicting the added noise $\epsilon_\theta(\boldsymbol{x}_t, y, t)$, conditioned on both the guidance signal y (e.g., target return) and the timestep t. Classifier-Free Guidance (CFG) [10, 11] avoids training an explicit classifier, making it lightweight for decision-making. CFG jointly trains a conditional model $\epsilon_\theta(\boldsymbol{x}_t, y, t)$ and an unconditional counterpart $\epsilon_\theta(\boldsymbol{x}_t, \emptyset, t)$ via random condition dropout. During sampling, the perturbed noise is given by: $\hat{\epsilon} = \epsilon_\theta(\boldsymbol{x}_t, \emptyset, t) + w \cdot (\epsilon_\theta(\boldsymbol{x}_t, y, t) - \epsilon_\theta(\boldsymbol{x}_t, \emptyset, t))$, where w is the guidance scale.

3 Motivation and Theoretical Analysis

In this section, we first introduce a motivating example that empirically reveals the self-locking bottleneck. We then establish a geometric optimization framework to demonstrate how an inherent SNR collapse causes self-locking, and how scarcity breaks self-locking via guidance handoff.

3.1 Motivating Example: Self-Locking in Trajectory Optimization

Generative auto-bidding can be viewed as a high-dimensional trajectory optimization problem. However, in real-world auction environments, the optimal bidding strategy is unknown and the return landscape is highly complex, strictly limiting rigorous theoretical analysis. To isolate the underlying optimization dynamics, we construct a tractable high-dimensional ($D = 768$) trajectory optimization task: $R(\boldsymbol{x}) = e^{-\beta\|\boldsymbol{x}-\boldsymbol{x}^*\|_2}$ ($\beta > 0$), where the global optimum $\boldsymbol{x}^*$ lies strictly outside the narrow training support. As illustrated in Figure 1 (left), Vanilla Self-Training exhibits a distinct dual-phase dynamic. While it initially extrapolates beyond the cold-start initialization, performance abruptly stagnates at a suboptimal boundary by Iteration 2. Crucially, a Full-Data Oracle with an identical architecture readily approaches the optimum, demonstrating that this stagnation is not a model capacity limitation but a data bottleneck. We term this ubiquitous failure mode *self-locking*, motivating our systematic theoretical analysis.

3.2 How Does the Diffusion Planner Optimize Trajectories?

To uncover the origins of self-locking, we cast the CFG-guided reverse denoising process as formal trajectory optimization. By leveraging the score-matching property, CFG effectively injects an implicit likelihood gradient $\nabla_{\boldsymbol{x}_t} \log p(y|\boldsymbol{x}_t)$ into the standard diffusion update. This reformulates the sampling step into a biased Langevin dynamic (see Appendix B.1 for detailed derivations):

$$\boldsymbol{x}_{t-1} = \mu_\theta(\boldsymbol{x}_t) + \underbrace{\lambda_t w \cdot \nabla_{\boldsymbol{x}_t} \log p(y|\boldsymbol{x}_t)}_{\text{Guidance Drift}} + \underbrace{\sigma_t \mathbf{z}}_{\text{Noise}}, \tag{1}$$

where $\mu_\theta(\boldsymbol{x}_t)$ is the unconditional mean that pulls the trajectory back toward the known data manifold, $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ provides isotropic sampling stochasticity, and the schedule-dependent coefficient $\lambda_t = (1 - \alpha_t)/\sqrt{\alpha_t}$ acts as the gradient step size. Equation (1) explicitly isolates the two interacting forces driving the planner's initial extrapolation: the isotropic noise ($\mathbf{z}$) provides the necessary stochastic capacity to venture beyond the narrow training support, while the guidance drift steers the optimization toward high-return regions.

3.3 Why Does Return-Guided Optimization Self-Lock?

As iteratively generated trajectories approach the suboptimal boundary of the training support, the delicate interplay between guidance drift and isotropic noise breaks down. We characterize this failure as an inherent SNR collapse. To formalize this, consider the flattened trajectory vector $\boldsymbol{x} \in \mathbb{R}^D$ with $D \gg 1$, and adopt an analytically tractable spherical return landscape: $R(\boldsymbol{x}) = h(d(\boldsymbol{x}))$, where $d(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{x}^*\|_2$ is the Euclidean distance to the OOD global optimum $\boldsymbol{x}^*$, and h is a decreasing function (as instantiated by $h(d) = e^{-\beta d}$ in the motivating example). This abstraction yields two crucial analytical advantages: (i) the optimal improvement direction $\mathbf{n}^* = (\boldsymbol{x}^* - \boldsymbol{x})/d(\boldsymbol{x})$ is explicitly defined as the inward radial vector, and (ii) the scalar distance $d(\boldsymbol{x})$ serves as a direct proxy for optimization progress. As detailed in Appendix B.2.3, this spherical landscape establishes a *conservative lower bound* for optimization difficulty in general smooth landscapes. Under this setting, we now demonstrate the fundamental SNR barrier.

Theorem 3.1 (SNR Barrier in High-Dimensional Optimization)**.** *Consider a trajectory $\boldsymbol{x}$ at distance $d = \|\boldsymbol{x} - \boldsymbol{x}^*\|_2$ from the optimum $\boldsymbol{x}^*$, and an update $\Delta\boldsymbol{x} = \mathbf{v}_{\text{det}} + \mathbf{v}_{\text{noise}}$, where $\mathbf{v}_{\text{det}}$ is a deterministic drift and $\mathbf{v}_{\text{noise}} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$ is isotropic noise. Let $v_{\text{rad}} = \langle \mathbf{v}_{\text{det}}, \mathbf{n}^* \rangle$ be the radial component of the drift. Then, the expected change in the squared distance satisfies the exact algebraic equality:*

$$\mathbb{E}[\Delta(d^2)] = \underbrace{-2dv_{\text{rad}}}_{\textit{Signal}} + \underbrace{\|\mathbf{v}_{\text{det}}\|^2 + D\sigma^2}_{\textit{Penalty}}. \tag{2}$$

Consequently, we define the geometric Signal-to-Noise Ratio (SNR) as the ratio of the radial signal to the total displacement penalty:

$$\textit{SNR} = \frac{2dv_{\text{rad}}}{\|\mathbf{v}_{\text{det}}\|^2 + D\sigma^2}. \tag{3}$$

Distance reduction ($\mathbb{E}[\Delta(d^2)] < 0$) is strictly achievable if and only if SNR > 1.

Theorem 3.1 (proof in Appendix B.2.1) reveals a fundamental trade-off: optimization succeeds only when the radial signal strongly dominates the curvature-induced penalty (SNR > 1). *Since the signal magnitude v_{rad} typically does not scale with dimension D, the D-dependent noise penalty effortlessly overwhelms the signal as d decreases.* The following proposition formally translates this geometric barrier to the diffusion planner, demonstrating why return-guided optimization inevitably self-locks in OOD regions.

Proposition 3.2 (Return Signal Collapse)**.** *Consider a trajectory $\boldsymbol{x}$ whose unconditional mean $\mu_\theta(\boldsymbol{x})$ is at a distance d from the OOD optimum $\boldsymbol{x}^*$. Let $\Delta\boldsymbol{x}_R = \lambda w_R \mathbf{g}_R + \sigma\mathbf{z}$ denote the return-guided step. Assuming the guidance scale w_R is chosen optimally to maximize the reduction of squared distance, the return-guided optimization stagnates ($\mathbb{E}[\Delta(d^2)]_R \geq 0$) when the radial signal falls below the threshold:*

$$\langle \mathbf{g}_R, \mathbf{n}^* \rangle \leq \frac{\sigma\sqrt{D}\,\|\mathbf{g}_R\|}{d} \quad (\textit{i.e., SNR} \leq 1). \tag{4}$$

Consequently, there exists a minimal achievable radius $d_{\min} > 0$ such that further reduction of d is statistically impossible under a vanishing radial signal outside the suboptimal training support.

Proposition 3.2 (proof in Appendix B.2.2) characterizes the mechanism of self-locking. As the planner extrapolates beyond the known training manifold, the implicit return gradient $\mathbf{g}_R$ becomes increasingly sparse and misaligned. Once its radial component $\langle \mathbf{g}_R, \mathbf{n}^* \rangle$ decays below the threshold required to combat the high-dimensional displacement penalty, the expected geometric progress halts. The diffusion planner is physically locked at the suboptimal boundary, incapable of sustaining further OOD exploration.

3.4 Breaking Self-Locking via Scarcity Guidance Handoff

To break this self-locking, the diffusion planner requires a guidance *handoff*: a transition from the collapsed return signal to a robust auxiliary signal that persists in the OOD vacuum. Crucially, since the training data is confined to a suboptimal region, advancing toward the OOD optimum inherently requires a stable outward impulse moving beyond the known data manifold. We identify scarcity as the ideal geometric signal to execute this handoff.

Definition 3.3 (Scarcity via Optimal DAE)**.** Consider a denoising autoencoder (DAE) mapping $r : \mathbb{R}^D \to \mathbb{R}^D$ trained to minimize the expected squared reconstruction error under isotropic Gaussian corruption $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \eta^2\mathbf{I})$. Let $r^*_\eta(\cdot)$ denote the theoretically optimal reconstruction function. We define the *scarcity score* $S(\boldsymbol{x})$ of any state $\boldsymbol{x}$ as its expected reconstruction error:

$$S(\boldsymbol{x}) := \mathbb{E}_{\boldsymbol{\epsilon}}\left[\|\boldsymbol{x} - r^*_\eta(\boldsymbol{x} + \boldsymbol{\epsilon})\|^2\right]. \tag{5}$$

Intuitively, a DAE denoises a sample by pulling it back toward the high-density data manifold (a direction mathematically aligned with the score function $\nabla \log p_{\text{data}}$ [8]). In this sense, $S(\boldsymbol{x})$ naturally acts as a proxy for the geometric distance to the training data: it remains small within the data distribution and strictly increases as the trajectory ventures outward. Therefore, its gradient $\nabla S(\boldsymbol{x})$ inherently points away from the known data, providing the outward repulsion field needed to escape the suboptimal boundary. Formally, we present the following properties.

Proposition 3.4 (Properties of the Scarcity Gradient)**.** *Let $\mathcal{M}$ be the suboptimal training manifold. Following the theory of optimal DAEs, the explicit scarcity gradient $\mathbf{v}_S(\boldsymbol{x}) := \nabla_{\boldsymbol{x}} S(\boldsymbol{x})$ satisfies two critical geometric properties in the local OOD neighborhood $\mathcal{N}(\mathcal{M}) \setminus \mathcal{M}$:*

1. ***Normal Alignment:*** *$\mathbf{v}_S(\boldsymbol{x})$ is consistently aligned with the outward-pointing normal vector $\mathbf{n}_\perp$ of the manifold $\mathcal{M}$.*
2. ***Non-Vanishing Magnitude:*** *There exists a constant $\delta > 0$ such that $\|\mathbf{v}_S(\boldsymbol{x})\| \geq \delta$, ensuring the signal does not collapse at the boundary.*

Furthermore, by integrating this signal via CFG, the implicit scarcity gradient $\mathbf{g}_S(\boldsymbol{x}) := \nabla_{\boldsymbol{x}} \log p(y_S|\boldsymbol{x})$ preserves these properties, acting as a non-vanishing outward impulse.

Proposition 3.4 (proof in Appendix B.3) formally justifies that the implicit scarcity gradient $\mathbf{g}_S$ reliably provides a non-vanishing, outward-pointing signal. This naturally raises the question: *does an outward impulse facilitate optimization?* While moving outward is not a universal guarantee of improvement in arbitrary landscapes, it serves as a geometrically sound heuristic under the self-locking condition. At this self-locked frontier, all improvement directions within the known suboptimal support are exhausted. The cone of improvement directions towards the OOD optimum $\boldsymbol{x}^*$ necessarily extends outward beyond the frontier. Consequently, the outward-pointing scarcity gradient $\mathbf{g}_S$ exhibits a high likelihood of aligning with this cone, naturally yielding a positive inner product ($\langle \mathbf{g}_S, \mathbf{n}^* \rangle > 0$). We validate this positive alignment via dedicated sanity checks in Section 5.1. Assuming this alignment holds, we now demonstrate how $\mathbf{g}_S$ takes over the optimization process.

Theorem 3.5 (Optimization Handoff via Scarcity)**.** *Consider the trajectory $\boldsymbol{x}$ from Proposition 3.2 at the critical self-locking boundary, where $\mathbb{E}[\Delta(d^2)]_R = 0$. Let $\Delta\boldsymbol{x}_{dual} = \lambda(w_R \mathbf{g}_R + w_S \mathbf{g}_S) + \sigma \mathbf{z}$ denote the dual-guided exploration step. Assuming the scarcity gradient aligns positively with the optimal direction ($\langle \mathbf{g}_S, \mathbf{n}^* \rangle > 0$, with full assumptions detailed in Appendix B.4), the optimization handoff is successfully executed ($\mathbb{E}[\Delta(d^2)]_{dual} < 0$) if the scarcity guidance scale w_S satisfies:*

$$w_S \in \left(0, \frac{2d\langle \mathbf{g}_S, \mathbf{n}^* \rangle}{\lambda \|\mathbf{g}_S\|^2}\right). \tag{6}$$

Furthermore, by choosing the scarcity scale $w_S^* = d\langle \mathbf{g}_S, \mathbf{n}^* \rangle / \lambda \|\mathbf{g}_S\|^2$, the marginal SNR of the scarcity guidance is exactly 2 ($SNR_\Delta(w_S^*) = 2$), maximizing the lower bound on optimization progress.

Theorem 3.5 (proof in Appendix B.4) formally guarantees that by injecting scarcity guidance, the planner breaks self-locking, uncovers trajectories with both high return and high scarcity. We denote such trajectories as *gift samples*. Appending these gift samples to the training dataset permanently restores the missing radial component of the return gradient (Appendix B.5), directly motivating the iterative framework.

Discussion. Our theoretical framework relies on two primary abstractions. (i) We adopt a tractable spherical return landscape. Intuitively, any smooth optimum admits a quadratic expansion; this spherical abstraction establishes a conservative lower bound for analyzing general smooth landscapes (Appendix B.2.3). (ii) We assume Frontier Alignment ($\langle \mathbf{g}_S, \mathbf{n}^* \rangle > 0$). Intuitively, since a self-locked policy has exhausted in-distribution improvements, the true optimal direction must inherently point outward, which we empirically validate in Section 5.1.

4 Iterative Scarcity-Guided Exploration

Building upon our theoretical insights, we present Iterative Scarcity-Guided Exploration (ISGE). As illustrated in Figure 2, ISGE breaks self-locking through a dynamic "Generate-Judge-Explore-Update" cycle, advancing the policy π_k to π_{k+1} by aggregating novel exploratory trajectories with conservative exploitation data. In contrast, vanilla self-training naively executes a "Generate-Update" loop by omitting the Judge (green panel) and Explore (red panel) modules.

Scarcity Evaluation via Reconstruction. To operationalize Definition 3.3, we introduce a Scarcity Judger $\mathcal{J}_\phi$ (an unconditional diffusion model) trained on the current exploitation data $\mathcal{D}_{gen}$. It estimates the scarcity of a trajectory τ via the empirical reconstruction residual: $S(\tau) = \|\text{Mask}(\tau -$

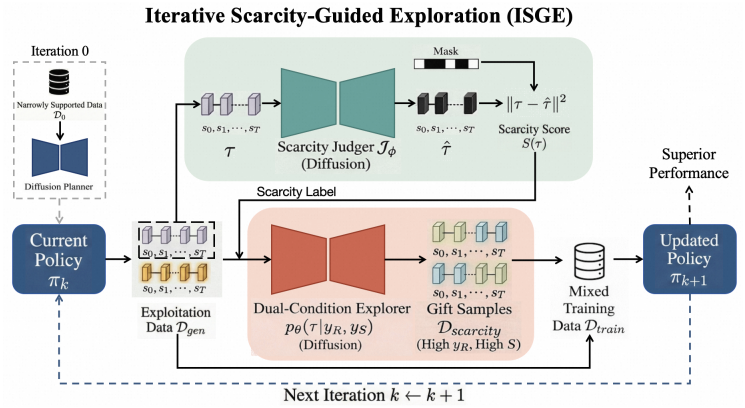


Figure 2: ISGE Framework

$\hat{\tau})\|^2$, where $\hat{\tau}$ is the Judger's reconstruction (see implementation details in Appendix C.2). Crucially, the Mask$(\cdot)$ operation restricts evaluation strictly to strategic dimensions (e.g., historical bids) while ignoring uncontrollable environmental stochasticity (e.g., traffic fluctuations). This isolates true strategic novelty from mere environmental randomness [12].

Scarcity-Guided Diffusion for Exploration. To discover high-return OOD "gift samples", we propose a *Dual-Condition Diffusion* model acting as our Explorer, parameterized as $p_\theta(\tau|y_R, y_S)$. During sampling, we employ disentangled classifier-free guidance to execute the optimization handoff:

$$\hat{\epsilon} = \epsilon_\theta(\tau_t, \emptyset, t) + w_R \underbrace{\big(\epsilon_\theta(\tau_t, y_R, t) - \epsilon_\theta(\tau_t, \emptyset, t)\big)}_{\propto \mathbf{g}_R \text{ (Return Signal)}} + w_S \underbrace{\big(\epsilon_\theta(\tau_t, y_S, t) - \epsilon_\theta(\tau_t, \emptyset, t)\big)}_{\propto \mathbf{g}_S \text{ (Scarcity Signal)}}, \tag{7}$$

where w_R and w_S control the respective guidance scales. The return term ($\mathbf{g}_R$) drives the trajectory to the suboptimal support boundary. At this point, the scarcity term ($\mathbf{g}_S$) takes over, injecting a robust outward impulse that steers the trajectory through the cone of improvement directions into unexplored, high-value regions.

Iterative Training and Dynamic Recalibration. The overall framework ties these components together into a self-improving loop. At iteration k, we sample exploitation data $\mathcal{D}_{gen}$ from π_k. The Judger learns this current manifold density and scores $\mathcal{D}_{gen}$. Next, the Explorer is trained on these labeled samples and then deployed to generate a novel exploration batch $\mathcal{D}_{scarcity}$ conditioned on high return and high scarcity targets (e.g., the 99th percentile). Finally, the updated policy π_{k+1} is trained on $\mathcal{D}_{gen} \cup \mathcal{D}_{scarcity}$. Each iteration dynamically recalibrates by specifically seeking trajectories that are scarce relative to the *current* policy, ensuring efficient expansion without redundant exploration.

5 Experiments

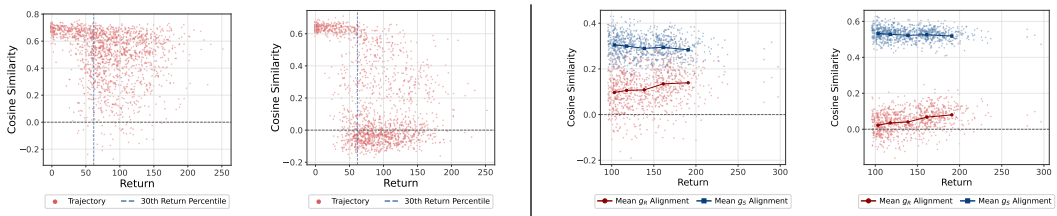
In this section, we evaluate ISGE on AuctionNet [9], a large-scale industrial auto-bidding benchmark that served as the official environment for the NeurIPS 2024 Auto-bidding Competition [13] and remains widely adopted across the auto-bidding community [14–19]. To simulate the severe cold-start challenge, we construct cold-start datasets by retaining only trajectories within the 10% to 30% return percentile range. We evaluate our framework across two distinct settings: AuctionNet-Dense (with strict CPA constraints) and AuctionNet-Sparse (with sparse reward signals). We aim to answer the following research questions: (1) Do the empirical dynamics in real-world auctions align with our theoretical predictions of return signal collapse and optimization handoff? (2) Can ISGE bootstrap superior policies from low-quality, narrowly supported data? (3) Is ISGE efficient and safe for practical auto-bidding? (4) How do the specific design choices contribute to the overall success?

5.1 Sanity Check: Empirical Validation of Theoretical Abstractions

To bridge our idealized geometric abstractions with the complex, non-convex AuctionNet environment, we conduct empirical sanity checks to validate the two core theoretical predictions: the return signal

Table 1: **Main Results on AuctionNet.** Results on the AuctionNet-Dense and AuctionNet-Sparse are reported as mean ± std over 7 evaluation episodes. Metrics include Score, Reward, ER (CPA exceedance ratio), and WR (win rate). Improvements are reported relative to Cold-Start Base based on the Score. Results that outperform the Full-Data Oracle are highlighted in blue, and the best results are highlighted in bold. Supplementary baselines [20, 14, 15, 21, 5, 16] are provided in Table 6.

Dataset	Metrics	Reference		Vanilla Self-Training			ISGE (Ours)		
		Full-Data Oracle	Cold-Start Base	Iter1	Iter2	Iter3	Iter1	Iter2	Iter3
Dense	Score ↑	102.0 ± 3.5	65.3 ± 3.2	74.9 ± 3.4	77.5 ± 3.1	77.8 ± 3.9	103.4 ± 5.3	104.1 ± 6.7	$\mathbf{108.4 \pm 8.2}$
	Reward ↑	208.3 ± 4.4	181.6 ± 2.8	192.5 ± 2.8	196.7 ± 4.0	198.4 ± 4.8	219.1 ± 4.6	220.3 ± 6.1	$\mathbf{223.1 \pm 7.0}$
	ER ↓	$74.9 \pm 6.8\%$	$97.4 \pm 3.5\%$	$85.2 \pm 3.2\%$	$81.7 \pm 4.1\%$	$79.8 \pm 4.7\%$	$58.8 \pm 3.6\%$	$58.3 \pm 4.8\%$	$54.3 \pm 5.3\%$
	WR ↑	$5.8 \pm 0.2\%$	$5.4 \pm 0.2\%$	$5.6 \pm 0.1\%$	$5.7 \pm 0.2\%$	$5.7 \pm 0.2\%$	$6.1 \pm 0.1\%$	$6.1 \pm 0.1\%$	$\mathbf{6.2 \pm 0.1}\%$
	Improve ↑	56.2%	–	14.7%	18.7%	19.1%	58.3%	59.4%	**66.0%**
Sparse	Score ↑	20.5 ± 1.1	15.7 ± 1.0	16.9 ± 0.5	16.6 ± 0.6	16.0 ± 0.6	20.0 ± 0.9	20.6 ± 0.7	$\mathbf{21.1 \pm 1.0}$
	Reward ↑	27.6 ± 0.8	26.1 ± 0.9	27.2 ± 0.5	27.1 ± 0.5	26.6 ± 0.5	29.6 ± 0.6	30.1 ± 0.6	$\mathbf{30.3 \pm 0.8}$
	ER ↓	$10.8 \pm 2.9\%$	$41.4 \pm 4.6\%$	$35.3 \pm 3.3\%$	$35.5 \pm 2.5\%$	$38.5 \pm 2.2\%$	$23.7 \pm 3.3\%$	$21.1 \pm 3.2\%$	$20.9 \pm 4.3\%$
	WR ↑	$7.4 \pm 0.1\%$	$7.8 \pm 0.2\%$	$8.1 \pm 0.1\%$	$8.2 \pm 0.1\%$	$8.2 \pm 0.1\%$	$8.5 \pm 0.1\%$	$8.5 \pm 0.1\%$	$\mathbf{8.6 \pm 0.1}\%$
	Improve ↑	30.6%	–	7.6%	5.7%	1.9%	27.4%	31.2%	**34.4%**



(a) Step $t = 15$ (b) Step $t = 5$ (Collapse) (c) Step $t = 15$ (d) Step $t = 5$ (Handoff)

Figure 3: **Left:** At late denoising stage ($t = 5$), return gradient $\mathbf{g}_R$ maintains alignment within training support but sharply decays toward zero in the high-return OOD region, triggering self-locking. **Right:** Tracking high-return trajectories reveals that when return gradient $\mathbf{g}_R$ (red) collapses at $t = 5$, scarcity gradient $\mathbf{g}_S$ (blue) maintains a positive alignment with the true denoising gradient $\mathbf{g}^*$.

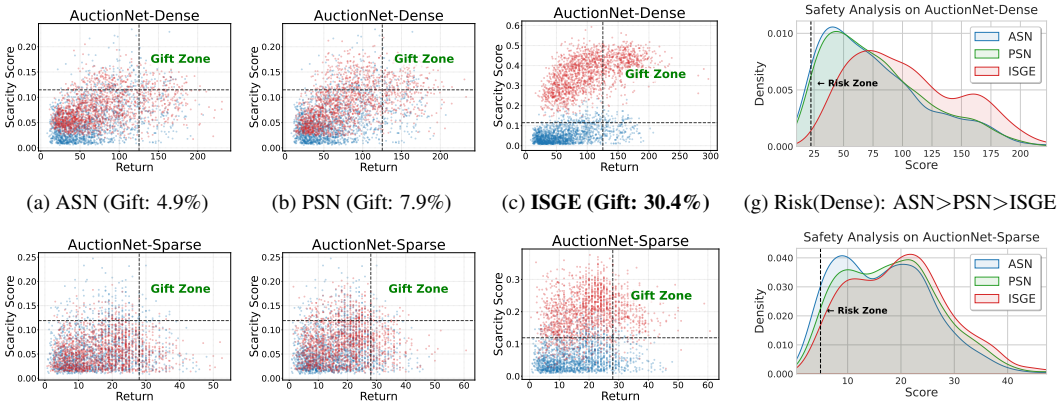
collapse (Proposition 3.2) and the optimization handoff via scarcity (Theorem 3.5). Comprehensive experimental setups and supplementary quantitative results are deferred to Appendix C.5.

Validating Return Signal Collapse (Proposition 3.2). To verify whether the return gradient loses its directional alignment when extrapolating into OOD regions, we evaluate the cosine similarity between the empirical return gradient $\mathbf{g}_R$ (from a cold-start planner trained only on the 10%–30% return quantile) and the empirical optimal direction $\tilde{\mathbf{n}}^*$ (derived from a Full-Data Oracle). As shown in Figure 3 (left), during the fine-grained late denoising stage ($t = 5$, where the total denoising steps run from 20 down to 0), $\cos(\tilde{\mathbf{n}}^*, \mathbf{g}_R)$ remains robust in lower-return regions but drops sharply to 0.11 in the high-return OOD region (>30% return quantile). This perfectly corroborates the theoretical self-locking condition: the return guidance is physically incapable of maintaining optimal directional alignment once it crosses the suboptimal boundary.

Validating Optimization Handoff via Scarcity (Theorem 3.5). To justify our critical assumption that the outward-pointing scarcity gradient aligns positively with the optimal improving direction ($\langle \mathbf{g}_S, \mathbf{n}^* \rangle > 0$), we analyze the top high-return gift samples discovered by the ISGE planner. We measure the alignment of both $\mathbf{g}_R$ and the scarcity gradient $\mathbf{g}_S$ against the true denoising gradient $\mathbf{g}^*$ required for the perfect recovery of these top gift samples. Figure 3 (right) illustrates a clear temporal handoff mechanism. At the late boundary-crossing stage ($t = 5$), the return gradient's alignment collapses ($\cos \approx 0.05$). In contrast, the scarcity gradient maintains a robust positive alignment ($\cos \approx 0.53$). Because $\mathbf{g}_R$ has successfully navigated the trajectory to the high-return frontier in earlier steps, the non-vanishing, outward-pointing $\mathbf{g}_S$ subsequently takes over, providing the necessary directional impulse to break self-locking.

5.2 Main Results: Bootstrapping from Narrow Support

We evaluate ISGE against a Vanilla Self-Training baseline. To rigorously isolate the efficacy of our iterative framework, all methods presented in Table 1 share the identical underlying diffusion planner architecture [4]. Specifically, the Cold-Start Base is trained exclusively on the low-quality cold-start data, serving as the shared starting point for both Vanilla Self-Training and ISGE. The Full-Data



(a) ASN (Gift: 4.9%) (b) PSN (Gift: 7.9%) (c) **ISGE (Gift: 30.4%)** (g) Risk(Dense): ASN>PSN>ISGE

(d) ASN (Gift: 0.1%) (e) PSN (Gift: 1.2%) (f) **ISGE (Gift: 16.2%)** (h) Risk(Sparse): ASN>PSN>ISGE

Figure 4: **Exploration Efficiency and Safety Analysis. Left (a)-(f):** Scarcity–Return landscapes for Dense and Sparse environments. Red and blue points denote exploration and baseline (V1) trajectories, respectively. The *Gift Zone* (top-right) represents high-value strategies exceeding the top 10% thresholds of both return and scarcity computed on the baseline dataset. **Right (g)-(h):** Kernel density estimates of exploration score distributions for ASN (blue), PSN (green), and ISGE (red). The *Risk Zone* indicates performance falling below the 10% quantile of the baseline (V1) distribution. Risk proportion (Dense, Sparse): ASN (4.7%, 6.8%), PSN (3.6%, 4.1%), **ISGE (0.1%, 2.6%)**.

Oracle serves as an empirical reference for the diffusion planner architecture, possessing global visibility into full offline data. Performance is measured by Score (primary metric), Total Reward (total conversions), CPA Exceedance Ratio (ER), and Win Rate (WR). Complete metric definitions, implementation details, and additional comparisons against non-diffusion baselines are provided in Appendix C. Robustness analysis on more iterations and an additional cold-start setting can be found in Appendix C.4.

Table 1 highlights a stark contrast in performance evolution. Vanilla Self-Training succumbs to self-locking: it saturates far below the Full-Data Oracle in the Dense setting (77.8 vs. 102.0) and regresses in the Sparse setting (Iter $1 \rightarrow 3$: $16.9 \rightarrow 16.0$). In contrast, ISGE effectively breaks this bottleneck. It not only outperforms baselines but surpasses the Full-Data Oracle in both settings. Notably, in AuctionNet-Dense, ISGE achieves a Score of 108.4 (vs. Full-Data Oracle 102.0) with significantly better constraint satisfaction (ER 54.3% vs. 74.9%), demonstrating Pareto-optimal improvement. By Iteration 3, ISGE attains relative gains of **+66.0%** (Dense) and **+34.4%** (Sparse). This consistent growth confirms that the guidance handoff mechanism successfully breaks self-locking, continuously repairing the optimization landscape and enabling robust bootstrapping from a severely suboptimal initialization.

5.3 ISGE Achieves Efficient and Safe Exploration

To validate that our gains stem from scarcity guidance rather than simple randomness, we compare ISGE against two standard undirected exploration baselines in auto-bidding: Action Space Noise (ASN) [22] and Parameter Space Noise (PSN) [23] (see Appendix C for details). Using Iteration 1 Vanilla (V1) as the baseline, Table 2 shows that ASN and PSN yield only marginal improvements ($\sim 9\%$ in Dense, $\sim 8\%$ in Sparse). This stagnation supports our hypothesis that in high-dimensional manifolds, isotropic noise is statistically orthogonal to sparse directions of improvement. In contrast, ISGE achieves a transformative leap, improving the Score by 38.1% in Dense and 18.3% in Sparse settings. Crucially, ISGE dominates in

Table 2: **Performance of Directed vs. Undirected Exploration.** Metrics are reported for Iteration 1 relative to Vanilla Self-Training (V1). ISGE significantly outperforms undirected baselines (ASN/PSN) across all metrics.

Dataset	Metrics	Vanilla (V1)	ASN	PSN	**ISGE**
Dense	Score ↑	74.9 ± 3.4	81.7 ± 4.0	81.8 ± 3.8	$\mathbf{103.4 \pm 5.3}$
	Reward ↑	192.5 ± 2.8	200.8 ± 3.9	201.0 ± 3.9	$\mathbf{219.1 \pm 4.6}$
	ER ↓	$85.2 \pm 3.2\%$	$77.7 \pm 4.0\%$	$77.4 \pm 4.1\%$	$\mathbf{58.8 \pm 3.6}\%$
	WR ↑	$5.6 \pm 0.1\%$	$5.8 \pm 0.1\%$	$5.8 \pm 0.1\%$	$\mathbf{6.1 \pm 0.1}\%$
	Improve ↑	–	9.1%	9.2%	**38.1%**
Sparse	Score ↑	16.9 ± 0.5	18.0 ± 0.5	18.2 ± 0.6	$\mathbf{20.0 \pm 0.9}$
	Reward ↑	27.2 ± 0.5	28.1 ± 0.5	28.2 ± 0.5	$\mathbf{29.6 \pm 0.6}$
	ER ↓	$35.3 \pm 3.3\%$	$30.9 \pm 2.9\%$	$30.4 \pm 3.0\%$	$\mathbf{23.7 \pm 3.3}\%$
	WR ↑	$8.1 \pm 0.1\%$	$8.3 \pm 0.1\%$	$8.2 \pm 0.1\%$	$\mathbf{8.5 \pm 0.1}\%$
	Improve ↑	–	6.5%	7.7%	**18.3%**

both reward maximization and constraint satisfaction, successfully navigating the trade-off between aggressive exploration and strict CPA limits.

Mechanism Analysis: Efficiency and Safety. We further investigate this success by visualizing the *Scarcity-Return landscape* of the explored trajectories in Figure 4 (a)-(f). We define the "Gift Zone" (top-right quadrant) as the target region containing high-scarcity, high-return trajectories. As observed in the scatter plots, undirected baselines (ASN/PSN) are inefficient, capturing negligible samples in the Gift Zone (e.g., $\leq 1.2\%$ in Sparse), proving that random perturbations fail to stumble upon high-value strategies in data vacuums. Conversely, ISGE's dual-condition explorer functions as a directed guide, efficiently achieving capture rates of **30.4%** (Dense) and **16.2%** (Sparse). Furthermore, regarding *safety* (downside risk), Figure 4 (g)-(h) reveals that undirected methods exhibit heavy left-tail distributions falling into the "Risk Zone" (up to 6.8% risk). In contrast, ISGE explicitly truncates this downside risk ($\leq 0.1\%$ in Dense, $\leq 2.6\%$ in Sparse) via return conditioning. This demonstrates that ISGE robustly expands the performance upper bound without compromising the reliable lower bound.

Furthermore, to demonstrate the superiority of our trajectory-level scarcity guidance over traditional step-wise intrinsic motivation, we provide additional comparisons against transition-level curiosity-driven exploration (e.g., ICM [12] and PGR [24]) in Appendix C.7.

5.4 Ablation Studies

Effect of Strategic Masking. ISGE calculates scarcity solely on controllable policy states, masking out uncontrollable environmental dimensions. Removing this mask degrades performance and significantly reduces the gift sample capture rate from 30.4% to 19.1% (Table 3a). This decline stems from a fundamental distraction: granular analysis (Table 3b) reveals that environmental features like *Traffic Value* exhibit negligible scarcity growth (i.e., from $\mathcal{D}_{gen}$ to $\mathcal{D}_{scarcity}$) because the agent cannot "invent" market conditions. Consequently, including these immutable dimensions introduces futile noise that distracts the explorer from controllable strategic features (e.g., *Budget Left*), thereby diluting the gradient signal of scarcity for valid exploration.

Table 3(a): **Masking vs. No Masking.**

Metrics	No Masking	**Masking**
Score	94.7 ± 4.5	$\mathbf{103.4 \pm 5.3}$
Gift %	19.1%	$\mathbf{30.4\%}$

Table 3(b): **Feature Scarcity Growth.**

	State Feature	$\mathcal{D}_{gen}$	$\mathcal{D}_{scarcity}$
Strat.	Budget Left	0.056	0.105
	History Bid	0.025	0.486
Env.	Traffic Volume	0.052	0.062
	Traffic Value	0.229	0.235

Hyperparameter Sensitivity & Mechanism Validation. We analyze the impact of guidance scales and data mixture ratios (Table 4) to validate the core design principles of ISGE.

(a) The Necessity of Proper Scarcity Guidance. Table 4(a) reveals that weak scarcity guidance scale ($w_S = 1$) fails to effectively capture gift samples (19.7%), dropping performance to 90.2. This confirms that scarcity identification alone is insufficient; a sufficiently large w_S is required to mechanically "push" the trajectory away from the high-density manifold against the strong attraction of the diffusion prior. This empirically validates our theoretical analysis (Theorem 3.5) that proper guidance scale w_S is necessary for scarcity-guided exploration.

Table 4: **Hyperparameter Analysis.** We study the impact of (a) Guidance Scale balance and (b) Data Mixture Ratios. The default settings are highlighted in gray.

(a) CFG Guidance Balance ($w_R : w_S$)

Setting (w_R, w_S)	Ratio	Exploration Data Quality			Final Policy
		Avg Score	Avg Scarcity	**Gift %**	**Score**
Reward-Heavy $(3, 1)$	3:1	88.6	0.14	19.7%	90.2 ± 3.8
Balanced $(3, 3)$	1:1	102.9	0.36	30.4%	103.4 ± 5.3
Scarcity-Heavy $(3, 5)$	3:5	**105.4**	**0.59**	**32.2%**	$\mathbf{105.5 \pm 6.1}$

(b) Data Mixture Ratio (Explore : Base)

Mixture Ratio	**Score**
Base-Only $(0 : 1)$	74.9 ± 3.4
Conservative $(1 : 2)$	103.3 ± 6.9
Balanced (5 : 4)	$\mathbf{103.4 \pm 5.3}$
Aggressive $(2 : 1)$	101.6 ± 5.3
Explore-Only $(1 : 0)$	96.7 ± 6.0

(b) The Necessity of Data Mixing. Table 4(b) demonstrates that while *Explore-Only* (96.7) significantly outperforms *Base-Only* (74.9), it still lags behind the mixed policy (103.4). This highlights the

complementary nature of the datasets: exploration data fixes policy "blind spots" (high-value/sparse regions) but sacrifices global coverage due to its targeted bias. Consequently, mixing is essential to combine the evolutionary signal of exploration with the structural stability of the base distribution.

6 Related Work

Auto-bidding and Generative Auto-bidding Traditionally, auto-bidding has been formulated as a constrained optimization problem, typically addressed through PID controllers [25, 26] or primal-dual frameworks [27, 28]. To handle dynamic market environments, Reinforcement Learning (RL) methods were subsequently introduced to maximize cumulative returns [29, 30, 3, 31, 32]. However, they struggle with ineffective credit assignment in non-Markovian long-horizon scenarios. Recently, the field has undergone a paradigm shift toward AIGB, which reformulates auto-bidding as a generative sequence modeling task. Pioneered by DiffBid [4], this paradigm utilizes conditional diffusion models to capture the joint distribution of bidding trajectories [6], while subsequent Transformer-based approaches like GAVE [15], GAS [14] and GRAD [16] employ autoregressive planning with return-to-go conditioning. To mitigate the limitations of static offline datasets, recent research has focused on offline improvement mechanisms [5, 33, 34]. However, these methods typically rely on high-quality expert demonstrations or broad data coverage. In contrast, our work targets the iterative cold-start setting, focusing on bootstrapping policies purely from low-quality, narrowly supported data without expert priors.

Decision Making and Exploration with Diffusion Models Diffusion models have emerged as a versatile paradigm for decision-making. Diffuser [35], Decision Diffuser [11], and DiffuserLite [36], directly treat planning as a conditional trajectory generation problem. Alternatively, methods such as Diffusion-QL [37], DTQL [38], and MetaDiffuser [39] leverage diffusion for policy representation. Beyond policy fitting, recent studies investigate diffusion-based data augmentation [40] and exploration. Approaches like PGR [24], ICM [12], and ExDM [41] rely on novelty or uncertainty scores to guide exploration. These methods predominantly operate at the *transition level* $(s_t, a_t \rightarrow s_{t+1})$ for RL policies, utilizing forward prediction error or diffusion noise prediction error as local signals. However, auto-bidding is a strictly non-Markovian problem governed by global constraints: the budget spent at step t limits the feasible actions at step T, thereby transition-level metrics are insufficient for capturing the long-horizon strategic structure. In contrast, ISGE adopts trajectory reconstruction error to perform *trajectory-level* exploration, which inherently aligns with the sequential modeling nature of diffusion planners. (We provide an empirical comparison demonstrating the advantage of our trajectory-level scarcity over transition-level curiosity in Appendix C.7.) Furthermore, regarding the utilization of exploration signals, AdaptDiffuser [42] achieves self-evolution by filtering generated plans via a learned discriminator, but lacks exploration beyond the support of the initial generation. Conversely, ISGE actively injects the scarcity signal as a condition during the diffusion sampling process via CFG. This mechanism allows our explorer to extrapolate beyond the current support, purposefully seeking "gift samples" in high-scarcity regions that filtering methods would rarely encounter.

7 Conclusion

In this paper, we addressed the critical challenge of bootstrapping auto-bidding policies from narrowly supported data. We revealed that diffusion planners suffer from a self-locking bottleneck due to an inherent SNR collapse in OOD regions. To break this limit, we proposed ISGE, a novel iterative framework. Our geometric analysis proves that scarcity provides a robust outward impulse, executing a successful guidance handoff when the return signal fails. By iteratively exploring and aggregating high-return "gift samples", ISGE continuously repairs the optimization landscape. Extensive experiments on AuctionNet demonstrate that ISGE efficiently navigates the OOD vacuum, achieving transformative performance growth beyond cold-start initialization.

While highly effective for bootstrapping, ISGE assumes local landscape continuity, and its outward exploration may become less effective near optimality (see Appendix D for detailed discussions). Furthermore, our framework is currently designed for diffusion planners. Future work will focus on extending ISGE's core philosophy of trajectory-level scarcity guidance to other generative decision-making architectures, such as autoregressive Transformers.

Acknowledgments and Disclosure of Funding

This work is supported by the National Natural Science Foundation of China (Grant No. 62572010), Alibaba Group through the Alibaba Innovative Research Program, and Peking University-Alimama Joint Laboratory of AI Innovation. We thank all anonymous reviewers for their helpful feedback.

References

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

[2] Weinan Zhang, Shuai Yuan, and Jun Wang. Optimal real-time bidding for display advertising. KDD '14, page 1077–1086, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450329569. doi: 10.1145/2623330.2623633. URL https://doi.org/10.1145/2623330.2623633.

[3] Yue He, Xiujun Chen, Di Wu, Junwei Pan, Qing Tan, Chuan Yu, Jian Xu, and Xiaoqiang Zhu. A unified solution to constrained bidding in online display advertising. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 2993–3001, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383325. doi: 10.1145/3447548.3467199. URL https://doi.org/10.1145/3447548.3467199.

[4] Jiayan Guo, Yusen Huo, Zhilin Zhang, Tianyu Wang, Chuan Yu, Jian Xu, Bo Zheng, and Yan Zhang. Generative auto-bidding via conditional diffusion modeling. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 5038–5049, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704901. doi: 10.1145/3637528.3671526. URL https://doi.org/10.1145/3637528.3671526.

[5] Yunshan Peng, Wenzheng Shu, Jiahao Sun, Yanxiang Zeng, Jinan Pang, Wentao Bai, Yunke Bai, Xialong Liu, and Peng Jiang. Expert-guided diffusion planner for auto-bidding. CIKM '25, pages 5963–5970, 2025.

[6] Yewen Li, Jingtong Gao, Nan Jiang, Shuai Mao, Ruyi An, Fei Pan, Xiangyu Zhao, Bo An, Qingpeng Cai, and Peng Jiang. Generative auto-bidding in large-scale competitive auctions via diffusion completer-aligner, 2025. URL https://arxiv.org/abs/2509.03348.

[7] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 1096–1103, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605582054. doi: 10.1145/1390156.1390294. URL https://doi.org/10.1145/1390156.1390294.

[8] Guillaume Alain and Yoshua Bengio. What regularized auto-encoders learn from the data-generating distribution. *J. Mach. Learn. Res.*, 15(1):3563–3593, January 2014. ISSN 1532-4435.

[9] 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 *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2025. Curran Associates Inc. ISBN 9798331314385.

[10] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

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

[12] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR, 2017.

[13] Jian Xu, Zhilin Zhang, Zongqing Lu, Xiaotie Deng, Michael P. Wellman, Chuan Yu, Shuai Dou, Yusen Huo, Zhiwei Xu, Zhijian Duan, Shaopan Xiong, Chuang Liu, Ningyuan Li, Kefan Su, Wei Gong, and Bo Zheng. Auto-bidding in large-scale auctions: Learning decision-making in uncertain and competitive games. In *NeurIPS 2024 Competition Track*, 2024. URL https://openreview.net/forum?id=ZejUjZUF6i.

[14] 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 on Web Conference 2025*, pages 315–324, 2025.

[15] 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. SIGIR '25, page 244–254, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400715921. doi: 10.1145/3726302.3729987. URL https://doi.org/10.1145/3726302.3729987.

[16] Yu Lei, Jiayang Zhao, Yilei Zhao, Zhaoqi Zhang, Linyou Cai, Qianlong Xie, and Xingxing Wang. Generative large-scale pre-trained models for automated ad bidding optimization, 2025. URL https://arxiv.org/abs/2508.02002.

[17] 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*, WWW '26, page 75–86, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400723070. doi: 10.1145/3774904.3792202. URL https://doi.org/10.1145/3774904.3792202.

[18] Yuejia Dou, Hesong Wang, Xinyu Zhang, Tianyu Wang, Zhilin Zhang, Chuan Yu, Jian Xu, Bo Zheng, and Qi Qi. Aigb-r1: Self-evolving generative auto-bidding via hierarchical planner-executor optimization, 2026. URL https://arxiv.org/abs/2607.17281.

[19] Jiaqi Ding, Chuan Yang, Linghui Meng, Shengsheng Niu, Jie He, Zhangang Lin, Ching Law, and Xiaolin Fang. Langbp: Language-guided reasoning and acting for joint bidding and pricing. *arXiv preprint arXiv:2608.30343*, 2026.

[20] 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, 2021. URL https://arxiv.org/abs/2106.01345.

[21] Kaiyuan Li, Pengyu Wang, Yunshan Peng, Pengjia Yuan, Yanxiang Zeng, Rui Xiang, Yanhua Cheng, Xialong Liu, and Peng Jiang. Ebaret: Expert-guided bag reward transformer for auto bidding. In *Companion Proceedings of the ACM on Web Conference 2025*, WWW '25, page 1104–1108, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400713316. doi: 10.1145/3701716.3715512. URL https://doi.org/10.1145/3701716.3715512.

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

[23] Haoming Li, Yusen Huo, Shuai Dou, Zhenzhe Zheng, Zhilin Zhang, Chuan Yu, Jian Xu, and Fan Wu. Trajectory-wise iterative reinforcement learning framework for auto-bidding. In *Proceedings of the ACM Web Conference 2024*, pages 4193–4203, 2024.

[24] Renhao Wang, Kevin Frans, Pieter Abbeel, Sergey Levine, and Alexei A Efros. Prioritized generative replay. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=5IkDAfabuo.

[25] Weinan Zhang, Yifei Rong, Jun Wang, Tianchi Zhu, and Xiaofan Wang. Feedback control of real-time display advertising. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 407–416, 2016.

[26] Xun Yang, Yasong Li, Hao Wang, Di Wu, Qing Tan, Jian Xu, and Kun Gai. Bid optimization by multivariable control in display advertising. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1966–1974, 2019.

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

[28] Gagan Aggarwal, Ashwinkumar Badanidiyuru, Santiago R Balseiro, Kshipra Bhawalkar, Yuan Deng, Zhe Feng, Gagan Goel, Christopher Liaw, Haihao Lu, Mohammad Mahdian, et al. Auto-bidding and auctions in online advertising: A survey. *ACM SIGecom Exchanges*, 22(1): 159–183, 2024.

[29] Han Cai, Kan Ren, Weinan Zhang, Kleanthis Malialis, Jun Wang, Yong Yu, and Defeng Guo. Real-time bidding by reinforcement learning in display advertising. In *Proceedings of the tenth ACM international conference on web search and data mining*, pages 661–670, 2017.

[30] Di Wu, Xiujun Chen, Xun Yang, Hao Wang, Qing Tan, Xiaoxun Zhang, Jian Xu, and Kun Gai. Budget constrained bidding by model-free reinforcement learning in display advertising. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1443–1451, 2018.

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

[32] Weitong Ou, Bo Chen, Xinyi Dai, Weinan Zhang, Weiwen Liu, Ruiming Tang, and Yong Yu. A survey on bid optimization in real-time bidding display advertising. *ACM Trans. Knowl. Discov. Data*, 18(3), December 2023. ISSN 1556-4681. doi: 10.1145/3628603. URL `https://doi.org/10.1145/3628603`.

[33] Zhiyu Mou, Yiqin Lv, Miao Xu, Qi Wang, Yixiu Mao, Qichen Ye, Chao Li, Rongquan Bai, Chuan Yu, Jian Xu, and Bo Zheng. Enhancing generative auto-bidding with offline reward evaluation and policy search, 2025. URL `https://arxiv.org/abs/2509.15927`.

[34] Hao Jiang, Yongxiang Tang, Yanxiang Zeng, Pengjia Yuan, Yanhua Cheng, Teng Sha, Xialong Liu, and Peng Jiang. Optimal return-to-go guided decision transformer for auto-bidding in advertisement. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 1033–1037, 2025.

[35] Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

[36] Zibin Dong, Jianye Hao, Yifu Yuan, Fei Ni, Yitian Wang, Pengyi Li, and Yan Zheng. Diffuserlite: Towards real-time diffusion planning. *Advances in Neural Information Processing Systems*, 37: 122556–122583, 2024.

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

[38] Tianyu Chen, Zhendong Wang, and Mingyuan Zhou. Diffusion policies creating a trust region for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 37: 50098–50125, 2024.

[39] Fei Ni, Jianye Hao, Yao Mu, Yifu Yuan, Yan Zheng, Bin Wang, and Zhixuan Liang. Metadiffuser: Diffusion model as conditional planner for offline meta-rl, 2023. URL `https://arxiv.org/abs/2305.19923`.

[40] Cong Lu, Philip Ball, Yee Whye Teh, and Jack Parker-Holder. Synthetic experience replay. *Advances in Neural Information Processing Systems*, 36:46323–46344, 2023.

[41] Chengyang Ying, Huayu Chen, Xinning Zhou, Zhongkai Hao, Hang Su, and Jun Zhu. Exploratory diffusion model for unsupervised reinforcement learning. *arXiv preprint arXiv:2502.07279*, 2025.

[42] Zhixuan Liang, Yao Mu, Mingyu Ding, Fei Ni, Masayoshi Tomizuka, and Ping Luo. Adaptdiffuser: Diffusion models as adaptive self-evolving planners. In *International Conference on Machine Learning*, 2023.

[43] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021. URL https://arxiv.org/abs/2011.13456.

[44] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL https://arxiv.org/abs/2006.11239.

[45] Herbert Federer. Curvature measures. *Transactions of the American Mathematical Society*, 93 (3):418–491, 1959.

[46] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J. Andrew Bagnell, Pieter Abbeel, and Jan Peters. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1–2):1–179, March 2018. ISSN 1935-8261. doi: 10.1561/2300000053. URL http://dx.doi.org/10.1561/2300000053.

[47] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration, 2019. URL https://arxiv.org/abs/1812.02900.

[48] Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

[49] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

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

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

[52] Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. *Advances in neural information processing systems*, 34:28954–28967, 2021.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state our contributions, and the claims regarding self-locking and guidance handoff are fully supported by theoretical proofs (Section 3) and empirical evaluations (Section 5).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All theoretical results, including the geometric assumptions for the optimal DAE and the optimization handoff, are formally stated in Section 3 and rigorously proven in Appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide detailed descriptions of the experimental setup, network architectures, and evaluation metrics in Section 5 and the Appendix to ensure full reproducibility.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We submit the code in the supplementary material. Upon acceptance, we will make the code publicly available.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All training details, hyperparameters, and dataset construction methods (e.g., AuctionNet-Dense/Sparse splits) are comprehensively specified in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report the performance metrics aggregated over multiple runs and trajectories, providing a robust evaluation of the proposed framework.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The hardware specifications (e.g., GPU types) and estimated compute times for training the diffusion planners are detailed in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in this paper conforms entirely to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the potential societal and economic impacts in Appendix E.

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The proposed auto-bidding diffusion models operate within specific market simulations and do not pose a high risk for malicious dual-use or societal harm.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We properly cite the AuctionNet benchmark and respect its original licensing and terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper does not introduce new datasets or large-scale assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were only used for writing assistance and editing, and do not form a core, non-standard methodological component of this research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.

Contents

1 Introduction **1**

2 Preliminaries **2**

3 Motivation and Theoretical Analysis **3**

3.1 Motivating Example: Self-Locking in Trajectory Optimization 3

3.2 How Does the Diffusion Planner Optimize Trajectories? 3

3.3 Why Does Return-Guided Optimization Self-Lock? 3

3.4 Breaking Self-Locking via Scarcity Guidance Handoff 4

4 Iterative Scarcity-Guided Exploration **5**

5 Experiments **6**

5.1 Sanity Check: Empirical Validation of Theoretical Abstractions 6

5.2 Main Results: Bootstrapping from Narrow Support 7

5.3 ISGE Achieves Efficient and Safe Exploration 8

5.4 Ablation Studies 9

6 Related Work **10**

7 Conclusion **10**

A Detailed Formulation of Auto-Bidding and MDP Modeling **24**

A.1 Problem Formulation 24

A.2 Auto-Bidding as Decision-Making 24

B Missing Proofs **25**

B.1 Derivation of the CFG Biased Langevin Update 25

B.2 Proof of Geometric Barrier and Dynamics Failure (Section 3.3) 26

B.3 Analysis for Proposition 3.4 28

B.4 Proof of Theorem 3.5 29

B.5 Restoring the Return Gradient via Fine-Tuning on Gift Samples 30

C Experimental Details and Additional Results **32**

C.1 Setup 32

C.2 Implementation Details of ISGE 35

C.3 Computational Cost 35

C.4 Robustness Analysis 36

C.5 Experimental Setup and Extended Results for Sanity Checks 38

C.6 Implementation Details of Undirected Exploration Baselines 39

C.7 Comparison with Transition-Level Curiosity Exploration 40

D Limitations and Future Work **41**

E Broader Impacts **41**

A Detailed Formulation of Auto-Bidding and MDP Modeling

Following DiffBid [4], we provide a comprehensive formalization of the auto-bidding problem and its translation into a sequential decision-making process, following established settings in the literature.

A.1 Problem Formulation

During a specific advertising period, suppose there are N impression opportunities arriving sequentially, indexed by i. Advertisers submit bids to compete for each impression opportunity. An advertiser wins the impression if its bid b_i surpasses all others, subsequently incurring a cost c_i for winning and obtaining the corresponding value.

The fundamental 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 $o_i \in \{0, 1\}$ is a binary variable indicating whether the advertiser wins impression i. Furthermore, the ad delivery is subject to budget and various performance constraints. The budget constraint is simply formulated as $\sum_i o_i c_i \leq B$, where B is the total budget. To unify other complex key performance indicator (KPI) constraints, we adopt the generalized formulation $\frac{\sum_i c_{ij} o_i}{\sum_i p_{ij} o_i} \leq C_j$, where C_j is the upper bound of the j-th constraint, p_{ij} represents any performance indicator (e.g., conversions), and c_{ij} represents the cost incurred for impression i under the j-th constraint (e.g., the cost used to compute the CPA for that constraint).

Given J constraints, the Multi-constrained Bidding (MCB) problem is formally defined as:

$$\begin{aligned} \underset{o_i}{\text{maximize}} \quad & \sum_{i=1}^{N} o_i v_i \\ \text{s.t.} \quad & \sum_{i=1}^{N} o_i c_i \leq B \\ & \frac{\sum_{i=1}^{N} c_{ij} o_i}{\sum_{i=1}^{N} p_{ij} o_i} \leq C_j, \quad \forall j \in \{1, \ldots, J\} \\ & o_i \in \{0, 1\}, \quad \forall i \end{aligned} \tag{8}$$

Previous studies [3] have shown that the optimal solution to this constrained optimization takes the form:

$$b_i^* = \lambda_0 v_i + C_j \sum_{j=1}^{J} \lambda_j p_{ij}, \tag{9}$$

where b_i^* is the predicted optimal bid for impression i, and λ_j ($j \in \{0, \ldots, J\}$) are the optimal bidding parameters. For instance, optimizing with only the budget constraint corresponds to Max Return bidding, whereas incorporating the CPA constraint forms Target-CPA bidding.

A.2 Auto-Bidding as Decision-Making

While Eq. (9) provides the theoretical formulation of the optimal bid, the highly random and complex nature of real-world advertising environments prevents the direct calculation of the optimal bidding parameters λ_j. Instead, these parameters must be carefully calibrated to adapt to the environment and dynamically adjusted as it evolves. This dynamic adjustment inherently transforms auto-bidding into a sequential decision-making problem.

To model this process, we define the key components of the automated bidding agent interacting with the industrial online advertising system:

- **State** $\mathbf{s}_t \in \mathcal{S}$: Describes the real-time advertising status at time period t, which encapsulates features such as: 1) the remaining time of the advertiser; 2) the remaining budget; 3) the budget spend speed; 4) real-time cost-efficiency; and 5) average cost-efficiency.
- **Action** $\mathbf{a}_t \in \mathcal{A}$: Indicates the adjustment made to the bidding parameters at time period t. The action space has the dimension corresponding to the number of bidding parameters λ_j, modeled as a vector $(a_t^{\lambda_0}, \ldots, a_t^{\lambda_J})$.
- **Reward** $\mathbf{r}_t \in \mathcal{R}$: The specific value contributed to the overall objective obtained within the time period t.

- **Trajectory** τ: Represents the complete sequence of states, actions, and rewards within an episode. Note that when formulating this as a generative modeling problem in subsequent sections, we denote this trajectory sequence as $\boldsymbol{x}$ to align with standard diffusion formulation conventions.

In real-world online advertising systems, learning a policy through direct trial-and-error interaction with the live environment is unfeasible due to severe financial and safety concerns. Consequently, policies must be learned entirely from offline historical bidding logs, which contain trajectories τ generated by a variety of behavior strategies. This offline sequential decision-making formulation perfectly aligns with our generative paradigm, where we explicitly model and optimize the generation of these high-return trajectories (re-denoted as $\boldsymbol{x}$) via diffusion planners.

B Missing Proofs

B.1 Derivation of the CFG Biased Langevin Update

Let $\epsilon_\theta(\boldsymbol{x}_t)$ denote the unconditional noise prediction. By leveraging the score matching property, we first reformulate the Classifier-Free Guidance (CFG) modified noise prediction $\hat{\epsilon}_\theta(\boldsymbol{x}_t)$ into a score-based gradient. Substituting this into the standard DDPM update reveals that the sampling process at step t manifests as a biased Langevin update:

$$\boldsymbol{x}_{t-1} \approx \mu_\theta(\boldsymbol{x}_t) + \lambda_t w \cdot \nabla_{\boldsymbol{x}_t} \log p(y \mid \boldsymbol{x}_t) + \sigma_t \mathbf{z}, \tag{10}$$

where $\lambda_t = \frac{1-\alpha_t}{\sqrt{\alpha_t}}$ is a positive coefficient derived from the noise schedule. **Under the idealized assumption of perfect density estimation (commonly adopted in theoretical analysis of diffusion models), the approximation becomes exact.**

Step 1: Reformulating CFG via Score Matching. Recall the definition of CFG. The modified noise prediction is a linear interpolation between the unconditional prediction $\epsilon_\theta(\boldsymbol{x}_t)$ and the conditional prediction $\epsilon_\theta(\boldsymbol{x}_t, y)$:

$$\hat{\epsilon}_\theta(\boldsymbol{x}_t) = \epsilon_\theta(\boldsymbol{x}_t) + w \cdot \big(\epsilon_\theta(\boldsymbol{x}_t, y) - \epsilon_\theta(\boldsymbol{x}_t)\big). \tag{11}$$

According to the denoising score matching objective [43], the optimal noise prediction network approximates the score of the data distribution (scaled by the noise level):

$$\epsilon_\theta(\boldsymbol{x}_t) \approx -\sqrt{1-\bar{\alpha}_t}\, \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t), \tag{12}$$

$$\epsilon_\theta(\boldsymbol{x}_t, y) \approx -\sqrt{1-\bar{\alpha}_t}\, \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t \mid y). \tag{13}$$

Note on the approximation ($\approx$): These relationships are approximate due to the finite capacity and training imperfections of the neural network; they become exact only in the limit of perfect density estimation.

Applying Bayes' rule to the score function yields $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t \mid y) = \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t) + \nabla_{\boldsymbol{x}_t} \log p(y \mid \boldsymbol{x}_t)$. Substituting Eqs. (12) and (13) into the CFG difference term cancels the unconditional score:

$$\begin{aligned} \epsilon_\theta(\boldsymbol{x}_t, y) - \epsilon_\theta(\boldsymbol{x}_t) &\approx -\sqrt{1-\bar{\alpha}_t}\, (\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t \mid y) - \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)) \\ &= -\sqrt{1-\bar{\alpha}_t} \cdot \nabla_{\boldsymbol{x}_t} \log p(y \mid \boldsymbol{x}_t). \end{aligned} \tag{14}$$

Substituting this back into Eq. (11), we obtain the effective noise prediction:

$$\hat{\epsilon}_\theta(\boldsymbol{x}_t) \approx \epsilon_\theta(\boldsymbol{x}_t) - w\sqrt{1-\bar{\alpha}_t} \cdot \nabla_{\boldsymbol{x}_t} \log p(y \mid \boldsymbol{x}_t). \tag{15}$$

Step 2: Deriving the Biased Langevin Update. In the standard DDPM sampling process [44], the transition from $\boldsymbol{x}_t$ to $\boldsymbol{x}_{t-1}$ is given by:

$$\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \hat{\epsilon}_\theta(\boldsymbol{x}_t) \right) + \sigma_t \mathbf{z}, \tag{16}$$

where $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$. Substituting Eq. (15) into Eq. (16):

$$\boldsymbol{x}_{t-1} \approx \frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \left[\epsilon_\theta(\boldsymbol{x}_t) - w\sqrt{1-\bar{\alpha}_t}\, \nabla_{\boldsymbol{x}_t} \log p(y \mid \boldsymbol{x}_t) \right] \right) + \sigma_t \mathbf{z}. \tag{17}$$

We define $\mu_\theta(\boldsymbol{x}_t) = \frac{1}{\sqrt{\alpha_t}}\big(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\boldsymbol{x}_t)\big)$ as the standard reconstruction mean toward the unconditional manifold. For the remaining guidance term, the coefficient $\sqrt{1-\bar{\alpha}_t}$ elegantly cancels out:

$$\begin{aligned}\text{Drift Term} &= \frac{1}{\sqrt{\alpha_t}}\left(-\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\right)\cdot\left(-w\sqrt{1-\bar{\alpha}_t}\,\nabla_{\boldsymbol{x}_t}\log p(y\mid\boldsymbol{x}_t)\right)\\ &= \frac{1-\alpha_t}{\sqrt{\alpha_t}}\,w\cdot\nabla_{\boldsymbol{x}_t}\log p(y\mid\boldsymbol{x}_t).\end{aligned} \tag{18}$$

Defining $\lambda_t = \frac{1-\alpha_t}{\sqrt{\alpha_t}} > 0$, the update rule simplifies to the biased Langevin dynamic:

$$\boldsymbol{x}_{t-1} \approx \mu_\theta(\boldsymbol{x}_t) + \lambda_t w\nabla_{\boldsymbol{x}_t}\log p(y\mid\boldsymbol{x}_t) + \sigma_t\mathbf{z}. \tag{19}$$

Step 3: From Approximation to Equality. To decouple our subsequent geometric analysis and theorems from the empirical approximation errors of neural networks, we adopt the standard theoretical abstraction of perfect density estimation. Under this idealized setting, the score-matching mapping becomes exact, allowing us to drop the approximation ($\approx$) and rigorously formalize the sampling step as an exact equality:

$$\boldsymbol{x}_{t-1} = \mu_\theta(\boldsymbol{x}_t) + \lambda_t w\nabla_{\boldsymbol{x}_t}\log p(y\mid\boldsymbol{x}_t) + \sigma_t\mathbf{z}. \tag{20}$$

This exact formulation serves as the foundational equation for analysis of trajectory optimization in Section 3.

B.2 Proof of Geometric Barrier and Dynamics Failure (Section 3.3)

In this section, we provide the proofs for Theorem 3.1 and Proposition 3.2. We first establish the general geometric condition for distance reduction in high-dimensional space (Theorem 3.1) and then apply this condition to the specific Langevin dynamics of diffusion planners to demonstrate the self-locking mechanism (Proposition 3.2).

B.2.1 Proof of Theorem 3.1 (Geometric Barrier)

Proof. Without loss of generality, place the optimum at the origin ($\boldsymbol{x}^* = \mathbf{0}$), so that $d^2 = \|\boldsymbol{x}\|^2$ and the unit inward radial direction is $\mathbf{n}^* = -\boldsymbol{x}/d$. The update is $\Delta\boldsymbol{x} = \mathbf{v}_{\text{det}} + \mathbf{v}_{\text{noise}}$. The expected squared distance of the new state $\boldsymbol{x}' = \boldsymbol{x} + \Delta\boldsymbol{x}$ is:

$$\begin{aligned}\mathbb{E}[\|\boldsymbol{x}'\|^2] &= \mathbb{E}[\|\boldsymbol{x} + \mathbf{v}_{\text{det}} + \mathbf{v}_{\text{noise}}\|^2]\\ &= \|\boldsymbol{x}\|^2 + 2\langle\boldsymbol{x}, \mathbf{v}_{\text{det}}\rangle + \|\mathbf{v}_{\text{det}}\|^2 + \mathbb{E}[\|\mathbf{v}_{\text{noise}}\|^2] + 2\mathbb{E}[\langle\boldsymbol{x} + \mathbf{v}_{\text{det}}, \mathbf{v}_{\text{noise}}\rangle].\end{aligned} \tag{21}$$

Since $\mathbf{v}_{\text{noise}} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$, the cross-term vanishes ($\mathbb{E}[\mathbf{v}_{\text{noise}}] = 0$), and $\mathbb{E}[\|\mathbf{v}_{\text{noise}}\|^2] = D\sigma^2$. Substituting $\boldsymbol{x} = -d\mathbf{n}^*$:

$$\mathbb{E}[\Delta(d^2)] = \mathbb{E}[\|\boldsymbol{x}'\|^2] - d^2 = -2d\langle\mathbf{v}_{\text{det}}, \mathbf{n}^*\rangle + \|\mathbf{v}_{\text{det}}\|^2 + D\sigma^2. \tag{22}$$

Letting $v_{\text{rad}} = \langle\mathbf{v}_{\text{det}}, \mathbf{n}^*\rangle$, the exact condition for distance reduction ($\mathbb{E}[\Delta(d^2)] < 0$) is $2dv_{\text{rad}} > \|\mathbf{v}_{\text{det}}\|^2 + D\sigma^2$, i.e., SNR $= \frac{2dv_{\text{rad}}}{\|\mathbf{v}_{\text{det}}\|^2 + D\sigma^2} > 1$. □

2D Geometric Intuition Figure 5 provides a geometric illustration of Theorem 3.1. Consider two points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ located at distances d_1 and d_2 from the optimum, respectively, with $d_2 \ll d_1$. The update direction at any point is decomposed into a deterministic drift component $\mathbf{v}_{\text{det}}$ and an isotropic noise component $\mathbf{v}_{\text{noise}}$. In the outer region (Large Sphere Shell, radius d_1), the combined effect of drift and noise produces a gradient $\Delta\boldsymbol{x}_1 = \mathbf{v}_{\text{det}} + \mathbf{v}_{\text{noise}}$ that reduces the distance to the optimum ($\Delta(d^2) < 0$), resulting in convergent behavior. In contrast, in the inner region (Small Sphere Shell, radius d_2), the same decomposition produces $\Delta\boldsymbol{x}_2 = \mathbf{v}_{\text{det}} + \mathbf{v}_{\text{noise}}$ that increases the distance from the optimum ($\Delta(d^2) > 0$), leading to divergent behavior.

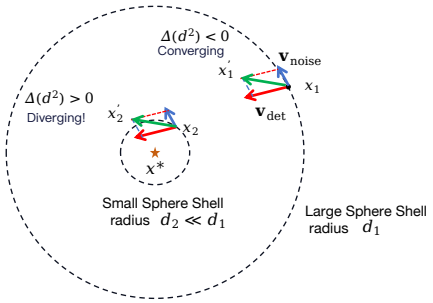


Figure 5: 2D Geometric illustration of Theorem 3.1

B.2.2 Proof of Proposition 3.2

Proof. We isolate the return-guided step from the unconditional mean: $\mathbf{v}_{\text{det}} = \lambda w_R \mathbf{g}_R$ and $\mathbf{v}_{\text{noise}} = \sigma \mathbf{z}$. We define the velocity of squared distance reduction as $V_{sq}(w_R) := -\mathbb{E}[\Delta(d^2)]_R$. Substituting the terms into Theorem 3.1:

$$V_{sq}(w_R) = 2d\lambda w_R \langle \mathbf{g}_R, \mathbf{n}^* \rangle - \lambda^2 w_R^2 \|\mathbf{g}_R\|^2 - D\sigma^2. \tag{23}$$

To find the theoretical limit of convergence, we solve for the optimal scale w_R^* that maximizes $V_{sq}(w_R)$:

$$\frac{dV_{sq}}{dw_R} = 2d\lambda \langle \mathbf{g}_R, \mathbf{n}^* \rangle - 2\lambda^2 w_R \|\mathbf{g}_R\|^2 = 0 \implies w_R^* = \frac{d\langle \mathbf{g}_R, \mathbf{n}^* \rangle}{\lambda \|\mathbf{g}_R\|^2}. \tag{24}$$

Substituting w_R^* back into $V_{sq}(w_R)$ gives the maximum possible convergence velocity:

$$V_{sq}(w_R^*) = \frac{d^2 \langle \mathbf{g}_R, \mathbf{n}^* \rangle^2}{\|\mathbf{g}_R\|^2} - D\sigma^2. \tag{25}$$

Optimization stagnates when even this maximum velocity is non-positive ($V_{sq}(w_R^*) \leq 0$):

$$\frac{d^2 \langle \mathbf{g}_R, \mathbf{n}^* \rangle^2}{\|\mathbf{g}_R\|^2} \leq D\sigma^2 \implies \langle \mathbf{g}_R, \mathbf{n}^* \rangle \leq \frac{\sigma\sqrt{D}\,\|\mathbf{g}_R\|}{d} \text{(i.e., SNR} \leq 1). \tag{26}$$

□

B.2.3 Generalization to Local Quadratic Landscapes

In Section 3, we derived the geometric barrier under an idealized *spherical* return landscape to geometrically define the Signal-to-Noise Ratio (SNR). Here, we justify its applicability to general smooth optimization landscapes by extending the analysis to *general quadratic basins*. Since any smooth non-degenerate local optimum can be approximated by a quadratic form via Taylor expansion, this generalization mathematically bridges our theoretical abstraction with complex continuous environments.

1. Local Quadratic Model and Expected Return Dynamics. Let $\boldsymbol{x}^*$ be a strict local maximum of the return function $R(\boldsymbol{x})$. By Taylor expansion,

$$R(\boldsymbol{x}) \approx R(\boldsymbol{x}^*) - \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^*)^\top \mathbf{A} (\boldsymbol{x} - \boldsymbol{x}^*), \tag{27}$$

where $\mathbf{A} = -\nabla^2 R(\boldsymbol{x}^*) \succ 0$ is the positive definite Hessian matrix.

Consider an optimization update $\Delta \boldsymbol{x} = \mathbf{v}_{\text{det}} + \mathbf{v}_{\text{noise}}$, consistent with the biased Langevin dynamics of diffusion planners, where $\mathbf{v}_{\text{noise}} \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$. The expected change in the return is given by:

$$\begin{aligned} \mathbb{E}[\Delta R] &= \mathbb{E}\left[R(\boldsymbol{x} + \Delta \boldsymbol{x}) - R(\boldsymbol{x})\right] \\ &= -(\boldsymbol{x} - \boldsymbol{x}^*)^\top \mathbf{A} \mathbf{v}_{\text{det}} - \frac{1}{2} \mathbf{v}_{\text{det}}^\top \mathbf{A} \mathbf{v}_{\text{det}} - \frac{1}{2} \mathbb{E}\left[\mathbf{v}_{\text{noise}}^\top \mathbf{A} \mathbf{v}_{\text{noise}}\right]. \end{aligned} \tag{28}$$

2. The Universality of the $O(D)$ Noise Penalty. The critical term inhibiting optimization is the noise-induced penalty: $\mathcal{P}_{\text{noise}} = \frac{1}{2}\mathbb{E}\left[\mathbf{v}_{\text{noise}}^\top \mathbf{A}\mathbf{v}_{\text{noise}}\right]$. Using the properties of the trace operator and isotropic noise, this evaluates exactly to:

$$\mathcal{P}_{\text{noise}} = \frac{1}{2}\sigma^2 \text{Tr}(\mathbf{A}) = \frac{1}{2}\sigma^2 \sum_{i=1}^{D} \lambda_i = \frac{1}{2} D\sigma^2 \bar{\lambda}, \tag{29}$$

where λ_i are the eigenvalues of $\mathbf{A}$ and $\bar{\lambda}$ is the average manifold curvature.

Crucially, this result demonstrates that the noise penalty is fundamentally governed by the trace of the Hessian, scaling linearly with the dimensionality D, regardless of how ill-conditioned or anisotropic the matrix $\mathbf{A}$ is.

Conclusion. In our spherical model (Theorem 3.1), the optimization enjoys perfect signal alignment while suffering the standard $O(D)$ noise penalty. This represents the best-case scenario for optimization progress. Because real-world anisotropic landscapes feature the same $O(D)$ trace-driven penalty but strictly weaker radial signals, the SNR barrier derived from the hyperspherical model constitutes a mathematically rigorous and *conservative lower bound*. If the diffusion planner self-locks in this idealized setting, it will more easily self-lock in more complex, ill-conditioned auto-bidding environments.

B.3 Analysis for Proposition 3.4

We now formalize the properties of the scarcity gradient. Our analysis proceeds in two steps: first, we establish the geometric properties of the explicit scarcity gradient $\nabla_{\boldsymbol{x}} S(\boldsymbol{x})$ under the framework of Denoising Autoencoders (DAEs); second, we prove that the implicit CFG gradient $\mathbf{g}_S$ used in ISGE aligns with this explicit gradient via the chain rule.

B.3.1 Geometry of Explicit Scarcity Gradient

Our analysis builds upon the foundational result of [8], which connects reconstruction error to the underlying data manifold.

Assumption B.1 (Scarcity as Squared Distance Proxy)**.** Let the training data lie on or near a smooth D-dimensional manifold $\mathcal{M} \subset \mathbb{R}^D$. Let $S(\boldsymbol{x})$ be the expected squared reconstruction error of a DAE trained with small isotropic Gaussian corruption noise of variance η^2. Then, in the limit $\eta \to 0$, there exists a tubular neighborhood $\mathcal{N}_r(\mathcal{M}) = \{\boldsymbol{x} \in \mathbb{R}^D : \text{dist}(\boldsymbol{x}, \mathcal{M}) < r\}$ for some $r > 0$, such that

$$S(\boldsymbol{x}) = c \cdot \text{dist}(\boldsymbol{x}, \mathcal{M})^2 + o(\text{dist}(\boldsymbol{x}, \mathcal{M})^2),$$

where $c > 0$ is a constant independent of $\boldsymbol{x}$, and $\text{dist}(\boldsymbol{x}, \mathcal{M}) = \inf_{\mathbf{y} \in \mathcal{M}} \|\boldsymbol{x} - \mathbf{y}\|$.

Lemma B.2 (Geometric Properties of ∇S)**.** *In the tubular neighborhood $\mathcal{N}_r(\mathcal{M})$, the explicit scarcity gradient $\mathbf{v}_S(\boldsymbol{x}) := \nabla_{\boldsymbol{x}} S(\boldsymbol{x})$ satisfies:*

1. ***Normal Alignment:*** *For any $\boldsymbol{x} \in \mathcal{N}_r(\mathcal{M}) \setminus \mathcal{M}$, let $\mathbf{y} = \Pi_{\mathcal{M}}(\boldsymbol{x})$ be the unique projection of $\boldsymbol{x}$ onto $\mathcal{M}$. Then $\mathbf{v}_S(\boldsymbol{x})$ lies in the normal space of $\mathcal{M}$ at $\mathbf{y}$, i.e., $\mathbf{v}_S(\boldsymbol{x}) \in \mathcal{N}_{\mathbf{y}}\mathcal{M}$.*
2. ***Non-Vanishing Magnitude:*** *There exists $\delta > 0$ such that for all $\boldsymbol{x}$ with $\text{dist}(\boldsymbol{x}, \mathcal{M}) \geq \epsilon_0 > 0$ (for some fixed $\epsilon_0 < r$), we have $\|\mathbf{v}_S(\boldsymbol{x})\| \geq \delta$.*

Proof. By Assumption B.1, we have $S(\boldsymbol{x}) = c \cdot \text{dist}(\boldsymbol{x}, \mathcal{M})^2 + o(\text{dist}(\boldsymbol{x}, \mathcal{M})^2)$. It is a classical result in differential geometry [45] that the squared distance function to a smooth manifold is differentiable in its tubular neighborhood, and its gradient is given by

$$\nabla_{\boldsymbol{x}} \text{dist}(\boldsymbol{x}, \mathcal{M})^2 = 2(\boldsymbol{x} - \Pi_{\mathcal{M}}(\boldsymbol{x})).$$

The vector $(\boldsymbol{x} - \Pi_{\mathcal{M}}(\boldsymbol{x}))$ is orthogonal to the tangent space $T_{\Pi_{\mathcal{M}}(\boldsymbol{x})}\mathcal{M}$ by definition of the projection, and thus lies in the normal space $\mathcal{N}_{\Pi_{\mathcal{M}}(\boldsymbol{x})}\mathcal{M}$. Therefore,

$$\mathbf{v}_S(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} S(\boldsymbol{x}) = 2c(\boldsymbol{x} - \Pi_{\mathcal{M}}(\boldsymbol{x})) + o(\|\boldsymbol{x} - \Pi_{\mathcal{M}}(\boldsymbol{x})\|),$$

which establishes the orthogonality property. For the magnitude, note that $\|\boldsymbol{x} - \Pi_{\mathcal{M}}(\boldsymbol{x})\| = \text{dist}(\boldsymbol{x}, \mathcal{M}) \geq \epsilon_0 > 0$. Hence,

$$\|\mathbf{v}_S(\boldsymbol{x})\| = \|2c(\boldsymbol{x} - \Pi_{\mathcal{M}}(\boldsymbol{x})) + o(\epsilon_0)\| \geq 2c\epsilon_0 - o(\epsilon_0).$$

Choosing ϵ_0 sufficiently small such that the higher-order term is dominated by the leading term, we obtain $\|\mathbf{v}_S(\boldsymbol{x})\| \geq \delta$ for some $\delta > 0$. □

B.3.2 Alignment of Implicit CFG Gradient

We now bridge the theoretical result above to the actual implementation in ISGE. The ISGE framework utilizes an implicit gradient via Classifier-Free Guidance (CFG):

$$\mathbf{g}_S(\boldsymbol{x}) := \nabla_{\boldsymbol{x}} \log p(y_S|\boldsymbol{x}),$$

where y_S represents the condition of "high scarcity".

Justification of the Chain Rule. Crucially, during the construction of the exploration dataset, the scarcity label y_S for any trajectory $\boldsymbol{x}$ is derived *exclusively* from the scarcity metric $S(\boldsymbol{x})$ (i.e., the reconstruction error). This establishes a deterministic dependency structure: $\boldsymbol{x} \to S(\boldsymbol{x}) \to y_S$. By the Markov property of this labeling process, y_S is conditionally independent of $\boldsymbol{x}$ given $S(\boldsymbol{x})$. Therefore, the likelihood satisfies the exact relation:

$$p(y_S|\boldsymbol{x}) = p(y_S|S(\boldsymbol{x})).$$

Consequently, the gradient calculation follows the chain rule strictly without approximation:

$$\mathbf{g}_S(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} \log p(y_S|S(\boldsymbol{x})) = \underbrace{\frac{\partial \log p(y_S|S)}{\partial S}}_{\alpha(S)} \cdot \underbrace{\nabla_{\boldsymbol{x}} S(\boldsymbol{x})}_{\mathbf{v}_S(\boldsymbol{x})}. \tag{30}$$

Directional Consistency. The scalar coefficient $\alpha(S) = \frac{\partial \log p(y_S|S)}{\partial S}$ determines the alignment. In our formulation, y_S is set to a high quantile (e.g., the 99th percentile) of the scarcity distribution. This implies that the likelihood $p(y_S|S)$ is a monotonically increasing function of S within the region of interest (as higher S indicates the sample is closer to the target condition y_S). Consequently, the derivative is strictly positive:

$$\alpha(S) > 0.$$

Thus, $\mathbf{g}_S(\boldsymbol{x})$ is a positive scaling of $\nabla_{\boldsymbol{x}} S(\boldsymbol{x})$. By Lemma B.2, the implicit CFG gradient $\mathbf{g}_S(\boldsymbol{x})$ inherits the geometric properties of **Orthogonality** (aligning with the manifold normal $\mathbf{n}_\perp$) and **Non-Vanishing Magnitude** (scaled by $\alpha(S)$). This justifies the use of $\mathbf{g}_S$ as a valid repulsion field in Theorem 3.5.

B.4 Proof of Theorem 3.5

Assumption B.3 (Geometry of Scarcity Guidance)**.** We make the following geometric assumptions based on the theory of regularized autoencoders and boundary optimization dynamics:

1. **Frontier Alignment (Selection-Induced Optimality):** For self-locked trajectories at the suboptimal frontier, the outward-pointing scarcity gradient aligns positively with the OOD global optimum: $\mathcal{V}_S := \langle \mathbf{g}_S(\boldsymbol{x}), \mathbf{n}^* \rangle > 0$.

   Justification: In the general case of an arbitrary return landscape, self-locking implies the exhaustion of improvement directions within the manifold's tangent space (i.e., $\nabla_{\text{tan}} R(\boldsymbol{x}) \approx 0$). Consequently, the true gradient of improvement $\mathbf{n}^*$ is dominated by its normal component. Since the training data represents a suboptimal subset, the direction of higher return at this boundary naturally points towards the unexplored exterior (outward polarity) rather than the interior. Thus, the scarcity gradient $\mathbf{g}_S$, which drives outward traversal, shares the same outward polarity with $\mathbf{n}^*$ with high probability. This geometric correlation ensures that scarcity-guided exploration reliably uncovers the cone of improving directions.

2. **Opposing Gradients at the Boundary:** By Proposition 3.4, the scarcity gradient $\mathbf{g}_S$ points strictly outward, normal to the known data manifold. Conversely, the implicit return gradient $\mathbf{g}_R$ under self-locking is confined by the empirical data support, exerting an inward pull to keep the trajectory within the known distribution. Geometrically, these two vector fields are structurally opposed (or at least orthogonal) at the OOD boundary, yielding a non-positive inner product: $\langle \mathbf{g}_R, \mathbf{g}_S \rangle \leq 0$.

   Note: See Appendix C.5.2 for detailed empirical validation of these two assumptions.

Proof. We analyze the exact change in squared distance. At the critical stagnation boundary, the optimal return-guided optimization is perfectly halted. Let $V_{sq}^R := -\mathbb{E}[\Delta(d^2)]_R = 0$. By introducing the scarcity guidance $\mathbf{v}_{\text{dual}} = \lambda w_R \mathbf{g}_R + \lambda w_S \mathbf{g}_S$, the new velocity of squared distance reduction is $V_{sq}^{\text{dual}}(w_S) = V_{sq}^R + \Delta V_{sq}(w_S)$. Our objective is to prove $\Delta V_{sq}(w_S) > 0$.

Step 1: Incremental Velocity Formulation Using the exact equation from Theorem 3.1, the dual-guided velocity is:

$$\begin{aligned} V_{sq}^{\text{dual}}(w_S) &= 2d\langle \lambda w_R \mathbf{g}_R + \lambda w_S \mathbf{g}_S, \mathbf{n}^* \rangle - \|\lambda w_R \mathbf{g}_R + \lambda w_S \mathbf{g}_S\|^2 - D\sigma^2 \\ &= \left(2d\lambda w_R \langle \mathbf{g}_R, \mathbf{n}^* \rangle - \lambda^2 w_R^2 \|\mathbf{g}_R\|^2 - D\sigma^2\right) \\ &\quad + 2d\lambda w_S \langle \mathbf{g}_S, \mathbf{n}^* \rangle - \lambda^2 w_S^2 \|\mathbf{g}_S\|^2 - 2\lambda^2 w_R w_S \langle \mathbf{g}_R, \mathbf{g}_S \rangle. \end{aligned} \tag{31}$$

The first parenthesis is exactly $V_{sq}^R = 0$. By Assumption B.3 (Opposing Gradients), we have $\langle \mathbf{g}_R, \mathbf{g}_S \rangle \leq 0$. Thus, the cross-term actually provides a non-negative velocity bonus ($-2\lambda^2 w_R w_S \langle \mathbf{g}_R, \mathbf{g}_S \rangle \geq 0$). To ensure robustness, we drop this bonus to establish a conservative lower bound for the exact velocity increment:

$$\Delta V_{sq}(w_S) \geq \underbrace{2d\lambda w_S \mathcal{V}_S}_{\text{Marginal Signal Gain}} - \underbrace{\lambda^2 w_S^2 \|\mathbf{g}_S\|^2}_{\text{Marginal Penalty Increase}}. \tag{32}$$

Step 2: Feasible Interval and Optimization Optimization is strictly revived when this lower bound is positive:

$$2d\lambda w_S \mathcal{V}_S - \lambda^2 w_S^2 \|\mathbf{g}_S\|^2 > 0 \implies w_S < \frac{2d\mathcal{V}_S}{\lambda \|\mathbf{g}_S\|^2}. \tag{33}$$

This defines the feasible interval. To maximize the guaranteed progress, we set the derivative of the lower bound to zero:

$$2d\lambda \mathcal{V}_S - 2\lambda^2 w_S \|\mathbf{g}_S\|^2 = 0 \implies w_S^* = \frac{d\mathcal{V}_S}{\lambda \|\mathbf{g}_S\|^2}. \tag{34}$$

Step 3: Marginal SNR Analysis We define the marginal SNR as the ratio of the marginal signal gain to the marginal penalty increase evaluated at w_S^*:

$$\text{SNR}_\Delta(w_S^*) = \frac{2d\lambda w_S^* \mathcal{V}_S}{\lambda^2 (w_S^*)^2 \|\mathbf{g}_S\|^2}. \tag{35}$$

Substituting w_S^*:

$$\text{SNR}_\Delta(w_S^*) = \frac{2d\lambda \left(\frac{d\mathcal{V}_S}{\lambda \|\mathbf{g}_S\|^2}\right) \mathcal{V}_S}{\lambda^2 \left(\frac{d\mathcal{V}_S}{\lambda \|\mathbf{g}_S\|^2}\right)^2 \|\mathbf{g}_S\|^2} = \frac{2d^2 \mathcal{V}_S^2 / \|\mathbf{g}_S\|^2}{d^2 \mathcal{V}_S^2 / \|\mathbf{g}_S\|^2} = 2. \tag{36}$$

This establishes that, at the optimal guidance scale, the incremental signal is exactly twice the incremental penalty ($\text{SNR}_\Delta \geq 2$ when considering the cross-term bonus), mathematically guaranteeing that the signal permanently dominates.

Step 4: Guaranteed Maximum Progress Substituting w_S^* into the conservative lower bound yields the guaranteed minimum velocity increment:

$$\Delta V_{sq}(w_S^*) \geq 2d\lambda \left(\frac{d\mathcal{V}_S}{\lambda \|\mathbf{g}_S\|^2}\right) \mathcal{V}_S - \lambda^2 \left(\frac{d\mathcal{V}_S}{\lambda \|\mathbf{g}_S\|^2}\right)^2 \|\mathbf{g}_S\|^2 = \frac{d^2 \mathcal{V}_S^2}{\|\mathbf{g}_S\|^2}. \tag{37}$$

Since $\mathcal{V}_S > 0$ (Frontier Alignment), we strictly have $\Delta V_{sq}(w_S^*) > 0$. Because $-\mathbb{E}[\Delta(d^2)]_{\text{dual}} = V_{sq}^R + \Delta V_{sq}(w_S^*) > 0$, we conclude that $\mathbb{E}[\Delta(d^2)]_{\text{dual}} < 0$. This proves that the guidance handoff breaks the geometric bottleneck. □

B.5 Restoring the Return Gradient via Fine-Tuning on Gift Samples

Definition B.4 (Gift Samples)**.** Given a generated dataset $\mathcal{D}$, the gift samples form a subset $\mathcal{D}_{\text{gift}} = \{\boldsymbol{x} \in \mathcal{D} \mid S(\boldsymbol{x}) \geq \gamma_S,\ R(\boldsymbol{x}) \geq \gamma_R\}$, where γ_S and γ_R are high percentiles (e.g., 90th percentile) of scarcity and return in $\mathcal{D}$.

Theorem B.5 (Return Gradient Restoration)**.** *Consider the functional gradient update induced by training on the exploratory gift samples $\mathcal{D}_{gift}$. For a test sample $\boldsymbol{x}$ located at the OOD boundary, let $\Delta \mathbf{g}_R(\boldsymbol{x}) := \mathbf{g}_{R,new}(\boldsymbol{x}) - \mathbf{g}_{R,old}(\boldsymbol{x})$ denote the expected contribution of $\mathcal{D}_{gift}$ to the return gradient field. We mathematically establish that this geometric update positively aligns with the true OOD optimum $\mathbf{n}^*$: $\mathbb{E}\big[\langle \Delta \mathbf{g}_R(\boldsymbol{x}), \mathbf{n}^* \rangle\big] > 0$.*

Problem Setup and NTK Derivation. We analyze the functional update at a test point $\boldsymbol{x}$ after fine-tuning on the gift sample set $\mathcal{D}_{\text{gift}} = \{\boldsymbol{x}_i\}_{i=1}^M$. The fine-tuning loss is $\mathcal{L}(\theta) = \frac{1}{2}\sum_{i=1}^M (f_\theta(\boldsymbol{x}_i) - y_{\text{gift}})^2$. We assume the generated gift samples represent a higher return target, yielding a positive initial prediction error $\delta_i = y_{\text{gift}} - f_{\text{old}}(\boldsymbol{x}_i) > 0$.

Under the continuous-time Neural Tangent Kernel (NTK) regime, the functional evolution follows gradient descent dynamics:

$$\Delta f(\boldsymbol{x}) \approx \eta \sum_{i=1}^M \delta_i \, \Theta(\boldsymbol{x}, \boldsymbol{x}_i). \tag{38}$$

Crucially, because the target optimum $\mathbf{n}^*$ possesses a fundamental radial/outward component, projecting the analysis onto a constrained tangent space (e.g., assuming $\boldsymbol{x} \in \mathbb{S}^{D-1}$) artificially nullifies radial updates. Instead, we evaluate the gradient field directly in the full Euclidean space $\mathbb{R}^D$ using a standard smooth, decreasing shift-invariant kernel $\Theta(\boldsymbol{x}, \boldsymbol{x}_i) = \kappa(\|\boldsymbol{x} - \boldsymbol{x}_i\|^2)$ with $\kappa'(\cdot) < 0$. Taking the spatial gradient w.r.t $\boldsymbol{x}$ yields the restored return signal:

$$\begin{aligned} \Delta \mathbf{g}_R(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} \Delta f(\boldsymbol{x}) &= \eta \sum_{i=1}^M \delta_i \cdot 2\kappa'(\|\boldsymbol{x} - \boldsymbol{x}_i\|^2) \cdot (\boldsymbol{x} - \boldsymbol{x}_i) \\ &= \eta \sum_{i=1}^M c_i(\boldsymbol{x}_i) \quad \underbrace{(\boldsymbol{x}_i - \boldsymbol{x})}_{\text{Full Displacement}} , \end{aligned} \tag{39}$$

where $c_i(\boldsymbol{x}_i) := -2\delta_i \, \kappa'(\|\boldsymbol{x} - \boldsymbol{x}_i\|^2) > 0$ acts as a strictly positive weighting coefficient. Notice that the update vector $\boldsymbol{x}_i - \boldsymbol{x}$ represents the *full displacement* in $\mathbb{R}^D$, perfectly capturing the necessary radial variation.

Structure of Gift Samples. According to the ISGE dual-guidance update rule (Theorem 3.5), each sample is generated by combining deterministic guidance and isotropic noise:

$$\boldsymbol{x}_i = \boldsymbol{x} + \underbrace{\lambda w_R \mathbf{g}_R(\boldsymbol{x}) + \lambda w_S \mathbf{g}_S(\boldsymbol{x})}_{\mathbf{d}} + \sigma \mathbf{z}_i. \tag{40}$$

Substituting the full displacement $\boldsymbol{x}_i - \boldsymbol{x} = \mathbf{d} + \sigma \mathbf{z}_i$ and taking the expectation over the standard Gaussian noise $\mathbf{z}_i$, the expected gradient update evaluates to:

$$\mathbb{E}[\Delta \mathbf{g}_R] = \eta M \Big(\mathbb{E}[c_i(\mathbf{z}_i)]\mathbf{d} + \sigma \mathbb{E}[c_i(\mathbf{z}_i)\mathbf{z}_i] \Big). \tag{41}$$

Projection onto the Optimal Direction $\mathbf{n}^*$. We project this expected update onto the true optimal direction $\mathbf{n}^*$:

$$\langle \mathbb{E}[\Delta \mathbf{g}_R], \mathbf{n}^* \rangle = \eta M \Big(\underbrace{\mathbb{E}[c_i(\mathbf{z}_i)] \langle \mathbf{d}, \mathbf{n}^* \rangle}_{\text{Term 1: Deterministic Signal}} + \underbrace{\sigma \mathbb{E}[c_i(\mathbf{z}_i) \langle \mathbf{z}_i, \mathbf{n}^* \rangle]}_{\text{Term 2: Covariance Drag}} \Big). \tag{42}$$

We rigorously analyze the geometric and asymptotic properties of both terms in the continuous-time diffusion limit ($\lambda \to 0$, with variance $\sigma^2 \propto \lambda$):

1. **Term 1 (Deterministic Signal):** Expanding the combined drift gives $\langle \mathbf{d}, \mathbf{n}^* \rangle = \lambda w_R \langle \mathbf{g}_R, \mathbf{n}^* \rangle + \lambda w_S \langle \mathbf{g}_S, \mathbf{n}^* \rangle$. At the self-locking boundary (Proposition 3.2), the valid radial component of the return guidance is virtually exhausted ($\langle \mathbf{g}_R, \mathbf{n}^* \rangle \approx 0$). In contrast, the scarcity gradient directly aligns with the outward optimum (Assumption B.3), ensuring $\langle \mathbf{g}_S, \mathbf{n}^* \rangle > 0$. Consequently, the deterministic signal $\langle \mathbf{d}, \mathbf{n}^* \rangle$ is strictly positive. Because $\mathbf{d}$ is linear with respect to the step size λ, the magnitude of Term 1 scales asymptotically as $\mathcal{O}(\lambda)$.
2. **Term 2 (Covariance Drag):** We formally evaluate the correlation between the positive weight $c_i(\mathbf{z}_i)$ and the noise $\mathbf{z}_i$. By applying a first-order Taylor expansion around the deterministic mean $\boldsymbol{\mu}_i = \boldsymbol{x} + \mathbf{d}$, we have $c_i(\mathbf{z}_i) \approx \bar{c} + \sigma \nabla_{\boldsymbol{x}} c_i(\boldsymbol{\mu}_i)^\top \mathbf{z}_i$. Since the zero-mean term vanishes ($\bar{c}\mathbb{E}[\mathbf{z}_i] = 0$), the expectation resolves to the covariance: $\sigma \mathbb{E}\left[(\nabla c_i^\top \mathbf{z}_i) \langle \mathbf{z}_i, \mathbf{n}^* \rangle \right]$. To determine the sign, recall that c_i is positively proportional to the prediction error $\delta_i = y_{\text{gift}} - f_{\text{old}}(\boldsymbol{x}_i)$. Moving along the true optimal direction $\mathbf{n}^*$ intrinsically increases the

underlying return, thereby reducing the prediction error. Consequently, the sensitivity gradient $\nabla_{\boldsymbol{x}} c_i$ is structurally anti-parallel to the optimal direction $\mathbf{n}^*$. This geometric opposition guarantees that the covariance expectation evaluates to a negative value. This term acts as a fundamental statistical drag force. Critically, because it arises from the quadratic noise expectation $\mathbb{E}[\mathbf{z}_i \mathbf{z}_i^\top] = \mathbf{I}$, its final magnitude strictly scales with σ^2. Given $\sigma^2 \propto \lambda$, Term 2 also scales precisely as $\mathcal{O}(\lambda)$.

Overcoming the Drag via Guidance Scale. Crucially, both the deterministic signal (Term 1) and the statistical covariance drag (Term 2) operate on the identical asymptotic order of $\mathcal{O}(\lambda)$. Consequently, one does not asymptotically vanish relative to the other.

Instead, the strict dominance between these terms is governed by their leading coefficients. The positive signal in Term 1 is explicitly scaled by the scarcity guidance weight w_S. Because w_S is a controllable hyperparameter in Classifier-Free Guidance (empirically set to a large magnitude for active OOD exploration), selecting a sufficiently large w_S mathematically guarantees that the outward scarcity impulse forcefully overrides the negative statistical drag:

$$\text{Term 1} > |\text{Term 2}|.$$

Conclusion. Therefore, we conclude that $\mathbb{E}[\langle \Delta \mathbf{g}_R, \mathbf{n}^* \rangle] > 0$. The fine-tuning process inherently leverages the full displacement vectors of the generated gift samples, robustly restoring the missing radial component of the return gradient and effectively unblocking the optimization landscape.

□

C Experimental Details and Additional Results

C.1 Setup

Benchmark and Dataset We evaluate our method on **AuctionNet** [9], a large-scale industrial auto-bidding benchmark derived from real-world advertising campaigns. AuctionNet provides a flexible and researcher-friendly experimental environment for large-scale auto-bidding studies. To enable efficient iterative experimentation, we adopt a lightweight configuration of the environment. Specifically, we reduce the total traffic volume from 500,000 to 50,000 per evaluation while scaling the mean traffic value by a factor of ten, thereby preserving the overall order of magnitude of total conversions.

The environment consists of 48 competing agents with varying budgets and CPAs, instantiated from a diverse set of bidding strategies, including PID, OnlineLP, BC [46], BCQ [47], TD3+BC [48], IQL [49], CQL [50], MBRL-MOPO [51], and MBRL-COMBO [52]. These strategies span a wide spectrum from rule-based methods to reinforcement learning approaches, covering both conservative and aggressive bidding behaviors. This diversity is designed to faithfully emulate the complexity and heterogeneity of real-world online advertising markets. Since the environment is operated under a lightweight configuration, we retrain all built-in agents to ensure that their policies remain sufficiently competitive under the modified traffic dynamics. In each evaluation episode, our method sequentially interacts with the environment by assuming the role of each of the 48 agents. The final performance of an episode is reported as the average score across all agents.

To faithfully simulate the cold-start dilemma discussed in Section 1, we construct a *cold-start dataset* by rigorously filtering the full offline logs, retaining only trajectories in the return percentile range $[10, 30]$. This creates a challenging learning environment where the data is both low-quality (far below optimality) and narrowly supported (lacking both high-performing demonstrations and diverse failure cases), forcing the model to extrapolate into the unknown. We establish two distinct experimental settings based on traffic characteristics: (1) **AuctionNet-Dense**, characterized by high traffic value ($p^d_{avg} \approx 0.05$) under strict safety constraints; and (2) **AuctionNet-Sparse**, representing a harder exploration landscape with extremely low traffic value ($p^s_{avg} \approx 0.005$) and sparse reward signals.

For each iteration, we conduct 63 online evaluation episodes and log the complete trajectory data generated by all agents. To ensure strict data volume parity across iterations, the training dataset for each new iteration is fixed at 3024 generated trajectories for all methods. For Vanilla Self-Training, all 3024 trajectories are generated by the diffusion planner. For ISGE, the dataset is constructed via a compositional split: 1680 trajectories (5/9) are exploratory samples provided by the ISGE

Explorer, while the remaining 1344 trajectories (4/9) are uniformly downsampled from the planner's standard generation. This data collection procedure requires approximately 2 hours using four GPUs in parallel, whereas running the same protocol under the default online AuctionNet configuration would require over 20 hours per iteration. Detailed parameters are shown in Table 5.

Table 5: The Parameters of AuctionNet-Dense and AuctionNet-Sparse.

Params	AuctionNet-Dense	AuctionNet-Sparse
Impressions (each episode)	50000	50000
Time steps in a trajectory	48	48
Advertiser Number	48	48
State dimension	16	16
Action dimension	1	1
CPA range	[6, 13]	[60, 130]
Mean Traffic Value	0.05	0.005
Trajectories (full)	5376	5040
Max Score (full)	276.6	63.0
Avg Score (full)	86.1	13.5
Trajectories (cold start)	1092	1318
Max Score (cold start)	111.1	21.0
Avg Score (cold start)	55.7	5.1
Trajectories for Iter 1-3	3024	3024

In detail, the state includes the following information:

- time_left: The remaining time steps left in the current advertising period.
- budget_left: The remaining budget that the advertiser has available to spend in the current advertising period.
- historical_bid_mean: The average values of bids made by the advertiser over past time steps.
- last_three_bid_mean: The average values of bids over the last three time steps.
- historical_LeastWinningCost_mean: The average of the least cost required to win an impression over previous time steps.
- historical_pValues_mean: The average of historical p-values over past time steps.
- historical_conversion_mean: The average number of conversions (e.g., sales, clicks, etc.) the advertiser achieved in previous time steps.
- historical_xi_mean: The average winning status of advertisers in impression opportunities, where 1 represents winning and 0 represents not winning.
- last_three_LeastWinningCost_mean: The average of the least winning costs over the last three time steps.
- last_three_pValues_mean: The average of conversion probability of advertising exposure to users over the last three time steps.
- last_three_conversion_mean: The average number of conversions over the last three time steps.
- last_three_xi_mean: The average winning status of advertisers over the last three time steps.
- current_pValues_mean: The mean of p-values at the current time step.
- current_pv_num: The number of impressions served at the current time step.
- last_three_pv_num_total: The total number of impressions served over the last three time steps.
- historical_pv_num_total: The total number of impressions served over past time steps.

Baselines In the main results (Table 1), all methods employ an identical diffusion planner architecture (based on [4]), equipped with a simulation-based inverse controller (see Section C.1), which serves as a strong offline baseline. Here, we provide supplementary baseline results on AuctionNet under both the cold-start and full-data settings in Table 6. The baselines include BC [46], BCQ [47],

CQL [50], DT [20], GAS [14], GAVE [15], EBaReT [21], EGDB [5] and GRAD [16]. Our Full-Data Oracle achieves strong performance on both Dense and Sparse AuctionNet, serving as a reliable empirical reference for the diffusion planner architecture. Meanwhile, the Cold-Start Base remains a dependable baseline, as most baseline methods suffer a similar performance collapse in the cold-start setting.

Table 6: **Supplementary Baselines.** Results are shown for cold-start (trained on cold-start data) and full-data (trained on full data) settings. Diffusion Planner (cold-start) corresponds to "Cold-Start Base" in Table 1; Diffusion Planner (full-data) corresponds to "Full-Data Oracle".

Dataset	Diffusion Planner	BC	BCQ	CQL	DT	GAVE	GAS	EBaReT	EGDB	GRAD
Dense (cold-start)	65.3 ± 3.2	56.0 ± 7.6	68.7 ± 1.2	70.4 ± 3.0	68.9 ± 7.4	$56.0 \pm 4.0^*$	70.1 ± 5.5	56.8 ± 3.9	60.4 ± 3.6	65.5 ± 4.5
Dense (full-data)	102.0 ± 3.5	70.0 ± 5.1	96.3 ± 0.5	106.8 ± 6.0	110.8 ± 6.4	$73.1 \pm 3.2^*$	110.8 ± 6.2	104.6 ± 4.7	105.3 ± 4.3	106.3 ± 6.7
Sparse (cold-start)	15.7 ± 1.0	0.5 ± 0.3	17.0 ± 0.7	10.6 ± 1.3	7.4 ± 0.3	17.2 ± 0.9	14.4 ± 0.7	9.3 ± 1.0	11.1 ± 1.2	15.9 ± 1.1
Sparse (full-data)	20.5 ± 1.1	15.9 ± 0.4	21.8 ± 0.9	17.5 ± 1.2	19.4 ± 1.1	22.7 ± 1.2	21.2 ± 1.1	20.9 ± 0.7	21.4 ± 1.1	22.8 ± 1.5

**:* For GAVE, the original paper reported results on AuctionNet (offline) showing that its performance surpassed that of DT [15]. However, in our AuctionNet-Dense (online) scenario, we could not reproduce similar gains using either version of the loss function provided in the source code.

Metrics We report four key metrics averaged among 48 advertisers over 7 evaluation episodes (total 336 bidding trajectories):

- **WR** (Win Rate): The fraction of page views (PVs) in which the advertiser wins the auction, $\text{WR} = \frac{1}{N}\sum_{i=1}^{N} o_i$, where N is the total number of participated impressions.
- **Reward**: The cumulative conversion value (GMV) $\sum_i o_i v_i$;
- **ER** (Exceeding ratio of the CPA constraints): The exceedance ratio of Cost-Per-Action (CPA) constraints $\text{ER} = \frac{1}{J}\sum_j \frac{C_j^{real} - C_j}{C_j} = \frac{1}{J}\sum_j \frac{\sum_i c_{ij} o_i / \sum_i p_{ij} o_i - C_j}{C_j}$, where C_j is the CPA limit.
- **Score**: A composite metric balancing revenue and constraints, $\text{Score} = (\sum_i o_i v_i) \times \min\{\text{penalty}_j\}_{j=1,\dots,J}$, where $\text{penalty}_j = \min\{(\frac{C_j}{\sum_i c_{ij} o_i / \sum_i p_{ij} o_i})^\beta, 1\}, \beta = 2$.

Simulation-based Inverse Control In our experiments, we employ a *Simulation-based Inverse Control* mechanism for diffusion planner. This design choice allows us to isolate the trajectory planning capability of the diffusion model from the execution errors typically introduced by heuristic controllers. The simulation-based inverse controller solves for the precise action required to realize the planner's state transitions.

Formally, let $\tau_{plan} = \{\hat{s}_t, \hat{s}_{t+1}, \dots, \hat{s}_{t+H}\}$ denote the trajectory generated by the diffusion planner at time step t. The planner dictates that the remaining budget should transition from the current state b_t to $\hat{b}_{t+1}$ in the next step. Consequently, the target expenditure for the current step is derived as $c^*_{target} = \max(0, b_t - \hat{b}_{t+1})$. The objective of the inverse control is to find a bidding scalar α_t such that the actual cost incurred in the auction environment, $\mathcal{C}(\alpha_t, \mathbf{B}_{-i})$, approximates c^*_{target} as closely as possible, where $\mathbf{B}_{-i}$ represents the bids of competitors. We leverage the reproducibility of the simulation environment. During the evaluation of this baseline, we grant the controller oracle access to a "frozen" copy of the current environment state, including the bids of all competing agents. Leveraging the monotonicity of the cost function with respect to the bid scaling factor α, we employ a **Simulation-based Binary Search** to numerically invert the cost function. This mechanism ensures that the realized trajectory aligns almost perfectly with the planned trajectory, barring stochastic deviations in impression exposure (e.g., Bernoulli trials). By effectively eliminating control latency and execution variance, this setup enables a rigorous evaluation of the *planning quality* itself, validating that performance improvements are attributable to the diffusion planner rather than the low-level controller.

Remark on Real-World Feasibility. We acknowledge that the oracle access used in the simulation-based binary search represents an idealized control setting. However, this design is intentional and serves to rigorously decouple the evaluation of our generative planning framework from the noise of low-level execution. In practical production environments where precise environment inversion is unavailable, this module is modular and can be substituted with standard engineering approximations without altering the core diffusion planning mechanism:

- **Learned Inverse Dynamics:** A separate lightweight network (Inverse Dynamics Model [11]) can be trained to predict the action a_t required to achieve the state transition $s_t \to s_{t+1}$.
- **Feedback Control (PID):** Rule-based PID controllers, which are ubiquitous in industrial auto-bidding systems, can dynamically adjust the bidding scalar α_t to track the target expenditure c^*_{target} with bounded error.

While replacing the simulation-based controller with these approximate methods may introduce minor execution noise, the *relative performance advantage* provided by the superior strategic guidance of the diffusion planner remains robust. The planner determines *where* to go, which is the dominant factor in performance, while the controller merely handles *how* to get there. Table 7 shows the performance difference between simulation-based inverse controller and standard inverse dynamic controller. Remarkably, equipped with the standard controller, ISGE (Iter 1) still matches the Full-Data Oracle.

Table 7: Comparison of Scores with Different Controllers.

Dataset	Method	Controller Type	
		Simulation-based	**Standard Inverse**
Dense	**Full-Data Oracle**	102.0 ± 3.5	95.1 ± 3.6
	Vanilla Self-Training (Iter 1)	74.9 ± 3.4	69.1 ± 2.7
	ISGE (Iter 1, Ours)	103.4 ± 5.3	97.4 ± 3.3
Sparse	**Full-Data Oracle**	20.5 ± 1.1	19.3 ± 1.5
	Vanilla Self-Training (Iter 1)	16.9 ± 0.5	15.5 ± 0.9
	ISGE (Iter 1, Ours)	20.0 ± 0.9	19.1 ± 1.0

C.2 Implementation Details of ISGE

We list the hyperparameter details in Table 8 for reproduction. All three models share the same Temporal U-Net backbone and inverse dynamics head, but differ in their conditioning configurations: **DFUSER** (Diffusion planner) is conditioned on return R; **DFUSER** (Judger) is trained *without* any return conditioning to serve as an unconditional quality evaluator; and **DFUSER$_{\text{SC}}$** (Explorer) is conditioned on both return R and scarcity score S.

To compute the scarcity score for a trajectory τ, the Judger applies a fixed-level forward corruption followed by unconditional reconstruction. Specifically, given a clean trajectory $\boldsymbol{x}_0$, we corrupt it to noise level $t^* = \lfloor 0.75 \times T \rfloor$ via the forward process $\boldsymbol{x}_{t^*} = \sqrt{\bar{\alpha}_{t^*}}\, \boldsymbol{x}_0 + \sqrt{1 - \bar{\alpha}_{t^*}}\, \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$ and $\bar{\alpha}_{t^*}$ follows a cosine schedule with $T = 20$ total diffusion steps. The corrupted trajectory is then reconstructed by running the reverse denoising chain from $t^* - 1$ down to 0 without any return conditioning, using only the initial state s_0 as the boundary condition. In practice, the scarcity score is defined as the weighted mean absolute error (MAE) between x_0 and the reconstructed $\hat{x}_0$, computed over a selected subset of state dimensions (budget_left, historical_bid_mean, last_three_bid_mean) that capture the most behaviorally distinctive features of a bidding trajectory.

Discussion on Judger Approximation Errors. In practice, the Judger may introduce approximation discrepancies into the estimated scarcity scores, primarily stemming from finite network capacity and residual denoising variance. Nevertheless, the impact of these approximation errors may be mitigated through two main factors. First, strategic masking explicitly restricts the residual evaluation to controlled strategic dimensions, helping reduce estimation errors caused by uncontrollable environmental stochasticity. Second, the Judger evaluates trajectories directly from $\mathcal{D}_{gen}$, which is the exact exploitation dataset it was just trained on. This in-distribution evaluation may help reduce fitting error and false positives (i.e., non-scarce trajectories misclassified as scarce due to network imperfection). Consequently, these factors may help limit the influence of mislabeled samples on the aggregate scarcity gradient during CFG exploration.

C.3 Computational Cost

All experiments are conducted on a server equipped with four NVIDIA L40S GPUs (48 GB VRAM each) and two Intel Xeon Gold 6462C CPUs (32 cores per socket, 64 physical cores in total).

Table 8: Hyperparameter Settings for DFUSER and $\text{DFUSER}_{\text{SC}}$

Category	Parameter	Value
Diffusion Process	Diffusion Steps (T)	20
	Horizon (H)	48
	Beta Schedule	Cosine
	Prediction Target	ϵ
	Guidance Weight (w)	3.0 ($[w_R, w_S]$=[3, 3] for $\text{DFUSER}_{\text{SC}}$)
	Condition Dropout	0.25 (0.30 for $\text{DFUSER}_{\text{SC}}$)
	Replanning Prob.	0.5 (N/A for $\text{DFUSER}_{\text{SC}}$)
	Inference Target Cond.	$\hat{R}$=$R_{99\%}$; $\hat{S}$=$S_{\max}$ ($\text{DFUSER}_{\text{SC}}$ only)
Network Architecture	Model Type	Temporal U-Net
	Base Channel Dim	64
	Channel Multipliers	$(1, 2)$
	Kernel Size	5
	Activation	Mish
	InvDyn Hidden Dim	256
Optimization	Optimizer	Adam
	Learning Rate	1×10^{-4}
	Discount Factor (γ)	1.0

Training uses a single GPU. Data collection is parallelized across four GPUs. Table 9 summarizes the wall-clock time for each key step of one ISGE iteration.

Table 9: Wall-clock time of one ISGE iteration. Steps above the double rule also appear in vanilla iterative training; steps below are ISGE-specific overhead.

Step	Description	Time
Policy Updating	Train DFUSER planner on $\mathcal{D}$	$\sim$0.2 h
Data Collection	Deploy DFUSER to collect $\mathcal{D}_{\text{gen}}$ (3024 samples)	$\sim$2 h
Judger Training	Train unconditional DFUSER on $\mathcal{D}$	$\sim$0.2 h
Explorer Training	Train $\text{DFUSER}_{\text{SC}}$ with dual conditioning	$\sim$0.2 h
Exploration Data Collection	Deploy $\text{DFUSER}_{\text{SC}}$ to collect $\mathcal{D}_{\text{exp}}$ (1680 samples)	$\sim$1.2 h
ISGE Overhead	Judger + Explorer + Exploration Collection	$\sim$1.6 h

As detailed in Table 9, we acknowledge that the Judge-Explore module naturally introduces additional computational overhead (approximately 1.6 hours per iteration). However, breaking down this overhead reveals two key insights. First, the extra *offline training cost* (training the Judger and Explorer) requires only about 0.4 hours. Because the underlying diffusion planner is relatively lightweight compared to modern Large Language Models, this offline training overhead is practically negligible in industrial practice. Second, while ISGE adds an extra *online* exploration step ($\sim$1.2 hours for data collection), it exhibits a highly favorable efficiency-performance tradeoff globally. As demonstrated in our main results (Table 1), a single iteration of ISGE already significantly surpasses the performance of the vanilla model's third iteration. While executing three iterations of the vanilla method takes over 6.6 hours of cumulative computation, it still falls significantly short of the performance ISGE achieves in a single 3.8-hour iteration.

C.4 Robustness Analysis

C.4.1 Robustness to More Iterations

In the main paper, we demonstrated that ISGE reaches near-oracle performance within 3 iterations. To investigate the long-term dynamics of our iterative framework, we extend the training process to 5 iterations on the AuctionNet-Sparse setting.

As detailed in Table 10, performance improves consistently and dramatically from Iteration 0 to Iteration 3. However, beyond Iteration 3, the performance largely plateaus with only minor statistical fluctuations (peaking at 21.1 ± 1.0 in Iter 3 and maintaining 20.8 ± 1.2 in Iter 5). In practice, this suggests that 3 iterations provide an optimal balance between performance maximization and computational efficiency.

Table 10: **Performance over extended iterations on AuctionNet-Sparse.** ISGE rapidly converges to the near-oracle plateau within 3 iterations and maintains stability thereafter.

Dataset	Cold-Start Base	Iter 1	Iter 2	Iter 3	Iter 4	Iter 5
AuctionNet-Sparse	15.7 ± 1.0	20.0 ± 0.9	20.6 ± 0.7	21.1 ± 1.0	20.4 ± 1.1	20.8 ± 1.2

Theoretical Analysis: The Breakdown of Frontier Alignment. This slight performance saturation can be elegantly explained through the lens of our geometric framework, specifically reflecting the gradual breakdown of the *Frontier Alignment* assumption (Assumption B.3). This assumption mechanically depends on the suboptimality of the current training support:

- Early Iterations (Base – Iter 2): The Suboptimal Support. Initially, the data manifold is narrowly supported and highly suboptimal. When trajectories reach the high-return frontier, the return gradient $\mathbf{g}_R$ collapses (Proposition 3.2). However, because the entire manifold is suboptimal, pointing "outwards" from this boundary via the scarcity gradient $\mathbf{g}_S$ inherently points towards the unexplored optimal region with high probability. Mathematically, the Frontier Alignment ($\langle \mathbf{g}_S, \mathbf{n}^* \rangle > 0$) holds rigorously, guaranteeing the efficacy of dual-guidance (Theorem 3.5) and yielding significant performance leaps.
- Late Iterations (Iter 3 – Iter 5): The Near-Optimal Support. By Iteration 3, the training manifold has expanded sufficiently to encompass the near-optimal regions. At this stage, the unconditional prior μ_θ has internalized a highly capable policy, serving as a reliable baseline. Crucially, because the manifold now already contains near-optimal strategies, pushing further outwards into deeper OOD regions is no longer guaranteed to be a universally better direction. The Frontier Alignment assumption gradually breaks down, meaning $\langle \mathbf{g}_S, \mathbf{n}^* \rangle \leq 0$ occurs with non-negligible probability. As a result, some generated OOD trajectories may reflect unlearnable or excessively risky patterns, leading to the observed plateau.

Therefore, such saturation is theoretically expected as the policy approaches the environmental ceiling. The primary value of ISGE lies precisely in its ability to quickly shatter the self-locking bottleneck and lift a low-quality cold-start model to this near-optimal plateau within just a few iterations.

C.4.2 Robustness to Extreme Cold-Start Settings

To further challenge our framework, we evaluate ISGE under an even more extreme cold-start scenario. Instead of the default $[10\%, 30\%]$ return quantile used in the main experiments, we restrict the initial training data strictly to the $[0\%, 10\%]$ return quantile on AuctionNet-Sparse. This creates an exceptionally poor initialization, placing the starting policy significantly further from the global optimum.

Table 11: **Robustness under Extreme Cold-Start ($[0\%, 10\%]$ quantile on AuctionNet-Sparse).** While Vanilla Self-Training is heavily self-locked, ISGE effectively bridges the massive gap to the Full-Data Oracle.

Method	Full-Data Oracle	Cold-Start Base	Iter 1	Iter 2	Iter 3
Vanilla Self-Training	20.5 ± 1.1	12.7 ± 0.9	14.8 ± 1.1	15.1 ± 0.6	15.0 ± 0.7
ISGE (Ours)			19.7 ± 0.8	20.2 ± 0.7	$\mathbf{20.6 \pm 1.0}$

As shown in Table 11, the Cold-Start Base model yields a severely degraded score of 12.7 ± 0.9. Under this extreme condition, Vanilla Self-Training completely fails to escape the low-quality region, becoming permanently self-locked around a score of 15.0 across all subsequent iterations. In contrast, ISGE demonstrates remarkable robustness. By merely Iteration 1, the scarcity guidance handoff propels the policy score to 19.7 ± 0.8. By Iteration 3, ISGE achieves 20.6 ± 1.0, successfully matching the Full-Data Oracle (20.5 ± 1.1). This result reinforces that ISGE's geometric exploration is resilient to the quality of the initial data support.

C.5 Experimental Setup and Extended Results for Sanity Checks

This section provides the detailed experimental setup, gradient computation formulations, and supplementary tabular data for the empirical sanity checks presented in Section 5.1 of the main text.

C.5.1 Experimental Setup and Data for Return Signal Collapse

Detailed Setup. To rigorously define the empirical proxy for the theoretical optimal direction $\mathbf{n}^*$ in OOD exploration, we utilize a Full-Data Oracle diffusion planner trained on the entire AuctionNet-Dense dataset. Because this oracle possesses full coverage of the offline data distribution, its return gradient reliably points toward empirically optimal regions. We evaluate this oracle return gradient, denoted as $\tilde{\mathbf{n}}^*$, on a subset of 1,613 samples exhibiting the lowest reconstruction errors under the Full-Data Oracle.

In contrast, the empirical return gradient $\mathbf{g}_R(\boldsymbol{x}_t, t)$ is evaluated using the Cold-Start Base diffusion planner, which was deliberately trained strictly on a narrow suboptimal support (the 10%–30% return quantile). During evaluation, we inject ground-truth forward noise to obtain the noisy state $\boldsymbol{x}_t$ at varying denoising steps t. The respective gradients are computed via their score estimates:

$$\tilde{\mathbf{n}}^*(\boldsymbol{x}_t, t) = \epsilon_R^{\text{Oracle}}(\boldsymbol{x}_t, t) - \epsilon_{\text{uncond}}^{\text{Oracle}}(\boldsymbol{x}_t, t), \tag{43}$$

$$\mathbf{g}_R(\boldsymbol{x}_t, t) = \epsilon_R^{\text{Vanilla}}(\boldsymbol{x}_t, t) - \epsilon_{\text{uncond}}^{\text{Vanilla}}(\boldsymbol{x}_t, t). \tag{44}$$

Quantitative Results. Table 12 reports the cosine similarity, radial signal (inner product), and collapse ratio across three return quantiles and denoising steps. All three metrics consistently degrade as the trajectory moves toward high-return (OOD) regions and as denoising progresses to later steps. For instance, in the high-OOD quantile at $t = 5$, the cosine similarity drops to 0.1151, the radial signal collapses to 0.0647, and 63.3% of trajectories exhibit near-zero similarity ($\cos < 0.1$), quantitatively confirming the signal collapse predicted in Section 3.

Table 12: **Diagnosis of Return Signal Collapse.** We report the cosine similarity $\cos(\tilde{\mathbf{n}}^*, \mathbf{g}_R)$, radial signal $\langle \tilde{\mathbf{n}}^*, \mathbf{g}_R \rangle$, and collapse ratio (% of trajectories with cosine alignment < 0.1) across return quantiles and three representative denoising steps ($t = 15, 10, 5$). All reported values are averages over 1613 trajectories.

	Cosine Similarity ↑			**Radial Signal ↑**			**Collapse Ratio ↓**		
Return Quantile	t=15	t=10	t=5	t=15	t=10	t=5	t=15	t=10	t=5
0% – 10% (Low-OOD)	0.6653	0.6916	0.6112	0.2486	0.2882	0.3443	0.0%	0.0%	1.2%
10% – 30% (In-Distribution)	0.5199	0.3923	0.2658	0.1938	0.1632	0.1497	5.0%	23.9%	45.3%
30% – 100% (High-OOD)	0.4566	0.2858	**0.1151**	0.1702	0.1183	**0.0647**	6.1%	29.5%	**63.3%**

C.5.2 Experimental Setup and Data for Optimization Handoff

Detailed Setup. To analyze the guidance handoff mechanism, we examine the top 50% highest-return gift samples discovered by the ISGE Explorer model (Return range: $[96.6, 296.4]$, totaling 840 samples, on AuctionNet-Dense). For these successfully explored high-return trajectories, accurately recovering the true forward noise ($\epsilon_{\text{pred}} \approx \epsilon_{\text{true}}$) at any given denoising step t corresponds to successfully moving toward a valid, high-return OOD optimum. Therefore, we define the ground-truth target gradient for trajectory recovery as the *true denoising gradient*:

$$\mathbf{g}^*(\boldsymbol{x}_t, t) = \epsilon_{\text{true}}(\boldsymbol{x}_t, t) - \epsilon_{\text{uncond}}^{\text{Explorer}}(\boldsymbol{x}_t, t). \tag{45}$$

We then evaluate the empirical return gradient $\mathbf{g}_R$ and the scarcity gradient $\mathbf{g}_S$ produced by the Explorer model on these same trajectories:

$$\mathbf{g}_R(\boldsymbol{x}_t, t) = \epsilon_R^{\text{Explorer}}(\boldsymbol{x}_t, t) - \epsilon_{\text{uncond}}^{\text{Explorer}}(\boldsymbol{x}_t, t), \tag{46}$$

$$\mathbf{g}_S(\boldsymbol{x}_t, t) = \epsilon_S^{\text{Explorer}}(\boldsymbol{x}_t, t) - \epsilon_{\text{uncond}}^{\text{Explorer}}(\boldsymbol{x}_t, t). \tag{47}$$

Quantitative Results. Table 13 details, for $\mathbf{g}_R$ and $\mathbf{g}_S$, the cosine similarity and inner product with $\mathbf{g}^*$, as well as their individual norms and mutual inner product $\langle \mathbf{g}_S, \mathbf{g}_R \rangle$. At the critical boundary-crossing stage ($t = 5$), we find $\cos(\mathbf{g}^*, \mathbf{g}_S) > 0$ and $\langle \mathbf{g}_S, \mathbf{g}_R \rangle < 0$, validating Assumptions B.3.

Table 13: **Gradient alignment and signal decomposition for high-return exploratory trajectories (Golden Gift Zone, Top 50%).** For the return gradient $\mathbf{g}_R$ and the scarcity gradient $\mathbf{g}_S$, we report their cosine similarity and inner product with the true denoising gradient $\mathbf{g}^*$, their individual norms and their mutual inner product. All reported values are averages over 840 trajectories.

Step	**Return Gradient $\mathbf{g}_R$**			**Scarcity Gradient $\mathbf{g}_S$**			
	$\cos(\mathbf{g}^*, \mathbf{g}_R)$	$\langle \mathbf{g}^*, \mathbf{g}_R \rangle$	$\|\mathbf{g}_R\|$	$\cos(\mathbf{g}^*, \mathbf{g}_S)$	$\langle \mathbf{g}^*, \mathbf{g}_S \rangle$	$\|\mathbf{g}_S\|$	$\langle \mathbf{g}_S, \mathbf{g}_R \rangle$
$t = 15$	0.1169	0.0820	0.6948	0.2946	0.2372	0.8057	0.3275
$t = 10$	0.1204	0.1094	0.9031	0.4700	0.4916	1.0466	0.1487
$\mathbf{t = 5}$	**0.0493**	**0.0578**	**1.1745**	**0.5255**	**0.8093**	**1.5406**	**-0.1139**

C.6 Implementation Details of Undirected Exploration Baselines

To evaluate the effectiveness of our proposed method, we compare it against two standard exploration strategies: Action Space Noise (ASN) [22] and Parameter Space Noise (PSN) [23]. We introduce these noises during the inference phase to simulate different levels of exploration behavior. The implementation details and hyperparameter tuning processes are described below.

Action Space Noise (ASN) ASN is a straightforward exploration strategy commonly used in reinforcement learning, where perturbation is directly applied to the agent's output action. In our automated bidding scenario, the action is represented by the bidding coefficient α. At each decision step t, after the model calculates the baseline bidding coefficient α_t, we add a noise term ϵ_{asn} sampled from a Gaussian distribution. Since bids must be non-negative, we apply a rectification operation:

$$\alpha'_t = \max(0, \alpha_t + \epsilon_{\text{asn}}), \quad \text{where } \epsilon_{\text{asn}} \sim \mathcal{N}(0, \sigma^2_{\text{asn}}) \tag{48}$$

This corresponds to the logic where the final calculated bid coefficient is perturbed before interacting with the auction environment.

Since ASN operates directly on the action value, its scale has a direct physical interpretation (i.e., modifying the bid multiplier). Consequently, a relatively larger noise scale is required to induce significant exploration. We performed a grid search for σ_{asn} over the set $\{0.01, 0.05, 0.1, 0.5, 1.0\}$. We observed that smaller scales had negligible impact on the bidding outcome, while larger scales led to unstable performance. We selected $\sigma_{\text{asn}} = 0.1$ as the optimal configuration for both AuctionNet-Dense and AuctionNet-Sparse.

Parameter Space Noise (PSN) PSN induces state-dependent exploration by perturbing the weights of the neural network during the planning phase. This approach allows the agent to "imagine" diverse trajectories consistent with the model's internal uncertainty.

We introduce noise into the Diffusion U-Net during the **inference-time planning** process. Before generating a trajectory, we inject Gaussian noise into the learnable parameters θ of the Temporal U-Net:

$$\tilde{\theta} = \theta + \epsilon_{\text{psn}}, \quad \text{where } \epsilon_{\text{psn}} \sim \mathcal{N}(0, \sigma^2_{\text{psn}}) \tag{49}$$

Crucially, our implementation utilizes a *transient injection* mechanism: the noise is applied solely for the current planning step's reverse diffusion loop and is immediately removed (weights are restored) afterwards. This ensures that the model parameters remain stable for subsequent evaluations, isolating the exploration effect to the current decision. The noise is applied only to the trajectory generation model (U-Net), leaving the Inverse Dynamics model untouched to preserve action execution precision.

Deep neural networks are highly sensitive to weight perturbations, as small changes are amplified through multiple layers of non-linear transformations. Therefore, the scale for PSN must be significantly smaller than that of ASN. We tuned σ_{psn} over the set $\{1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}, 5 \times 10^{-3}, 1 \times 10^{-2}\}$. Our experiments showed that $\sigma_{\text{psn}} > 10^{-2}$ caused trajectory collapse (e.g., generating invalid or negative budgets), rendering the planning infeasible. We selected $\sigma_{\text{psn}} = 1 \times 10^{-3}$ for AuctionNet-Dense and $\sigma_{\text{psn}} = 5 \times 10^{-4}$ for AuctionNet-Sparse to balance exploration diversity with trajectory validity.

Table 14: Summary of Baseline Implementation and Hyperparameters

Method	Injection Target	Noise Type	Search Candidates (σ)	Selected
ASN	Final Action (α)	Additive Gaussian	$\{0.01, 0.05, \mathbf{0.1}, 0.5, 1.0\}$	0.1
PSN	U-Net Weights (θ)	Additive Gaussian	{1e-4, **5e-4**, **1e-3**, 5e-3, 1e-2}	1e-3 (Dense), 5e-4 (Sparse)

C.7 Comparison with Transition-Level Curiosity Exploration

To provide a direct empirical comparison with recent exploration methods based on curiosity (ICM [12], PGR [24]), we adapt their transition-level exploration mechanism into our generative planning framework.

Formalization of Curiosity-Guided Exploration. A well-known challenge in curiosity-driven RL is that the forward prediction error can be dominated by unpredictable environmental stochasticity (e.g., the "noisy TV" problem). While original methods like ICM utilize a learned feature encoder to extract predictable representations, we align the implementation with our ISGE framework for a rigorous and fair comparison: we apply the identical *strategic masking* operation $\text{Mask}(\cdot)$ (as defined in Section 4) to filter out random environmental dimensions.

We train a forward dynamics model f_ψ to predict the masked next state given the masked current state and action. The transition-level curiosity score c_t is then defined as the forward prediction error on the strategic dimensions:

$$c_t = \| f_\psi(\text{Mask}(s_t), a_t) - \text{Mask}(s_{t+1}) \|_2^2 \tag{50}$$

Since our diffusion planner operates at the trajectory level, we aggregate these local signals to form a trajectory-level curiosity condition $C(\tau)$:

$$C(\tau) = \sum_{t=0}^{H-1} c_t \tag{51}$$

where H is the horizon length. Similar to our ISGE framework, we inject $C(\tau)$ alongside the return $R(\tau)$ as conditions during the diffusion process, utilizing classifier-free guidance to steer the generation toward high-return and high-curiosity regions. For fairness, we apply a guidance scale of 3.0 for the curiosity condition and set the target curiosity score to the normalized maximum curiosity score observed in the dataset.

Performance Comparison. We evaluate this Curiosity-guided baseline against ISGE on the AuctionNet-Dense. As shown in Table 15, ISGE significantly outperforms the transition-level curiosity approach.

Table 15: Performance comparison between Curiosity-guided exploration and ISGE on AuctionNet-Dense.

Iteration	Curiosity-Guided (Transition-level)	ISGE (Trajectory-level)
Iter 1	84.8 ± 2.6	$\mathbf{103.4 \pm 5.3}$
Iter 2	90.6 ± 5.4	$\mathbf{104.1 \pm 6.7}$
Iter 3	91.0 ± 6.6	$\mathbf{108.4 \pm 8.2}$

Correlation Analysis and Discussion. To gain a deeper understanding of the performance gap, we analyze the correlation between our trajectory-level scarcity score (reconstruction error) and the aggregated curiosity score across different return quantiles of the dataset in Iteration 1.

As presented in Table 16, while the two metrics exhibit moderate correlation overall, the correlation drops significantly in the highest return quartile (Pearson = 0.354). We attribute the superiority of ISGE to two fundamental reasons:

1. **Compounding Errors:** Aggregating local, transition-level curiosity scores amplifies estimation errors over the long horizon. Consequently, transition-level aggregation struggles to reliably identify trajectories that are simultaneously high-return and highly novel in the OOD region (as evidenced by the sharp drop in correlation for the top 25% samples).

Table 16: Correlation between Scarcity and Curiosity scores across return quantiles.

Return Quantile	Pearson Correlation	Spearman Correlation
All Data	0.516	0.466
0% - 25%	0.591	0.445
25% - 50%	0.551	0.423
50% - 75%	0.469	0.450
75% - 100% (High-Return)	**0.354**	**0.338**

2. **Architectural Alignment:** The transition-level forward model f_ψ operates independently of the diffusion planner. In contrast, our scarcity signal is derived directly from the reconstruction error of the diffusion Judger. This trajectory-level metric is inherently aligned with the sequence modeling nature of diffusion architectures, providing a more reliable and cohesive geometric guidance for the sampling process.

D Limitations and Future Work

First, ISGE is fundamentally designed as a *bootstrapping* mechanism to bridge the massive gap between severely suboptimal initialization (self-locking) and near-optimal performance. It is not intended for fine-grained asymptotic convergence. As demonstrated in Appendix C.4.1, once the policy reaches a near-optimal plateau, the *Frontier Alignment* assumption gradually breaks down. Pushing further into high-scarcity OOD regions no longer guarantees a positive gradient toward the global optimum. In scenarios where broad, expert-level data is already abundant, fine-tuning on existing AIGB [4, 33, 14] is both sufficient and more stable.

Second, the efficacy of our guidance handoff relies on the *local continuity* of the return landscape near the training manifold. While this assumption generally holds in continuous tasks like auto-bidding, it may fail in highly discontinuous or pathologically rugged environments (e.g., cryptographic or perfectly hashed reward functions), where outward exploration might yield pure noise. Adapting ISGE to topologically complex, discontinuous landscapes remains an open problem for future research.

Third, regarding architectural scope, the current ISGE iterative framework is specifically designed for diffusion planners, elegantly leveraging CFG to execute the geometric handoff. Extending the core philosophy of trajectory-level scarcity guidance to other generative decision-making paradigms, such as autoregressive Transformers, represents a highly promising direction for generalizing iterative self-improvement across broader decision-making architectures.

E Broader Impacts

Our research introduces a powerful generative exploration framework for auto-bidding, which carries significant economic and systemic implications for the online advertising ecosystem. We outline these broader impacts, particularly focusing on the multi-agent dynamics and market efficiency.

Democratization of Advanced Bidding Capabilities. In the current industrial landscape, sophisticated bidding strategies are often monopolized by large advertisers with massive historical data. The cold-start problem serves as a severe barrier to entry for small and medium-sized businesses (SMBs) or new business sectors. By demonstrating that near-optimal policies can be bootstrapped efficiently from narrowly supported, low-quality data, ISGE democratizes access to highly competitive bidding capabilities, fostering a fairer and more accessible market ecosystem.

Multi-Agent Implications and Market Efficiency. From a multi-agent perspective, the concept of "gift samples" essentially represents transitory arbitrage opportunities—high-return, low-competition bidding strategies that exist in the unexplored OOD region. If ISGE is adopted globally, and all bidders become efficient "gift sample" miners, the market dynamics will fundamentally shift. As multiple generative agents actively explore and converge upon these highly profitable regions, the "gift" margins will be rapidly competed away. Rather than leading to a collapse, game-theoretic principles suggest that this collective exploration drives the auction market toward a more competitive and allocatively efficient equilibrium. Consequently, the automated allocation of advertising impressions

will more accurately reflect the true underlying valuation of the advertisers, thereby improving global market efficiency and social welfare.

Co-Evolution of Platform Mechanism Design. The widespread deployment of generative, exploratory agents like ISGE will inevitably necessitate adaptations from the platform side. If all agents are aggressively exploring the OOD parameter space, the auction environment could experience increased volatility or non-stationary "bidding wars". To maintain stability, advertising platforms will need to evolve from passive auctioneers to active market designers. Future mechanisms may need to adapt by introducing smoothed pricing rules, dynamic reserve prices, or incentive-compatible information-sharing protocols that are explicitly designed to be robust against generative exploratory agents. This highlights a promising future research direction: the co-evolution of AI-generated bidding strategies and automated mechanism design.