Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

CATALIST: Structurally Guided Sequence Learning withModel-Based Augmentation for Offline Auto-Bidding

Liyu Huang
Jiangsu University
Haiyu Liu
Jingxiong Branch, Hebei Expressway Group Co., Ltd.
Jinghui Wang
wjinghui96@gmail.com

Jiangsu University

Research Article

Keywords:

Posted Date: November 4th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11412157/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

CATALIST: Structurally Guided Sequence Learning with Model-Based Augmentation for Offline Auto-Bidding

Liyu Huang[1], Haiyu Liu[2], Jinghui Wang[1*]

[1*]School of Mathematical Sciences, Jiangsu University, 301 Xuefu Road, Zhenjiang, 212013, Jiangsu, China.
[2]Jingxiong Branch, Hebei Expressway Group Co., Ltd., Xiong'an, Hebei, China.

*Corresponding author(s). E-mail(s): wjinghui96@gmail.com;
Contributing authors: 18770938501@163.com; 18913113295@163.com;

Abstract

Offline auto-bidding learns bidding policies from logged auction trajectories to optimize resource allocation under budget and cost-per-acquisition (CPA) constraints. Temporal dependencies among auction features and limited action coverage in the logs complicate policy learning. This paper proposes CATALIST, a structurally guided sequence-learning framework that combines dependency priors with local model-based augmentation. CATALIST constructs structured representations using lagged and instantaneous dependency priors and models decision histories through token-type and temporal attention biases. Alternative actions are generated within the same historical context and selected through resource-feasibility checks and model-predicted gain screening for action supervision. Training jointly incorporates action learning from observed and generated samples, outcome and value prediction using only observed records, and context-consistency regularization. On AuctionNet, CATALIST achieves a Score of 260.95 at the nominal budget, improving over CDT's 175.48 by 48.7%. At the 150% budget ratio, it achieves 339.24, exceeding GAS and GAVE, which obtain 294.13 and 304.32, by 15.3% and 11.5%, respectively. The experimental results show that combining structurally guided sequence learning with model-based augmentation improves offline auto-bidding performance.

Keywords: Auto-bidding, Structural Priors, Structured Attention, Model-Based Augmentation, Offline Sequential Decision Making

1 Introduction

Auto-bidding allocates a finite campaign budget across sequential auctions while pursuing value under business requirements such as cost per acquisition (CPA). A bid determines current participation and expenditure and changes the resources available for later opportunities. Auto-bidding is therefore a resource-constrained sequential decision problem that requires auction context, resource states, and historical outcomes. Offline learning uses existing auction records to train policies, providing a way to reduce training-time interaction with live auctions [35, 68].

Decision Transformer formulates policy learning as return-conditioned sequence modeling and predicts actions from sequences of states, actions, and returns [9]. In auto-bidding, history captures both resource evolution and changes in traffic: expenditure affects subsequent participation, while auction context affects opportunity value. The basic Decision Transformer can learn these interactions through sequence attention, but does not include the feature-dependency prior and explicit type/time bias introduced here. Feature dependencies describe relationships among coordinates within and across control intervals, whereas token interactions describe how return, state, and action tokens exchange information over the decision history. CATALIST encodes the former in state representations and the latter through token-type and temporal attention biases, making both levels explicit [47, 49].

A second challenge is the action coverage of offline data. Logs contain actions executed by the collection policy and their outcomes, rather than feedback for every alternative bid at the same history. Locally generated examples can supplement training information, but their predictions depend on observed data and a fitted model and can be inaccurate in poorly covered regions [35, 64]. Augmentation consequently needs to address candidate ranges, outcome-based screening, and generated labels without treating predicted alternatives as observed auction feedback. Structured representation and model-based augmentation act on the organization of historical information and the construction of training examples, respectively, connecting two aspects of offline bidding learning.

We propose CATALIST, which integrates structurally guided sequence learning and local model-based augmentation in one framework. The model combines lagged and instantaneous dependency priors, forms state representations through feature weighting and dependency aggregation, and adjusts autoregressive attention with token-type and temporal biases. At a common decision history, it constructs alternative actions from policy outputs and screens generated examples using empirical action ranges, predicted resource states, predicted return gains, and policy dispersion. Observed and generated examples supervise action prediction; outcome/value prediction and context consistency use observed records only.

The main contributions are:

1. We propose CATALIST, a structurally guided sequence-learning framework for offline auto-bidding. The framework combines a fixed dependency-informed state parameterization with learned token-type and temporal attention biases, separating relationships among auction-state features from interactions across the decision sequence.

2. We design a local model-based augmentation mechanism trained jointly with the structurally guided sequence model. Alternative actions are generated at the same decision history and screened using action ranges, resource feasibility, predicted gains, and policy dispersion. Joint training combines action supervision from observed and generated samples with outcome and value prediction and context-consistency regularization on observed records.
3. We evaluate CATALIST on AuctionNet through comparisons across budget ratios, backbone integrations, component ablations, and hyperparameter analyses. CATALIST achieves a Score of 260.95 at the nominal budget, improving over DT and GAVE by 65.4% and 10.2%, respectively. At the 150% budget ratio, it achieves 339.24, exceeding GAVE by 11.5%, demonstrating its performance across different budget conditions.

2 Related Work

2.1 Auto-Bidding Methods

Auto-bidding methods connect opportunity value to campaign-level resource allocation. Auction theory provides a reference for allocation and payment [59], while real-time bidding and cost-per-click optimization translate advertiser objectives into bid functions [68, 73]. Budget pacing and unified constrained bidding couple current expenditure with future opportunities and business requirements such as CPA [6, 24]. Reinforcement-learning approaches formulate this coupling as sequential decision making [61, 69]. More recent conditional diffusion, value-guided generation, and post-training search methods learn or refine bidding trajectories [18, 21, 36].

Offline learning fits policies from fixed auction records and limits additional training-time exploration [35]. BCQ and BEAR address action support and bootstrapping error [17, 32], whereas CQL and IQL provide conservative or implicit policy-improvement mechanisms [31, 33]. Behavior-regularized policy learning, TD3+BC, critic-weighted regression, and advantage-weighted fitting further study how policy updates remain connected to logged experience [15, 16, 40, 60]. Explicit stochastic-policy models provide action distributions [22], but policy dispersion alone does not quantify the error of a learned transition model. For bidding, a global action bound is consequently different from support conditional on campaign history. Sequence models provide an alternative interface for using that history. Decision Transformer and Trajectory Transformer model return-conditioned actions or trajectories [9, 26], while RvS and related analyses examine the role of supervised learning and return conditioning [7, 13]. Online adaptation, masked sequence models, and diffusion planning extend the range of inference and generation tasks [3, 8, 27, 70]. In auto-bidding, these choices interact with resource summaries, the target-return horizon, and the available action coverage. CATALIST follows the offline sequence-learning setting and studies structural history representations together with locally generated training examples.

2.2 Structurally Guided Sequence Modeling

Structural learning describes relationships that can guide representation. Granger-style prediction, nonlinear temporal association estimation, and contemporaneous/lagged structure methods concern different sources of dependency [19, 42, 47, 54]. NOTEARS, DAG-GNN, and nonparametric structure learning also differ in assumptions and score interpretation [66, 71, 72]. Potential-outcome, structural-causal, invariant-prediction, and causal-representation research distinguish such observational information from intervention identification [5, 43–46, 48, 49]. CATALIST uses estimated dependencies as representation priors. Graph convolution, neighborhood aggregation, graph attention, and graph Transformers provide mechanisms for incorporating relational information [23, 30, 57, 63]. Sequence attention and relative-position methods incorporate token interactions and temporal structure [12, 50, 56]. CATALIST separates these levels: a feature-level prior enters state encoding, while token-type and temporal biases enter autoregressive attention.

2.3 Model-Based Augmentation and Consistency Learning

Model-based learning supplies another source of training information. Probabilistic dynamics, model-based policy optimization, MOPO, MOReL, and COMBO investigate generated experience and prediction error through uncertainty-aware, pessimistic, or conservative constructions [11, 25, 29, 64, 65]. CATALIST constructs local alternative-action labels at an observed history instead of recursively expanding synthetic rollouts. It combines empirical action-range and resource-feasibility checks with predicted-gain and policy-dispersion screening, and reserves observed outcomes for predictor supervision.

Consistency methods regularize predictions across selected views, transformations, or interpolations. Mean Teacher, virtual adversarial training, UDA, FixMatch, Mixup, and interpolation consistency supply related ideas from supervised and semi-supervised learning [39, 51, 55, 58, 62, 67]. Ensemble, calibration, and dataset-shift studies distinguish confidence from reliable uncertainty, while multi-task weighting addresses losses with different scales [20, 28, 34, 41]. For bidding, output consistency must distinguish context noise from decision-relevant changes in budget or time. Constraint-aware policy optimization and safe sequential learning further motivate evaluating CPA outcomes directly [1, 10, 37]. A higher utility score need not imply fewer campaign-level violations. Offline benchmarks and statistical evaluation research also emphasize dataset definitions, independent training replicates, and uncertainty in finite-run comparisons [2, 14]. AuctionNet supplies the advertising-specific setting here [52], but the benchmark name does not determine the subset, evaluator, or replication protocol of a particular experiment. CATALIST combines these mechanisms by applying consistency to exogenous context views of observed histories and using accepted alternatives only for action supervision. Structural encoding organizes the history used by the policy and predictors, while local augmentation supplies additional action targets at that history.

3 Problem Formulation

The task is to learn a history-conditioned bidding policy from fixed campaign trajectories and evaluate its resource allocation under budget and CPA requirements. We first introduce a value-scaled bidding abstraction, then distinguish sequential campaign outcomes from offline sequence-training labels.

3.1 Budget- and CPA-Constrained Bidding

Consider a single-slot second-price reference model with opportunity value $v_i \geq 0$, bid $b_i \geq 0$, and competing price $m_i \geq 0$. Let $p_i \in \{0, 1\}$ be an explicit participation flag. Allocation and payment are

$$x_i = p_i \mathbf{1}[b_i \geq m_i], \qquad c_i = m_i x_i. \tag{1}$$

The skip command sets $p_i = 0$ regardless of price; a participating zero bid may win a zero-price opportunity. The static optimization below is over bid and participation choices. For initial budget $B \geq 0$ and target expenditure per unit value $C_v \geq 0$, the static allocation problem is

$$\begin{aligned} \max_{\{b_i, p_i\}} \ & \sum_i v_i x_i, \\ \text{s.t.} \ & \sum_i c_i \leq B, \\ & \sum_i c_i \leq C_v \sum_i v_i x_i. \end{aligned} \tag{2}$$

Here C_v denotes the target expenditure per unit value, and the static constraint uses C_v in place of the campaign CPA target C. If v_i represents acquisition probability, value is expected acquisitions and C_v can equal C. In general, value reward and realized acquisitions are recorded separately. The second-price abstraction defines the value-scaled bidding interface; campaign CPA uses the acquisition counts defined below [52].

For fixed nonnegative cost-constraint and budget-constraint multipliers λ and μ, winning contributes $(1+\lambda C_v)v_i - (\lambda+\mu)m_i$ to the relaxed objective. When $\lambda+\mu > 0$, the corresponding threshold is

$$b_i^{\text{relax}} = \frac{1+\lambda C_v}{\lambda+\mu} v_i. \tag{3}$$

This motivates a scalar bidding coefficient a_t, with $b_i = a_t v_i$ for positive-action participation in control interval t, subject to the common budget executor. The interval skip command sets $p_i = 0$; a positive action permits participation only while the executor allows it. Zero is an explicit nonparticipation command, rather than participation with a zero bid. Budget pacing and future opportunity uncertainty require a sequential policy [6].

3.2 Sequential Resource Allocation

A campaign contains H control intervals indexed by $t = 0, \ldots, H-1$. Before choosing a_t, the policy receives s_t, containing resource summaries, accumulated outcomes, and available traffic information. Opportunities in interval t form $\mathcal{I}_t$. Under the reference model, interval value and expenditure are

$$r_t = \sum_{i \in \mathcal{I}_t} v_i x_i, \qquad d_t = \sum_{i \in \mathcal{I}_t} c_i. \tag{4}$$

These outcomes occur after the action. The next state reflects realized expenditure and elapsed time; in the accounting abstraction, $B_{t+1} = B_t - d_t$. Resource variables are decision-relevant: changing current expenditure changes the resources available to later actions.

Let n_t be the realized acquisitions recorded in interval t, separately from the value reward r_t. Campaign spend and acquisitions are $S^{\text{spend}} = \sum_t d_t$ and $A = \sum_t n_t$. The CPA target C has units of expenditure per acquisition. A campaign violates the target when $S^{\text{spend}} > CA$; positive spend with zero acquisitions counts as a violation, while zero spend with zero acquisitions does not. The expected resource-allocation objective is

$$\begin{aligned} &\max_{\pi} \mathbb{E}_{\pi}\left[\sum_{t=0}^{H-1} r_t\right], \\ &\text{s.t. } \mathbb{E}_{\pi}\left[\sum_{t} d_t\right] \le B, \\ &\mathbb{E}_{\pi}\left[\sum_{t} (d_t - C n_t)\right] \le 0. \end{aligned} \tag{5}$$

This formulation describes the resource-allocation goal. CATALIST learns history-conditioned actions through supervised objectives: return conditions guide value-seeking action prediction, the common executor enforces the available budget, and candidate screening uses predicted spend and acquisitions to control CPA excess. Observed outcome and value losses train the predictors used in screening. Evaluation reports value, penalized Score, campaign-level CPA violations, and budget utilization [4]. A state-sufficiency assumption would concern its information about future transitions conditional on the past, not independence of adjacent states; history conditioning can retain information omitted by a compact state summary [53].

3.3 Offline Sequence Learning

Let $\mathcal{D} = \{\tau_n\}_{n=1}^{N}$ be a fixed dataset of campaign trajectories, where τ_n contains (s_t, a_t, r_t, s_{t+1}) records and a terminal marker. At decision t, the observed prefix is

$$u_t = (s_0, a_0, r_0, \ldots, s_{t-1}, a_{t-1}, r_{t-1}, s_t). \tag{6}$$

The decision prefix contains states up to s_t and actions and rewards up to interval $t-1$. For a window of K control steps, let $\ell_t = \max(0, t-K+1)$ and denote its physical history by $u_t^{(K)} = (s_{\ell_t}, a_{\ell_t}, r_{\ell_t}, \ldots, s_{t-1}, a_{t-1}, r_{t-1}, s_t)$; when $\ell_t = t$, this is simply s_t.

Training return labels use the complete logged reward sequence up to the campaign endpoint T: $T = H$ for a full-horizon campaign and $T < H$ for a recorded terminal campaign. The endpoint is shared by all labels in that trajectory; a history window limits the input length, not the return horizon. For $t < T$, define

$$R_t^{(T)} = \sum_{j=t}^{T-1} \gamma^{j-t} r_j, \qquad 0 < \gamma \leq 1. \tag{7}$$

Return labels contain future training outcomes, but they are conditioning targets rather than runtime observations. A return-conditioned sequence policy has the interface

$$p_\psi(a_t \mid u_t^{(K)}, \mathcal{R}_t^{(K)}), \tag{8}$$

where $\mathcal{R}_t^{(K)} = (R_{\ell_t}^{(T)}, \ldots, R_t^{(T)})$ denotes the return conditions aligned to the same control indices as the history window. We write $R_t = R_t^{(T)}$ below. Labels satisfy $R_j = r_j + \gamma R_{j+1}$ with $R_T = 0$. Records without a known campaign endpoint are excluded from value supervision and augmentation. At deployment, aligned conditions are reconstructed from the desired current return and observed past rewards, as specified in Section 4.4. During training, known logged actions provide supervision, while current action targets and future physical observations are excluded from their prediction inputs [9].

At deployment, the desired return is a policy-conditioning input, whereas the realized future campaign return is unavailable. The distinction is also important for augmentation: replacing an action changes its candidate outcome, so generated action labels and their return conditions require a declared continuation and reconstruction rule.

4 Methodology

CATALIST combines structurally guided sequence modeling with local model-based augmentation. It predicts bids from historical auction information and supplements policy training with screened alternative actions. Figure 1 illustrates the overall framework.

4.1 Structural Priors and Sequence Attention

CATALIST incorporates structure at two levels: feature dependencies guide state encoding, while token-type and temporal biases adjust attention over the decision history. Let $x_t \in \mathbb{R}^{d_s}$ denote the standardized state. We estimate lagged and

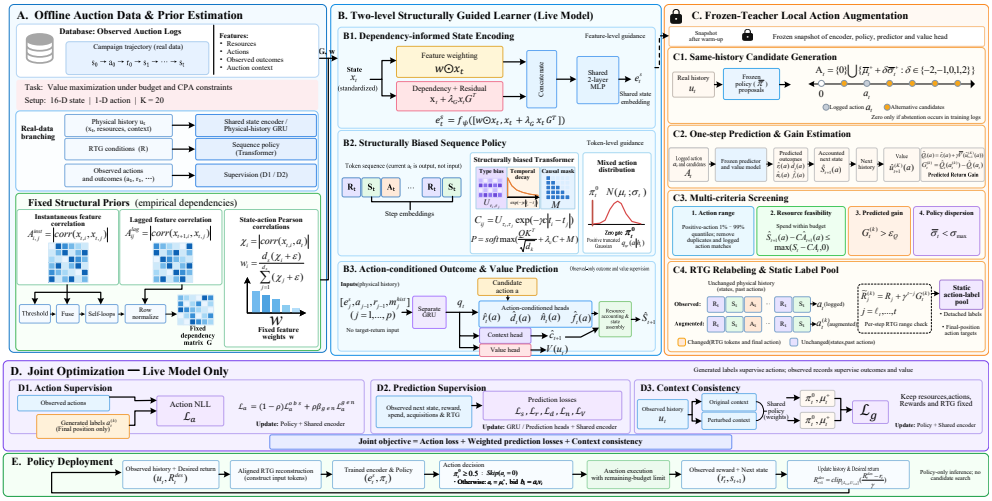


Fig. 1 CATALIST architecture for structurally guided sequence learning, frozen-teacher local action augmentation, joint optimization, and policy deployment.

instantaneous dependencies from observed training trajectories:

$$\begin{aligned} A^{\text{lag}}_{ij} &= |\operatorname{corr}(x_{t+1,i}, x_{t,j})|, \\ A^{\text{inst}}_{ij} &= |\operatorname{corr}(x_{t,i}, x_{t,j})|. \end{aligned} \tag{9}$$

Here corr is Pearson correlation, and lagged pairs are consecutive observations from the same campaign. For $q \in \{\text{lag}, \text{inst}\}$, retain off-diagonal scores above threshold θ_G as $G^q_{ij} = A^q_{ij}\mathbf{1}[A^q_{ij} \geq \theta_G]$, with $G^q_{ii} = 0$. The two matrices are fused, augmented with self-loops, and row-normalized:

$$\begin{aligned} A &= \alpha_G G^{\text{lag}} + (1-\alpha_G) G^{\text{inst}}, \\ G &= D^{-1}(A+I), \quad D_{ii} = \sum_j (A+I)_{ij}. \end{aligned} \tag{10}$$

The coefficient $\alpha_G \in [0,1]$ balances cross-time and within-step dependencies. Row i aggregates information from source features j. These correlations provide dependency priors rather than identified causal relations; the estimates remain fixed during training.

To connect the prior to the state representation, we combine action-associated feature weighting with dependency aggregation. Define $\chi_i = |\operatorname{corr}(x_{t,i}, a_t)|$ and

$$w_i = \frac{d_s(\chi_i + \varepsilon)}{\sum_j (\chi_j + \varepsilon)}, \tag{11}$$

where $\varepsilon = 10^{-6}$. The weights have mean one and reflect associations with logged actions. The state embedding is

$$z_t = x_t G^\top, \qquad e_t^s = f_\psi([w \odot x_t,\ x_t + \lambda_G z_t]). \tag{12}$$

Here $\odot$ denotes elementwise multiplication, λ_G controls aggregation strength, and f_ψ is a two-layer MLP with GELU activation and hidden/output dimension d_h. The first branch weights individual features; the second combines each feature with its dependency-weighted neighbors. Both branches use the current state, with the lagged matrix supplying a cross-time prior. In bidding, resource consumption and changing traffic are represented by related state features; this parameterization makes their empirical dependencies explicit at the encoder input. It does not expand the function class of an unrestricted MLP of the same width. Its role is to organize feature information, whereas the learned type/time bias below directly modifies interactions between sequence tokens.

The return-conditioned token ordering and causal mask follow Decision Transformer [9]; the additions here are the dependency-informed state encoder and learned type/time bias. The state embeddings enter a decision sequence ordered as (R_j, S_j, A_j) at each control step. Return and action tokens use linear embeddings after scaling by $c_R = \max(\mathrm{std}_{\mathrm{tr}}\, R_t, \varepsilon)$ and $c_a = \max_{\mathrm{tr}}\, a_t > 0$, respectively; state tokens use e_j^s. Tokens from the same step share a learned step embedding. To distinguish token interactions and their temporal separation, we add

$$C_{ij} = U_{\tau_i, \tau_j} \exp(-\gamma_C |t_i - t_j|) \tag{13}$$

to the attention logits. The trainable matrix $U \in \mathbb{R}^{3\times 3}$ encodes interactions among return, state, and action tokens, is initialized to zero, and is shared across attention layers. The fixed coefficient $\gamma_C \geq 0$ attenuates the bias with temporal distance. Attention becomes

$$P_{ij} = \mathrm{softmax}_j\left(\frac{q_i^\top k_j}{\sqrt{d_k}} + \lambda_C C_{ij} + M_{ij}\right), \tag{14}$$

where λ_C controls bias strength. The mask M_{ij} is zero for valid keys with $j \leq i$ and $-\infty$ otherwise; padded queries are excluded. For decision t, the input ends at S_t, and its representation predicts a_t. Thus the model uses prior returns, states, and actions while excluding the current action target and future observations. The bias modulates type-specific interactions; it does not guarantee larger attention weights for nearby tokens.

4.2 Bidding Policy and Outcome Prediction

The sequence representation supports two complementary tasks: predicting a bidding action and evaluating its possible outcomes. The policy conditions on the decision history and target-return tokens, $h_t = (u_t^{(K)}, \mathcal{R}_t^{(K)})$. To represent both abstention and

positive bids, we combine a point mass at zero with a Gaussian density truncated to $(0, c_a]$:

$$\begin{aligned} p_\psi(da \mid h_t) &= \pi_t^0 \delta_0(da) + (1 - \pi_t^0) q_\psi(a \mid h_t)\, da, \\ q_\psi(a \mid h_t) &= \frac{\phi((a - m_t)/s_t^a)}{s_t^a Z_t}, \quad 0 < a \le c_a, \\ Z_t &= \Phi((c_a - m_t)/s_t^a) - \Phi(-m_t/s_t^a). \end{aligned} \tag{15}$$

Here π_t^0 is the abstention probability, δ_0 is a point mass at zero, and ϕ and Φ are the standard Gaussian density and distribution functions. Three linear heads produce b_t^0, b_t^m, b_t^s, parameterized as

$$\begin{aligned} \pi_t^0 &= \mathrm{sigm}(b_t^0), \qquad m_t = c_a \,\mathrm{sigm}(b_t^m), \\ s_t^a &= c_a[10^{-3} + (1 - 10^{-3})\,\mathrm{sigm}(b_t^s)]. \end{aligned} \tag{16}$$

The conditional positive-action mean and variance are $\mu_t^+ = \mathbb{E}_{q_\psi}[a]$ and $(\sigma_t^+)^2 = \mathrm{Var}_{q_\psi}(a)$. The full policy moments are

$$\begin{aligned} \mu_t &= (1 - \pi_t^0)\mu_t^+, \\ \sigma_t^2 &= (1 - \pi_t^0)[(\sigma_t^+)^2 + (\mu_t^+)^2] - \mu_t^2. \end{aligned} \tag{17}$$

These quantities characterize the action proposals and their dispersion in Section 4.3. At deployment, the policy outputs zero if $\pi_t^0 \ge 0.5$ and μ_t^+ otherwise.

To evaluate candidate actions, a separate prediction branch shares the state encoder f_ψ and summarizes physical history with a GRU. Its inputs are the state embedding, previous action, previous reward, and a history-availability indicator:

$$q_t = \mathrm{GRU}_\theta\big(\{[e_j^s, a_{j-1}/c_a, r_{j-1}/c_r, m_j^{\mathrm{hist}}]\}_{j=\ell_t}^t\big)\,. \tag{18}$$

At the first step of the window, previous inputs and the indicator are zero; later indicators are one. This branch excludes target-return tokens, so outcome predictions depend on observed history and the candidate action rather than the desired return. Let $\zeta_t(a) = \mathbf{1}[a > 0 \wedge B_t > 0]$ indicate participation with available budget. Action-conditioned heads predict reward, expenditure, and acquisitions:

$$\begin{aligned} \hat{r}_t(a) &= \zeta_t(a) c_r \,\mathrm{softplus}(g_r(q_t, a/c_a)), \\ \hat{d}_t(a) &= \zeta_t(a) B_t \,\mathrm{clip}_{[0,1]}(g_d(q_t, a/c_a)), \\ \hat{n}_t(a) &= \zeta_t(a) c_n \,\mathrm{softplus}(g_n(q_t, a/c_a)). \end{aligned} \tag{19}$$

Here $c_r = \max(\mathrm{std_{tr}}\, r_t, \varepsilon)$, $c_n = \max(\mathrm{std_{tr}}\, n_t, \varepsilon)$, and $\mathrm{clip}_{[0,1]}(z) = \min(1, \max(0, z))$. For zero action or zero budget, $\zeta_t(a) = 0$ makes all three outcomes exactly zero; for participating actions, Softplus gives nonnegative reward/acquisition predictions and expenditure is bounded by B_t. The next context is predicted as $\hat{c}_{t+1} = g_{\mathrm{ctx}}(q_t)$, shared by candidates at the same history. The win fraction is $\hat{f}_t(a) = \zeta_t(a)\,\mathrm{sigm}(g_{\mathrm{win}}(q_t, a/c_a))$.

The next state combines these predictions with deterministic resource updates:

$$\begin{aligned}(\hat{B}_{t+1}, \hat{S}_{t+1}) &= (B_t - \hat{d}_t,\ S_t + \hat{d}_t),\\ (\hat{V}^{\mathrm{cum}}_{t+1}, \hat{A}_{t+1}) &= (V^{\mathrm{cum}}_t + \hat{r}_t,\ A_t + \hat{n}_t),\\ \hat{s}_{t+1}(a) &= \Pi(s_t, \hat{d}_t, \hat{r}_t, \hat{n}_t, \hat{c}_{t+1}, \hat{f}_t).\end{aligned} \tag{20}$$

Here $S_t, V^{\mathrm{cum}}_t, A_t$ are cumulative expenditure, value, and acquisitions, respectively. The state-construction map Π applies these updates, advances the control step, inserts predicted context and the latest interval statistics, retains campaign constants, and recomputes derived ratios. Context predictions are converted to physical units with the training-state scales before assembly. The same training-state standardization maps $\hat{s}_{t+1}$ to $\hat{x}_{t+1}$ for the state loss.

A value head estimates the remaining discounted return from the same physical-history representation:

$$V_{\psi,\theta}(u^{(K)}_t) = c_R \,\mathrm{softplus}(g_V(q_t)). \tag{21}$$

The outcome and value heads are supervised only by observed records. For each candidate, $\hat{u}^{(K)}_{t+1}(a) = \mathrm{Tail}_K(u^{(K)}_t, a, \hat{r}_t(a), \hat{s}_{t+1}(a))$ appends its action and predicted transition, then retains the most recent K control steps. The GRU encodes this window for continuation-value estimation. This predicted window is used only for candidate scoring; exhausted-budget or known terminal next states have zero continuation value. The value head estimates logged return conditional on the available history window, providing a model-based score for the augmentation procedure in Section 4.3.

4.3 Local Action Augmentation

Logged trajectories provide only one executed action at each decision history. To supplement action supervision, CATALIST generates alternatives at the same history and retains candidates with favorable model-predicted outcomes. After training on observed records, the policy, outcome models, value head, and shared encoder are frozen as a teacher $(\bar{\psi}, \bar{\theta})$.

Candidate actions are constructed around the teacher's conditional positive-action mean, with offsets determined by its standard deviation:

$$\mathcal{A}_t = \{0\} \cup \{\bar{\mu}^+_t + \delta\bar{\sigma}^+_t : \delta \in \{-2, -1, 0, 1, 2\}\}. \tag{22}$$

The positive candidates must lie within the training-action interval $[Q_{0.01}(a \mid a > 0), Q_{0.99}(a \mid a > 0)]$. Zero is included only when abstention is observed in the training data. Duplicate candidates and matches to the logged action are removed.

Predicted-return-gain screening compares each remaining candidate with the logged action using the same frozen outcome and value models:

$$\begin{aligned}\hat{Q}_t(a) &= \hat{r}_t(a) + \gamma\bar{V}(\hat{u}^{(K)}_{t+1}(a)),\\ G^{(k)}_t &= \hat{Q}_t(a^{(k)}_t) - \hat{Q}_t(a_t).\end{aligned} \tag{23}$$

Here $\hat{u}_{t+1}^{(K)}(a)$ is the predicted next-history window defined in Section 4.2. A candidate is retained when its gain exceeds ϵ_Q, the teacher's full-policy dispersion satisfies $\bar{\sigma}_t < \sigma_{\max}$, and its predicted outcome passes resource checks. These checks bound expenditure by B_t and prevent an increase in the positive part of cumulative CPA excess:

$$\hat{S}_{t+1}(a) - C\hat{A}_{t+1}(a) \leq \max(S_t - CA_t, 0). \tag{24}$$

The global quantile interval limits action magnitude but does not establish support conditional on the current history; policy dispersion describes the action distribution, not outcome-model uncertainty. The gain $G_t^{(k)}$ measures predicted discounted raw value, not CPA-penalized Score. Selection therefore combines range restrictions, predicted-return gain, dispersion control, and predicted resource feasibility, without guaranteeing improved realized Score or CPA compliance.

For each retained candidate, the predicted gain adjusts the observed return conditions across the history window:

$$\widetilde{R}_j^{(k)} = R_j + \gamma^{t-j} G_t^{(k)}, \qquad j = \ell_t, \ldots, t. \tag{25}$$

The factor γ^{t-j} aligns the adjustment with each earlier return token. The physical history remains unchanged; only the return conditions and final action label are replaced. Reconstructed return tokens must remain within the training-return range at their respective control indices. The resulting sample is

$$\begin{aligned} \xi_t^{(k)} &= (\widetilde{h}_t^{(k)}, a_t^{(k)}), \\ \widetilde{h}_t^{(k)} &= (u_t^{(K)}, \widetilde{\mathcal{R}}_t^{(K,k)}). \end{aligned} \tag{26}$$

Accepted samples form a static augmentation pool used for action supervision in joint training. Predicted states, rewards, and values are not used as outcome or value targets, and generated samples do not become new generation roots. If no candidate is accepted, training uses observed records only.

4.4 Joint Training and Inference

CATALIST jointly trains the bidding policy and prediction models with action supervision, observed-outcome prediction, and context-consistency regularization. For the mixed policy in Section 4.2, the action loss distinguishes abstention from positive bids:

$$\ell_a(a, h) = \begin{cases} -\log \pi_\psi^0(h), & a = 0, \\ -\log(1 - \pi_\psi^0(h)) - \log q_\psi(a \mid h), & a > 0. \end{cases} \tag{27}$$

Observed and augmented samples contribute to the policy objective as

$$\begin{aligned} \mathcal{L}_a = (1-\rho)\mathbb{E}_{\text{obs}}\ell_a(a_t, h_t) \\ + \rho\beta_{\text{gen}}\mathbb{E}_{\text{gen}}\ell_a(a_t^{(k)}, \widetilde{h}_t^{(k)}). \end{aligned} \tag{28}$$

Here ρ controls the contribution of augmented samples and β_{gen} weights their labels. Each group loss is averaged separately.

The prediction models use observed records only. Their losses supervise the next state, reward, expenditure, acquisitions, and discounted return:

$$\begin{aligned}
\mathcal{L}_s &= \mathbb{E}_{\text{obs}_s} \frac{\|\hat{x}_{t+1} - x_{t+1}\|_2^2}{d_s}, \\
\mathcal{L}_r &= \mathbb{E}_{\text{obs}} \left(\frac{\hat{r}_t - r_t}{c_r} \right)^2, \\
\mathcal{L}_d &= \mathbb{E}_{\text{obs}_d} \left(\frac{B_t g_d(q_t, a_t/c_a) - d_t}{c_d} \right)^2, \\
\mathcal{L}_n &= \mathbb{E}_{\text{obs}} \left(\frac{\hat{n}_t - n_t}{c_n} \right)^2, \\
\mathcal{L}_V &= \mathbb{E}_{\text{obs}_V} \left(\frac{V(u_t^{(K)}) - R_t}{c_R} \right)^2.
\end{aligned} \tag{29}$$

The scales are those defined in Sections 4.1–4.2, with $c_d = \max(\text{std}_{\text{tr}}\, d_t, \varepsilon)$. Each loss uses records with the corresponding observed target; expenditure supervision uses positive-budget, positive-action records without mandatory skips. It acts on the pre-clipping output to retain corrective gradients. Generated outcomes are excluded from these prediction losses.

To regularize the policy against small traffic-context variations, we perturb continuous context features while keeping resource variables, actions, rewards, and return conditions fixed. For context coordinate i with training standard deviation above ε, sample $\nu_{j,i} \sim \mathcal{N}(0, \kappa^2/(1+w_i)^2)$, add it to $x_{j,i}$, and constrain the perturbed input to its valid range. The resulting view $\widetilde{h}_t^{\text{ctx}}$ is matched to the original through

$$\begin{aligned}
\mathcal{L}_g = \mathbb{E}_{\text{obs},\nu} \Bigg[& (\pi_\psi^0(h_t) - \pi_\psi^0(\widetilde{h}_t^{\text{ctx}}))^2 \\
& + \left(\frac{\mu_\psi^+(h_t) - \mu_\psi^+(\widetilde{h}_t^{\text{ctx}})}{c_a} \right)^2 \Bigg].
\end{aligned} \tag{30}$$

This term penalizes changes in both abstention probability and positive-action mean. The overall objective is

$$\begin{aligned}
\mathcal{L} = \mathcal{L}_a &+ \eta_s \mathcal{L}_s + \eta_r \mathcal{L}_r \\
&+ \eta_d \mathcal{L}_d + \eta_n \mathcal{L}_n + \eta_V \mathcal{L}_V + \eta_g \mathcal{L}_g,
\end{aligned} \tag{31}$$

with nonnegative auxiliary weights. Training first uses observed records with $\rho = 0$, then incorporates the static augmentation pool from Section 4.3. Policy and prediction losses update their respective branches and the shared encoder; the frozen teacher and

generated labels receive no gradients. If the pool is empty, training remains observed-only.

At inference, the initial desired return is selected by validation from training-campaign return quantiles $\{Q_{0.5}(R_0), Q_{0.75}(R_0), Q_{0.9}(R_0)\}$. After observing reward r_t, the next target is updated and projected to the training-return range at that step:

$$R_{t+1}^{\text{des}} = \text{clip}_{[L_{t+1}, U_{t+1}]} \left(\frac{R_t^{\text{des}} - r_t}{\gamma} \right). \tag{32}$$

Here L_j, U_j are the training-return minimum and maximum at control step j. Before each policy call, the return tokens in the history window are reconstructed from realized past rewards and the current target:

$$R_{j|t}^{\text{in}} = \sum_{k=j}^{t-1} \gamma^{k-j} r_k + \gamma^{t-j} R_t^{\text{des}}, \quad \ell_t \le j \le t. \tag{33}$$

This reconstruction preserves the discounted-return recursion within the window. The policy then produces the action using the rule in Section 4.2, without access to future outcomes.

5 Experiments

This section compares CATALIST across budget conditions, component configurations, backbone integrations, and hyperparameter settings. Relative improvements are calculated from the experimental results.

5.1 Experimental Setup

Dataset and budget settings. The experiments use a subset of the standard AuctionNet benchmark [52]. The state and action dimensions are 16 and 1, respectively. Each bidding campaign contains 48 control steps, and the policy takes a historical sequence of length 20 as input. To compare performance under different budget conditions, the evaluation budget is set to 50%, 75%, 100%, 125%, and 150% of the nominal budget.

Baselines and implementation settings. The compared methods comprise three groups: behavior cloning (BC), offline reinforcement learning methods IQL [31], BCQ [17], and CQL [33], and sequence/generative methods DT [9], CDT [37], GAS [36], and GAVE [18]. The overall performance comparison uses DT as the base backbone of CATALIST. The backbone integration experiment further combines CATALIST with DT, CDT, GAS, and GAVE individually and compares each integrated configuration with its original backbone. The computational platform is a dual-node NVIDIA H100 GPU cluster. Table 1 lists the main training settings.

Metrics and interpretation. Tables 2 and 3 report the original performance metric; the component comparison reports Score and the original CPA diagnostic (%). Comparisons use the same metric within each table. Campaign value is

Table 1 Experimental parameter settings.

Hyperparameters	**Value**
Batch size	128
Number of steps	400000
Sequence length	20
Learning rate	1.00E-04
Hidden size	64
Number of attention layers	3
Number of heads	1
Optimizer	AdamW [38]
Weight decay	1.00E-04
Gamma	0.99
graph_strength	0.5
cf_aug_start	50000
cf_no_search_action	5

Table 2 Reported performance on standard AuctionNet at different evaluation budget ratios. Best values within each budget are bold. See Section 5.1 for metric interpretation.

Dataset: AuctionNet.

Method	50%	75%	100%	125%	150%
BC	93.73	109.96	120.84	128.92	134.66
IQL	95.12	113.37	124.56	133.32	140.42
BCQ	69.47	96.42	125.83	144.85	168.29
CQL	90.39	110.90	128.48	143.68	155.46
DT	99.92	132.78	157.78	180.10	196.46
CDT	99.59	141.37	175.48	209.84	235.30
GAS	123.65	175.94	235.58	271.20	294.13
GAVE	**151.49**	193.13	236.83	279.30	304.32
CATALIST (DT)	148.48	**203.65**	**260.95**	**301.57**	**339.24**

$V_n = \sum_i x_{n,i} v_{n,i}$, and penalized Score is $S_n = V_n \min_j p_{n,j}$, where $p_{n,j}$ is a constraint penalty. The CPA diagnostic is reported separately from the explicitly defined campaign violation frequency in the constraint analysis. The nominal-budget CATALIST result is 260.95 throughout the main comparison, component comparisons, and constraint analysis. Reported error terms and shaded deviations summarize variation; the comparisons do not claim statistical significance.

5.2 Overall Performance and Backbone Integration

Figure 2 visualizes the budget-dependent ranking and the comparison at the tightest budget.

Table 2 shows that CATALIST has the highest reported performance at four of the five budget ratios, from 75% to 150%. At nominal budget, it reaches 260.95,

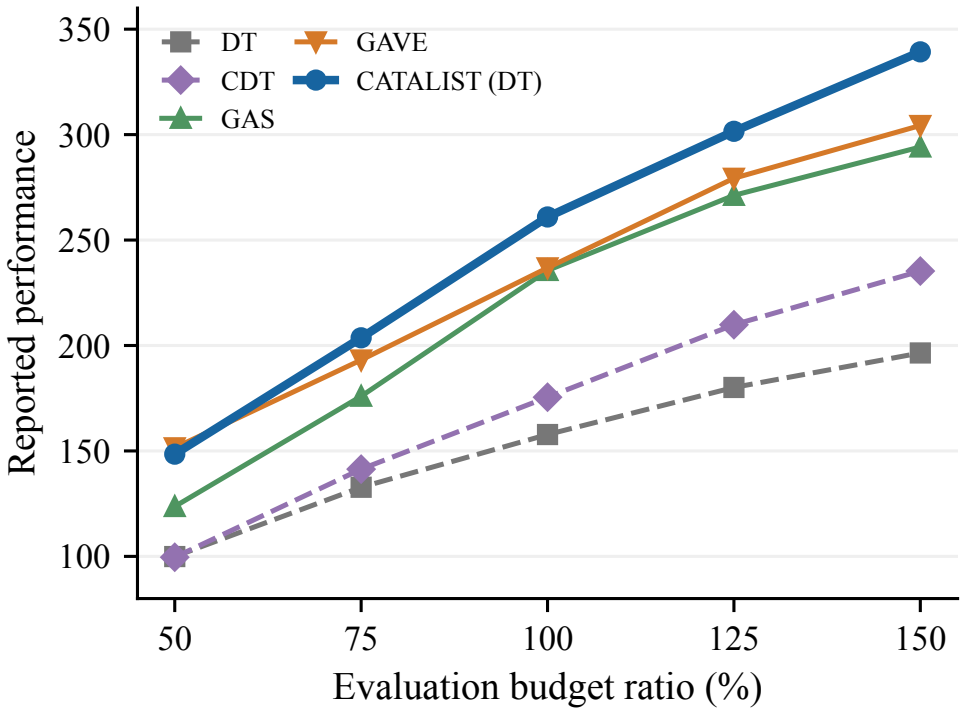


Fig. 2 Bidding performance under different evaluation budgets.

compared with 157.78 for DT and 236.83 for GAVE, the strongest listed alternative. The relative improvements are 65.4% and 10.2%, respectively. Against GAVE, the relative differences across the five ratios are −2.0%, 5.4%, 10.2%, 8.0%, and 11.5%. Thus the advantage is present across the nominal and higher-budget conditions rather than being confined to one operating point.

The 50% setting identifies a boundary of this comparison: GAVE reaches 151.49, exceeding CATALIST's 148.48. The complete CATALIST configuration therefore does not dominate every tested resource condition. At 150%, CATALIST reaches 339.24, exceeding GAS and GAVE by 15.3% and 11.5%. This budget-dependent ranking motivates reporting the full set of conditions instead of only the nominal-budget result. It does not by itself identify whether the differences arise from budget utilization, action selection, or evaluator penalties.

Backbone integration.

Figure 3 summarizes the relative gains across backbones and budget ratios. All displayed gains are positive, while their magnitudes differ across configurations.

Table 3 asks whether the favorable performance pattern extends beyond the DT backbone. Every displayed backbone/budget pair has a higher point estimate after integration. At nominal budget, the relative increases are 65.4% for DT, 48.1% for CDT, 16.4% for GAS, and 24.5% for GAVE. Improvements therefore appear for both

Table 3 Backbone-specific integration results under different budget ratios. Bold identifies the higher member of each paired comparison; the rows do not isolate equal-cost component effects.

Dataset: AuctionNet.

Method	50%	75%	100%	125%	150%
DT: Baseline	99.92	132.78	157.78	180.10	196.46
DT: CATALIST	**148.48**	**203.65**	**260.95**	**301.57**	**339.24**
CDT: Baseline	99.59	141.37	175.48	209.84	235.30
CDT: CATALIST	**152.07**	**211.57**	**259.85**	**313.32**	**345.40**
GAS: Baseline	123.65	175.94	235.58	271.20	294.13
GAS: CATALIST	**155.62**	**216.85**	**274.15**	**323.11**	**359.79**
GAVE: Baseline	151.49	193.13	236.83	279.30	304.32
GAVE: CATALIST	**167.07**	**235.36**	**294.79**	**339.26**	**378.77**

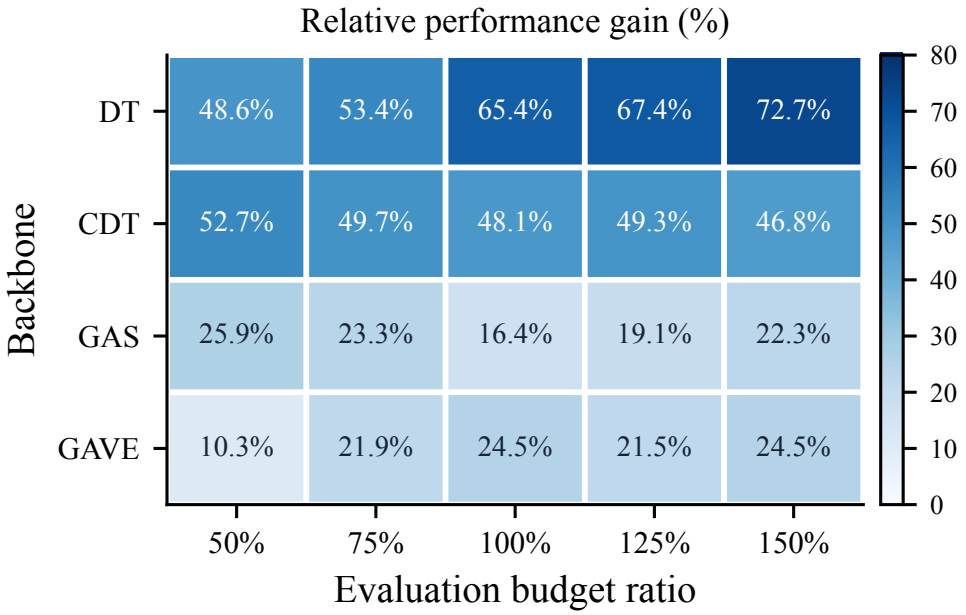


Fig. 3 Relative performance gains across policy backbones and evaluation budgets.

return-conditioned Transformer baselines and the domain-specific generative baselines in the reported configurations.

Relative improvement and final performance answer different questions. DT has the largest percentage increase at nominal budget, whereas integrated GAVE has the highest final value, 294.79, compared with 260.95, 259.85, and 274.15 for integrated DT, CDT, and GAS. At 50% budget, integrated GAVE reaches 167.07 versus its baseline's 151.49, even though CATALIST trails baseline GAVE in the main comparison.

Table 4 Cumulative component configurations. Each row adds one component to the preceding configuration.

Variant	Score	CPA diagnostic (%)
Base DT	157.78 ± 38.12	52.68 ± 8.39
+ graph prior	240.08 ± 34.38	62.20 ± 7.67
+ consistency	251.18 ± 27.92	58.93 ± 6.47
+ augmentation (full)	260.95 ± 19.41	59.82 ± 6.17

There is no contradiction: these are different backbone configurations. The 20 favorable pairs describe coverage across the table, rather than 20 independent replications or evidence of identical implementation and computational cost across backbones.

5.3 Component Contributions and Screening

Table 4 examines the reported progression from DT to graph prior, graph plus consistency, and full CATALIST.

Score progression. Adding the graph-prior configuration raises Score from 157.78 to 240.08, an increase of 82.30 (52.2%). Adding consistency raises it by a further 11.10 (4.6%) to 251.18; the full configuration adds 9.77 (3.9%) to reach 260.95. Relative to DT, the total increase is 103.17 (65.4%). The largest displayed increment accompanies the structural configuration, while consistency and the final augmentation-containing configuration yield further increases. These cumulative comparisons describe conditional changes along the reported sequence. They do not separately identify graph encoding versus attention bias, or generation versus screening and additional training-data exposure.

CPA diagnostic. The diagnostic changes from 52.68 for Base DT to 62.20 with graph prior, 58.93 with consistency, and 59.82 with augmentation. The consistency step combines a higher Score with a lower diagnostic than graph prior. The full configuration has the highest Score, while its diagnostic is higher than that of Base DT. These values describe the component comparison; Campaign-level violation frequency is discussed separately in Section 5.5.

Independent component ablation. Table 5 compares the full model with configurations that remove structure, augmentation, or consistency. Structure denotes dependency-guided encoding together with type/time attention bias. Removing these components lowers Score by 78.54, 23.29, and 12.23, respectively. The comparisons assess the combined modules and do not separately isolate graph encoding from attention bias or establish statistical significance.

Figure 4 separates the component-removal results from the screening comparison and displays performance and candidate acceptance on separate axes.

Augmentation screening. Full screening yields a Score of 260.95 and acceptance percentage of 7.2%, compared with 244.30 and 18.7% for range-only screening. Full screening therefore combines a higher Score with fewer accepted candidates. A lower acceptance percentage alone does not demonstrate better generated-label quality.

Table 5 Component ablation and screening variants at the nominal budget. Str.: structure; Aug.: augmentation; Cons.: consistency; Range: range-only screening.

Variant	Score	Accepted (%)
CATALIST	260.95 ± 19.41	7.2
w/o Str.	182.41 ± 31.03	7.4
w/o Aug.	237.66 ± 22.15	N/A
w/o Cons.	248.72 ± 20.84	7.1
w/ Range	244.30 ± 21.66	18.7

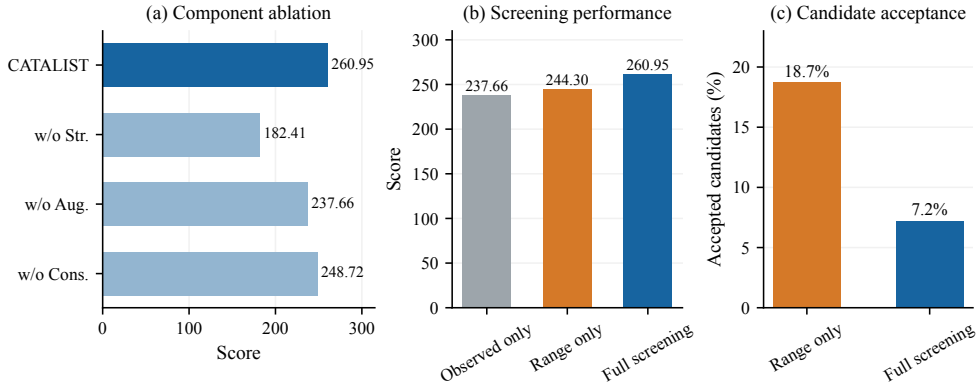


Fig. 4 Component ablation and augmentation screening performance. (a) Component ablation. (b) Screening performance. (c) Candidate acceptance rate.

The screening comparison contrasts observed-only training, range-only augmentation, and full screening. The range-only variant omits the resource-excess, gain, and dispersion gates. The observed-only configuration is the without-augmentation row in Table 5.

5.4 Hyperparameter Sensitivity

Figure 5 compares graph threshold, graph strength, regularization weight, searched-action count, mix ratio, and action-search step. The displayed means are highest at candidate count five, graph threshold 0.03, graph strength 0.5, and regularization weight 0.03 in their respective sweeps. These marginal maxima do not establish that their combination is jointly optimal.

The curves show sensitivity to the tested parameter settings. The favorable settings are specific to the displayed ranges and do not establish a universal parameter choice.

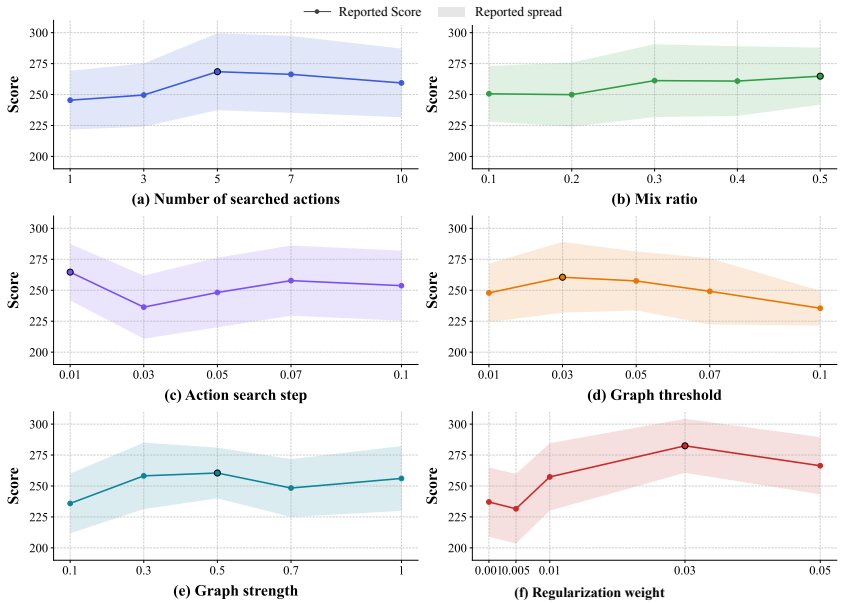


Fig. 5 Sensitivity of bidding performance to key hyperparameters.

Table 6 Value, Score, CPA violation frequency, and budget utilization at nominal budget.

Policy	Value	Score	CPA viol. (%)	Utiliz. (%)
Base DT	212.74 ± 42.35	157.78 ± 38.12	37.2	81.4
Base GAVE	314.69 ± 35.61	236.83 ± 24.07	42.5	84.1
CATALIST	348.22 ± 30.27	260.95 ± 19.41	39.1	87.6

5.5 Constraint Analysis and Discussion

Table 6 compares Value, Score, CPA violation frequency, and budget utilization. For campaign n, realized acquisitions are A_n, expenditure is S_n^{spend}, the CPA target is C_n, and initial effective budget is $B_{0,n} > 0$. Violation frequency is the percentage of campaigns satisfying $S_n^{\text{spend}} > C_n A_n$: zero acquisitions with positive expenditure count as a violation, whereas zero acquisitions with zero expenditure do not. Budget utilization is the mean of $100 S_n^{\text{spend}} / B_{0,n}$ across the same eligible campaigns. These explicitly defined quantities are kept separate from the CPA diagnostic in Table 4.

Table 6 reports the four evaluation metrics. CATALIST has higher Value and Score point estimates than both baselines, and budget utilization is 87.6%, compared with 81.4% for DT and 84.1% for GAVE. The CPA violation frequency is 39.1%, below GAVE by 3.4 percentage points but above DT by 1.9 percentage points. Thus the results show a utility–constraint trade-off rather than uniformly improved compliance. The CATALIST Score of 260.95 agrees with the full-model component result in Table 5.

Result interpretation. The results show three descriptive patterns: CATALIST leads the main comparison at 75%–150% budgets; Score increases along the reported component progression; and the integrated variants exceed their corresponding backbones at all displayed budget ratios. The tightest-budget exception and the CPA diagnostic qualify the interpretation: higher performance is not uniform dominance across resource conditions and does not establish improved constraint compliance.

The comparisons characterize performance within the evaluated settings. Cumulative component configurations do not isolate individual mechanisms, and backbone integration results do not quantify computational efficiency. Budget variation tests resource conditions rather than general distribution shifts. The dependency prior represents observational associations, and generated action labels depend on model predictions; the results therefore support neither causal identification nor campaign-level compliance guarantees.

6 Conclusion

This paper addresses historical dependency modeling and limited logged-action coverage in offline auto-bidding through CATALIST, a structurally guided sequence-learning framework. CATALIST incorporates feature-dependency priors into state representations, organizes information exchange over decision histories through token-type and temporal attention biases, and supplements policy supervision with local alternative actions screened for resource feasibility and predicted gains. Joint training grounds outcome and value prediction in observed records, connecting structured history modeling with model-based augmentation in a unified framework for learning bidding policies from limited auction logs.

Experiments on AuctionNet show that CATALIST achieves the highest reported performance among the compared methods over the 75%–150% evaluation-budget range. At the nominal budget, its Score reaches 260.95, exceeding DT and GAVE by 65.4% and 10.2%, respectively. Backbone integrations and component comparisons further exhibit consistent performance-improvement trends, supporting the design of combining structural guidance with local model-based augmentation for offline bidding. Meanwhile, the comparison at the 50% budget and the CPA violation frequencies indicate that policy utility remains sensitive to resource conditions and that performance gains should be evaluated together with constraint outcomes.

References

[1] Achiam J, Held D, Tamar A, et al (2017) Constrained policy optimization. In: Proc. Int. Conf. Machine Learning (ICML), vol 70. PMLR, pp 22–31

[2] Agarwal R, Schwarzer M, Castro PS, et al (2021) Deep Reinforcement Learning at the Edge of the Statistical Precipice. arXiv preprint arXiv:210813264

[3] Ajay A, Du Y, Gupta A, et al (2022) Is Conditional Generative Modeling all you need for Decision-Making? arXiv preprint arXiv:221115657

[4] Altman E (2021) Constrained Markov decision processes. Routledge

[5] Arjovsky M, Bottou L, Gulrajani I, et al (2019) Invariant Risk Minimization. arXiv preprint arXiv:190702893

[6] Balseiro SR, Gur Y (2019) Learning in repeated auctions with budgets: Regret minimization and equilibrium. Management Science 65(9):3952–3968

[7] Brandfonbrener D, Bietti A, Buckman J, et al (2022) When does return-conditioned supervised learning work for offline reinforcement learning? arXiv preprint arXiv:220601079

[8] Carroll M, Lin J, Paradise O, et al (2022) Towards Flexible Inference in Sequential Decision Problems via Bidirectional Transformers. arXiv preprint arXiv:220413326

[9] Chen L, Lu K, Rajeswaran A, et al (2021) Decision transformer: Reinforcement learning via sequence modeling. Advances in neural information processing systems 34:15084–15097

[10] Chow Y, Nachum O, Duenez-Guzman E, et al (2018) A Lyapunov-based Approach to Safe Reinforcement Learning. arXiv preprint arXiv:180507708

[11] Chua K, Calandra R, McAllister R, et al (2018) Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models. arXiv preprint arXiv:180512114

[12] Dai Z, Yang Z, Yang Y, et al (2019) Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. arXiv preprint arXiv:190102860

[13] Emmons S, Eysenbach B, Kostrikov I, et al (2021) RvS: What is Essential for Offline RL via Supervised Learning? arXiv preprint arXiv:211210751

[14] Fu J, Kumar A, Nachum O, et al (2020) D4RL: Datasets for Deep Data-Driven Reinforcement Learning. arXiv preprint arXiv:200407219

[15] Fujimoto S, Gu SS (2021) A Minimalist Approach to Offline Reinforcement Learning. arXiv preprint arXiv:210606860

[16] Fujimoto S, Conti E, Ghavamzadeh M, et al (2019) Benchmarking Batch Deep Reinforcement Learning Algorithms. arXiv preprint arXiv:191001708

[17] Fujimoto S, Meger D, Precup D (2019) Off-policy deep reinforcement learning without exploration. In: International conference on machine learning, PMLR, pp 2052–2062

[18] Gao J, Li Y, Mao S, et al (2025) Generative auto-bidding with value-guided explorations. In: Proceedings of the 48th International ACM SIGIR Conference

on Research and Development in Information Retrieval, pp 244–254

[19] Granger CW (1969) Investigating causal relations by econometric models and cross-spectral methods. Econometrica: journal of the Econometric Society pp 424–438

[20] Guo C, Pleiss G, Sun Y, et al (2017) On calibration of modern neural networks. In: Proc. Int. Conf. Machine Learning (ICML), vol 70. PMLR, pp 1321–1330

[21] Guo J, Huo Y, Zhang Z, et al (2024) Generative auto-bidding via conditional diffusion modeling. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp 5038–5049

[22] Haarnoja T, Zhou A, Abbeel P, et al (2018) Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: Proc. Int. Conf. Machine Learning (ICML), vol 80. PMLR, pp 1861–1870

[23] Hamilton WL, Ying R, Leskovec J (2017) Inductive Representation Learning on Large Graphs. arXiv preprint arXiv:170602216

[24] He Y, Chen X, Wu D, et al (2021) A unified solution to constrained bidding in online display advertising. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp 2993–3001

[25] Janner M, Fu J, Zhang M, et al (2019) When to Trust Your Model: Model-Based Policy Optimization. arXiv preprint arXiv:190608253

[26] Janner M, Li Q, Levine S (2021) Offline reinforcement learning as one big sequence modeling problem. Advances in neural information processing systems 34:1273–1286

[27] Janner M, Du Y, Tenenbaum J, et al (2022) Planning with diffusion for flexible behavior synthesis. In: Proc. Int. Conf. Machine Learning (ICML), vol 162. PMLR, pp 9902–9915

[28] Kendall A, Gal Y, Cipolla R (2018) Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7482–7491

[29] Kidambi R, Rajeswaran A, Netrapalli P, et al (2020) MOReL : Model-Based Offline Reinforcement Learning. arXiv preprint arXiv:200505951

[30] Kipf TN, Welling M (2016) Semi-Supervised Classification with Graph Convolutional Networks. arXiv preprint arXiv:160902907

[31] Kostrikov I, Nair A, Levine S (2021) Offline reinforcement learning with implicit q-learning. arXiv preprint arXiv:211006169

[32] Kumar A, Fu J, Tucker G, et al (2019) Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction. arXiv preprint arXiv:190600949

[33] Kumar A, Zhou A, Tucker G, et al (2020) Conservative q-learning for offline reinforcement learning. Advances in neural information processing systems 33:1179–1191

[34] Lakshminarayanan B, Pritzel A, Blundell C (2017) Simple and scalable predictive uncertainty estimation using deep ensembles. Advances in neural information processing systems 30

[35] Levine S, Kumar A, Tucker G, et al (2020) Offline reinforcement learning: Tutorial, review, and perspectives on open problems. arXiv preprint arXiv:200501643

[36] Li Y, Mao S, Gao J, et al (2025) Gas: Generative auto-bidding with post-training search. In: Companion Proceedings of the ACM on Web Conference 2025, pp 315–324

[37] Liu Z, Guo Z, Yao Y, et al (2023) Constrained decision transformer for offline safe reinforcement learning. In: Proc. Int. Conf. Machine Learning (ICML), vol 202. PMLR, pp 21611–21630

[38] Loshchilov I, Hutter F (2017) Decoupled weight decay regularization. arXiv preprint arXiv:171105101

[39] Miyato T, ichi Maeda S, Koyama M, et al (2017) Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. arXiv preprint arXiv:170403976

[40] Nair A, Gupta A, Dalal M, et al (2020) AWAC: Accelerating Online Reinforcement Learning with Offline Datasets. arXiv preprint arXiv:200609359

[41] Ovadia Y, Fertig E, Ren J, et al (2019) Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift. arXiv preprint arXiv:190602530

[42] Pamfil R, Sriwattanaworachai N, Desai S, et al (2020) Dynotears: Structure learning from time-series data. In: Proc. Int. Conf. Artificial Intelligence and Statistics (AISTATS), vol 108. PMLR, pp 1595–1605

[43] Pearl J, Mackenzie D (2018) The book of why: the new science of cause and effect. Basic books

[44] Peters J, Buhlmann P, Meinshausen N (2015) Causal inference using invariant prediction: identification and confidence intervals. arXiv preprint arXiv:150101332

[45] Peters J, Janzing D, Scholkopf B (2017) Elements of causal inference: foundations and learning algorithms. MIT press

[46] Rubin DB (1974) Estimating causal effects of treatments in randomized and nonrandomized studies. Journal of educational Psychology 66(5):688–701

[47] Runge J, Nowack P, Kretschmer M, et al (2019) Detecting and quantifying causal associations in large nonlinear time series datasets. Science advances 5(11):eaau4996

[48] Schoelkopf B, Janzing D, Peters J, et al (2012) On Causal and Anticausal Learning. arXiv preprint arXiv:12066471

[49] Schölkopf B, Locatello F, Bauer S, et al (2021) Toward causal representation learning. Proceedings of the IEEE 109(5):612–634

[50] Shaw P, Uszkoreit J, Vaswani A (2018) Self-Attention with Relative Position Representations. arXiv preprint arXiv:180302155

[51] Sohn K, Berthelot D, Li CL, et al (2020) FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. arXiv preprint arXiv:200107685

[52] Su K, Huo Y, Zhang Z, et al (2024) Auctionnet: A novel benchmark for decision-making in large-scale games. Advances in Neural Information Processing Systems 37:94428–94452

[53] Sutton RS, Barto AG (2018) Reinforcement Learning: An Introduction, 2nd edn. MIT Press

[54] Tank A, Covert I, Foti N, et al (2018) Neural Granger Causality. arXiv preprint arXiv:180205842

[55] Tarvainen A, Valpola H (2017) Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. arXiv preprint arXiv:170301780

[56] Vaswani A, Shazeer N, Parmar N, et al (2017) Attention is all you need. Advances in neural information processing systems 30

[57] Velickovic P, Cucurull G, Casanova A, et al (2017) Graph Attention Networks. arXiv preprint arXiv:171010903

[58] Verma V, Kawaguchi K, Lamb A, et al (2019) Interpolation Consistency Training for Semi-Supervised Learning. arXiv preprint arXiv:190303825

[59] Vickrey W (1961) Counterspeculation, auctions, and competitive sealed tenders. The Journal of finance 16(1):8–37

[60] Wang Z, Novikov A, Zolna K, et al (2020) Critic Regularized Regression. arXiv preprint arXiv:200615134

[61] Wu D, Chen X, Yang X, et al (2018) Budget constrained bidding by model-free reinforcement learning in display advertising. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp 1443–1451

[62] Xie Q, Dai Z, Hovy E, et al (2020) Unsupervised data augmentation for consistency training. Advances in neural information processing systems 33:6256–6268

[63] Ying C, Cai T, Luo S, et al (2021) Do Transformers Really Perform Bad for Graph Representation? arXiv preprint arXiv:210605234

[64] Yu T, Thomas G, Yu L, et al (2020) Mopo: Model-based offline policy optimization. Advances in neural information processing systems 33:14129–14142

[65] Yu T, Kumar A, Rafailov R, et al (2021) COMBO: Conservative Offline Model-Based Policy Optimization. arXiv preprint arXiv:210208363

[66] Yu Y, Chen J, Gao T, et al (2019) DAG-GNN: DAG structure learning with graph neural networks. In: Proc. Int. Conf. Machine Learning (ICML), vol 97. PMLR, pp 7154–7163

[67] Zhang H, Cisse M, Dauphin YN, et al (2017) mixup: Beyond Empirical Risk Minimization. arXiv preprint arXiv:171009412

[68] Zhang W, Yuan S, Wang J (2014) Optimal real-time bidding for display advertising. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 1077–1086

[69] Zhao J, Qiu G, Guan Z, et al (2018) Deep reinforcement learning for sponsored search real-time bidding. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, pp 1021–1030

[70] Zheng Q, Zhang A, Grover A (2022) Online decision transformer. In: Proc. Int. Conf. Machine Learning (ICML), vol 162. PMLR, pp 27042–27059

[71] Zheng X, Aragam B, Ravikumar PK, et al (2018) Dags with no tears: Continuous optimization for structure learning. Advances in neural information processing systems 31

[72] Zheng X, Dan C, Aragam B, et al (2020) Learning sparse nonparametric dags. In: Proc. Int. Conf. Artificial Intelligence and Statistics (AISTATS), vol 108. PMLR, pp 3414–3425

[73] Zhu H, Jin J, Tan C, et al (2017) Optimized cost per click in taobao display advertising. In: Proceedings of the 23rd ACM SIGKDD international conference

on knowledge discovery and data mining, pp 2191–2200