C2:CROSS LEARNING MODULE ENHANCED DECISION TRANSFORMER WITH CONSTRAINT-AWARE LOSS FOR AUTO-BIDDING

Jinren Ding
Kuaishou Technology
Beijing, China
dingjinren@kuaishou.com

Xuejian Xu
Kuaishou Technology
Beijing, China
xuxuejian@kuaishou.com

Shen Jiang
Kuaishou Technology
Beijing, China
jiangshen@kuaishou.com

Zhitong Hao
Kuaishou Technology
Beijing, China
haozhitong@kuaishou.com

Jinhui Yang
Kuaishou Technology
Beijing, China
yangjinhui@kuaishou.com

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

ABSTRACT

Decision Transformer (DT) shows promise for generative auto-bidding by capturing temporal dependencies, but suffers from two critical limitations: insufficient cross-correlation modeling among state, action, and return-to-go (RTG) sequences, and indiscriminate learning of optimal/suboptimal behaviors. To address these, we propose C2, a novel framework enhancing DT with two core innovations: (1) a Cross Learning Block (CLB) via cross-attention to strengthen inter-sequence correlation modeling; (2) a Constraint-aware Loss (CL) incorporating budget and Cost-Per-Acquisition (CPA) constraints for selective learning of optimal trajectories. Extensive offline evaluations on the AuctionNet dataset demonstrate consistent performance gains (up to 3.23% over state-of-the-art GAVE) across diverse budget settings; ablation studies verify the complementary synergy of CLB and CL, confirming C2's superiority in auto-bidding. The code for reproducing our results is available at: https://github.com/Dingjinren/C2.

1 INTRODUCTION

Auto-bidding has become an indispensable component of modern online advertising ecosystems (Zhao et al., 2018; 2020; Gao et al., 2024b;a), bridging advertisers' diverse objectives (e.g., maximizing conversions, controlling cost-per-action) and dynamic market environments. With the exponential growth of ad impressions and increasing user behavior complexity (Aggarwal et al., 2024), traditional

manual bid management and static rule-based strategies (Chen et al., 2011; Yu et al., 2017) are impractical, failing to adapt to real-time market fluctuations and evolving advertiser preferences (Jha et al., 2024; Liu et al., 2024). This drives the shift to data-driven solutions, with reinforcement learning (RL)-based methods widely explored by framing bidding as a Markov Decision Process (MDP) (Ye et al., 2019; Zhao et al., 2019).

However, RL-based approaches (Ye et al., 2019; Cai et al., 2017; Gao et al., 2023; Zhao et al., 2021) have a critical limitation: the MDP's state independence assumption ignores long-range temporal dependencies in bidding sequences. In real scenarios, subsequent outcomes correlate strongly with historical trajectories (e.g., past bids, user engagement, budget dynamics), making MDP-based state abstraction overly simplistic and suboptimal. To tackle this issue, the dominant generative solution lies in Decision Transformer (DT), a sequence-modeling framework tailored to capture complex temporal correlations and contextual information inherent in bidding processes.

Specifically, Chen et al. (2021) established DT as a powerful auto-bidding framework, boasting dual strengths: it captures intricate temporal dependencies in bidding sequences and seamlessly integrates multi-dimensional inputs (states, actions, return-to-go rewards) to generate optimal bids, perfectly matching the dynamic requirements of advertising scenarios. Its ability to model long-range dependencies and fuse multi-factor information addresses limitations of isolated-factor models, advancing practical bidding performance.

Yet, DT-based models (Gao et al., 2025; Li et al., 2024; 2025) face real-world challenges. First, conventional DTs simply stack state, action, and return-to-go embeddings before inputting to the Transformer, insufficiently modeling their correlations—especially for complex objectives with interdependent parameters like cost thresholds. Second, DTs mimic historical bidding sequences indiscriminately, learning both optimal and suboptimal behaviors, restricting to documented patterns and risking behavioral collapse.

To tackle these issues, we propose C2: a **C**ross learning block-enhanced decision transformer with **C**onstraint-aware loss. We introduce Cross Learning Block-enhanced DT (CLB-DT) to deepen correlation modeling among state, action, and return-to-go sequences via cross-attention (Brown et al., 2020; Vaswani et al., 2017; Melnychuk et al., 2022), and a Constraint-aware Loss (CL) to guide targeted learning—promoting optimal behaviors while suppressing suboptimal ones (Edelman et al., 2007; Aggarwal et al., 2009).

Our contributions are summarized as follows:

- We identify key limitations of conventional DT in auto-bidding: inadequate correlation modeling among state-action-return-to-go sequences and indiscriminate learning of historical behaviors, highlighting the need for targeted improvements.
- We propose the novel C2 framework with two core innovations: (1) A Cross Learning Block (CLB) with cross-attention that replaces traditional Transformer blocks to enhance multi-dimensional sequential correlation modeling; (2) A Constraint-aware Loss that prioritizes optimal bidding behaviors via budget and KPI constraints.
- We validate C2 through comprehensive offline experiments. Evaluations on AuctionNet (Su et al., 2024) show C2 outperforms state-of-the-art baselines (up to 3.23% gain over GAVE) across 50%-150% budget settings; ablation studies confirm CLB and CL contribute 10.2% and 7.2% gains respectively, with a 15.3% synergetic improvement over vanilla DT, verifying C2's effectiveness.

2 PRELIMINARY

This section first formalizes the auto-bidding problem with core constraints in real-world advertising scenarios, then introduces the fundamental DT-based sequential decision-making method, laying the foundation for subsequent improvements.

2.1 AUTO-BIDDING PROBLEM

Consider a discrete-time auto-bidding process where N impression opportunities arrive sequentially ($i = 1, 2, \ldots, N$). An advertiser submits a bid b_i for each impression to participate in real-time

auctions; under the industry-standard generalized second-price (GSP) mechanism (Edelman et al., 2007; Aggarwal et al., 2009), the advertiser wins if b_i exceeds the highest competing bid b_i^-. Let $x_i \in \{0, 1\}$ denote the auction outcome (1 for winning, 0 otherwise), and $c_i = b_i^-$ (cost of winning in GSP auctions).

The core objective is to maximize total value from won impressions ($v_i \in \mathbb{R}^+$, e.g., CTR/CVR-weighted value), with bidding strategies satisfying budget and CPA constraints (Multi-Constrained Bidding setting). The auto-bidding problem is formally formulated as:

$$\begin{aligned} \max_{\{b_i\}_{i=1}^N} \quad & \sum_{i=1}^N x_i v_i \\ \text{s.t.} \quad & \sum_{i=1}^N x_i c_i \leq B, \\ & \frac{\sum_{i=1}^N x_i c_i}{\sum_{i=1}^N x_i v_i} \leq C, \\ & x_i = \begin{cases} 1 & \text{if } b_i > b_i^-, \\ 0 & \text{otherwise}, \end{cases} \end{aligned} \tag{1}$$

where $B \in \mathbb{R}^+$ is the total campaign budget, and $C \in \mathbb{R}^+$ is the maximum allowable CPA threshold. As shown in prior work, the optimal bid b_i^* can be parameterized as a linear combination of v_i and constraint-related terms:

$$b_i^* = \lambda_0 v_i + \lambda_1 C v_i, \tag{2}$$

where λ_0 (regulating value-budget trade-off) and λ_1 (adjusting CPA adherence) balance objectives and constraints. However, real-world dynamics (e.g., fluctuating competition and user behavior) hinder direct parameter computation, so DT models are needed to adaptively learn optimal bidding strategies from historical data.

2.2 DT-BASED METHOD

By framing sequential decision-making as conditional sequence modeling, Chen et al. (2021) has established Decision Transformer (DT) as an effective framework to capture long-range temporal dependencies in auto-bidding sequences. Unlike traditional RL methods, DT avoids restrictive MDP assumptions (which neglect trajectory correlation structures) and leverages the Transformer's superior sequence modeling capacity to map historical segments to optimal actions.

For auto-bidding tasks, DT characterizes the decision process via four core components forming a trajectory segment $\tau = \{\mathbf{R}, \mathbf{S}, \mathbf{A}\} = \{(r_t, s_t, a_t)\}_{t-M}^t$, where M is the input sequence length. Detailed definitions are as follows:

- **State Sequence** $\mathbf{S} = (s_{t-M}, \ldots, s_t)$: Multi-dimensional feature vectors depicting the advertising ecosystem's real-time state at each t, including: (1) remaining budget ($B_t = B - \sum_{k=1}^{t-1} x_k c_k$); (2) remaining campaign duration; (3) historical bidding metrics (e.g., hourly average win rate); and (4) real-time cost-rate $\frac{\sum_{k=1}^{t-1} x_k c_k}{\sum_{k=1}^{t-1} x_k v_k}$ for constraint compliance.
- **Action Sequence** $\mathbf{A} = (a_{t-M}, \ldots, a_t)$: Bidding parameters adjusted at each t. Consistent with Equation equation 2, $a_t = \lambda_t = \lambda_{0,t} + \lambda_{1,t} C$, the unified parameter determining actual bids $b_t = \lambda_t v_i$.
- **Reward Sequence** $\mathbf{RW} = (rw_{t-M}, \ldots, rw_t)$: Cumulative value from impressions in $[t, t+1]$, computed as $rw_t = \sum_{k \in \mathcal{I}_t} x_k v_k$ ($\mathcal{I}_t$ is the set of impressions in this window).
- **Return-to-Go (RTG) Sequence** $\mathbf{R} = (r_{t-M}, \ldots, r_t)$: $r_t = \sum_{k=t}^T rw_k$, the total remaining reward from t to campaign end. As a critical conditional signal, RTG guides DT to learn long-term goal-aligned strategies.

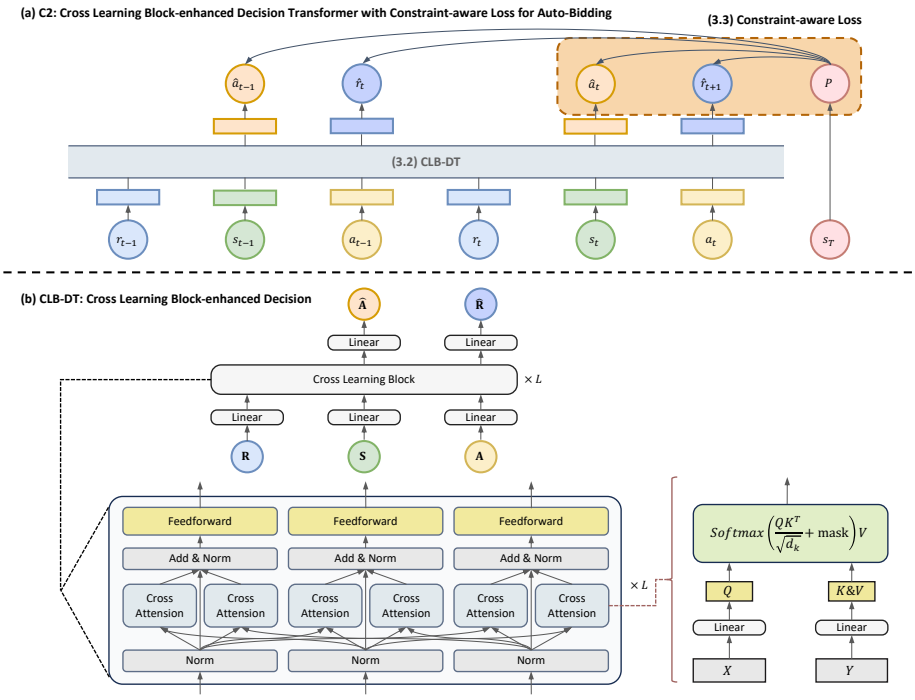


Figure 1: Overall structure of C2.

The standard DT training pipeline has two key steps: concatenate r_t, s_t, a_t embeddings into a sequential vector, then feed it into a causal Transformer encoder to model inter-component dependencies. The model is optimized to predict a_{t+1} from the historical segment $(r_{t-M}, s_{t-M}, a_{t-M}, \ldots, r_t, s_t, a_t)$. Despite its strengths, this paradigm has two drawbacks: (1) naive concatenation fails to model cross-correlations between r_t, s_t, a_t (exacerbated in complex scenarios) (Guo et al., 2024; Li et al., 2024); (2) lack of constraint-aware objectives leads to indiscriminate learning of optimal/suboptimal behaviors (Fujimoto et al., 2019; Kostrikov et al.).

DT-based methods are often evaluated on AuctionNet (Su et al., 2024), which provides large-scale simulated trajectories for fair comparison. Implementations follow prior work (Li et al., 2024; Guo et al., 2024) using causal Transformers with fixed hyperparameters (e.g., layers, attention heads) optimized via AdamW (Loshchilov, 2017), ensuring consistency with evaluation protocols.

3 Methodology

Here, we will detail C2's overview and key components.

3.1 C2 Overview

As illustrated in Figure 1(a), the proposed C2 framework adopts a DT-like architecture, which takes sequences of states, actions, and RTGs as inputs and outputs the optimal subsequent bidding action. Unlike conventional DT models that suffer from two critical limitations—insufficient capacity to capture cross-correlations among state, action, and RTG sequences and indiscriminate learning of all historical bidding samples—C2 introduces two key improvements to address these issues. Specifically, the Cross Learning Block-enhanced DT (CLB-DT) optimizes the model architecture to strengthen the modeling of correlations among the three input sequences, while the Constraint-aware Loss (CL) leverages the constraints of Multi-Constrained Bidding (MCB) to compute adaptive penalty terms. These penalty terms are incorporated into the loss function to guide the model toward selective learning of high-quality bidding trajectories while suppressing suboptimal ones.

3.2 CLB-DT

Our Cross Learning Block-enhanced Decision Transformer (CLB-DT) adopts a multi-input unified architecture, which takes three distinct sequences (each of length M) as inputs: the state sequence $\mathbf{S}$, action sequence $\mathbf{A}$, and return-to-go (RTG) sequence $\mathbf{R}$. Specifically, we introduce the Cross Learning Block (CLB) to replace the vanilla Transformer blocks in the original DT architecture, thereby enabling the model to fully capture the cross-correlations among the three input sequences. The architecture is illustrated in Figure 1(b).

Cross Learning Block. Let $b = 1, \ldots, B$ index the distinct Cross Learning Blocks. Each block takes three parallel hidden state sequences $\mathbf{S}^b, \mathbf{A}^b, \mathbf{R}^b$ as inputs (outputs of the b-th block) and outputs updated sequences $\mathbf{S}^{b+1}, \mathbf{A}^{b+1}, \mathbf{R}^{b+1}$, where the hidden state dimension is denoted by d_h. For the first block ($b = 1$), the input sequences are initialized with positional encodings derived from time steps $\mathbf{T}$, combined with sequence-specific linear encodings:

$$\begin{aligned} \mathbf{T}^0 &= \mathrm{Linear}_T(\mathbf{T}), \quad \mathbf{S}^1 = \mathrm{Linear}_S(\mathbf{S}) + \mathbf{T}^0, \\ \mathbf{A}^1 &= \mathrm{Linear}_A(\mathbf{A}) + \mathbf{T}^0, \quad \mathbf{R}^1 = \mathrm{Linear}_R(\mathbf{R}) + \mathbf{T}^0, \end{aligned} \tag{3}$$

where $\mathrm{Linear}_T, \mathrm{Linear}_S, \mathrm{Linear}_A, \mathrm{Linear}_R$ are independent learnable linear layers for positional encoding and sequence-specific encoding, respectively. For blocks with $b \geq 2$, the inputs $\mathbf{S}^b, \mathbf{A}^b, \mathbf{R}^b$ are directly the outputs of the preceding block ($b - 1$).

Each Cross Learning Block integrates three core components: (i) masked cross-attention, (ii) feed-forward layer (FF), and (iii) layer normalization (LN), which collaborate to model cross-sequence dependencies and non-linear feature transformations (Radford et al., 2019; Brown et al., 2020; Vaswani et al., 2017; Melnychuk et al., 2022).

(i) Masked cross-attention extends standard cross-attention with an attention mask to regulate interactions between two distinct sequences (denoted as X and Y). Its core consists of three learnable components—Query (Q), Key (K), and Value (V)—derived via linear transformations:

$$Q = Q(\mathrm{X}^b) = \mathrm{X}^b W_Q + \mathbf{1} b_Q^\top, \tag{4}$$

$$K = K(\mathrm{Y}^b) = \mathrm{Y}^b W_K + \mathbf{1} b_K^\top, \tag{5}$$

$$V = V(\mathrm{Y}^b) = \mathrm{Y}^b W_V + \mathbf{1} b_V^\top, \tag{6}$$

where $W_Q, W_K, W_V \in \mathbb{R}^{d_h \times d_k}$ are learnable weight matrices, $b_Q, b_K, b_V \in \mathbb{R}^{d_k}$ are bias vectors, $\mathbf{1}$ is a column vector of ones for bias broadcasting, and d_k denotes the dimension of Query/Key/Value (typically $d_k = d_h$). The superscript b indicates the block index (consistent with input sequences), and batch dimensions are omitted for brevity.

The attention mask is a binary matrix that suppresses invalid interactions: invalid positions are assigned a large negative value (e.g., -10^4) to inhibit attention flow, while valid positions are set to zero. The masked cross-attention operation is defined as:

$$\mathrm{Attn}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + \mathrm{mask}\right)V, \tag{7}$$

which ensures the model focuses only on relevant cross-sequence information. The computation proceeds in three steps: (1) compute a similarity matrix via $QK^\top/\sqrt{d_k}$ to quantify relevance between X and Y; (2) apply the mask to suppress invalid positions; (3) normalize scores via softmax to generate attention weights, then compute a weighted sum of V to obtain the attention output. This process enables modeling intricate cross-sequence dependencies under mask constraints.

(ii) Feedforward layer (FF) is a position-wise component that acts on the output of masked cross-attention (denoted as $\mathbf{H}^b$) to model complex non-linear mappings. It is formally defined as:

$$\mathrm{FF}(\mathbf{H}^b) = \mathrm{Linear}_2\left(\mathrm{ReLU}\left(\mathrm{Linear}_1(\mathbf{H}^b)\right)\right), \tag{8}$$

where $\mathrm{Linear}_1 \in \mathbb{R}^{d_h \times d_{ff}}$ and $\mathrm{Linear}_2 \in \mathbb{R}^{d_{ff} \times d_h}$ are successive learnable linear layers, and d_{ff} (typically $4d_h$) is the intermediate hidden dimension. A dropout layer (dropout rate = 0.1 in our implementation) is inserted after the second linear projection as a standard regularization strategy to

mitigate overfitting. Notably, the FF layer preserves the input dimensionality d_h, ensuring consistency in sequence processing within the block.

(iii) Layer normalization (LN) is an indispensable component applied to the hidden state matrix $\mathbf{H}^b$ at the start of each block to stabilize hidden state distributions and accelerate training. It normalizes features across the hidden dimension d_h for each position (instead of across the sequence dimension), formulated as:

$$\mathrm{LN}(\mathbf{H}^b) = \gamma \cdot \frac{\mathbf{H}^b - \mathbb{E}[\mathbf{H}^b]}{\sqrt{\mathrm{Var}[\mathbf{H}^b] + \epsilon}} + \beta, \tag{9}$$

where $\mathbb{E}[\mathbf{H}^b]$ and $\mathrm{Var}[\mathbf{H}^b]$ are the mean and variance of the hidden state matrix $\mathbf{H}^b$ computed along the hidden dimension d_h, $\epsilon = 10^{-5}$ is a small constant for numerical stability, and $\gamma, \beta \in \mathbb{R}^{d_h}$ are learnable affine parameters (scale and shift). This strategy alleviates internal covariate shift and preserves meaningful feature information via learnable parameters.

Collectively, the three components collaborate to update the input sequences. Taking $\mathbf{A}^{b+1}$ as an example, its computation first aggregates residual cross-attention outputs from interactions between $\mathbf{A}^b$ and the other two sequences ($\mathbf{S}^b, \mathbf{R}^b$):

$$\begin{aligned}\mathbf{A}^b_{\text{attn}} = \mathbf{A}^b &+ \mathrm{Attn}\left(\mathrm{LN}(\mathbf{S}^b), \mathrm{LN}(\mathbf{A}^b), \mathrm{LN}(\mathbf{A}^b)\right) \\ &+ \mathrm{Attn}\left(\mathrm{LN}(\mathbf{R}^b), \mathrm{LN}(\mathbf{A}^b), \mathrm{LN}(\mathbf{A}^b)\right),\end{aligned} \tag{10}$$

where the residual connection ($\mathbf{A}^b + \cdots$) mitigates vanishing gradients. The aggregated output is then processed by LN and FF to generate $\mathbf{A}^{b+1}$:

$$\mathbf{A}^{b+1} = \mathrm{FF}\left(\mathrm{LN}(\mathbf{A}^b_{\text{attn}})\right), \tag{11}$$

where $\mathrm{LN}(\cdot)$ follows Eq. (9), $\mathrm{Attn}(\cdot)$ follows Eq. (7), and $\mathrm{FF}(\cdot)$ follows Eq. (8). The computation of $\mathbf{S}^{b+1}$ and $\mathbf{R}^{b+1}$ adheres to the same logic (cross-attention with the other two input sequences + LN + FF). All operations preserve the hidden dimension d_h and sequence length M, ensuring seamless information flow across consecutive blocks ($b = 1, \ldots, B$).

3.3 Constraint-aware Loss

The Constraint-aware Loss (CL) is designed to enable the model to adaptively perceive and comply with core business constraints (i.e., CPA cost constraint and budget consumption constraint) in advertising auto-bidding during training (Edelman et al., 2007; Aggarwal et al., 2009). Unlike vanilla loss functions that solely optimize for reward maximization, this loss function introduces a dynamic penalty term P to penalize constraint-violating predictions, thereby achieving the dual goals of optimal performance and constraint compliance.

Construction of Penalty Term. The penalty term P combines two sub-terms (P_{CPA} for excessive CPA and P_{BC} for rapid budget consumption) via multiplication to enforce joint constraint compliance. The detailed formulation is as follows:

First, we compute the actual cost per action (CPA) at the end of a trajectory (i.e., the last time step T of the trajectory):

$$CPA_T = \frac{\sum_{i=1}^{I_T} x_i c_i}{\sum_{i=1}^{I_T} x_i v_i}, \tag{12}$$

where I_T denotes the total number of impression/click samples up to the last time step T of a trajectory, x_i is the delivery decision (e.g., bid or not) for the i-th sample, c_i is the delivery cost (e.g., click cost) of the i-th sample, and v_i is the number of valid conversions for the i-th sample. An indicator function is used to identify CPA constraint violations at the last time step T of the trajectory:

$$\mathbb{I}_{CPA_T > \theta} = \begin{cases} 1, & \text{if } CPA_T > \theta \\ 0, & \text{if } CPA_T \leq \theta \end{cases}, \tag{13}$$

where θ is the advertiser-specified maximum acceptable CPA threshold (consistent with C in Eq. equation 1 for constraint consistency).

The CPA penalty term P_{CPA} is activated only when the CPA at the last time step T of the trajectory exceeds the threshold, with adaptive intensity:

$$P_{\text{CPA}} = \begin{cases} \left(\frac{CPA_T}{C}\right)^{\alpha_1}, & \text{if } \mathbb{I}_{CPA_T > \theta} = 1 \\ 1.0, & \text{otherwise} \end{cases}, \tag{14}$$

where C is the maximum allowable cost-rate threshold (Eq. equation 1) and $\alpha_1 > 1$ is the CPA penalty strength coefficient (larger α_1 leads to heavier penalties for more severe violations; set to 2 in our experiments).

For the budget consumption constraint, we first calculate the budget consumption rate at the last time step T of a trajectory:

$$BC_T = \frac{\sum_{i=1}^{I_T} x_i c_i}{B}, \tag{15}$$

where B is the total budget allocated to a single trajectory (Eq. equation 1). The budget consumption penalty term is defined as:

$$P_{\text{BC}} = (BC_T)^{\alpha_2}, \tag{16}$$

where $\alpha_2 > 1$ is the budget penalty coefficient (higher consumption rates result in more significant penalties; set to 2 in our experiments).

The total penalty term P is the product of P_{CPA} and P_{BC}, ensuring joint punishment for both constraints at the end of a trajectory:

$$P = P_{\text{CPA}} \times P_{\text{BC}}. \tag{17}$$

Notably, $P \geq 1.0$ for all scenarios: $P = 1.0$ (no penalty) only when the CPA constraint is satisfied and the budget consumption rate is 0 at the last time step T of the trajectory; $P > 1.0$ if either constraint is violated or the budget is consumed rapidly, with larger P indicating more severe violations.

Application to Action Loss and RTG Loss. The Constraint-aware Loss integrates the penalty term P (denoted as cpa_p in code) into Action Loss ($\mathcal{L}_a$, bid prediction loss) and RTG Loss ($\mathcal{L}_r$, remaining reward prediction loss) via element-wise multiplication, enforcing constraint-aware weighting at the sample level.

The Action Loss (MSE loss for bid prediction) is defined as:

$$\mathcal{L}_a = \frac{1}{N} \sum_{n=1}^{N} P_n \times (\hat{a}_n - a_n^*)^2, \tag{18}$$

where N is the total number of samples, P_n is the penalty term (derived from the last time step T of the trajectory) for the n-th sample, $\hat{a}_n$ is the model-predicted bid parameter λ_n, and a_n^* is the ground-truth target bid parameter. When the n-th sample's bidding strategy leads to constraint violations at the end of the trajectory, $P_n > 1.0$ amplifies the prediction error, forcing the model to prioritize correcting non-compliant bids.

The RTG Loss (MSE loss for remaining reward prediction) is formulated as:

$$\mathcal{L}_r = \frac{1}{N} \sum_{n=1}^{N} P_n \times (\hat{r}_n - r_n^*)^2, \tag{19}$$

where $\hat{r}_n$ is the model-predicted remaining reward for the n-th sample, and r_n^* is the ground-truth remaining reward. The penalty term P_n (calculated at the last time step T of the trajectory) ensures the model learns high-reward RTG distributions while complying with CPA and budget constraints for each sample by the end of the trajectory.

Total Loss. The final total loss function unifies the optimization of bid prediction and reward prediction under constraint awareness, defined as the sum of Action Loss and weighted RTG Loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_a + \lambda \cdot \mathcal{L}_r, \tag{20}$$

where $\lambda > 0$ is a hyperparameter that modulates the contribution of RTG Loss ($\mathcal{L}_r$); $\mathcal{L}_{\text{total}}$ balances bid accuracy and reward prediction quality, with both components constrained by the adaptive penalty term P_n (derived from the last time step T of the trajectory) to ensure adherence to advertising business rules by the end of each trajectory.

Table 1: **Performance comparison**. Bold values are the highest scores, underlined values denote the best baseline results.

Dataset	Budget	USCB	CQL	BCQ	IQL	CDT	DT	GAS	GAVE	C2	*Improve*
AuctionNet	50%	13.1	14.3	16.3	15.8	13.2	15.9	18.4	19.6	**20.2**	3.06%
	75%	16.2	17.5	23.1	24.7	20.0	22.3	27.5	28.3	**28.5**	0.71%
	100%	19.5	24.6	30.3	31.5	32.1	33.3	36.1	37.2	**38.4**	3.23%
	125%	25.8	28.4	35.2	36.4	34.3	36.2	40.0	42.7	**43.0**	0.70%
	150%	33.7	34.8	36.9	40.4	38.6	39.0	46.5	47.4	**48.4**	2.11%

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Dataset We conduct experiments on **AuctionNet** (Su et al., 2024), a public large-scale real-world bidding dataset released by Alibaba and hosted on GitHub[1]. Unlike private non-open-source bidding logs, **AuctionNet** is derived from a manually built offline real advertising system, replicating the complexity of real-world ad auctions while ensuring public accessibility for research.

As the largest available public bidding dataset to our knowledge, it has a substantial scale: totaling over 500 million records with multiple advertising periods. Each period contains more than 500,000 impression opportunities, divided into 48 decision steps, and each opportunity involves bids from 50 agents. Each record includes core information such as predicted conversion value, bid amount, auction results, and impression outcomes.

Metrics Following previous studies (Gao et al., 2025; Li et al., 2024; Su et al., 2024; Guo et al., 2024), we conduct evaluations using varying budget ratios (50%, 75%, 100%, 125%, 150%) from the original dataset to simulate real-world under-budget, full-budget, and over-budget scenarios. Performance is measured using the standard scoring metric for auto-bidding tasks:

$$\begin{aligned} penalty_j &= \min\left\{\left(\frac{C_j}{\sum_i c_{ij}o_i / \sum_i p_{ij}o_i}\right)^{\beta}, 1\right\}, \\ score &= \left(\sum_i o_i v_i\right) \times \min_{j=1}^{J}\{penalty_j\}, \end{aligned} \tag{21}$$

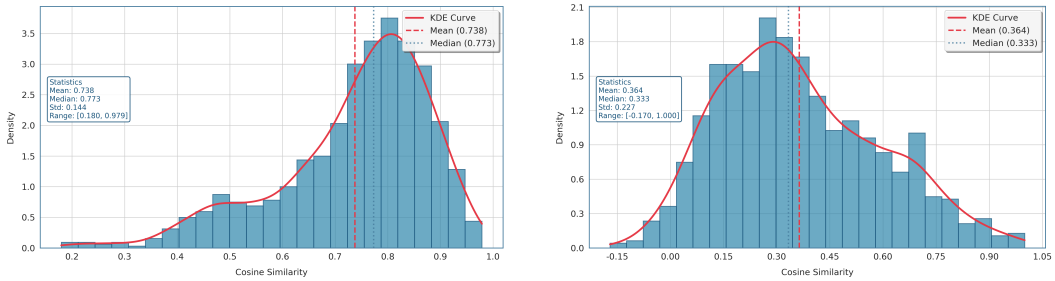
where we set $\beta = 2$ following the standard configuration in prior works, C_j is the constraint threshold for the j-th KPI, c_{ij} and p_{ij} are the cost and performance of the i-th sample under the j-th KPI, o_i is the delivery outcome indicator, and v_i is the sample valuation. Higher scores indicate better balance between performance and constraint compliance.

4.2 COMPARISON WITH STATE-OF-THE-ARTS

To thoroughly validate the effectiveness of our proposed method C2 for advertising auction bid optimization, we conduct a comprehensive comparison between C2 and a diverse set of baseline methods (including USCB (He et al., 2021), CQL (Kumar et al., 2020), BCQ (Fujimoto et al., 2019), CDT (Liu et al., 2023), IQL (Kostrikov et al.), Chen et al. (2021) (DT), GAS (Li et al., 2024), and the state-of-the-art baseline GAVE (Gao et al., 2025)) on the AuctionNet dataset, covering five typical budget ratio settings (50%, 75%, 100%, 125%, and 150%).

As illustrated in Table 1, our C2 method achieves superior performance over all competing baselines across all budget configurations. Specifically, GAVE emerges as the best-performing baseline (marked by underlined values) in every budget scenario, yet our C2 consistently delivers the highest overall scores (highlighted in bold) across all tested budget ratios. Quantitatively, the performance improvement of C2 over GAVE ranges from 0.70% (125% budget) to 3.23% (100% budget), with an average gain of 1.96% across all budget settings. Notably, the most substantial enhancement (3.23%) is observed under the 100% budget ratio—a standard operational scenario for advertising auctions—while C2 also maintains steady and meaningful gains in both under-budget (50%, 75%) and over-budget (125%, 150%) conditions.

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



(a) *DT: Matched vs. shuffled embedding similarity (1000 samples)*

(b) *C2: Matched vs. shuffled embedding similarity (1000 samples)*

Figure 2: **Cross-correlation learning ability: C2 vs. DT**

These results suggest that our C2 method can effectively balance bid prediction accuracy and reward estimation reliability under budget constraints, contributing to consistent performance improvements over the state-of-the-art method GAVE across various budget scenarios on the AuctionNet dataset.

4.3 Cross-correlations Analysis

To verify whether our C2 method alleviates the cross-correlation modeling deficiency of conventional Decision Transformer (DT) models (discussed in Section 1), we conduct a controlled experiment on the AuctionNet dataset. The core intuition is that a model with enhanced cross-correlation learning ability generates significantly distinct embeddings for **matched** (original) and **shuffled** (state sequence randomly permuted) sequences, which is quantified by low cosine similarity between the two embedding types.

Specifically, we randomly sample 1000 instances from the AuctionNet dataset and extract two embedding types for each instance: *matched embeddings* from the first transformer block with original sequential auction data as input, and *shuffled embeddings* from the same block with the state sequence randomly permuted (other components unchanged). We compute cosine similarity between matched and shuffled embeddings for all samples, with the distribution shown in Figure 2.

Statistical results confirm that C2 effectively enhances cross-correlation modeling compared to DT. For the vanilla DT model (Subfigure 2a), the cosine similarity between matched and shuffled embeddings has a **high central tendency** (mean = 0.738, median = 0.773, Std = 0.144, range = $[0.180, 0.979]$), indicating limited ability to distinguish between original and shuffled sequences and thus insufficient cross-correlation modeling. For our C2 method (Subfigure 2b), the cosine similarity distribution shows a **marked reduction in central tendency** (mean = 0.364, 50.7% lower than DT; median = 0.333, Std = 0.227, range = $[-0.170, 1.000]$). The substantially lower mean/median values prove C2 generates distinct embeddings for original and shuffled sequences, verifying successful mitigation of DT's cross-correlation modeling issue.

These results validate that C2 effectively captures interdependencies among sequential auction components—a key improvement over DT. By modeling cross-correlations in auction trajectories, C2 leverages temporal relational features of sequential data to make more optimal bid decisions in auto-bidding tasks.

4.4 Ablation Study

Table 2: **Ablation study of our method components**. (a) DT baseline; (b) DT + Constraint-aware Loss (CL); (c) CLB-DT backbone; (d) CLB-DT + CL (full model). Experiments on AuctionNet (100% budget).

	DT	CLB-DT	CL	Score
(a)	√			33.3
(b)	√		√	35.7
(c)		√		36.7
(d)		√	√	**38.4**

We perform an ablation study on the AuctionNet dataset under 100% budget to evaluate the contributions of cross learning block-enhanced DT and Constraint-aware Loss in our C2 method (Table 2).

The baseline DT model (a) achieves a score of 33.3. Adding CL to DT (b) improves the score to 35.7 (**7.2%** gain), verifying CL's effectiveness in enforcing business constraints. Replacing DT with CLB-DT (c) yields a score of 36.7 (**10.2%** gain over baseline), demonstrating CLB-DT's superiority in capturing auction trajectory correlations. Combining CLB-DT and CL (d) achieves the highest score of **38.4** (**15.3%** gain), confirming the complementary synergy of the two components for optimal performance.

5 Conclusion

This paper proposes C2, a novel framework addressing two key limitations of Decision Transformer (DT) in auto-bidding: insufficient cross-correlation modeling among state-action-RTG sequences and indiscriminate behavior learning. C2 integrates two core components: Cross Learning Block (CLB) for enhanced inter-sequence dependencies via cross-attention, and Constraint-aware Loss (CL) for constraint-guided optimal trajectory learning.

Extensive offline evaluations on AuctionNet validate C2's effectiveness, with up to 3.23% gain over SOTA GAVE across diverse budget settings. Ablation studies confirm CL and CLB contribute 20.6% and 24.0% gains respectively over vanilla DT, achieving a 29.7% combined improvement. Cross-correlation analysis further verifies CLB's superiority in modeling sequence dependencies.

Future work will explore dynamic penalty adjustment for better adaptability and optimize the model structure to better adapt to auto-bidding scenarios.

References

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

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*, pp. 159–183, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

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 *Proc. of WSDM*, pp. 661–670, 2017.

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

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

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

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

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

Jingtong Gao, Bo Chen, Menghui Zhu, Xiangyu Zhao, Xiaopeng Li, Yuhao Wang, Yichao Wang, Huifeng Guo, and Ruiming Tang. Hierrec: Scenario-aware hierarchical modeling for multi-scenario recommendations. In *Proc. of CIKM*, pp. 653–662, 2024a.

Jingtong Gao, Xiangyu Zhao, Muyang Li, Minghao Zhao, Runze Wu, Ruocheng Guo, Yiding Liu, and Dawei Yin. Smlp4rec: an efficient all-mlp architecture for sequential recommendations. *ACM Transactions on Information Systems*, pp. 1–23, 2024b.

Jingtong Gao, Yewen Li, Shuai Mao, Peng Jiang, Nan Jiang, Yejing Wang, Qingpeng Cai, Fei Pan, Kun Gai, Bo An, et al. Generative auto-bidding with value-guided explorations. *arXiv preprint arXiv:2504.14587*, 2025.

Jiayan Guo, Yusen Huo, Zhilin Zhang, Tianyu Wang, Chuan Yu, Jian Xu, Bo Zheng, and Yan Zhang. Aigb: Generative auto-bidding via conditional diffusion modeling. In *Proc. of KDD*, pp. 5038–5049, 2024.

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 *Proc. of KDD*, pp. 2993–3001, 2021.

Arti Jha, Harshit Jain, Parikshit Sharma, Yashvardhan Sharma, and Kamlesh Tiwari. Optimizing real-time bidding strategies: An experimental analysis of reinforcement learning and machine learning techniques. *Procedia Computer Science*, pp. 2017–2026, 2024.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *Proc. of ICLR*.

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

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*, pp. 1104–1108, 2025.

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. *arXiv preprint arXiv:2412.17018*, 2024.

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

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

I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Valentyn Melnychuk, Dennis Frauen, and Stefan Feuerriegel. Causal transformer for estimating counterfactual outcomes. In *International conference on machine learning*, pp. 15293–15329. PMLR, 2022.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

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 *Proc. of NeurIPS*, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Yujian Ye, Dawei Qiu, Mingyang Sun, Dimitrios Papadaskalopoulos, and Goran Strbac. Deep reinforcement learning for strategic bidding in electricity markets. *IEEE Transactions on Smart Grid*, pp. 1343–1355, 2019.

Hao Yu, Michael Neely, and Xiaohan Wei. Online convex optimization with stochastic constraints. *Proc. of NeurIPS*, 2017.

Jun Zhao, Guang Qiu, Ziyu Guan, Wei Zhao, and Xiaofei He. Deep reinforcement learning for sponsored search real-time bidding. In *Proc. of KDD*, pp. 1021–1030, 2018.

Xiangyu Zhao, Long Xia, Jiliang Tang, and Dawei Yin. " deep reinforcement learning for search, recommendation, and online advertising: a survey" by xiangyu zhao, long xia, jiliang tang, and dawei yin with martin vesely as coordinator. *ACM sigweb newsletter*, pp. 1–15, 2019.

Xiangyu Zhao, Xudong Zheng, Xiwang Yang, Xiaobing Liu, and Jiliang Tang. Jointly learning to recommend and advertise. In *Proc. of KDD*, pp. 3319–3327, 2020.

Xiangyu Zhao, Changsheng Gu, Haoshenglun Zhang, Xiwang Yang, Xiaobing Liu, Jiliang Tang, and Hui Liu. Dear: Deep reinforcement learning for online advertising impression in recommender systems. In *Proc. of AAAI*, pp. 750–758, 2021.