AUTO-BIDDING WITH DISENTANGLED ADVERTISER PROFILES AND TRAINING-FREE ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Auto-bidding is a key component of modern advertising systems that provides a personalized bidding strategy for each advertiser. By characterizing each individual, profile-based methods achieve personalization and have proven effective in domains such as recommendation. However, despite the diverse bidding behavior of advertisers, their application to auto-bidding remains limited. A primary reason is that constructing and leveraging advertiser profiles face several challenges: extracting pure profiles is non-trivial, modeling common and private information simultaneously is difficult, and profile updating and cold-start adaptation remain challenging. To tackle these issues, we propose **ADAPT**, an **A**uto-bidding framework with **D**isentangled **A**dvertiser **P**rofiles and **T**raining-free adaptation. ADAPT introduces a two-stage training paradigm and supports training-free adaptation. Specifically, (i) the stage 1 extracts pure static and dynamic profiles via contrastive learning over the advertiser memory bank; (ii) the stage 2 disentangles the dynamic profile into a common profile and a private profile, and combines them with the static profile to jointly condition the bidding strategy; (iii) once trained, ADAPT constructs profiles for new advertisers and updates profiles of existing advertisers without retraining. Our experiments on a large-scale auto-bidding benchmark demonstrate that ADAPT consistently achieves superior performance, and ablation studies further validate the effectiveness of each module.

1 INTRODUCTION

In online advertising systems, advertisers aim to secure billions of daily impression opportunities through bidding, yet the massive traffic volume and fierce competition Muthukrishnan (2009) make manual bidding by human experts infeasible Evans (2009); Wang & Yuan (2015). Auto-bidding addresses this challenge by generating precise bids for large-scale traffic in real time from the current state and environmental information. It formulates bidding as a constrained sequential decision-making problem, dynamically adjusting the bid throughout the delivery period to maximize conversions under advertiser-specified constraints such as cost-per-action (CPA) He et al. (2021); Mou et al. (2022). This makes auto-bidding a crucial mechanism for advertisers to achieve their business objectives in competitive markets Ren et al. (2017).

Existing auto-bidding methods can generally be divided into two categories, *i.e.*, offline reinforcement learning (RL) methods and generative auto-bidding methods. Specifically, offline RL methods learn bidding policies purely from logged datasets, estimating value functions without further interaction with the environment. For example, IQL Kostrikov et al. (2022) approximates the optimal value function via expectile regression and avoids querying out-of-distribution actions during training, thereby mitigating value overestimation and stabilizing policy learning. However, value estimation in long-horizon, sparse-reward bidding environments is often inaccurate, leading to unstable policy learning. To address this, generative auto-bidding methods reformulate auto-bidding as trajectory-conditioned action generation, bypassing value function learning and better capturing long-horizon dependencies Ajay et al. (2023). For instance, Decision Transformer (DT) Chen et al. (2021) employs a Transformer Vaswani et al. (2017) to model sequences of return-to-go (RTG), states, and actions, autoregressively generating bidding actions conditioned on a target return. Although generative methods have achieved remarkable success in auto-bidding, they typically train the model on historical trajectories to produce the current action, without personalizing to the underlying strategy of each advertiser. Meanwhile, profile-based methods have achieved remarkable success in domains such as

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

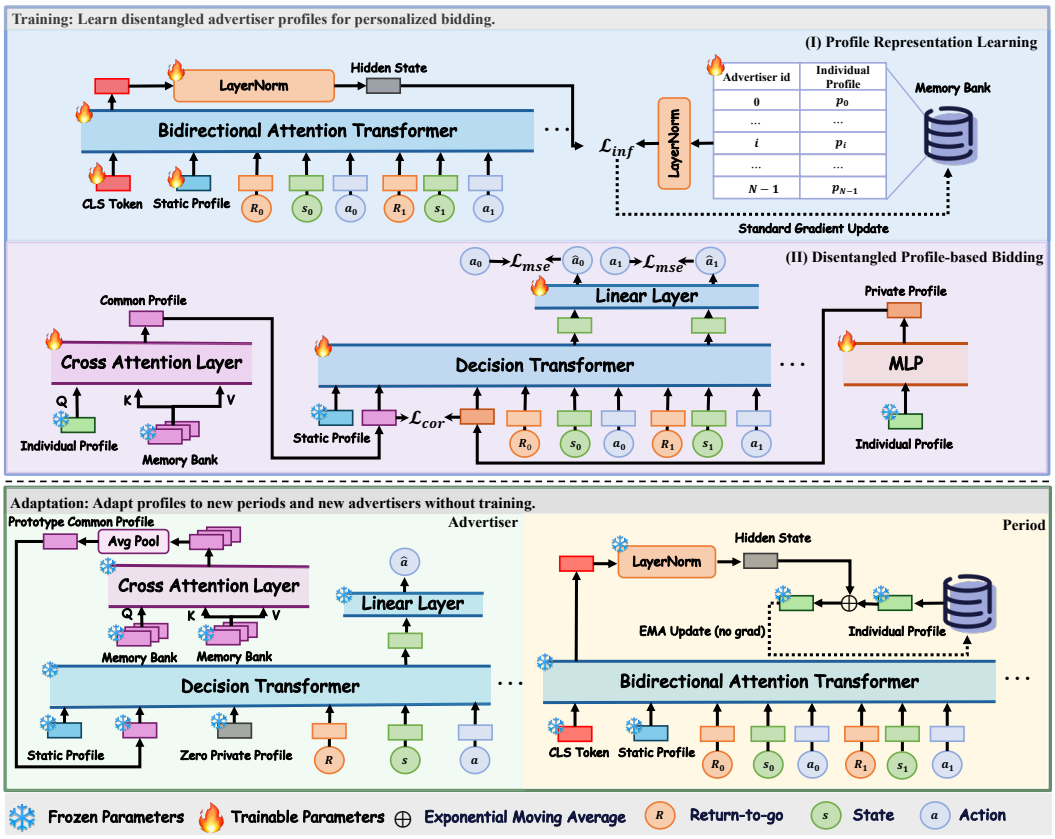


Figure 1: The overall flowchart of the proposed **ADAPT**. ADAPT consists of a two-stage training paradigm and an adaptation module. **Stage (I):** an advertiser's bidding-period sequence, prefixed with a learnable CLS token and its category static profile, is encoded by a trainable bidirectional attention transformer into the hidden state, against which the advertiser's dynamic profile in a trainable memory bank is optimized by an InfoNCE loss. **Stage (II):** the same sequence is first prefixed with the frozen static profile; the common profile is then obtained by cross-attention with the individual profile as query and the memory bank as keys and values, while the private profile is obtained by encoding the individual profile with a multi-layer perceptron (MLP); both are prepended as prefixes and decorrelated by a correlation loss, and a DT with a linear head predicts the action under an MSE loss. **Adaptation:** for a new advertiser, the memory bank is fed as query, key, and value into the Stage (II) cross-attention and average pooled into a prototype common profile; the static profile, this prototype, and a zero private profile are prepended as prefixes and decoded by the frozen DT with a linear head into the action. For a new period, ADAPT reuses the Stage (I) encoding to obtain the hidden state and updates the advertiser's individual profile via an exponential moving average (EMA).

recommendation Wang et al. (2025); Zhang et al. (2026b); Zhang (2024) and personalized generation and retrieval. Teevan et al. (2005); Salemi et al. (2024). A user profile summarizes the style and preference of each user, thereby enabling personalization tailored to each individual.

Although profile-based methods have shown potential in other domains, directly applying them to auto-bidding still faces several challenges. First, existing profile-based methods mostly operate on discrete historical behaviors from which preference is easy to identify Zhang (2024), whereas an auto-bidding trajectory consists of continuous-valued states and actions from which the underlying strategy is difficult to extract. Second, profiles in domains such as recommendation are typically expressed in semantically explicit discrete forms that are easy to understand and inherently interpretable Wang et al. (2025), whereas profiles in auto-bidding consist entirely of numerical information, making the encoded strategy difficult for the model to understand. Third, existing profile-based methods rely on side information or gradient updates to handle new users and profile drift Chen et al. (2018), whereas an advertiser exposes only its trajectory and online serving forbids retraining, forcing profile construction and updating to be training-free and trajectory-only.

To address the above challenges, we propose a novel **A**uto-bidding framework with **D**isentangled **A**dvertiser **P**rofiles and **T**raining-free adaptation, referred to as **ADAPT**. As illustrated in Figure 1, ADAPT is built upon a two-stage training paradigm followed by a training-free adaptation step. In the first stage, profile representation learning extracts pure static and dynamic profiles via contrastive learning over a shared advertiser memory bank, thereby addressing the first challenge. In the second stage, disentangled profile-based bidding applies a correlation loss to disentangle the dynamic profile into a common and a private profile for better profile understanding, which, together with the static profile, condition the final bidding decision, thereby addressing the second challenge. Finally, a training-free adaptation step cold-starts a new advertiser with a common profile distilled from existing advertisers and updates the dynamic profile of an existing advertiser without retraining, thereby addressing the third challenge. Compared with previous methods, the main contributions of our work are summarized as follows:

- We propose **ADAPT**, an auto-bidding framework that learns disentangled advertiser profiles via two-stage training and maintains them via training-free adaptation, thereby delivering personalized bidding decisions.
- We show that ADAPT consistently outperforms strong baselines on a large-scale auto-bidding benchmark, with ablations confirming the contribution of each design.

2 PRELIMINARIES

2.1 PROBLEM STATEMENT

Modern online advertising platforms host thousands of advertisers that simultaneously contend for a limited supply of impressions. Within a given horizon, the traffic an advertiser faces forms a chronologically ordered trajectory of D ad opportunities, and for the i-th opportunity ($i \in \{1, \dots, D\}$) the advertiser responds with a bid bid_i. Each opportunity is allocated through an auction: the highest bidder wins the impression, receives its value v_i, and is charged a cost c_i that, under the widely adopted second-price mechanism, is set by the runner-up bid. The advertiser aims to accumulate as much impression value as possible without exceeding a prescribed budget.

Beyond the budget, real-world campaigns frequently carry auxiliary requirements, a common one being the CPA constraint He et al. (2021). A competent auto-bidding strategy must therefore drive the advertiser's overall utility to its maximum while keeping every such constraint satisfied. This goal can be stated formally as:

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

where o_i denotes whether the advertiser wins the i-th impression, v_i denotes its value, B denotes the advertiser's budget, c_{ij} denotes the cost incurred under constraint j for impression i, C_j denotes the advertiser-specified upper bound of the j-th constraint, and p_{ij} denotes the corresponding performance indicator, *e.g.*, conversions. Prior work has shown that the constrained problem above admits a closed-form optimal bid He et al. (2021):

$$bid_i^* = \lambda_0 v_i + \sum_{j=1}^{J} \lambda_j p_{ij} C_j, \tag{2}$$

where bid_i^* denotes the optimal bid for the i-th impression, λ_0 denotes the optimal bidding parameter, J denotes the number of constraints, and λ_j denotes the Lagrange multiplier associated with the j-th constraint. Auto-bidding thus reduces to continually tuning these parameters so that the resulting bids stay optimal under multiple constraints.

2.2 SEQUENTIAL DECISION FOR AUTO-BIDDING

Because both competitor strategies and the distribution of arriving opportunities drift continuously, any fixed setting of the bidding parameters is soon rendered suboptimal. Optimal bidding therefore requires the parameters to be re-adjusted on the fly from real-time market feedback, which naturally casts auto-bidding as a sequential decision-making problem. We formalize this interaction through the following interconnected components, in which the model reads a state s_t, selects an action a_t, receives an immediate reward r_t as feedback, and optimizes toward the cumulative RTG R_t:

- State s_t: a feature vector summarizing the bidding environment at timestep t, such as the remaining budget, the remaining time, and historical bidding information.
- Action a_t: the action applied at timestep t, written as $a_t = (a_t^{\lambda_0}, a_t^{\lambda_1}, \ldots, a_t^{\lambda_J})$.
- Reward r_t: the total value of the impressions won within timestep t. With D_t candidate impressions in this period, it is defined as $r_t = \sum_{i=1}^{D_t} o_i v_i$, where D_t is the number of ad opportunities at timestep t.
- RTG R_t: the reward accumulated from timestep t to the terminal timestep T, defined as $R_t = \sum_{n=t}^{T-1} r_n$.

Under these definitions, the whole bidding procedure is represented as a trajectory $w = (R_0, s_0, a_0, \ldots, R_{T-1}, s_{T-1}, a_{T-1})$.

3 METHODOLOGY

3.1 MOTIVATION

Generative auto-bidding methods have advanced in optimizing bids under various constraint values Li et al. (2026; 2025b); Cai et al. (2026), but numerical constraints such as CPA cannot fully capture advertiser intent. Even the same CPA target may imply different bidding preferences across advertisers, highlighting the need for personalization. By modeling advertiser-specific strategies beyond constraint values, personalized auto-bidding can better align decisions with individual objectives, yet remains largely underexplored. Meanwhile, profile-based methods have shown strong results in other domains Wang et al. (2025); Salemi et al. (2024). In recommendation systems, for instance, user profiles are built to capture individual preferences, allowing the system to provide items that better match each user's interests and needs Zhang (2024). However, auto-bidding differs fundamentally from recommendation in both data structure and optimization objective. As a result, directly transferring existing profile-based methods to auto-bidding often fails to deliver the expected gains.

More specifically, applying profile-based methods to auto-bidding faces several challenges. First, unlike recommendation profiles built from discrete user-item interactions, advertiser profiles must be distilled from historical bidding dynamics, where bids, costs, conversions, and constraints interact over time. This raises the question: *how can we effectively extract multi-aspect profiles from an advertiser's bidding history?* Second, recommendation profiles often carry explicit semantic information, such as item attributes or category labels, which helps the model interpret them. In auto-bidding, however, bidding histories and advertiser categories usually lack clear semantics, leaving the extracted profile as a dense vector whose shared and personalized components are hard to distinguish. This raises the question: *how can the decision model better understand and use the profile?* Third, profile updates are more costly in auto-bidding. New advertisers require cold-start profiles, existing profiles must evolve as delivery continues, and retraining the model for every update is impractical. This raises the question: *how can advertiser profiles be updated efficiently without retraining?* These challenges raise a key yet underexplored question:

How can an auto-bidding framework systematically extract, exploit, and update advertiser profiles to enable personalized bidding decisions?

To this end, we propose **ADAPT**, which consists of a two-stage training paradigm and an adaptation module. During training, Stage 1 (Section 3.2.1) achieves effective profile extraction, while Stage 2 (Section 3.2.2) achieves effective profile exploitation and improves the model's understanding of the profile. The adaptation module (Section 3.3) then efficiently updates profiles whenever a new advertiser arrives or a new delivery period begins.

3.2 TRAINING

We organize the training process into two stages. Stage 1 learns clean and stable advertiser profiles, while stage 2 uses the learned profiles to guide personalized bidding decisions. By assigning each stage a distinct objective, the training process prevents profiles from being overly shaped by action labels while ensuring that they are effectively exploited for decision making.

3.2.1 STAGE 1: PROFILE REPRESENTATION LEARNING

Existing profile-based methods usually construct profiles by aggregating observed behaviors, which works well when user preferences are expressed through discrete and semantically meaningful interactions. In auto-bidding, however, an advertiser's bidding preference cannot be fully described by a single type of information. On the one hand, the advertiser category provides useful static prior knowledge about its business domain. On the other hand, the advertiser's actual bidding strategy is implicitly reflected in historical bidding dynamics, where returns, states, actions, and advertiser information interact over time. Therefore, we represent each advertiser from both static and dynamic perspectives: the static profile captures category-level prior information, while the dynamic profile is learned from bidding history to characterize advertiser-specific strategies.

Specifically, we divide the advertiser profile into a static profile and a dynamic profile. In this work, the static profile represents the advertiser category. For all advertiser categories, we build a trainable embedding table, denoted as $\boldsymbol{S} = \{\boldsymbol{s_0}, \boldsymbol{s_1}, \ldots, \boldsymbol{s_{C-1}}\} \in \mathbb{R}^{C\times d}$, where C is the number of categories and d is the embedding dimension. The dynamic profile captures the personalized bidding strategy extracted from each advertiser's historical delivery periods. We similarly initialize dynamic profiles with a trainable embedding table, denoted as $\boldsymbol{P} = \{\boldsymbol{p_0}, \boldsymbol{p_1}, \ldots, \boldsymbol{p_{N-1}}\} \in \mathbb{R}^{N\times d}$, where N is the number of advertisers, and refer to it as the advertiser memory bank. The dynamic profile of advertiser i, denoted as $\boldsymbol{p}_i$, is referred to as the individual profile. For advertiser i with category c and we sample a fixed-length segment of length K from its historical trajectory w, we first use a lightweight embedder to transform the trajectory into a sequence of trajectory embeddings $\boldsymbol{e} \in \mathbb{R}^{3K\times d}$. Since the goal is to extract the advertiser's strategy information from the entire trajectory rather than to predict the next action, each position should be allowed to attend to both past and future contexts. Therefore, we adopt a bidirectional attention Transformer as the encoder $\mathcal{T}(\cdot)$. To construct the input sequence, we prepend two types of prefix tokens before the trajectory embeddings. The first is the corresponding static profile, which provides category-level strategy information for the advertiser. The second is a learnable cls token $\boldsymbol{cls} \in \mathbb{R}^d$, inspired by ViT Dosovitskiy et al. (2021), which is used to aggregate the strategy information of the entire trajectory. Positional embeddings are then added to the trajectory tokens to encode their temporal order. Finally, the input embeddings $\boldsymbol{x}$ to the encoder are formulated as:

$$\boldsymbol{x} = [\boldsymbol{cls}, \boldsymbol{s_c}, \boldsymbol{e}], \quad \boldsymbol{x} \in \mathbb{R}^{(2+3K)\times d}. \tag{3}$$

After that, we feed the input embeddings into the encoder. The encoded feature corresponding to the cls token is then passed through layer normalization LayerNorm$(\cdot)$. This process is formulated as:

$$\boldsymbol{h} = \text{LayerNorm}(\mathcal{T}(\boldsymbol{x})_0), \quad \boldsymbol{h} \in \mathbb{R}^d, \tag{4}$$

where $\boldsymbol{h}$ denotes the hidden state of the cls token. For the input trajectory of advertiser i, we take the corresponding memory entry $\boldsymbol{p}_i$ from $\boldsymbol{P}$ as its individual profile. Finally, we train Stage 1 with an InfoNCE-style contrastive objective, where $\boldsymbol{p}_i$ serves as the positive sample for $\boldsymbol{h}$ and while the profiles of other advertisers serve as negatives. The optimization process is formulated as:

$$\mathcal{L}_{\text{inf}} = -\log \frac{\exp(\boldsymbol{h}^\top \text{LayerNorm}(\boldsymbol{p}_i)/\tau)}{\sum_{j=0}^{N-1} \exp(\boldsymbol{h}^\top \text{LayerNorm}(\boldsymbol{p}_j)/\tau)}, \tag{5}$$

where τ denotes the temperature coefficient. By optimizing Eq.(5), it learns advertiser-specific dynamic profiles that capture personalized bidding strategies and remain discriminative across advertisers. Overall, the profile representation learning stage extracts strategy-level information from bidding histories and forms advertiser-specific dynamic profiles. It also learns category-aware static profiles and a strategy-aware trajectory encoder. As a result, stage 1 produces clean and informative advertiser profiles, addressing the first limitation.

3.2.2 STAGE 2: DISENTANGLED PROFILE-BASED BIDDING

Stage 1 produces clean and informative advertiser profiles from bidding histories and advertiser categories, but effectively exploiting these profiles remains non-trivial. Unlike recommendation profiles, which often come with explicit semantics such as item categories, textual descriptions, or user interests, auto-bidding profiles are continuous numerical tokens Zhang (2024); Wang et al. (2025). In particular, a dynamic profile may encode both strategy patterns shared across advertisers and preferences specific to an individual advertiser. Directly using such dense profiles forces the decision model to interpret them implicitly, which can lead to insufficient profile utilization and redundant representations.

Based on the above analysis, it is necessary to transform the dense dynamic profile into semantically clearer components before using it for decision making. To do this, we further disentangle the dynamic profile into a common profile and a private profile, so that the decision model can better exploit the strategy information encoded in advertiser profiles. Specifically, the dynamic profile may contain strategy patterns that are commonly shared across advertisers. For example, most advertisers aim to obtain more conversions while satisfying CPA constraints, and such common bidding principles are useful for helping the model understand the general objective of auto-bidding. However, these shared strategies should not be represented by a single identical vector for all advertisers. Different advertisers may emphasize or express the same common strategy in different ways, depending on their own bidding histories and business preferences. Simply averaging all advertiser profiles or using a fixed shared profile would ignore such advertiser-dependent variations, weakening personalization and potentially introducing irrelevant information. Therefore, we use a cross-attention mechanism to extract the common profile. Given the individual profile of advertiser i, denoted as $\boldsymbol{p}_i$, we use it as the query and take the advertiser memory bank $\boldsymbol{P}$ as the keys and values. This process is formulated as:

$$\boldsymbol{p}_{\text{com}} = \text{CrossAttn}(Q = \boldsymbol{p}_i, K = \boldsymbol{P}, V = \boldsymbol{P}), \quad \boldsymbol{p}_{\text{com}} \in \mathbb{R}^d, \tag{6}$$

where $\boldsymbol{p}_{\text{com}}$ denotes the common profile, Q, K, and V represent the query, key, and value embeddings, and $\text{CrossAttn}(Q, K, V)$ denotes a cross-attention layer. By using Eq. (6), the common profile extracts shared bidding patterns in an advertiser-aware manner, rather than relying on a fixed shared representation for all advertisers.

Similarly, different advertisers also have their private strategies beyond the shared ones. These strategies arise from advertiser-specific business goals, budget conditions, or market environments. Without modeling such private strategies, the decision model may overlook individual differences and produce overly homogeneous bidding decisions. Therefore, we extract the private profile directly from the individual profile with an MLP. This process is formulated as:

$$\boldsymbol{p}_{\text{pri}} = \text{MLP}(\boldsymbol{p}_i), \quad \boldsymbol{p}_{\text{pri}} \in \mathbb{R}^d, \tag{7}$$

where $\boldsymbol{p}_{\text{pri}}$ denotes the private profile. By using Eq. (7), the private profile captures advertiser-specific preferences beyond shared bidding patterns. However, extracting the common and private profiles with the above modules does not guarantee that they are semantically separated. The two profiles may still contain overlapping or redundant information, causing the intended disentanglement to collapse into two similar representations. To avoid this, we introduce a correlation loss to encourage the common and private profiles to capture complementary information. Specifically, for each sample, we first center the common and private profiles along the hidden dimension, and then penalize the absolute cosine similarity between the centered vectors. The loss is formulated as:

$$\mathcal{L}_{\text{cor}} = \frac{1}{M} \sum_{z=1}^{M} \frac{\left|\text{Cov}\left(\boldsymbol{p}_{\text{com},z}, \boldsymbol{p}_{\text{pri},z}\right)\right|}{\sqrt{\text{Var}\left(\boldsymbol{p}_{\text{com},z}\right)} \sqrt{\text{Var}\left(\boldsymbol{p}_{\text{pri},z}\right)}}, \quad \boldsymbol{p}_{\text{com},z}, \boldsymbol{p}_{\text{pri},z} \in \mathbb{R}^d, \tag{8}$$

where M denotes the batch size, $\boldsymbol{p}_{\text{com},z}$ and $\boldsymbol{p}_{\text{pri},z}$ denote the common and private profiles of the z-th sample respectively, $\text{Cov}(\cdot,\cdot)$ and $\text{Var}(\cdot)$ denote the covariance and variance computed across the d hidden dimensions within each sample. By optimizing Eq. (8), ADAPT encourages the common and private profiles to be semantically separated, reducing redundancy and improving profile utilization. Next, for advertiser i with category c, we sample a fixed-length segment of length K from its historical trajectory w, and prepend the static, common, and private profiles to the trajectory embeddings $\boldsymbol{e}$:

$$\boldsymbol{y} = [\boldsymbol{s}_c, \boldsymbol{p}_{\text{com}}, \boldsymbol{p}_{\text{pri}}, \boldsymbol{e}], \quad \boldsymbol{y} \in \mathbb{R}^{(3+3K)\times d}. \tag{9}$$

The sequence $\boldsymbol{y}$ is encoded by a causal transformer as $\boldsymbol{H} = \mathrm{Backbone}(\boldsymbol{y})$, where $\boldsymbol{H} \in \mathbb{R}^{(3+3K)\times d}$. Then, the hidden state of each state token, denoted as $\boldsymbol{h}_t^s$, is linearly projected to predict the action, i.e., $\hat{a}_t = \mathrm{Linear}(\boldsymbol{h}_t^s)$. We use the mean squared error (MSE) loss to optimize action prediction:

$$\mathcal{L}_{\mathrm{mse}} = \frac{1}{K}\sum_{t=0}^{K-1} \|a_t - \hat{a}_t\|^2 . \tag{10}$$

The overall training objective of stage 2 is:

$$\mathcal{L} = \mathcal{L}_{\mathrm{mse}} + \alpha * \mathcal{L}_{\mathrm{cor}}, \tag{11}$$

where α denotes the weight factor. Overall, stage 2 enables exploiting advertiser profiles through semantically clearer common and private components, thereby addressing the second limitation.

3.3 Adaptation

Through the two-stage training process, ADAPT learns clean advertiser profiles and effectively exploits them for personalized bidding. However, real-world advertising environments are dynamic: existing advertisers continuously generate new delivery periods, while new advertisers may enter the system at any time. Without profile updates, existing profiles may become outdated and fail to reflect recent bidding strategies; without cold-start profiles, new advertisers cannot receive personalized decisions at the beginning of delivery. Therefore, efficient profile updating and cold-start construction are necessary for practical auto-bidding systems. To address this, we design a training-free adaptation mechanism. It updates the individual profile of an existing advertiser using newly observed delivery periods, and provides suitable strategy guidance for a new advertiser in a zero-shot manner when no learned individual profile is available.

New period. For existing advertisers, new periods may indicate recent strategy changes, making profile updating necessary. Retraining the profile extractor or fine-tuning the memory bank is a straightforward solution, but it is too costly for online advertising systems. Therefore, we update profiles in a training-free manner. Since the Stage 1 transformer has learned to summarize strategy information, we freeze it and encode the new period trajectory embedding $\boldsymbol{x}_{\mathrm{new}} = \{\boldsymbol{cls}, \boldsymbol{s}_c, \boldsymbol{e}_{\mathrm{new}}\}$ of advertiser i with category c: $\boldsymbol{h}_{\mathrm{new}} = \mathrm{LayerNorm}\left(\mathcal{T}(\boldsymbol{x}_{\mathrm{new}})_0\right)$. We then update its individual profile $\boldsymbol{p}_i$ in the memory bank using exponential moving average (EMA):

$$\boldsymbol{p}_i \leftarrow \lambda \boldsymbol{p}_i + (1-\lambda)\boldsymbol{h}_{\mathrm{new}}, \tag{12}$$

where λ controls the update strength. Eq. (12) enables the individual profile to adapt to new bidding strategies while preserving historical information, without requiring retraining.

New advertiser. New advertisers may enter the bidding platform at any time, but they have no historical trajectories during training and thus no individual profiles in the memory bank. Retraining a profile for each new advertiser is inefficient, so we provide zero-shot strategy guidance using available profile information. Specifically, for a new advertiser j with category c, we use its category static profile $\boldsymbol{s}_c$, average the common profiles of existing advertisers as prototype common profile $\bar{\boldsymbol{p}}_{\mathrm{com}}$, and set the private profile to a zero vector since no advertiser-specific strategy is available. The inference input is formulated as:

$$\boldsymbol{y} = [\boldsymbol{s}_c, \bar{\boldsymbol{p}}_{\mathrm{com}}, \boldsymbol{p}^0_{\mathrm{pri}}, \boldsymbol{e}], \quad \bar{\boldsymbol{p}}_{\mathrm{com}}, \boldsymbol{p}^0_{\mathrm{pri}} \in \mathbb{R}^d, \tag{13}$$

where $\boldsymbol{p}^0_{\mathrm{pri}}$ denotes the zero private profile. By using Eq. (13), ADAPT guides bidding decisions for new advertisers in a zero-shot manner, without requiring any additional training or model modification. Overall, the adaptation module enables efficient profile updating and zero-shot guidance without retraining, thereby addressing the third limitation.

Finally, ADAPT unifies profile extraction, exploitation, and updating into a complete advertiser profile modeling framework, addressing all three limitations discussed above.

4 Experiments

4.1 Experimental Details

Datasets. We conduct experiments on AuctionNet Su et al. (2024), a large-scale benchmark dataset for advertising bidding released by Alibaba. It provides two versions with different conversion

Table 1: Score comparison between our proposed method and previous baselines on the AuctionNet benchmark under different budget scales. Bold indicates the best performance, and underlined values denote the second-best performance.

Dataset	Budget	USCB	BCQ	CQL	IQL	DiffBid	DT	CDT	DT-score	GAS	ADAPT	Improve
AuctionNet-dense	50%	86	190	113	164	54	191	174	178	<u>193</u>	**210**	8.81%
	75%	135	259	139	232	100	265	242	268	<u>287</u>	**289**	0.70%
	100%	157	321	171	281	152	329	326	334	<u>359</u>	**367**	2.23%
	125%	220	379	201	355	193	396	378	395	<u>409</u>	**420**	2.69%
	150%	281	429	238	401	234	450	433	441	<u>461</u>	**469**	1.74%
AuctionNet-sparse	50%	11.5	17.7	12.8	16.5	9.9	14.8	11.2	17.7	<u>18.4</u>	**18.7**	1.63%
	75%	14.9	24.6	16.7	22.1	15.4	22.9	18.0	25.9	<u>27.5</u>	**29.5**	7.27%
	100%	17.5	31.1	22.2	30.0	19.5	29.6	31.2	33.2	<u>36.1</u>	**37.1**	2.77%
	125%	26.7	34.2	28.6	37.1	25.3	34.3	31.7	39.6	<u>40.0</u>	**42.8**	7.00%
	150%	31.3	37.9	35.8	43.1	30.8	44.5	39.1	44.7	<u>46.5</u>	**47.4**	1.94%

Table 2: Effectiveness of advertiser profiles.

Method	Score ↑	Conv. ↑
w/o Any Prefix (DT)	329	364
w/o Static Profile	348 (+5.78%)	371 (+1.92%)
w/o Dynamic Profile	334 (+1.52%)	379 (+4.12%)
ADAPT	**367** (+11.55%)	**405** (+11.26%)

Table 3: Effectiveness of disentanglement.

Method	Score ↑	Conv. ↑
w/o Disentanglement	344	371
w/o Private Profile	348 (+1.16%)	379 (+2.16%)
w/o Common Profile	358 (+4.07%)	382 (+2.96%)
ADAPT	**367** (+6.69%)	**405** (+9.16%)

sparsity levels, denoted as AuctionNet-dense and AuctionNet-sparse. Both versions contain records from 48 advertisers, and each advertising period involves over 500,000 impression opportunities. Additional dataset statistics are reported in Appendix A.2.

Comparison methods. The comparison methods include four offline RL methods (*i.e.*, USCB He et al. (2021), BCQ Fujimoto et al. (2019), CQL Kumar et al. (2020), IQL Kostrikov et al. (2022)), and five generative auto-bidding methods (*i.e.*, DiffBid Guo et al. (2024), DT Chen et al. (2021), CDT Liu et al. (2023), DT-score Li et al. (2025b), GAS Li et al. (2025b)).

Implementation details. Our method is implemented with a two-stage architecture. Stage 1 employs a bidirectional transformer with 8 layers to encode advertiser profiles via contrastive learning, while stage 2 uses a causal Transformer with 6 layers for sequential bidding decisions. The trajectory window length is K=10, and the correlation loss weight is set to α=1. In the adaptation stage, the EMA coefficient is λ=0.99. Further implementation details are provided in Appendix A.3.

Evaluation metrics. We adopt two metrics for evaluation:

- Conversions: the number of conversions achieved in a delivery period without considering additional constraints, computed as $\sum_i o_i v_i$.
- Score: the constraint-aware value under the CPA requirement. Specifically, we apply a penalty term $\min\left\{\left(\frac{1}{\text{ratio}}\right)^2, 1\right\}$, where ratio denotes the ratio of the achieved CPA to the target CPA. The final score is calculated as $\sum_i o_i v_i \times$ penalty.

4.2 Main Results

We report the experimental results of all methods on the AuctionNet benchmark, including its two variants, AuctionNet-dense and AuctionNet-sparse. Table 1 compares our proposed method with previous auto-bidding methods.

Comparison to baselines. ADAPT achieves the best results across all budget scales on both variants. For instance, under the 100% budget setting, ADAPT improves the score by 2.23% and 2.77% over the strongest baseline (*i.e.*, GAS) on AuctionNet-dense and AuctionNet-sparse, respectively, and by 141.45% and 90.26% over the weakest baseline (*i.e.*, DiffBid). The improvements mainly come from the advertiser profile modeling in ADAPT, which extracts strategy-level information from historical bidding trajectories and uses disentangled common and private profiles to provide personalized bidding guidance. This is especially helpful in sparse scenarios, where limited conversion feedback makes direct policy learning more difficult.

4.3 Ablation Studies

We conduct a series of ablation studies to analyze the main components of ADAPT. Specifically, Table 2 evaluates the effectiveness of advertiser profiles; Table 3 studies the effectiveness of profile

Table 4: Effectiveness of zero-shot new advertiser adaptation.

Method	Score ↑	Conv. ↑
DT	66	73
Zero Profiles	106	128
Static Profile	113 (+6.60%)	134 (+4.69%)
Common Profile	114 (+7.55%)	135 (+5.47%)
ADAPT (Zero-shot)	**134** (+26.42%)	**151** (+17.97%)

Table 5: Effectiveness of training-free new period update.

Method	Score ↑	Conv. ↑
w/o Update	167	194
ADAPT	**176** (+5.39%)	**200** (+3.09%)

disentanglement; Table 4 examines the effectiveness of zero-shot new advertiser adaptation; and Table 5 investigates the effectiveness of training-free new period update. In addition, profile sensitivity analysis, hyperparameter studies, and profile visualization, are provided in Appendix A.4.

Effectiveness of advertiser profiles. To evaluate the effectiveness of advertiser profiles, we compare ADAPT with variants that remove all profiles, use only the static profile, or use only the dynamic profile. As shown in Table 2, using only the dynamic profile improves score and conversions by 5.78% and 1.92% over the no-profile variant, while using only the static profile improves them by 1.52% and 4.12%. When both profiles are used, ADAPT achieves the best performance, with 11.55% and 11.26% improvements in score and conversions. These results show that both profiles provide useful advertiser information, and their combination offers more effective guidance for bidding decisions.

Effectiveness of profile disentanglement. To verify the effectiveness of profile disentanglement, we compare ADAPT with variants that directly use the original individual profile, use only the common profile, or use only the private profile. As shown in Table 3, using only the common profile improves the score and conversions by 1.16% and 2.16% over the non-disentangled variant, while using only the private profile improves them by 4.07% and 2.96%, respectively. When both common and private profiles are used, ADAPT achieves the best results, improving the score and conversions by 6.69% and 9.16%. These results indicate that common and private profiles capture complementary strategy information, and their disentanglement enables more effective profile utilization.

Zero-shot new advertiser adaptation. To evaluate whether the adaptation module can generalize to unseen advertisers, we randomly select advertisers from different categories, remove all their data during the two-stage training, and test the model on these unseen advertisers. For fair comparison, DT is trained and evaluated under the same setting, and we report the averaged results over all selected advertisers in Table 4. DT achieves a score of 66, indicating that previous methods lack explicit strategy guidance for unseen advertisers. In contrast, adding the static profile improves the score and conversions by 6.60% and 4.69%, while adding the common profile further improves them by 7.55% and 5.47%. With the full zero-shot design, ADAPT achieves the best performance, improving the score and conversions by 26.42% and 17.97% over the zero-profile setting. These results show that the proposed profiles provide effective guidance for new advertisers and enable zero-shot generalization beyond previous auto-bidding methods.

Training-free new period update. To verify whether ADAPT can adapt to newly observed bidding periods without training, we hold out 500 periods from the training set and do not use them during training. These periods are then treated as new periods to update advertiser profiles with the proposed EMA-based mechanism, while all model parameters remain frozen. As shown in Table 5, ADAPT improves the score and conversions by 5.39% and 3.09% over the variant without EMA update. These results indicate that the proposed update mechanism can incorporate recent bidding information into advertiser profiles without retraining, enabling efficient adaptation to new periods.

5 Conclusion

In this paper, we propose **ADAPT**, a profile-based framework for personalized auto-bidding. Specifically, ADAPT first learns static and dynamic advertiser profiles from advertiser categories and historical bidding trajectories. It then disentangles the dynamic profile into common and private profiles to better capture shared bidding patterns and advertiser-specific preferences. Finally, ADAPT introduces a training-free adaptation mechanism to update profiles for new periods and provide zero-shot guidance for new advertisers. Experimental results demonstrate the effectiveness of ADAPT in improving bidding performance and enhancing adaptability in dynamic advertising environments.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language polishing, grammar correction, readability improvement, and minor LaTeX formatting. We also used generative AI tools for limited code-related assistance, such as identifying syntax issues and improving code clarity. We did not use generative AI tools to generate datasets, design the core methodology, formulate mathematical claims, conduct experiments, or interpret experimental results. All AI-assisted text and code suggestions were carefully reviewed, revised, and verified by the authors. The authors take full responsibility for the final content of this work, including all claims, experimental results, and artifacts.

REFERENCES

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua B Tenenbaum, Tommi S Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision making? In *ICLR*, pp. 1–24, 2023.

Xiang Ao, Xiting Wang, Ling Luo, Ying Qiao, Qing He, and Xing Xie. Pens: A dataset and generic framework for personalized news headline generation. In *ACL*, pp. 82–92, 2021.

Songyue Cai, Lianyu Wang, Shan Gu, Ziru Xu, Jian Xu, Xiaofeng Zhu, and Bo Zheng. Strategy-aware parameter-efficient adaptation for llm-based auto-bidding. *arXiv*, 2026.

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. In *NeurIPS*, pp. 15084–15097, 2021.

Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. Sequential recommendation with user memory networks. In *WSDM*, pp. 108–116, 2018.

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

Miduo Cui, Haochen Wang, Shangqin Mao, Xun Yang, Qianlong Xie, Xingxing Wang, Xuri Ge, Ying Zhou, and Zhiwei Xu. Drive: Distributional and retrieval-augmented bidding with value evaluation. In *ICML*, pp. 1–23, 2026.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, pp. 1–21, 2021.

David S Evans. The online advertising industry: Economics, evolution, and privacy. *JEP*, pp. 37–60, 2009.

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

Jingtong Gao, Yewen Li, Shuai Mao, Peng Jiang, Nan Jiang, Yejing Wang, Qingpeng Cai, Fei Pan, Peng Jiang, Kun Gai, et al. Generative auto-bidding with value-guided explorations. In *SIGIR*, pp. 244–254, 2025a.

Yulong Gao, Wan Jiang, Mingzhe Cao, Xuepu Wang, Zeyu Pan, Haonan Yang, Ye Liu, and Xin Yang. Segb: Self-evolved generative bidding with local autoregressive diffusion. *arXiv*, pp. 1–11, 2025b.

Jiayan Guo, Yusen Huo, Zhilin Zhang, Tianyu Wang, Chuan Yu, Jian Xu, Bo Zheng, and Yan Zhang. Generative auto-bidding via conditional diffusion modeling. In *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 *KDD*, pp. 2993–3001, 2021.

Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *ICML*, pp. 9902–9915, 2022.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *ICLR*, pp. 1–13, 2022.

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

Yu Lei, Jiayang Zhao, Yilei Zhao, Zhaoqi Zhang, Linyou Cai, Qianlong Xie, and Xingxing Wang. Generative large-scale pre-trained models for automated ad bidding optimization. In *KDD*, pp. 2266–2277, 2026.

Yewen Li, Jingtong Gao, Nan Jiang, Shuai Mao, Ruyi An, Fei Pan, Xiangyu Zhao, Bo An, Qingpeng Cai, and Peng Jiang. Generative auto-bidding in large-scale competitive auctions via diffusion completer-aligner. *arXiv*, 2025a.

Yewen Li, Shuai Mao, Jingtong Gao, Nan Jiang, Yunjian Xu, Qingpeng Cai, Fei Pan, Peng Jiang, and Bo An. Gas: Generative auto-bidding with post-training search. In *WWW*, pp. 315–324, 2025b.

Yewen Li, Zhiyi Lyu, Peng Jiang, Qingpeng Cai, Fei Pan, Bo An, and Peng Jiang. Lbm: Hierarchical large auto-bidding model via reasoning and acting. In *WWW*, pp. 75–86, 2026.

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

Xiaowei Lv, Zhilin Zhang, Yijun Li, Yusen Huo, Siyuan Ju, Xuyan Li, Chunxiang Hong, Tianyu Wang, Yongcai Wang, Peng Sun, et al. Decisionllm: Large language models for long sequence decision exploration. *arXiv*, pp. 1–14, 2026.

Zhiyu Mou, Yusen Huo, Rongquan Bai, Mingzhou Xie, Chuan Yu, Jian Xu, and Bo Zheng. Sustainable online reinforcement learning for auto-bidding. In *NeurIPS*, pp. 2651–2663, 2022.

S. Muthukrishnan. Ad exchanges: Research issues. In *WINE*, pp. 1–12, 2009.

Kan Ren, Weinan Zhang, Ke Chang, Yifei Rong, Yong Yu, and Jun Wang. Bidding machine: Learning to bid for directly optimizing profits in display advertising. *TKDE*, pp. 645–659, 2017.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. Lamp: When large language models meet personalization. In *ACL*, pp. 7370–7392, 2024.

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 *NeurIPS*, pp. 94428–94452, 2024.

Jaime Teevan, Susan T Dumais, and Eric Horvitz. Personalizing search via automated analysis of interests and activities. In *SIGIR*, pp. 449–456, 2005.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pp. 6000–6010, 2017.

Jun Wang and Shuai Yuan. Real-time bidding: A new frontier of computational advertising research. In *WSDM*, pp. 415–416, 2015.

Lu Wang, Di Zhang, Fangkai Yang, Pu Zhao, Jianfeng Liu, Yuefeng Zhan, Hao Sun, Qingwei Lin, Weiwei Deng, Dongmei Zhang, et al. Lettingo: Explore user profile generation for recommendation system. In *KDD*, pp. 2985–2995, 2025.

Chao Wen, Miao Xu, Zhilin Zhang, Zhenzhe Zheng, Yuhui Wang, Xiangyu Liu, Yu Rong, Dong Xie, Xiaoyang Tan, Chuan Yu, et al. A cooperative-competitive multi-agent framework for auto-bidding in online advertising. In *WSDM*, pp. 1129–1139, 2022.

Fan Yang, Zheng Chen, Ziyan Jiang, Eunah Cho, Xiaojiang Huang, and Yanbin Lu. Palr: Personalization aware llms for recommendation. *arXiv*, 2023.

Xun Yang, Yasong Li, Hao Wang, Di Wu, Qing Tan, Jian Xu, and Kun Gai. Bid optimization by multivariable control in display advertising. In *KDD*, pp. 1966–1974, 2019.

Hao Yu, Michael Neely, and Xiaohan Wei. Online convex optimization with stochastic constraints. In *NeurIPS*, pp. 1–11, 2017.

Jiarui Zhang. Guided profile generation improves personalization with large language models. In *EMNLP*, pp. 4005–4016, 2024.

Mingming Zhang, Feiqing Zhuang, Na Li, Shengjie Sun, Xiaowei Chen, Junxiong Zhu, Fei Xiao, Keping Yang, Lixin Zou, and Chenliang Li. Generative auto-bidding with unified modeling and exploration. *arXiv*, pp. 1–11, 2026a.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a dog, do you have pets too? In *ACL*, pp. 2204–2213, 2018.

Weinan Zhang, Yifei Rong, Jun Wang, Tianchi Zhu, and Xiaofan Wang. Feedback control of real-time display advertising. In *WSDM*, pp. 407–416, 2016.

Yi Zhang, Yiwen Zhang, Yu Wang, Tong Chen, and Hongzhi Yin. Proex: A unified framework leveraging large language model with profile extrapolation for recommendation. In *KDD*, pp. 1940–1951, 2026b.

Guanyu Zhu, Jining Luan, Hanwen Du, Xinyu Fang, Sibo Xu, Ersheng Ni, Hongji Li, Jincheng Fang, Ronghao Chen, Huacan Wang, et al. On the role of language representations in auto-bidding: Findings and implications. *arXiv*, pp. 1–19, 2026.

A Appendix

A.1 Related Work

Auto-bidding methods. With the rapid growth of e-commerce platforms, auto-bidding for online advertising has become an important research topic. Early studies mainly relied on control-based methods Chen et al. (2011); Yu et al. (2017); Yang et al. (2019); Zhang et al. (2016), such as PID Chen et al. (2011) and OnlineLP Yu et al. (2017), which adjust bidding strategies through manually designed optimization or control rules. However, as auction environments become increasingly dynamic and the scale of advertising traffic continues to grow, these methods face significant challenges in scalability and adaptability.

Benefiting from the rapid development of reinforcement learning and generative modeling, recent auto-bidding methods can be broadly grouped into two categories: traditional offline RL methods and generative auto-bidding methods. Traditional offline RL methods learn bidding policies by optimizing long-term rewards through value estimation and policy optimization Kumar et al. (2020); Kostrikov et al. (2022); Mou et al. (2022); Wen et al. (2022). For example, SORL Mou et al. (2022) combines offline pretraining with online RL to improve bidding stability, while MAAB Wen et al. (2022) models auto-bidding as a cooperative-competitive multi-agent optimization problem.

Generative auto-bidding methods instead formulate bidding as a trajectory generation or sequence modeling problem and directly produce bidding actions from historical trajectories or auxiliary information Chen et al. (2021); Gao et al. (2025a); Lei et al. (2026); Zhang et al. (2026a). For example, GAS Li et al. (2025b) enhances generative bidding through post-training search, while DRIVE Cui et al. (2026) leverages distributional modeling and retrieval augmentation to improve decision quality. Some methods further incorporate planning mechanisms to guide current bidding decisions by modeling future dynamics Gao et al. (2025b); Guo et al. (2024); Janner et al. (2022); Li et al. (2025a). For instance, DiffBid Guo et al. (2024) models bidding behaviors through conditional diffusion processes, while SEGB Gao et al. (2025b) employs self-evolved autoregressive diffusion for long-horizon bidding planning. Recently, LLM-based multi-modal methods have extended generative auto-bidding by leveraging the reasoning and language understanding capabilities of LLMs. These methods use textual information to model advertiser objectives and constraints while benefiting from the long-context reasoning ability of LLMs Lv et al. (2026); Li et al. (2026); Zhu et al. (2026); Cai et al. (2026). For example, LBM Li et al. (2026) adopts a hierarchical reasoning-and-acting framework for bidding decision generation, SemBid Zhu et al. (2026) investigates the role of language representations in improving auto-bidding performance, and SAGE Cai et al. (2026) introduces a strategy-aware parameter-efficient adaptation framework for LLM-based auto-bidding.

Profile-based personalization. Profiles play an important role in personalized decision-making, as they summarize user preferences, behavioral patterns, and contextual characteristics into compact representations. They have been widely studied in recommendation systems, personalized generation, and personalized retrieval Wang et al. (2025); Zhang et al. (2026b); Zhang (2024); Teevan et al. (2005); Salemi et al. (2024). In recommendation, user profiles are commonly constructed from historical interactions or side information to better match users with items. Recent studies further leverage LLMs to generate more interpretable and semantically rich profiles. For example, LettinGo Wang et al. (2025) uses multiple LLMs to explore diverse user profiles and aligns profile generation with downstream recommendation performance through preference optimization. PALR Yang et al. (2023) incorporates user interaction histories into LLM-based recommendation and fine-tunes a large ranking model to select preferred items from retrieved candidates. GPG Zhang (2024) generates natural-language user profiles from sparse and complex personal contexts, enabling LLMs to better capture user habits and preferences. ProEx Zhang et al. (2026b) further constructs multiple profiles for each user or item to model different aspects of user intent and improve both discriminative and generative recommendation models.

Beyond recommendation, profiles have also been used for personalized generation and retrieval. In personalized search, user profiles or implicit behavioral histories, such as previous queries, browsing records, and clicked pages, are used to infer user intent and re-rank retrieval results. In personalized language generation, profiles provide user-specific context for generating outputs that better match individual preferences and writing styles Teevan et al. (2005); Salemi et al. (2024). For instance, LaMP Salemi et al. (2024) provides a benchmark for personalized language modeling and studies

retrieval-augmented methods that select relevant profile items to condition model outputs. Prior work on personalized dialogue, review generation, and headline generation also shows that incorporating user or persona profiles can improve the relevance and consistency of generated content Teevan et al. (2005); Zhang et al. (2018); Ao et al. (2021). Different from these domains, advertiser profiles in auto-bidding are usually derived from continuous bidding trajectories rather than explicit text or interaction semantics, making profile construction and utilization more challenging.

A.2 Dataset Details

Table 6: Parameters of AuctionNet-dense and AuctionNet-sparse.

Parameters	AuctionNet-dense	AuctionNet-sparse
Trajectories	479,376	479,376
Delivery periods	9,987	9,987
timestep in a trajectory	48	48
RTG dimension	1	1
State dimension	16	16
Action dimension	1	1
Action range	[0, 493]	[0, 589]
Impression's value range	[0, 1]	[0, 1]
CPA range	[6, 12]	[60, 130]
Total conversion range	[0, 1512]	[0, 57]

In our experiments, we adopt AuctionNet Su et al. (2024), a large-scale public advertising bidding benchmark proposed and constructed by Alibaba. Specifically, the dataset provides two variants with different conversion sparsity levels, referred to in this paper as AuctionNet-dense and AuctionNet-sparse. Each variant contains over 500,000 impression opportunities from 48 advertisers across 9987 delivery periods. Detailed dataset statistics are shown in Table 6. In addition, the state at each timestep in a trajectory is constructed by aggregating historical statistics and current budget-related information. Specifically, each dimension represents the following information:

time_left: The number of remaining decision intervals in the current advertising delivery period.

budget_left: The remaining budget available for allocation in the current delivery period.

historical_bid_mean: The average bid submitted by the advertiser over all previous timesteps.

last_three_bid_mean: The moving average bid over the last three timesteps.

historical_LeastWinningCost_mean: The average market clearing price over all previous timesteps, *i.e.*, the minimum cost required to win an auction.

last_three_LeastWinningCost_mean: The average market clearing price over the last three timesteps.

historical_pValues_mean: The average predicted conversion probability over all previous timesteps.

last_three_pValues_mean: The average predicted conversion probability over the last three timesteps.

current_pValues_mean: The average predicted conversion probability of impression opportunities arriving at the current timestep.

historical_conversion_mean: The average number of conversions achieved over all previous timesteps.

last_three_conversion_mean: The average number of conversions over the last three timesteps.

historical_xi_mean: The historical average winning rate, defined as the ratio between the number of won auctions and the total number of participated auctions.

last_three_xi_mean: The average winning rate over the last three timesteps.

current_pv_num: The total number of available impression opportunities at the current timestep.

last_three_pv_num_total: The cumulative number of impression opportunities over the last three timesteps.

historical_pv_num_total: The cumulative number of impression opportunities over all previous timesteps.

Table 7: Detailed hyperparameter settings for ADAPT.

Hyperparameter	**Value**
Layer of stage 1 transformer	8
Layer of stage 2 transformer	6
Head of stage 1 transformer	8
Head of stage 2 transformer	8
Batch size	128
Total training steps of stage 1	200000
Total training steps of stage 2	400000
Learning rate	1e-5
Weight decay	1e-4
Optimizer	AdamW
Stage-1/2 trajectory length K	10
Adaptation trajectory length K	48
Episode length	48

Table 8: Sensitivity analysis of advertiser profiles. At inference time we perturb the static and/or dynamic profile by replacing it with a zero vector (*Zero*) or by randomly reassigning it to another advertiser (*Shuffle*); *Intact* denotes the unperturbed profile.

Static Profile			**Dynamic Profile**			**Score ↑**	**Conv. ↑**
Intact	Zero	Shuffle	Intact	Zero	Shuffle		
	✓		✓			361	394
		✓	✓			341	390
✓				✓		347	373
✓					✓	348	386
	✓			✓		352	393
		✓			✓	334	377
✓			✓			**367**	**405**

A.3 Implementation Details

Table 7 presents the detailed hyperparameter configurations of the proposed ADAPT method. Notably, the hyperparameter settings remain largely consistent across different benchmark variants, which further demonstrates the robustness of our method.

A.4 Additional Ablation Studies and Visualizations

Profile sensitivity analysis. To further verify whether the learned profiles contain meaningful advertiser information, we perturb the static and dynamic profiles at inference time and report the results in Table 8. Specifically, we either replace a profile with a zero vector or randomly shuffle it across advertisers, while keeping the model parameters unchanged. The results show that perturbing either profile consistently degrades performance. For example, shuffling the static profile reduces the score from 367 to 341, while zeroing and shuffling the dynamic profile reduce the score to 347 and 348, respectively. When both profiles are shuffled, the score further drops to 334. These results indicate that the learned profiles are not redundant prefix tokens, but encode meaningful category-level and advertiser-specific strategy information for bidding decisions, further validating the effectiveness of our profile modeling design.

Hyperparameter sensitivity analysis. To analyze the effect of the correlation loss, we vary the loss weight α in $\mathcal{L} = \mathcal{L}_{\text{mse}} + \alpha * \mathcal{L}_{\text{cor}}$ and report the results in Figure 2. When $\alpha = 0$, the correlation loss is removed and the model achieves a score of 352. Introducing the correlation loss consistently improves performance, with the best score obtained at $\alpha = 1$. In addition, ADAPT maintains strong performance across a reasonable range of α, such as 0.1, 1, and 10, indicating that the method is not overly sensitive to this hyperparameter. These results verify the effectiveness of the correlation loss and demonstrate the robustness of ADAPT to the choice of α.

Profile Visualization and Similarity Analysis. To further examine whether the learned profiles are consistent with their expected semantics, we conduct visualization and similarity analysis for

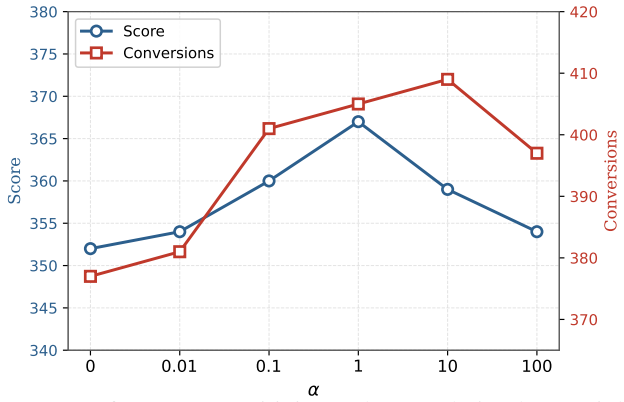


Figure 2: Performance sensitivity to the correlation loss weight α.

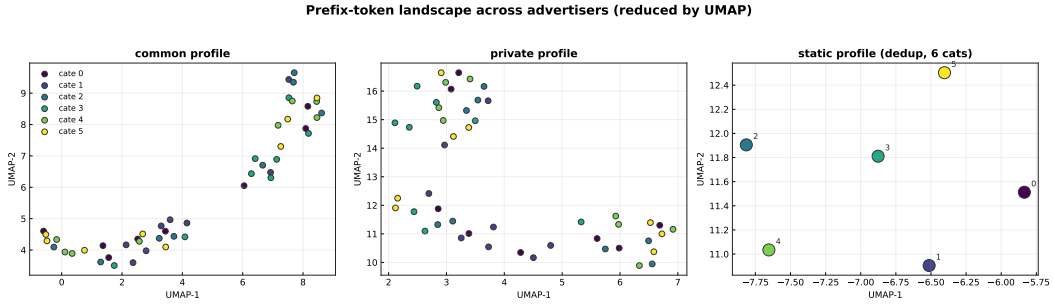


Figure 3: UMAP visualization of learned advertiser profiles.

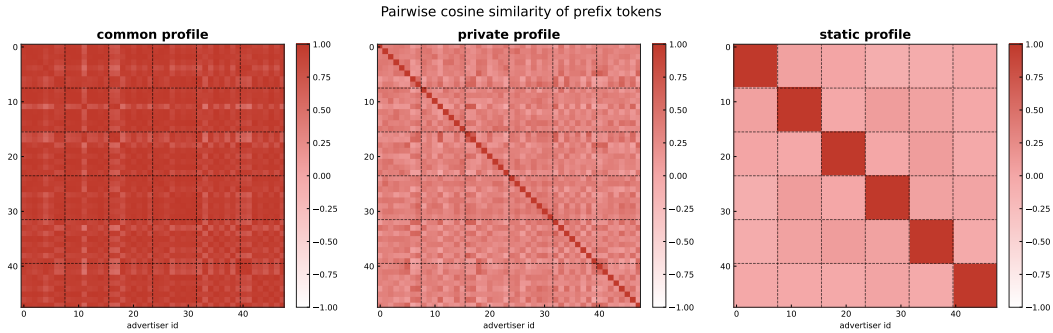


Figure 4: Visualization of pairwise cosine similarity among learned advertiser profiles.

the static, common, and private profiles. Specifically, we first project the learned profiles into a two-dimensional space using UMAP, as shown in Figure 3. We then visualize the pairwise cosine similarity between advertisers for each type of profile in Figure 4. In addition, we report the mean similarity, intra-category similarity, and inter-category similarity in Table 9.

From Figure 3, we observe that common profiles are relatively clustered, indicating that they capture shared bidding patterns across advertisers. In contrast, private profiles are more scattered, suggesting that they encode advertiser-specific strategy preferences. The static profiles form separated category-level representations and are clearly different from the dynamic profiles. Moreover, the common and private profiles occupy largely different regions, showing that the disentanglement process separates shared and individual strategy information.

Figure 4 further visualizes the pairwise similarity patterns. The common profiles show high similarity across advertisers, indicating that they represent general bidding knowledge without strong advertiser-specific preference. The private profiles exhibit high similarity mainly on the diagonal, while the similarities between different advertisers are much lower, confirming their personalized nature. For the static profiles, advertisers belonging to the same category share the same category-level representation and therefore show higher within-category similarity than cross-category similarity, indicating that static profiles encode category-specific information.

Table 9: Pairwise cosine similarity statistics of prefix profiles.

Profile	Mean	Intra-category	Inter-category
Common Profile	0.9027	0.8988	0.9034
Private Profile	0.3218	0.3158	0.3228
Static Profile	0.1532	1.0000	0.0050

Table 9 provides quantitative evidence for these observations. The common profiles have a high mean similarity of 0.9027, while the private profiles have a much lower mean similarity of 0.3218, confirming that common profiles capture shared strategies and private profiles preserve personalized information. Moreover, both common and private profiles show similar intra-category and inter-category similarities, indicating that they contain limited category-specific information after disentanglement. In contrast, the static profiles show high intra-category similarity and very low inter-category similarity, demonstrating their role in providing complementary category-level information. Overall, these results show that ADAPT learns meaningful profiles with distinct semantics, and the static, common, and private profiles are complementary and consistent with their expected roles.