CAAI Transactions on Intelligence Technology

WILEY

The Institution of Engineering and Technology

ORIGINAL RESEARCH OPEN ACCESS

AGT: Efficient Offline Reinforcement Learning With Advantage-Guided Transformer

Jiaye Wei[1] | Xin Xu[1] | Yixing Lan[1] | Tenglong Liu[1] | Yueying Wang[2]

[1]College of Intelligence Science and Technology, National University of Defense Technology, Changsha, China | [2]School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, China

Correspondence: Yixing Lan (lanyixing16@nudt.edu.cn)

Received: 30 June 2025 | **Revised:** 23 October 2025 | **Accepted:** 18 November 2025

Keywords: artificial intelligence | artificial neural network | learning (artificial intelligence) | machine learning

ABSTRACT

Offline reinforcement learning (RL) is a paradigm that seeks to train policies directly based on fixed datasets derived from previous interactions with the environment. However, offline RL faces critical challenges in environments characterised by sparse rewards and datasets dominated by suboptimal trajectories. In this article, we propose a simple and efficient offline RL framework via advantage-guided transformer (AGT). Different from previous works, AGT promotes an advantage-guided policy constraint into the loss function during policy training that adaptively learns the features of trajectories with high advantage values. Furthermore, AGT samples actions with high advantage values during action generation, replacing manual designs with advantage values to update the return-to-go (RTG) as conditional inputs, to ensure action consistency with expected returns. Theoretical analysis shows that AGT's policy improvement guarantees convergence to near-optimal policies in environments with low stochasticity. Extensive experiments on D4RL benchmarks, including robotic locomotion control, dexterous manipulation and maze navigation tasks, show AGT outperforms state-of-the-art (SOTA) offline RL methods.

1 | Introduction

Offline reinforcement learning (RL) [1] is a paradigm that seeks to train policies directly based on fixed datasets derived from previous interactions with the environment. Effective offline RL algorithms hold significant promise where online interaction is impractical or data collection is expensive [2], for example, in education [3], healthcare [4], autonomous driving [5]and robotics [6]. According to exiting approaches to policy learning, methods in offline RL can mainly be classified into two types: (1) value-based methods [7] and (2) conditional sequence modelling (CSM) methods [8].

Value-based offline RL methods learn both a policy and an action-value function and are typically executed in an iterative manner, alternating swiftly between the steps of policy evaluation and policy improvement [9]. Conservative Q-learning (CQL) [10] regularises Q-value estimation by incorporating a conservative penalty term. Implicit Q-Learning (IQL) [11] learns a Q-function by implicitly modelling the behaviour policy. Policy regularisation is pivotal research in these methods for addressing distribution shift challenges [2]. TD3+BC [12] enhances policy by incorporating a straightforward behaviour cloning term. BEAR [13] utilises maximum mean discrepancy (MMD) with a Gaussian kernel as a divergence regularisation. BCQ [14] and LAPO [15] utilise a conditional VAE [16] to model the behaviour policy, incorporating it as an implicit constraint.

Unlike value-based offline RL methods, CSM methods utilise the state-action-reward triplets derived from offline trajectories to generate action predictions by inputting triplets into the causal transformer [17]. The decision transformer (DT) [8] first integrates the conditional policy approach into a transformer

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2025 The Author(s). *CAAI Transactions on Intelligence Technology* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology and Chongqing University of Technology.

architecture, which shows competitive performance, especially on sparse rewards [18]. Nevertheless, the CSM methods lack the stitching abilities which refers to the capability of integrating segments of suboptimal trajectories to generate a near-optimal trajectory [7], especially in stochastic environments. The fundamental reason is that condition only on cumulative returns such as return-to-go (RTG) can lead to incorrect decision-making irrespective of the scale or volume of training data [19]. CSM methods implicitly assume that actions leading to a specific goal are optimal for achieving that goal. However, this assumption does not hold in stochastic environments.

To address the challenges associated with the lack of stitching capability in CSM methods within stochastic environments, alternative strategies have been suggested. For example, environment-stochasticity-independent representation (ESPER) [19] employs clustering techniques to group trajectories and utilises the average returns of these clusters as conditions to guide subsequent policy learning. Dichotomy of control (DoC) [20] utilises a latent variable representation of the future to condition the policy, thereby minimising mutual information to achieve a representation that is agnostic to stochasticity. The elastic decision transformer (EDT) [21] optimises the trajectory by preserving a longer history when the previous trajectory is optimal and a shorter history when it is suboptimal. Those methods above enhance the trajectory stitching capability of CSM approaches by reshaping estimated RTG from the entire distribution of future trajectories to guide policy generation. However, they often require complex representation learning objectives and statistical steps [22–24], complicating the training process and increasing the calculation burdens.

Inspired by the ideas of value-based methods, many approaches that utilise the value-based methods to strengthen the stitching abilities have achieved good performance [25]. For instance, Q-learning decision transformer (QDT) [26] firstly combines the CSM methods with Q-learning. It employs the results of dynamic programming to relabel the RTG in the training data, subsequently using the relabelled data to train the decision transformer. Critic-guided decision transformer (CGDT) [7] utilises a value function which is similar to the critic network to guide policy training and it ensures the expected returns of actions consistent with concrete RTG. Improvements above are essentially regarded as basic data augmentation in which 'stitched' trajectories with predicted RTG is integrated into the training set. Nonetheless, unmatched RTG is still encountered during inference, leading to the failure of surpassing existing benchmarks [27].

Based on the cognitive mechanism that value-based prioritisation in working memory enhances long-term retention of critical information [28], where selective attention to high-value data improves task performance, we propose the Advantage-Guided Transformer (AGT), which integrates the advantage function [29] into CSM methods. Our approach combines the ability of sequence modelling with the adaptive data augmentation while also leveraging the predictability of future returns based on insights from the actor-critic method. We learn both the value function and the action value function to calculate the advantage function of each sequence [30]. Then, the relevant trajectories with low advantage values will be masked, and the learning process will focus on trajectories with high advantage values. By utilising the advantage-guided policy constraint, the model adaptively learns sequences with high advantage functions in the trajectory and masks suboptimal action sequences to achieve stitching abilities. AGT also samples actions with high advantage values during action, effectively replacing manual designs with advantage values to update the return-to-go (RTG) as conditional inputs. This mechanism ensures action consistency with expected returns, enhancing the AGT's ability to make coherent decisions. By conditioning on the advantage values, the model emphasises actions that have the potential to lead to greater cumulative rewards, facilitating improved exploration and exploitation throughout the learning process. Evaluation results obtained on D4RL benchmark datasets verify the superior performance of the proposed AGT against the state-of-the-art (SOTA) CSM methods.

The contributions of this paper are summarised as follows:

1. A novel AGT algorithm is designed by leveraging the transformer architecture to achieve precise policy regularisation and advantage-guided policy constraints to prioritise actions that yield higher returns over the baseline, which guides adaptive data augmentation on suboptimal offline datasets while learning near-optimal policy.
2. AGT aims to obtain actions with high advantage consistent with the behaviour policy and replaces manual designs with advantage values to update the return-to-go (RTG) as conditional inputs, facilitating strong stitching capability and efficient sequence modelling in stochastic scenarios characterised by delayed and sparse rewards.
3. Through comprehensive comparisons with traditional value-based methods and recently proposed CSM algorithms on D4RL benchmark tasks, AGT achieves SOTA performance, especially in datasets that contain abundant suboptimal trajectories.

The remainder of this paper is organised as follows. Section 2 briefly reviews the related works. Section 3 introduces the research preliminaries. Section 4 provides detailed descriptions of the proposed algorithms along with the theoretical analysis. Section 5 presents a comprehensive simulation study. Section 6 concludes this paper and suggests future work.

2 | Related Work

2.1 | Offline RL

Offline RL enables agents to learn optimal policy from a static dataset, without the need for online interaction with the environment [2]. Due to the potential divergence between the learnt policy and the behaviour policy, offline algorithms must address the challenge of out-of-distribution [31]. The two commonly used approaches include policy regularisation and data reweighting. Policy regularisation constrains the learnt policy to the behaviour policy, such as MMD [13], Fisher divergence [32] and Wasserstein distance [33]. BEAR [13] employs MMD with a Gaussian kernel as divergence regularisation for policy improvement but it relies on the approximate nature of low-

sampled MMD. BCQ [14] implement the modelling of the behaviour policy using conditional VAE [16] to achieve proximity between the learnt and behaviour policies through an implicit constraint. Data reweighting approaches address the OOD by using an estimated advantage function as the weight without restricting the learnt policy. ReDS [34] emphasises the reweighting of the data distribution specifically for CQL, aiming to establish an approximate formulation of support constraints. AW [35]reweights trajectories based on their returns, a process that necessitates the consideration of entire trajectories. OPER [36] employs a priority function to prioritise the dataset, which improves the learnt policy. A2PR [37] samples actions with high advantage from an augmented behaviour policy combined with VAE to guide the learnt policy. Our methodology is not classified within above categories but it is associated with the concepts of learning pessimistic value functions and weighted imitation learning approaches.

2.2 | Conditional Sequence Modelling Methods

The CSM methods do not train the policy by rapidly alternating between policy evaluation and policy improvement steps. Instead, it leverages state-action-reward tuples derived from offline trajectories to generate action predictions by inputting these tuples into causal transformers. These methods effectively transform offline reinforcement learning into a supervised learning task. Chen [8] first integrates the conditional policy approach into a transformer architecture, which shows competitive performance, especially on sparse rewards. But CSM methods lack the stitching abilities especially in stochastic environments [38]. The inherent uncertainty and approximation errors in behavioural policies exacerbate this inconsistency, leading to poor performance in stitching optimal trajectories when suboptimal trajectories are present [7].

One of the approaches to solve the above problem is reshaping estimated RTG from the entire distribution of future trajectories to guide policy generation. ESPER [19] employs clustering techniques to group trajectories and utilises the average returns of these clusters as conditions to guide subsequent policy learning. DoC [20] utilises a latent variable representation of the future to condition the policy. EDT [21] optimises the trajectory by preserving a longer history when the previous trajectory is optimal and a shorter history when it is suboptimal. QDT [26] firstly combines the CSM methods with Q-learning, which employs the results of dynamic programming to relabel the RTG in the training data.

In comparison, the approaches improved by changing the training paradigm of the decision transformer also achieve good performance. To be specific, CGDT [7] utilises a value function which is similar to the critic network to guide policy training and it ensures the expected returns of actions consistent with concrete RTG. Q-value regularised transformer (QT) leverages the n-step Bellman equation to approximate the Q-value function based on sequential history [27]. Moreover, value-promped autotuned decision transformer (V-ADT) [39] firstly integrates hierarchical reinforcement learning (HRL) [40] into the CSM methods. Instead of utilising return as condition, V-ADT [39] leverages the prompts generated by high-level policy concatenated with states as the input. However, despite these modifications, both QT and V-ADT fundamentally utilise the causal transformer to generate actions directly, similar to the original DT framework. This approach results in the learnt policy closely fitting the behaviour policy of training datasets, even though those actions are suboptimal. QT and V-ADT merely alter the conditional input and loss of DT during the training process, without considering the intrinsic quality of the offline dataset. When training on the datasets dominated by dense low-return and sparse high-return trajectories, simply constraining the input and loss functions hinders policy performance.

In this article, we combine the sequence modelling capability of the transformer with adaptive data augmentation [41]. Advantage function is designed to guide the policy learning to focus more on trajectories with high advantage. Moreover, we use Q-value and value function to achieve regularisation during the training process. In addition, we use the RTG evaluated by the advantage function instead of manual designed by trial and error, which ensures the consistence between expected returns of the sampled actions and the optimal returns.

3 | Preliminaries

This section provides a concise introduction to the background and introduces some key notation. Advantage function is an important concept in the RL problem. An RL problem can be formalised as an MDP defined by a tuple $\langle S, A, P, R, \gamma\rangle$, where S represents state space, A denotes action space, $P : S\times S\times A\to[0,1]$ is the transition function, $R : S\times A\mapsto\mathbb{R}$ is a reward function that provides a one-step immediate reward $r_t = R(\boldsymbol{s}_t, \boldsymbol{a}_t)$ and $\gamma\in(0,1)$ is the discount factor. A mapping $\pi : S\to A$ denotes a policy. Given a policy π, the value function is defined as

$$V^{\pi}(\boldsymbol{s}) = \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty}\gamma^t R(\boldsymbol{s}_t, \pi(\boldsymbol{s}_t))|\boldsymbol{s}_0 = \boldsymbol{s}\right], \tag{1}$$

where $\mathbb{E}^{\pi}$ denotes the expectation with respect to π. Similarly, the action-value function is defined as

$$Q^{\pi}(\boldsymbol{s}, \boldsymbol{a}) = \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty}\gamma^t R(\boldsymbol{s}_t, \boldsymbol{a}_t)|\boldsymbol{s}_0 = \boldsymbol{s}, \boldsymbol{a}_0 = \boldsymbol{a}\right], \tag{2}$$

The advantage function of the action $\boldsymbol{a}$ is defined as

$$A^{\pi}(\boldsymbol{s}, \boldsymbol{a}) = Q^{\pi}(\boldsymbol{s}, \boldsymbol{a}) - V^{\pi}(\boldsymbol{s}). \tag{3}$$

Offline RL, also known as data-driven RL [42], creates a specialised category in RL. In the offline RL setting [2], a static dataset of transitions $\mathcal{D} = \{(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1}, r_t)_i\}$ is provided, where $\boldsymbol{i}$ indexes a transition in the dataset, the action $\boldsymbol{a}_t$ comes from the behaviour policy $\pi_\beta(\cdot|s_t)$, state $\boldsymbol{s}_t$ comes from a distribution induced by the behaviour policy $d^{\pi_\beta}(\cdot)$, the next state $\boldsymbol{s}_{t+1}$ is determined by the transition dynamics $s_{t+1}\sim P(\cdot|\mathbf{s}_t, \mathbf{a}_t)$ and the one-step reward is $r_t = R(\boldsymbol{s}_t, \boldsymbol{a}_t)$. In offline RL, the learning algorithm can only take samples from $\mathcal{D}$ without collecting new

data from the environment and the target is the same as in the online case: to find an optimal policy that maximises the expected return. This paradigm has potential prospects in case of the interaction with the environment being expensive or high risk.

DT architecture casts the RL problem as CSM [26]. In this work, we consider the following trajectory τ representation that enables autoregressive policy training and generation as input:

$$\tau = \left(\hat{R}_{t-K+1}, s_{t-K+1}, a_{t-K+1}, ..., \hat{R}_t, s_t, a_t\right). \tag{4}$$

where $\boldsymbol{K}$ is the context length, and $\hat{R}$ is RTG $\left(\hat{R}_t = \sum_{t'=t}^{T} r_{t'}\right)$. Then, we use the casual transformer to output the next predicted action.

4 | AGT Algorithm

In this section, the AGT algorithm is first proposed. Then, the advantage guided policy constraint is presented to address the issue of insufficient stitching capability, and the advantage mask mechanism is designed to adaptively enhance the policy by prioritising actions with high advantages. Next, an action sampling method that uses the learnt advantage function is designed with more details. Finally, some theoretical analyses are given to illustrate the effectiveness of the CSM method based on the advantage function.

4.1 | Overview of the AGT Algorithm

The intuitive motivation of AGT is to use the advantage function to effectively guide a continuous improvement of policy and enhance the stitching ability of CSM methods. When the advantage function of the trajectory is less than zero, the corresponding low advantage trajectories will be masked, enabling the learning process to concentrate on trajectories with high advantage values. Consider a robotic navigation task where the dataset contains various suboptimal paths. The advantage function acts like an experienced guide that can identify which path segments actually lead efficiently towards the destination. When the advantage value of a trajectory segment is negative (indicating worse-than-average performance), AGT masks out these 'poor directions', allowing the model to focus on learning from the more promising path segments with positive advantages.

The components of AGT are summarised in Figure 1, which includes a CSM module, an advantage function module and an advantage mask module. In the CSM module, a DT [8] architecture is utilised to output the predicted action tuple $\pi(\tau_t)$ from the history trajectory sequence τ_t, encompassing multiple state s_t, action a_t and RTG $\hat{R}_t$ tuples. Compared with value-based offline RL algorithms, DT demonstrates advantages in addressing issues related to reward sparsity and long-term sequence dependencies. The loss function for training the auto-encoder in DT is

$$\mathcal{L}_{DT} = \mathbb{E}_{\tau_t \sim \mathcal{D}}\left[\frac{1}{K}\sum_{i=t-K+1}^{t}\left(\mathbf{a}_i - \pi(\tau_t)_i\right)^2\right]. \tag{5}$$

where K is the sequence length, D is the offline dataset and $\pi(\tau_t)_i$ is the ith predicted action output of the transformer in an auto-regressive manner.

Then, we construct Q-network and V-network along with their respective target networks to calculate the advantage values of each sequence in the advantage function module. The input of

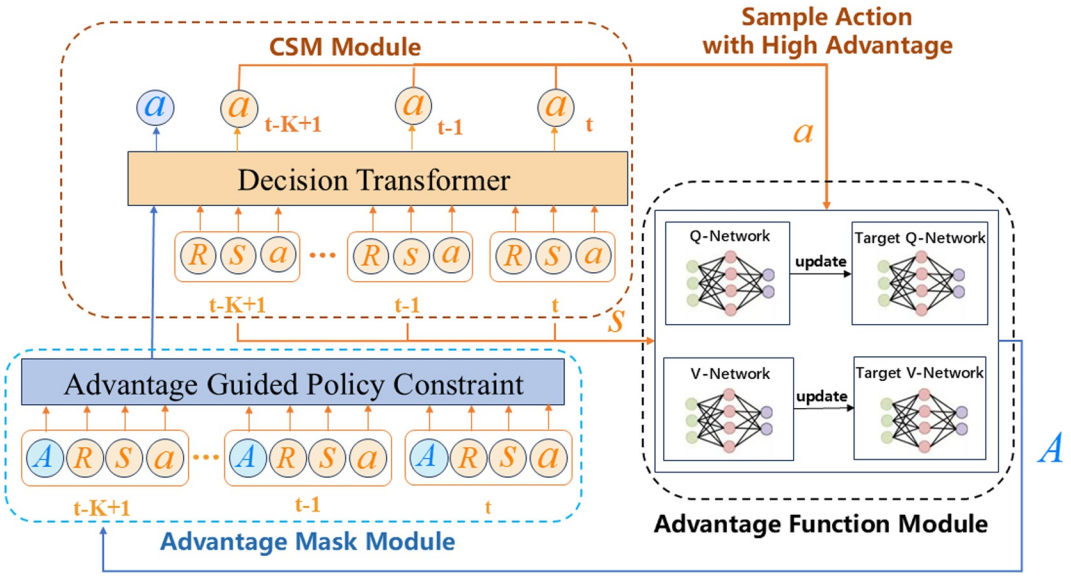


FIGURE 1 | Components of AGT. (i) In the CSM module, the trajectory input $\tau_t = \left(\hat{R}_{t-K+1}, s_{t-K+1}, a_{t-K+1}, ..., \hat{R}_t, s_t, a_t\right)$ is encoded into the decision transformer to the decoder predicted action tuple $\pi(\tau_t) = (\boldsymbol{a}_{t-k+1}, ..., \boldsymbol{a}_{t-1}, \boldsymbol{a}_t)$. (ii) In the advantage function module, we construct Q-network and V-network along with their respective target networks based on both the input and output of the CSM module and trajectory sequence τ_t. Then, we output the advantage function to the next module. (iii) In the advantage mask module, the trajectory with low advantage is masked and the high advantage trajectory is input to the CSM module for policy generation.

this module comes from the offline dataset D and predicted action tuple $\pi(\tau_t) = (\boldsymbol{a}_{t-k+1}, ..., \boldsymbol{a}_{t-1}, \boldsymbol{a}_t)$.

The advantage mask module in AGT uses the advantage values of the trajectory sequence from the advantage function module to determine the quality of each trajectory. When the advantage function of a trajectory is negative, the associated low-advantage trajectory sequence will be masked, facilitating the learning process's autonomously focus on trajectories with high advantage values, thereby ensuring that the final returns align with the expected rewards associated with the task.

AGT introduces two key innovations: a novel policy constraint method based on the advantage function, and an automated approach to generate expected returns. The following sections will discuss these methods and provide a theoretical analysis.

4.2 | Advantage-Guided Policy Constraint

In the previous studies, the loss function of DT (Equation 5) was often used in the training process of CSM methods. However, training solely with DT loss results in the lack of the stitching ability in CSM methods.

Lemma 1. *(Alignment with respect to the conditioning function) Consider an MDP $\langle S, A, P, R, \gamma\rangle$, behaviour policy $\pi(\beta)$, trajectory τ, return-to-go g and policy $\pi(\tau_t)$ learnt by the decision transformer. Assume return coverage: $P_\beta(g = f(s_1)|s_1) \geq \alpha_f$ (P_β refers to the distribution over states, actions and returns induced by β) and near determinism: $Pr \neq r(s,a)$ or $s' \neq \mathcal{T}(s,a)|s,a) \leq \epsilon$ at all s, a for some functions $\mathcal{T}$ and r (note that this does not constrain the stochasticity of the initial state). Let RTG $\hat{r}(\tau) = \sum_{t'=t}^{\mathcal{H}} r_{t'}$, when for all initial states, we have:*

$$\mathbb{E}_{\tau\sim\pi(\beta)}[\hat{r}(\tau)] - \mathbb{E}_{\tau\sim\pi_f}[\hat{r}(\tau)] \leq \epsilon\left(\frac{1}{\alpha_f} + 2\right)\mathcal{H}^2, \tag{6}$$

where $\mathcal{H}$ is the horizon of the MDP.

As shown in Lemma 1, training with the DT loss $\mathcal{L}_{DT}$ yields a gradual convergence of the learnt policy $\pi(\tau_t)$ towards the associated behaviour policy $\pi(\beta)$. However, this convergence restricts the ability of stitching suboptimal trajectories into an optimal policy in CSM methods, leading to learnt policy aligning with the actions observed in the training trajectories to a great extent [43]. Consequently, the learnt policy $\pi(\tau_t)$ struggles to exceed the performance of behaviour policy $\pi(\beta)$, resulting in the performance of DT being comparable to, or in some cases even inferior to, BC methods across various benchmarks.

To tackle the challenge of the learnt policy's incapacity to outperform the behaviour policy, as well as the difficulty of stitching optimal strategies from suboptimal trajectories, an advantage-guided policy constraint is utilised in the training process. By integrating a policy constraint that emphasises actions with high advantage values, along with considerations for policy improvement, AGT enhances the ability of the learnt policy to effectively acquire knowledge from adaptive data augmentation.

As shown in the advantage function module, we use the Q-value function and V-value function to calculate the advantage function. The value function is learnt by minimising the Bellman policy operator [14] and we use the technique of double Q-learning [44] to learn two Q-networks, Q_{θ_1}, Q_{θ_2} as well as the corresponding target network, $Q_{\theta_1'}$, $Q_{\theta_2'}$ and the target policy $\pi_{\omega'}$. Considering that the input of CSM module comprises the historical trajectory sequence, we employ the n-step Bellman equation to calculate the function:

$$\hat{Q}_n = \sum_{j=n}^{t-1} \gamma^{j-n} r_j + \gamma^{t-n} \min_{i=1,2} Q_{\theta_i'}(s_t, \hat{a}_t), \tag{7}$$

where γ is the discount factor and $\hat{a}_t$ is the predicted action from the target policy $\pi_{\omega'}$. θ_i where $i = 1, 2$ is updated by minimising the following equation:

$$\mathcal{L}_Q(\theta_i) = \mathbb{E}_{\tau_t\sim\mathcal{D}, \hat{\mathbf{a}}_t\sim\pi_\omega'} \sum_{n=t-K+1}^{t-1} \left\| \hat{Q}_n - Q_\theta(\mathrm{s}_n, a_n) \right\|^2, \tag{8}$$

where D is the offline dataset, τ_t is the trajectory sequence and K is the sequence length.

At the same time, we further learnt the V-network V_ϕ through the Q-network to approximate an expected value solely concerning the Q-function, and the optimisation of ϕ is conducted as follows:

$$\mathcal{L}_V(\phi) = \mathbb{E}_{(s,a)\sim D}\left[\left(Q_\theta(s,a) - V_\phi(s)\right)^2\right], \tag{9}$$

where V_ϕ is the value function. This design guarantees the circumvention of undue conservatism. Consequently, the equation for the advantage function is formulated as follows:

$$A(s,a) = Q_\theta(s,a) - V_\phi(s). \tag{10}$$

We enhance the policy constraint towards high advantage value functions by dynamically modifying the DT loss $\mathcal{L}_{DT}$ to an AGT loss $\mathcal{L}_{AGT}$ during the training phase. $\mathcal{L}_{AGT}$ adopt advantage mask mechanism, which computes the advantage function for the first action a_1 in K trajectories and exclude these trajectories if the advantage is less than zero. Consequently, the loss function is dynamically adjusted as follows:

$$\mathcal{L}_{AGT} = \begin{cases} \mathbb{E}_{\tau_t\sim\mathcal{D}}\left[\frac{1}{K}\sum_{i=t-K+1}^{t}\left(a_i - \pi(\tau_t)_i\right)^2\right], & \text{if } A(s_1, \hat{a}_1) \geq 0 \\ 0, & \text{if } A(s_1, \hat{a}_1) < 0 \end{cases} \tag{11}$$

where $\pi_{\omega'}(\tau_t)_i$ is i the predicted action from target policy $\pi_{\omega'}$. We also incorporate the loss of Q-value $\mathcal{L}_Q$ in the training process to facilitate the policy improvement instead of solely using DT loss $\mathcal{L}_{DT}$ for policy constraints. The final loss function of the training process is

$$\mathcal{L}(\omega) = \mathcal{L}_{AGT}(\omega) + \mathcal{L}_Q(\omega). \tag{12}$$

The policy derived from the training is represented as a linear combination of policy improvement and policy regularisation:

$$\pi = \arg\min_{\omega}\{\mathcal{L}(\omega) = \mathcal{L}_{AGT}(\omega) + \mathcal{L}_{Q}(\omega)\}. \tag{13}$$

By dynamically balancing the policy to constrain the value of $\mathcal{L}_{AGT}$ and adding $\mathcal{L}_Q$ to encourage the policy to improve towards actions with high advantage values, the model can autonomously learn optimal policy in the historical trajectory. Therefore, AGT enhances the stitching ability, leading to the consistence between final returns and expected returns for tasks.

4.3 | Sample Action With High Advantage

In the previous CSM methods, RTG is designed manually or updated through sampling, which is unable to ensure consistency between the final returns and the expected returns for tasks. We use the learnt advantage function module to update the RTG. According to the Bellman expectation equation [45], the Q-value function could be formulated as

$$Q^{\pi}(s,a) = r(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) \sum_{a' \in A} \pi(a'|s') Q^{\pi}(s',a'), \tag{14}$$

where $P(s'|s,a)$ represents the probability of transitioning from the state s to a new state s' after taking action a, $\pi(a'|s')$ represents the probability of taking action a' in the new state s'. $\gamma \sum_{s' \in S} P(s'|s,a) \sum_{a' \in A} \pi(a'|s') Q^{\pi}(s',a')$ represents the expected return, considering all possible future states and actions, given the current state s and action a under policy π.

Consequently, we use the expected return in the right side of Equation (14) to approximate the RTG of the last action in the sequence:

$$\hat{R}_t = Q^{\pi}(s_{t-1}, a_{t-1}) - r(s_{t-1}, a_{t-1}), \tag{15}$$

The output of the transformer is a predicted action tuple $\pi(\tau_t) = (\boldsymbol{a}_{t-k+1}, ..., \boldsymbol{a}_{t-1}, \boldsymbol{a}_t)$. Instead of directly utilising the a_{t-k+1} as the sampled action during evaluation, we preferentially sample action with high advantage, which could be represented as

$$\dot{a} = \underset{\hat{a} \in \{a, \pi_{\omega}(\tau_t)_0\}}{\operatorname{argmax}} A(s,\hat{a}), \tag{16}$$

where $\dot{a}$ is the action with high advantage values, opted between a and the first predicted action of the sequence $\pi_{\omega}(\tau_t)_0$.

Specifically, when the advantage value $A(s,\dot{a}) \geq 0$, we select actions $\dot{a}$ with high advantage from the sampled K-step trajectories. Conversely, if $A(s,\dot{a}) < 0$, we still utilise the first action $\pi_{\omega}(\tau_t)_0$ from the sequence predicted by the causal transformer as the action output of the policy. The sampled action is dynamically changed by the following equation:

$$\overline{a} = \begin{cases} \dot{a}, & A(s,\dot{a}) \geq 0 \\ \pi_{\omega}(\tau_t)_0, & A(s,\dot{a}) < 0. \end{cases} \tag{17}$$

By dynamically adjusting the sampled action based on the advantage function, the agent can preferentially select actions with high advantage values. This policy adjustment allows the agent to better extract optimal policies from suboptimal trajectories, significantly enhancing its learning capability, especially when dealing with offline datasets characterised by randomly generated suboptimal trajectories. The training and evaluation procedures are presented in Algorithm 1, which comprehensively summarises the involved processes.

ALGORITHM 1 | AGT.

Input: Offline datasets D, sequence length K, target network update rate ϵ, max training steps N, max iterations I.
Output: The learned parameters.

Randomly initialize policy π_ω, two Q networks Q_{θ_1}, Q_{θ_2} and target Q networks $Q_{\theta_1'}$, $Q_{\theta_2'}$, value function network V_ϕ, target policy $\pi_{\omega'}$.
for $iter = 1 \cdots I$ **do**
  for $step = 1 \cdots N$ **do**
    Sample mini-batch of trajectories $\{(\mathbf{s}_i, \mathbf{a}_i, r_i)_{i=t}^{t+K}\} \sim \mathcal{D}$.
    Compute return-to-go $\hat{R}_t = \sum_{t'=t}^{K} r_{t'}$.
    Predict action $\hat{\mathbf{a}}_{t+K} \sim \pi_{\omega'}(\cdot|\hat{R}_{t:t+K}, \mathbf{s}_{t:t+K}, \mathbf{a}_{t:t+K-1})$.
    `// Q-function and V-function learning`
    Update Q_{θ_1}, Q_{θ_2} by minimizing Equation (8).
    Update V_ϕ by minimizing Equation (9).
    Compute advantage function by Equation (10).
    `// Advantage guided policy learning`
    for $j = 1 \cdots K$ **do**
      $\hat{\mathbf{a}}_{t+j} \sim \pi_{\omega}(\cdot|\hat{R}_{t:t+K}, \mathbf{s}_{t:t+K}, \mathbf{a}_{t:t+K-1})$.
    end
    Update policy π by minimizing Equation (13).
    `// Update target networks`
    $\theta_i^{'} \leftarrow \epsilon\theta + (1-\epsilon)\theta_i^{'}, i = 1, 2.$
    $\omega^{'} \leftarrow \epsilon\omega + (1-\epsilon)\omega.$
  end
  `// Sample action with high advantage`
  while *Done is True* **do**
    for $i = 1 \cdots K$ **do**
      $\hat{\mathbf{a}}_t^i \sim \pi_{\omega}(\cdot|\hat{R}_{t-K+1:t}, \mathbf{s}_{t-K+1:t}, \mathbf{a}_{t-K+1:t-1})$.
      Compute advantage function by the Equation (10).
      Sample action $a_t \sim \{\hat{\mathbf{a}}_t^i\}_{i=1}^{K}$ by the Equation (17).
      Execute the action a_t and get the reward $r(s_t, a_t)$ and Done.
      Update RTG by the Equation (15).
    end
  end
end

4.4 | Theoretical Analysis

In this section, we analyse the performance of AGT and provide the theoretical analysis of why AGT realises the recovery of optimal policies.

Theorem 1. *(Policy improvement guarantee) Given precise Q value function, precise advantage function $A(s,a)$ and behaviour policy π_β, let $\tilde{\pi}$ be the learnt policy in Equation (13). Then, we have*

$$J(\tilde{\pi}) - J(\pi_\beta) \geq 0, \tag{18}$$

where $J(\pi_\beta) = \mathbb{E}_{\tau\sim\pi_\beta}\left[\sum_{t'=t}^{T} r_{t'}\right]$ represents the expected returns of a behaviour policy π_β.

The proof is provided in Appendix A. Theorem 1 illustrates the validity of Equation (13), further demonstrating that the optimal policy π^* learnt by AGT is continually improved towards a behaviour policy that surpasses the historical trajectories in terms of expected return.

It shows that the advantage-guided policy constraint effectively improves the policy by allowing the targeted sampling of actions with higher expected values. Therefore, the advantage function brings the learning dynamics into a closer alignment with optimal return outcomes. AGT not only promotes a more efficient exploration of the action space but also guarantees an enhancement over the initial behaviour policy, as the agent tends to prioritise actions that yield superior anticipated returns.

Theorem 2. *(Convergence to near-optimal policies) Let $J(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t'=t}^{T} r_{t'}\right]$, as shown in Equation (16), the target is maximum $J(\pi)$. Suppose that there exists a conditioning function f and an optimal policy π^*. Assume return coverage: $P_\beta(g = f(s_1)|s_1) \geq \alpha_f$ and near determinism: $P(r \neq r(s,a)$ or $s' \neq \mathcal{T}(s,a)|s,a) \leq \epsilon$ at all s, a for some functions $\mathcal{T}$ and r, such that the equation*

$$J(\pi_f) \geq J(\pi^*) - \epsilon\left(\frac{1}{\alpha_f} + \eta\right)\mathcal{H}^2, \tag{19}$$

where $\eta \geq 3$, π_f represents the learnt policy by the conditioning function and $\mathcal{H}$ represents the horizon of the MDP.

If $\alpha_f > 0$, $f(s_1) = A^{\pi^*}(s_1, a_1)$ for all initial state s_1, a_1, when $\epsilon \to 0$, we could derive the following equation:

$$J(\pi_f) \to J(\pi^*). \tag{20}$$

Theorem 2 shows that it is feasible for CSM methods to achieve near-optimal policy recovery in nearly deterministic settings when appropriate conditioning functions and comprehensive data coverage are utilised.

In Equation (15), we approximate the RTG, which is the conditional function of CSM methods, by expected returns. As the estimation of the value function becomes increasingly precise and the policy is continuously improved towards generating actions with higher advantage values, we can express $f(s_1) \sim A^{\pi^*}(s_1, a_1)$ and the learnt policy $\tilde{\pi} \sim \pi^*$. The detailed proof of Theorem 2 is provided in Appendix A.

In comparison, traditional CSM methods rely on manual design or learning of the RTG using other networks, which fails to meet the conditions necessary for finding the optimal policy. Furthermore, the manual engineering of RTG features frequently overlooks trajectory-level temporal correlations, resulting in credit assignment inaccuracies that propagate through subsequent decision steps. Consequently, these methods exhibit suboptimal performance in stochastic environments. The practical validation of these findings is shown in Section 5.2.

5 | Simulation Study

In this section, the performance of AGT is tested on offline RL benchmarks based on D4RL datasets. Specifically, the implementation details of AGT are first summarised. Then, AGT is evaluated on Gym tasks, Adroit tasks and Maze2D tasks and compared with SOTA methods. Then, each component of AGT is studied in an ablation study. Finally, the sensitivity of each hyperparameter is analysed.

5.1 | Implementation Details

In AGT, the Q-network and V-network are both implemented with 3 hidden-layer multilayer perceptrons (MLPs) containing 1024 neurons (256 hidden units for all networks). The architecture of the AGT is built upon minGPT open-source code (https://github.com/karpathy/minGPT). Through empirical validation, we employed an optimised configuration of 4 layers and 4 attention heads, which provides an effective balance between model capacity and computational efficiency while maintaining strong performance. We use the Adam [46] optimiser for training actor and critic networks. The detailed hyperparameters are listed in Table 1. The motivation for using ReLU (rectified linear unit) activation functions lies in their computational efficiency and simplicity, especially when integrated with the advantage function module of AGT, as they effectively alleviate the vanishing gradient problem.

TABLE 1 | Details of the hyperparameters used in the paper.

Hyperparameter	Value
Number of layers	4
Number of attention heads	4
Model learning rate	$3 \cdot 10^{-4}$
Batch size	256
Dropout	0.1
Context length K	20
Embedding dimension	256
Actor learning rate	$3 \cdot 10^{-4}$
Critic learning rate	$3 \cdot 10^{-4}$
Discount γ	0.99
Warmup steps	$1 \cdot 10^{4}$
Weight decay	$1 \cdot 10^{-4}$
Nonlinearity function	ReLU
Optimiser	Adam
Adam betas	(0.9 0.999)

5.2 | Evaluation on Gym

To determine the benchmarking performance achieved on Gym Mujoco, we compare the AGT results with the normalised scores [38] obtained by the traditional offline RL algorithm, including TD3+BC, CQL, IQL, BCQ, MoRel and BEAR.

i. TD3+BC [12] introduces behaviour cloning for improved stability.
ii. CQL [10] is a conservative algorithm that minimises value overestimation.
iii. IQL [11] stands for implicit Q-learning, optimising implicitly over actions.
iv. BCQ [14] combines off-policy learning with generative modelling for efficiency.
v. MoRel [47] is a model-based offline RL algorithm that utilises relational models to enhance learning efficiency.
vi. BEAR [13] employs behaviour regularisation to prevent extrapolation errors.
vii. IPC [48] utilises implicit policy constraints with energy-based models rather than explicitly learning the behaviour policy.

We also compare the AGT results with the AGT algorithm with the traditional CSM methods based on the transformer, including DT and QDT and the SOTA CSM methods and CGDT, V-ADT and QT.

i. DT [8] firstly combines the transformer architecture in offline RL.
ii. QDT [26] combines the CSM methods with Q-learning to reshape RTG.
iii. CGDT [7] utilises a value function which is similar to the critic network to guide policy training.
iv. QT [27] leverages the n-step Bellman equation to approximate the Q-value function to achieve value regularisation.

The simulation results obtained in Gym Mujoco are shown in Table 2. We can see that the AGT algorithm achieves the best performance in Gym Mujoco tasks. Moreover, AGT exhibits remarkable performance on both medium and medium-replay datasets, significantly outperforming other CSM methods.

In particular, these offline datasets consist of trajectories generated by an agent trained using the online soft actor-critic (SAC) [49] algorithm. This agent, trained to achieve approximately one-third of expert performance, includes a substantial amount of suboptimal data due to its inherent randomness, which results in a pronounced gap between traditional CSM methods and value-based offline RL algorithms, especially evident on the medium dataset. As described in Section 4, the introduction of advantage functions in AGT facilitates guided policy generation, enabling the agent to learn high-quality policies that align with expected returns in random environments with suboptimal trajectories. This policy enhancement significantly contributes to the impressive empirical performance of AGT.

Mechanistically, AGT diverges from prior SOTA methods. Unlike CGDT and QT, which uses a critic to guide learning but lacks our adaptive trajectory-level filtering, AGT's advantage mask actively curates the training data. Contrasted with QDT, which uses dynamic programming for one-time RTG relabelling, AGT's advantage-based RTG update (Equation 15) provides a dynamic, policy-dependent conditioning signal that evolves during training. This integrated advantage-guided approach in both the loss function and the conditioning input is the key to its superior performance on suboptimal datasets.

5.3 | Evaluation on Adroit

As shown in Figure 2, the Adroit domain encompasses the control of a 24-degree-of-freedom (DoF) simulated shadow hand robot, which is designed to perform a variety of intricate tasks, including hammering a nail, opening a door, twirling a pen or picking up and moving a ball [38]. Unlike those of Gym Mujoco tasks, the datasets of Adroit tasks are collected by the limited scope of human demonstrations. The sparse rewards and exploration challenges in Adroit tasks make it challenging for online RL to solve. The selection of the domain in Adroit is used to assess the impact of narrow expert data distributions and human demonstrations on a high-dimensional robotic manipulation task characterised by sparse rewards [38].

The evaluation tasks in Adroit are shown in the second part of Table 2. The results illustrate that AGT outperforms the compared algorithms in each task. As shown in the Figure 3, AGT achieves a remarkable performance improvement of 163% over SOTA value-based methods and 20% over SOTA CSM approaches on pen-cloned datasets, which mixes the demonstration data of human operators with an imitation policy on the human demonstration data.

By utilising the advantage function, we effectively implement a dynamic adjustment of the loss function $\mathcal{L}_{AGT}$ during training, as shown in Equation (11). This allows our algorithm to filter out a significant amount of suboptimal data while robustly learning a near-optimal policy in the presence of noise. Additionally, the incorporation of a regularisation term based on the action value function further enhances our algorithm's ability to exploit high-quality demonstrations. This combination not only improves sample efficiency but also ensures that our model can perform adaptive data augmentation, leading to superior performance on the cloned dataset.

In addition to traditional value-based reinforcement learning methods and CSM approaches, we also compare our work with D-QL [50], a method based on diffusion models used in offline RL. By comparison, D-QL, which learns policies based on the diffusion process, is unable to learn effective policies in Adroit tasks. This may demonstrate that our transformer-based architecture approaches is able to leverage rich contextual

TABLE 2 | The performance of AGT and SOTA baselines on D4RL datasets (Gym, Adroit, Maze2D). Results for AGT related to the mean and standard errors of normalised scores over 30 random rollouts (3 independently trained models and 10 trajectories per model) for all tasks, which typically enable the maintenance of low variance in performance.

Gym tasks	TD3+BC	CQL	IQL	BCQ	MoRel	IPC	BC	DT	QDT	CGDT	V-ADT	QT	AGT (ours)
Halfcheetah-medium-v2	48.4	49.2	47.4	41.5	42.1	49.0	43.1	42.6	42.3	43.0	48.7	51.4	**57.8** ± 0.2
Hopper-medium-v2	59.3	58.0	66.2	65.1	95.4	73.1	63.9	67.6	66.5	96.9	60.6	96.9	**102.9** ± 0.2
Walker2d-medium-v2	83.7	83.0	78.3	52.0	77.8	85.3	77.3	74.0	67.1	79.1	80.9	88.8	**93.1** ± 0.0
Halfcheetah-medium-replay-v2	44.6	45.5	44.2	34.8	40.2	45.3	4.3	36.6	35.6	40.4	42.8	48.9	**52.0** ± 0.1
Hopper-medium-replay-v2	60.9	95.0	94.7	31.1	93.6	101.6	27.6	82.7	52.1	93.4	83.5	102.0	**103.5** ± 0.3
walker2d-medium-replay-v2	81.8	77.2	73.9	13.7	49.8	92.2	36.9	66.6	58.2	78.1	86.3	98.5	**104.1** ± 0.3
Halfcheetah-medium-expert-v2	90.7	91.6	86.7	69.6	53.3	94.7	59.9	86.8	79.0	93.1	91.7	**96.1**	95.1 ± 0.4
Hopper-medium-expert-v2	98.0	105.4	91.5	109.1	108.7	107.4	79.6	107.6	94.2	108.6	101.6	113.4	**114.1** ± 0.2
walker2d-medium-expert-v2	110.1	108.8	109.6	67.3	95.6	109.1	36.6	108.1	101.7	109.2	112.1	112.6	**113.6** ± 0.1
Sum	677.5	717.3	692.5	484.2	656.5	757.7	429.2	672.6	596.7	741.8	708.2	808.6	**836.3**

Adroit tasks	O-RL	CQL	IQL	BCQ	MoRel	BEAR	BC	DT	GDT	StAR	D-QL	QT	AGT (ours)
Pen-human-v1	90.7	37.5	71.5	66.9	−3.2	−1.0	63.9	79.5	92.5	77.9	72.8	129.6	**137.6** ± 3.3
Hammer-human-v1	0.2	4.4	1.4	0.9	2.7	0.3	1.2	3.7	5.5	3.7	0.2	35.6	**41.5** ± 1.1
Door-human-v1	−0.1	9.9	4.3	−0.05	2.2	−0.3	2.0	14.8	20.6	1.5	0.0	28.7	**30.1** ± 1.8
Pen-cloned-v1	60.0	39.2	37.3	50.9	−0.2	26.5	37.0	75.8	86.2	33.1	57.3	125.0	**150.3** ± 5.0
Hammer-cloned-v1	2.0	2.1	2.1	0.4	2.3	0.3	0.6	3.0	8.9	0.3	3.1	23.0	**53.7** ± 3.0
Door-cloned-v1	0.4	0.4	1.6	0.0	2.3	−0.1	0.0	16.3	19.8	0.0	0.0	20.6	**28.9** ± 3.1
Sum	153.2	93.5	118.2	119.1	6.1	25.7	104.7	193.1	233.5	116.5	133.4	362.5	**442.2**

Maze2D tasks	COMBO	CQL	IQL	BCQ	TD3+BC	BEAR	BC	DT	QDT	DD	Diffuser	QT	AGT (ours)
maze2d-umaze-v1	76.4	94.7	42.1	49.1	14.8	65.7	88.9	31.0	57.3	116.2	113.9	105.4	**167.0** ± 3.4
maze2d-medium-v1	68.5	41.8	34.9	17.1	62.1	25.8	38.3	8.2	13.3	122.3	121.5	172.0	**188.8** ± 1.1
maze2d-large-v1	14.1	49.6	61.7	30.8	88.6	81.0	1.5	2.3	31.0	125.9	123.0	240.1	**250.3** ± 1.2
Sum	159.0	186.1	138.7	97.0	165.5	172.5	128.7	41.5	101.6	364.4	358.4	517.5	**606.1**

Kitchen tasks	O-RL	CQL	IQL	BCQ	TD3+BC	BEAR	BC	DT	DD	GDT	V-ADT	QT	AGT (ours)
Kitchen-complete-v0	2.0	43.8	62.5	8.1	0.0	0.0	65.0	50.8	65.0	43.8	55.1	81.7	**83.8** ± 0.7
Kitchen-partial-v0	35.5	49.8	42.1	18.9	0.0	13.1	33.8	57.9	57.0	73.3	46.0	75.0	**80.8** ± 0.4
Sum	37.5	93.6	108.8	27.0	0.0	13.1	98.8	108.7	122.0	117.1	101.1	156.7	**164.0**

Note: Bold values indicate the highest performance score on the corresponding task.

information and effectively generalise from limited demonstrations, thereby achieving improved performance in high-dimensional robotic manipulation tasks.

5.4 | Evaluation on Maze2D

To understand the stitching ability of AGT, we conduct an evaluation on Maze2D tasks. The Maze2D domain is a navigation task in which a 2D agent is required to reach a fixed target location. These tasks are designed to provide a straightforward test of an offline RL algorithm's ability to stitch together previously collected sub-goals in order to find the shortest path to the evaluation target. The trajectories of the dataset in the Maze2D task are illustrated in Figure 4.

This navigation task imposes high demands on the ability to stitch together previously collected sub-goals in order to reach the evaluation target via the shortest path. The prior CSM method exhibits only moderate performance on this task. Even under dense reward conditions, it performs worse than traditional offline reinforcement learning methods. AGT enhances the algorithm's stitching capability by promoting an advantage

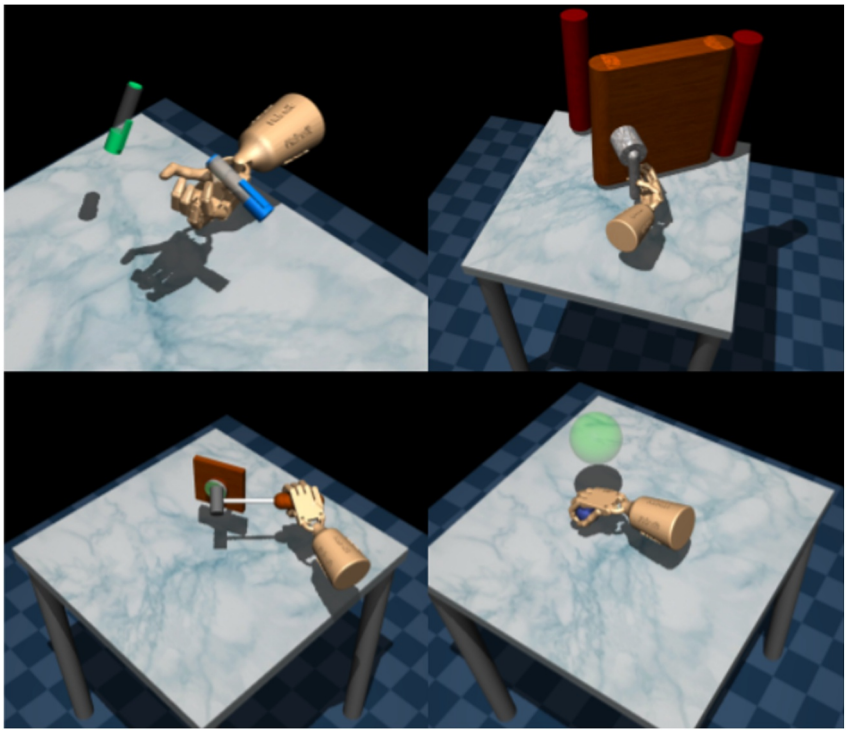


FIGURE 2 | The tasks of Adroit (pen, door, hammer and relocate).

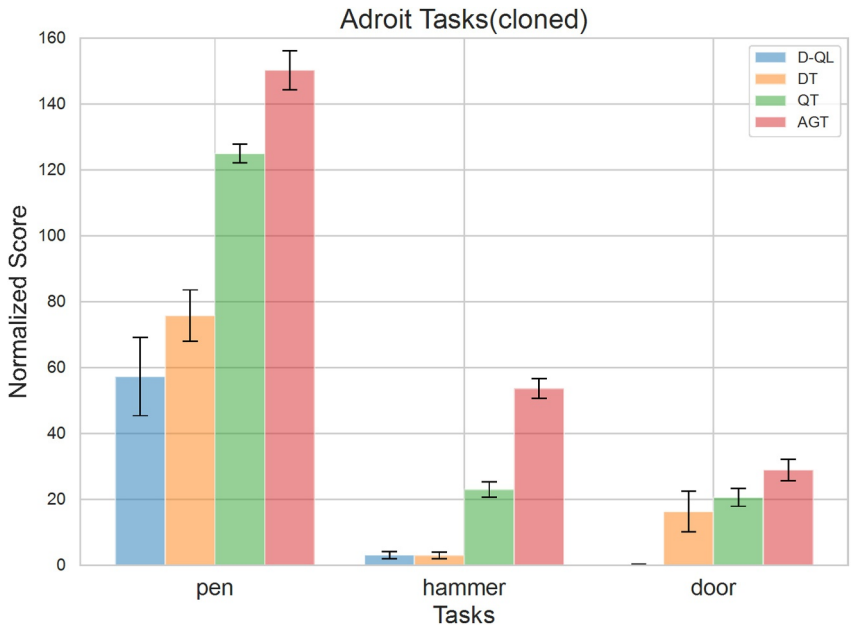


FIGURE 3 | Evaluation results for D-QL, DT, QT and AGT in the Adroit tasks (three cloned-v1 tasks).

function to guide the learning and generation of policies, enabling it to quickly learn policies for trajectories with high advantage values.

Across the 3 Maze2D tasks, AGT achieves optimal performance in each, particularly in the maze-umaze task, where it outperforms the second-best offline RL algorithm based on CSM by 59% (as shown in Table 2).

5.5 | Evaluation on Kitchen

To thoroughly evaluate the capability of AGT in long-horizon, sparse-reward scenarios, we conducted comprehensive experiments on the challenging Franka Kitchen domain. This environment presents a more complex and realistic testbed for assessing trajectory stitching and generalisation capabilities compared to the simpler Maze2D domain. The Franka Kitchen

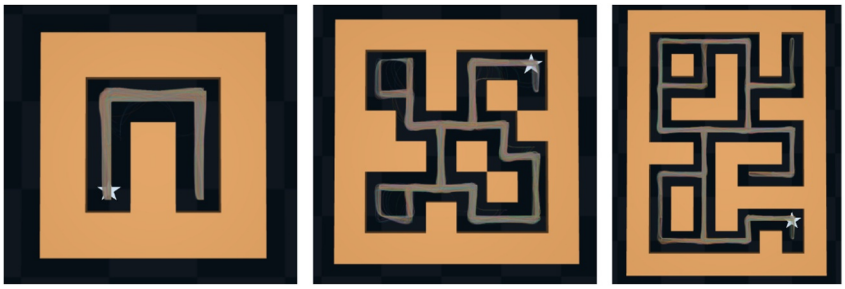


FIGURE 4 | Three tasks in the Maze2D for the simulation study, including 'maze2d-umaze', 'maze2d-medium' and 'maze2d-large'.

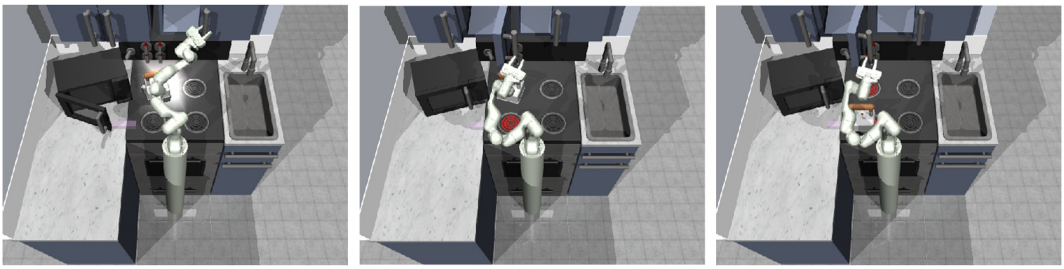


FIGURE 5 | The different compound goals in the kitchen for the simulation study.

domain, first proposed by Gupta et al. [51], involves controlling a 9-DoF Franka robot in a kitchen environment containing several common household items: a microwave, a kettle, an overhead light, cabinets and an oven. The goal requires executing a sequence of subtasks to achieve desired goal configurations, such as having the microwave open, kettle on the burner and light turned on. Examples of these compound goals are illustrated in Figure 5.

This domain benchmarks multitask behaviour in a realistic, nonnavigation environment, where the 'stitching' challenge is particularly demanding. The collected trajectories represent complex paths through the high-dimensional state space, requiring algorithms to generalise beyond seen trajectories and effectively combine suboptimal segments. We specifically evaluated on the kitchen-complete and kitchen-partial datasets, where the latter contains undirected demonstrations that only partially solve the task, thus imposing stronger requirements on the algorithm's ability to identify and stitch relevant sub-trajectories.

As shown in Table 2, the results confirm that AGT's advantage-guided mechanism effectively identifies high-value trajectory segments and synthesises them into optimal behaviours, even in this complex, long-horizon task with sparse rewards. This capability stems from the dynamic masking of low-advantage trajectories during training and the advantage-based action sampling during evaluation, which collectively enable the model to focus on and combine the most promising behavioural segments from the suboptimal dataset.

5.6 | Ablation Study

In this subsection, we first conduct an ablation study to understand the main novel components of our proposed algorithm, advantage-guided policy constraint (AGPC), sample action with high advantage (SAHA) and advantage-based RTG update (ARTG). Then, we conduct further ablation study to evaluate whether AGT effectively combines the advantages of both CSM and advantage functions while addressing their limitations.

5.6.1 | *Main Novel Components*

The AGT algorithm without AGPC is denoted as w/o AGPC, which only uses the conditional decision transformer (CDT) for policy learning. The AGT algorithm that samples actions without selecting high advantage is denoted as w/o SAHA. The AGT algorithm that uses manual RTG without using ARTG is denoted as w/o ARTG. The AGT algorithm without CDT is denoted as w/o CDT, which uses behaviour cloning (BC) instead of CDT for policy learning. Moreover, we test a version without leveraging AGPC, SAHA and CDT, only using BC for policy learning. The ablation study is conducted on the walker2d-medium-replay task. The results are shown in Table 3.

Firstly, w/o AGPC reduces the score to 95.2, indicating that the adaptive masking mechanism of AGPC, which suppresses suboptimal trajectories while preserving high-advantage sequences, is critical for refining policy learning. Without AGPC, the policy lacks explicit guidance to prioritise transitions aligned with the physical constraints of tasks. Similarly, w/o SAHA results in a score of 101.3, demonstrating that SAHA's advantage-driven sequence augmentation is essential for synthesising coherent locomotion patterns from fragmented data. The absence of SAHA forces the policy to rely on static trajectories, limiting its ability to exploit high-reward transitions. The replacement of CDT with behaviour cloning (BC) in w/o CDT further degrades performance to 83.2, highlighting the role of CDT in modelling temporal dependencies and long-horizon action sequences. Though BC learns deterministic action distributions, it fails to capture the sequential nature of locomotion tasks, resulting in

inflexible policies. Furthermore, the baseline using only BC (without AGPC, SAHA or CDT) achieves merely 36.9, underscoring the limitations of omitting all three components. The complete ablation reveals that the superiority of AGT stems from the synergistic integration of its components. For instance, w/o AGPC+SAHA (CDT-only) achieves 79.4, significantly lower than AGT. This synergy enables AGT to dynamically balance exploration (via SAHA's advantage-guided augmentation) and regularisation (via AGPC's trajectory masking), whereas CDT provides the foundational architecture for sequence modelling. w/o ARTG underscores the essential role of ARTG in dynamically conditioning the policy on accurate return estimates derived from the advantage function. Without this component, the model relies on static or manually designed RTG values, which fail to capture the evolving policy's expected returns.

5.6.2 | *Stitching Ability*

The stitching ability of offline RL algorithms, which is defined as their capacity to synthesise suboptimal trajectory segments into optimal paths is critically evaluated in the Maze2D domain. This domain is characterised by sparse rewards and complex geometric constraints. It adopts two reward functions: sparse and dense. Specifically, the sparse reward setting delivers feedback solely upon the completion of the task, whereas the dense reward mechanism provides more detailed feedback by allocating rewards according to the agent's distance to the target, thus requiring effective sequential decision-making.

As illustrated in Table 4, CQL demonstrates strong performance in dense rewards settings, leveraging its pessimistic value estimation to mitigate extrapolation errors [10]. Although DT effectively captures trajectory dependencies through sequence prediction, it faces challenges in segment stitching, causing its performance to lag behind that of CQL in Maze2D tasks. QT combines the Q-value module with the transformer policy, resulting in performance improvements that surpass CQL. However, the enhancement is limited, with only an 11% increase in maze2d-umaze tasks.

In contrast, AGT achieves superior stitching performance across all Maze2D configurations, as evidenced by its consistent superior performance in both sparse and dense reward settings. For instance, AGT outperforms DT (438.7%), CQL (76.3%) and QT (59.0%) in maze2d-umaze task. This improvement is attributed to AGT's ability to identify and stitch critical trajectory segments, such as turns or corridor traversals that correlate with high advantage values, even when such segments are sparsely distributed in the dataset.

5.6.3 | *Adaptive Data Augmentation*

The effectiveness of the proposed AGT algorithm is further validated through ablation studies on adaptive data augmentation, focusing on the Walker2D locomotion task. As a high-dimensional control task, Walker2D requires precise coordination of multiple joints (torso, thighs, legs and feet) to generate continuous and stable walking motions, where uncoordinated and suboptimal actions can disrupt balance and degrade performance.

As shown in Table 5, AGT achieves a remarkable score of 23.6 on the random dataset, surpassing A2PR by 386.9% and outperforming other baselines. This stark contrast underscores AGT's unique capability to synthesise coherent locomotion

TABLE 3 | Ablation on the different components of AGT. The results are averaged over three random seeds for the walker2d-medium-replay task.

Algorithm	Policy	AGPC	SAHA	ARTG	Performance
BC	BC	×	×	×	36.9 ± 7.6
w/o CDT	BC	✓	✓	×	83.2 ± 0.6
w/o AGPC+ SAHA	CDT	×	×	×	79.4 ± 2.0
w/o AGPC	CDT	×	✓	✓	95.2 ± 1.1
w/o SAHA	CDT	✓	×	✓	101.3 ± 0.5
w/o ARTG	CDT	✓	✓	×	99.6 ± 0.7
AGT	CDT	✓	✓	✓	104.1 ± 0.3

TABLE 4 | Ablation on the stitching ability. The results are averaged over three random seeds for the Maze2D tasks.

Reward	Dataset	CQL	DT	QT	AGT
Sparse	maze2d-umaze-v1	94.7	31.0	105.4	**167.0** ± 3.4
	maze2d-medium-v1	41.8	8.2	172.0	**188.8** ± 1.1
	maze2d-large-v1	49.6	2.3	240.1	**250.3** ± 1.2
Dense	maze2d-umaze-v1	72.7	−6.8	103.1	**138.4** ± 2.5
	maze2d-medium-v1	70.9	31.5	111.9	**142.0** ± 1.5
	maze2d-large-v1	90.9	45.3	177.29	**199.3** ± 7.5

Note: Bold values indicate the highest performance score on the corresponding task.

TABLE 5 | Ablation on the adaptive data augmentation. The results are averaged over three random seeds for Walker2d tasks.

Task	TD3+BC	A2PR	DT	QT	AGT
walker2d-random-v2	1.6	6.1	6.1	7.0	**23.6** ± 0.7
walker2d-medium-v2	83.7	90.3	74.0	88.8	**93.1** ± 0.0
walker2d-medium-replay-v2	81.8	96.0	66.6	98.5	**104.1** ± 0.3

Note: Bold values indicate the highest performance score on the corresponding task.

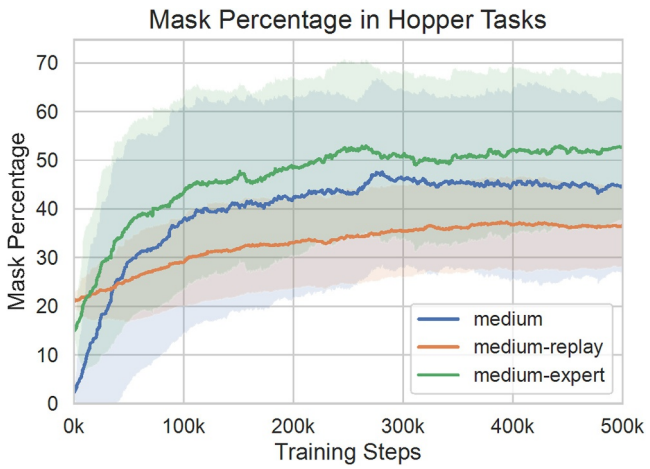


FIGURE 6 | The mask percentage in 3 hopper tasks, including 'hopper-medium', 'hopper-medium-replay' and 'hopper-medium-expert'.

trajectories from highly stochastic data. Although A2PR, a SOTA value-based method, employs advantage functions for regularisation, it fails to capture the sequential dependencies required for leg coordination. In contrast, AGT's advantage-guided augmentation mechanism identifies and reinforces temporally extended high-advantage action sequences, enabling it to reconstruct reasonable motion patterns even from random or noisy trajectories.

Furthermore, AGT surpasses DT and QT, indicating its adaptive data augmentation ability during the training process. The superiority of AGT extends to datasets with mixed-quality trajectories. CSM methods like DT and QT fail to filter suboptimal trajectories due to sequence learning. AGT addresses this challenge through its adaptive advantage-guided masking mechanism, which selectively reinforces high-advantage sequences while suppressing inconsistent or suboptimal actions. For example, AGT outperforms QT by 5.6 normalised points in medium-replay settings, demonstrating its ability to learn nuanced locomotion features.

As depicted in Figure 6, about 45% of the trajectories in the medium datasets, 35% in the medium—replay datasets and 50% in the medium-expert datasets are masked. This is because the medium dataset is collected from a partially trained policy after early stopping the training. Consequently, a large proportion of trajectories are likely to have low advantage values since the policy has not fully learnt the optimal behaviour. In the medium-replay dataset, samples are accumulated in the replay buffer during the training process until the medium performance level is reached. As the training progresses, the policy improves and the quality of samples in the replay buffer gradually enhances. For the medium-expert dataset, which consists of 50% expert data and 50% suboptimal data, the masking proportion of low advantage trajectories is set at 50%. These findings further validate the effectiveness of our adaptive data augmentation.

5.7 | Hyperparameter Study

We further investigate the sensitivity of AGT to the hyperparameter K, which defines the context length in the transformer architecture. We test AGT with different context lengths K. Table 6 shows the results. It is observed that AGT achieves optimal performance at $K = 20$, with normalised scores of 104.1 and 188.8 on the walker2d-medium-replay and maze2d-medium tasks, respectively. Notably, the performance variance across the values of K remains relatively small, indicating AGT's robustness to context length selection. Although shorter contexts ($K = 10$) may inadequately capture long-horizon dependencies, and excessively long contexts ($K = 70$) add noise from irrelevant historical states, AGT maintains competitive performance across all settings. For instance, even at $K = 70$, AGT achieves 96.5 on Walker2D and 177.6 on Maze2D, demonstrating its capability to model extended temporal dependencies without severe degradation. This resilience highlights the superiority of AGT in long-term tasks compared to conventional value-based methods, which often face challenges due to cumulative errors in sequential decision-making [52].

Table 7 shows the results of different discount factors γ. It is observed that AGT achieves optimal performance at $\gamma = 0.99$, with normalised scores of 104.1 and 188.8 on the walker2d-medium-replay and maze2d-medium tasks, respectively. Notably, the performance variance across different γ values remains relatively small, indicating AGT's robustness to the choice of the discount factor. This can be attributed to the advantage-guided mechanism, which dynamically prioritises high-value trajectories based on relative action advantages rather than relying heavily on the precise discounting of future rewards.

5.8 | Computational Cost Study

To evaluate the training time of AGT compared to baseline methods, we conducted systematic experiments measuring training time and computational resources on the hopper-medium-v2 task from the D4RL benchmark for 200K steps. All experiments were performed on a consistent hardware setup including one Nvidia RTX 4090 GPU with the AMD EPYC 7543 CPU. As demonstrated in Table 8, AGT achieves a training time of 7.89 h, which is comparable to the standard decision transformer (DT, 7.32 h) and significantly faster than other advanced methods including QT (9.04 h) and CQL (10.31 h). This

TABLE 6 | Sensitivity of AGT to the hyperparameter K. The results are averaged over three random seeds for Walker2D (walker2d-medium-replay) and Maze2D (maze2d-medium) tasks.

Task	$K = 10$	$K = 20$	$K = 30$	$K = 50$	$K = 70$
Walker2D	102.8 ± 0.8	**104.1** ± 0.3	101.4 ± 0.3	99.9 ± 0.7	98.2 ± 1.7
Maze2D	185.5 ± 3.0	**188.8** ± 1.1	182.8 ± 3.8	176.5 ± 6.6	180.8 ± 5.5

Note: Bold values indicate the highest performance score on the corresponding task.

TABLE 7 | Sensitivity of AGT to the discount γ. The results are averaged over three random seeds for Walker2D (walker2d-medium-replay) and Maze2D (maze2d-medium) tasks.

Task	$\gamma = 0.93$	$\gamma = 0.95$	$\gamma = 0.97$	$\gamma = 0.99$
Walker2D	89.7 ± 1.2	90.3 ± 0.8	92.1 ± 0.6	**104.1** ± 0.3
Maze2D	152.6 ± 2.4	172.7 ± 1.3	176.2 ± 3.3	**188.8** ± 1.1

Note: Bold values indicate the highest performance score on the corresponding task.

TABLE 8 | The training time of the different methods.

Methods	CQL	DT	CGDT	QT	AGT
Train time	10.31 h	7.32 h	8.43 h	9.04 h	7.89 h

efficiency can be attributed to AGT's adaptive trajectory filtering mechanism guided by the advantage function, which effectively reduces the amount of data processed during training by focusing only on high-advantage transitions.

6 | Conclusion

In this paper, we present the AGT algorithm, which is designed for offline RL. To our knowledge, AGT represents the first method to integrate CSM and the advantage function, offering direct and efficient methods to enhance the behaviour policy. By utilising the advantage-guided policy constraint, AGT adaptively learns sequences with high advantage functions in the trajectory and masks suboptimal action sequences to achieve an automatic balance between policy improvement and policy regularisation. This approach effectively overcomes the challenge posed by noisy data prevalent in the offline dataset, which often hinders the learning of optimal policies. Through sampling the action with high advantage, AGT demonstrates robust performance in trajectories with a significant amount of suboptimal data. In the D4RL benchmark, AGT achieves SOTA performance, emphasising its potential to improve the performance of offline RL.

In future work, we plan to study an autonomous hyperparameter tuning approach for the AGT framework. We also aim to extend the AGT algorithm to multi-agent settings, further evaluating its performance and adaptability in environments where multiple agents interact and learn simultaneously under offline constraints.

Funding

This work was supported by the National Natural Science Foundation of China under Grants 62403483, U24A20279, T2521006, U21A20518 and 62533021.

Conflicts of Interest

Xin Xu is a Deputy Editors-in-Chief for the journal, and was not involved in peer review process or the decision to publish this article. The authors declare that they have no conflict of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

1. F. Hêche, O. Barakat, T. Desmettre, T. Marx, and S. Robert-Nicoud, "Offline Reinforcement Learning in High-Dimensional Stochastic Environments," *Neural Computing & Applications* 36, no. 2 (2024): 585–598, https://doi.org/10.1007/s00521-023-09029-3.

2. R. F. Prudencio, M. R. O. A. Máximo, and E. L. Colombini, "A Survey on Offline Reinforcement Learning: Taxonomy, Review, and Open Problems," *IEEE Transactions on Neural Networks and Learning Systems* 35, no. 8 (2024): 10237–10257, https://doi.org/10.1109/TNNLS.2023.3250269.

3. S. Lamsiyah, A. E. Mahdaouy, A. Nourbakhsh, and C. Schommer, "Fine-Tuning a Large Language Model With Reinforcement Learning for Educational Question Generation," in *Artificial Intelligence in Education—25th International Conference, AIED 2024, Recife, Brazil, July 8–12, 2024, Proceedings, Part I, Vol. 14829 of Lecture Notes in Computer Science* (Springer, 2024), 424–438.

4. M. M. Islam, M. K. Hasan, S. Islam, et al., "Enabling Pandemic-Resilient Healthcare: Narrowband Internet of Things and Edge Intelligence for Real-Time Monitoring," *CAAI Transactions on Intelligence Technology* (2024): cit2.12314, https://doi.org/10.1049/cit2.12314.

5. H. Lin, W. Ding, Z. Liu, et al., "Safety-Aware Causal Representation for Trustworthy Offline Reinforcement Learning in Autonomous Driving," *IEEE Robotics and Automation Letters* 9, no. 5 (2024): 4639–4646, https://doi.org/10.1109/lra.2024.3379805.

6. J. Zhang, K. Wang, and C. Mu, "Multi-Station multi-robot Task Assignment Method Based on Deep Reinforcement Learning," *CAAI Transactions on Intelligence Technology* 10, no. 1 (2025): 134–146, https://doi.org/10.1049/cit2.12394.

7. Y. Wang, C. Yang, Y. Wen, Y. Liu, and Y. Qiao, "Critic-Guided Decision Transformer for Offline Reinforcement Learning," in *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20–27, 2024, Vancouver, Canada* (AAAI Press, 2024), 15706–15714.

8. L. Chen, K. Lu, A. Rajeswaran, et al., "Decision Transformer: Reinforcement Learning via Sequence Modeling," in *Advances in Neural Information Processing Systems*, Vol. 34 (Curran Associates Inc., 2021), 15084–15097.

9. Z.-M. M. Y. L. Chongchong Li, Y. Wang, Z. Ma, and Y. Liu, "A Criterion for Selecting the Appropriate One From the Trained Models for Model-Based Offline Policy Evaluation," *CAAI Transactions on Intelligence Technology* 10, no. 1 (2025): 223–234, https://doi.org/10.1049/cit2.12376.

10. A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative q-Learning for Offline Reinforcement Learning," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6–12, 2020, Virtual* (2020), 1179–1191.

11. I. Kostrikov, A. Nair, and S. Levine, "Offline Reinforcement Learning With Implicit q-Learning," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25–29, 2022* (OpenReview.net, 2022).

12. S. Fujimoto and S. S. Gu, "A Minimalist Approach to Offline Reinforcement Learning," in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6–14, 2021, Virtual* (2021), 20132–20145.

13. A. Kumar, J. Fu, M. Soh, G. Tucker, and S. Levine, "Stabilizing Off-Policy q-Learning via Bootstrapping Error Reduction," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8–14, 2019, Vancouver, BC, Canada* (2019), 11761–11771.

14. S. Fujimoto, D. Meger, and D. Precup, "Off-Policy Deep Reinforcement Learning Without Exploration," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9–15 June 2019, Long Beach, California, USA*, eds. K. Chaudhuri and R. Salakhutdinov, Vol. 97 *Proceedings of Machine Learning Research*, (PMLR, 2019), 2052–2062.

15. X. Chen, A. Ghadirzadeh, T. Yu, et al., "Latent-Variable Advantage-Weighted Policy Optimization for Offline RL," *CoRR* abs/2203 (2022): 08949, https://doi/org/10.48550/arXiv.2203.08949.

16. K. Sohn, H. Lee, and X. Yan, "Learning Structured Output Representation Using Deep Conditional Generative Models," in *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7–12, 2015, Montreal, Quebec, Canada* (2015), 3483–3491.

17. A. Vaswani, N. Shazeer, N. Parmar, et al., "Attention Is all You Need," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4–9, 2017, Long Beach, CA, USA* (2017), 5998–6008.

18. L. Wei, H. Huang, and X. Yu, "Intersection Over Union Similarity Based Non-Maximum Suppression for Human Pose Estimation in Crowded Scenes," *IEEE Transactions on Cognitive and Developmental Systems* (2023): 1, https://doi/org/10.1109/TCDS.2023.3276372.

19. K. Paster, S. A. McIlraith, and J. Ba, "You Can't Count on Luck: Why Decision Transformers and Rvs Fail in Stochastic Environments," in *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28–December 9, 2022* (2022), 38966–38979.

20. S. Yang, D. Schuurmans, P. Abbeel, and O. Nachum, "Dichotomy of Control: Separating What You Can Control From What You Cannot," in *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1–5, 2023* (OpenReview.net, 2023).

21. Y. Wu, X. Wang, and M. Hamaya, "Elastic Decision Transformer," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10–16, 2023* (2023), 18532–18550.

22. H. Jung, Y. Kim, and Y. Kim, "Maneuver-Conditioned Decision Transformer for Tactical in-Flight Decision-Making," *IEEE Robotics and Automation Letters* 9, no. 6 (2024): 5322–5329, https://doi.org/10.1109/lra.2024.3391691.

23. S. Orfanoudakis, N. K. Panda, P. Palensky, and P. P. Vergara, "GNN-DT: Graph Neural Network Enhanced Decision Transformer for Efficient Optimization in Dynamic Environments," *CoRR* abs/2502.01778 (2025), https://doi/org/10.48550/arXiv.2502.01778.

24. Z. Wang, X. Qu, J. Xiao, B. Chen, and J. Wang, "P2DT: Mitigating Forgetting in Task-Incremental Learning With Progressive Prompt Decision Transformer," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2024, Seoul, Republic of Korea, April 14–19, 2024* (IEEE, 2024), 7265–7269.

25. M. H. Modirrousta, P. F. Arani, R. Kazemi, and M. A. Shoorehdeli, "Analysis of Anomalous Behaviour in Network Systems Using Deep Reinforcement Learning With Convolutional Neural Network Architecture," *CAAI Transactions on Intelligence Technology* 9, no. 6 (2024): 1467–1484, https://doi/org/10.1049/CIT2.12359.

26. T. Yamagata, A. Khalil, and R. Santos-Rodríguez, "Q-Learning Decision Transformer: Leveraging Dynamic Programming for Conditional Sequence Modelling in Offline RL," in *International Conference on Machine Learning, ICML 2023, 23–29 July 2023, Honolulu, Hawaii, USA, Vol. 202 of Proceedings of Machine Learning Research* (PMLR, 2023), 38989–39007.

27. S. Hu, Z. Fan, C. Huang, et al., "Q-Value Regularized Transformer for Offline Reinforcement Learning," in *Forty-First International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21–27, 2024* (OpenReview.net, 2024).

28. R. J. A. A. L. Atkinson, A. H. Waterman, and R. J. Allen, "Does Value-Based Prioritization at Working Memory Enhance Long-Term Memory?," *Memory & Cognition* 52, no. 8 (2024): 1–16, https://doi.org/10.3758/s13421-024-01532-9.

29. J. Schulman, P. Moritz, S. Levine, M. I. Jordan, and P. Abbeel, "High-Dimensional Continuous Control Using Generalized Advantage Estimation," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2–4, 2016, Conference Track Proceedings* (2016).

30. S. Wei, X. Wang, X. Feng, and H. Yu, "Mast: Multi-Agent Safe Transformer for Reinforcement Learning," *IEEE Transactions on Cognitive and Developmental Systems* 17, no. 4 (2025): 1–10, https://doi.org/10.1109/tcds.2025.3533744.

31. C. Robertson, N. A. Tong, T. T. Nguyen, Q. V. Hung Nguyen, and J. Jo, "Resource-Adaptive and OOD-Robust Inference of Deep Neural Networks on Iot Devices," *CAAI Transactions on Intelligence Technology* 10, no. 1 (2025): 115–133, https://doi.org/10.1049/cit2.12384.

32. I. Kostrikov, R. Fergus, J. Tompson, and O. Nachum, "Offline Reinforcement Learning With Fisher Divergence Critic Regularization," in *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18–24 July 2021, Virtual Event*, Vol. 139 *Proceedings of Machine Learning Research* (PMLR, 2021). 5774–5783.

33. Y. Wu, G. Tucker, and O. Nachum, "Behavior Regularized Offline Reinforcement Learning," *CoRR* abs/1911.11361 (2019), https://doi.org/10.48550/arXiv.1911.11361.

34. A. Singh, A. Kumar, Q. Vuong, Y. Chebotar, and S. Levine, "Reds: Offline RL With Heteroskedastic Datasets via Support Constraints," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10–16, 2023* (2023), 23921–23933.

35. Z. Hong, P. Agrawal, R. T. des Combes, and R. Laroche, "Harnessing Mixed Offline Reinforcement Learning Datasets via Trajectory Weighting," in *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1–5, 2023* (OpenReview.net, 2023).

36. Y. Yue, B. Kang, X. Ma, G. Huang, S. Song, and S. Yan, "Offline Prioritized Experience Replay," *CoRR* abs/2306.05412 (2023), https://doi.org/10.48550/arXiv.2306.05412.

37. T. Liu, Y. Li, Y. Lan, H. Gao, W. Pan, and X. Xu, "Adaptive Advantage-Guided Policy Regularization for Offline Reinforcement Learning," in *Forty-First International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21–27, 2024* (OpenReview.net, 2024).

38. J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine, "D4RL: Datasets for Deep Data-Driven Reinforcement Learning," *CoRR* abs/2004.07219 (2020), https://doi.org/10.48550/arXiv.2004.07219.

39. Y. Ma, J. Hao, H. Liang, and C. Xiao, "Rethinking Decision Transformer via Hierarchical Reinforcement Learning," in *Forty-First International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21–27, 2024* (OpenReview.net, 2024).

40. O. Nachum, S. Gu, H. Lee, and S. Levine, "Data-Efficient Hierarchical Reinforcement Learning," in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3–8, 2018, Montréal, Canada* (2018), 3307–3317.

41. L. Guo, Y. Ren, R. Li, and B. Jiang, "Prior Knowledge-Augmented Broad Reinforcement Learning Framework for Fault Diagnosis of Heterogeneous Multiagent Systems," *IEEE Transactions on Cognitive and Developmental Systems* 16, no. 1 (2024): 115–123, https://doi.org/10.1109/tcds.2023.3266791.

42. S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems," *CoRR* abs/2005 (2020): 01643, https://doi.org/10.48550/arXiv.2005.01643.

43. D. Brandfonbrener, A. Bietti, J. Buckman, R. Laroche, and J. Bruna, "When Does Return-Conditioned Supervised Learning Work for Offline Reinforcement Learning?," in *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28–December 9, 2022* (2022), 1542–1553.

44. H. van Hasselt, "Double q-Learning," in *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a Meeting Held 6–9 December 2010, Vancouver, British Columbia, Canada* (Curran Associates Inc., 2010), 2613–2621.

45. R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction* (MIT press, 2018).

46. D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings* (2015).

47. R. Kidambi, A. Rajeswaran, P. Netrapalli, and T. Joachims, "Morel: Model-Based Offline Reinforcement Learning," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6–12, 2020, Virtual* (2020), 21810–21823.

48. Z. Peng, Y. Liu, C. Han, and Z. Zhou, "Implicit Policy Constraint for Offline Reinforcement Learning," *CAAI Transactions on Intelligence Technology* 9, no. 4 (2024): 973–981, https://doi.org/10.1049/cit2.12304.

49. T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning With a Stochastic Actor," in *ICML* (2018), 1861–1870.

50. Z. Wang, J. J. Hunt, and M. Zhou, "Diffusion Policies as an Expressive Policy Class for Offline Reinforcement Learning," in *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1–5, 2023* (OpenReview.net, 2023).

51. A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman, "Relay Policy Learning: Solving Long-Horizon Tasks via Imitation and Reinforcement Learning," in *3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30–November 1, 2019, Proceedings*, eds L. P. Kaelbling, D. Kragic, and K. Sugiura, vol. 100 *Proceedings of Machine Learning Research* (PMLR, 2019), 1025–1037.

52. P. Bhargava, R. Chitnis, A. Geramifard, S. Sodhani, and A. Zhang, "Sequence Modeling Is a Robust Contender for Offline Reinforcement Learning," *CoRR* abs/2305.14550 (2023), https://doi.org/10.48550/arXiv.2305.14550.

53. J. Schulman, S. Levine, P. Abbeel, M. I. Jordan, and P. Moritz, "Trust Region Policy Optimization," in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6–11 July 2015, Vol. 37 of JMLR Workshop and Conference Proceedings* (JMLR.org, 2015), 1889–1897.

54. L. Sun, L. Wang, W. Ding, Y. Qian, and J. Xu, "Neighborhood Multi-Granulation Rough Sets-Based Attribute Reduction Using Lebesgue and Entropy Measures in Incomplete Neighborhood Decision Systems," *Knowledge-Based Systems* 192 (2020): 105373, https://doi.org/10.1016/j.knosys.2019.105373.

Appendix A

Theorem A1. *(Policy improvement guarantee):*

Proof. Based on (Ref. [53], Lemma 1), given any two policies π_1 and π_2, we have □

$$J(\pi_1) - J(\pi_2) = \int_s d_{\pi_1}(s)\big(Q_{\pi_2}(s, \pi_1(s)) - V_{\pi_2}(s)\big)\mathrm{d}s. \tag{A1}$$

Then, π_β is defined as behaviour policy, and $\tilde{\pi}$ is defined as the learnt policy. Substituting these into Equation (A1), we have

$$J(\tilde{\pi}) - J(\pi) = \int_s d_{\pi^*}(s)(Q_\pi(s, \pi^*(s)) - V_\pi(s))\mathrm{d}s, \tag{A2}$$

$$J(\pi_\beta) - J(\pi) = \int_s d_{\pi_\beta}(s)\big(Q_\pi\big(s, \pi_\beta(s)\big) - V_\pi(s)\big)\mathrm{d}s, \tag{A3}$$

Subtracting Equation (A2) from Equation (A3) yields the following:

$$J(\tilde{\pi}) - J(\pi_\beta) = J(\tilde{\pi}) - J(\pi) + J(\pi) - J(\pi_\beta) \tag{A4}$$

$$= (J(\tilde{\pi}) - J(\pi)) - \big(J(\pi_\beta) - J(\pi)\big) \tag{A5}$$

$$= \int_s d_{\tilde{\pi}}(s)(Q(s, \tilde{\pi}) - V(s))\mathrm{d}s - \int_s d_{\pi_\beta}(s)\big(Q(s, \pi_\beta(s)) - V(s)\big)\mathrm{d}s. \tag{A6}$$

Under the proposed policy update mechanism, the state visitation distributions satisfy $d_{\pi^*} \approx d_{\pi_\beta}$, as the algorithm iteratively modifies policies only for a finite subset of states within the continuous state space. Formally, let $\mathcal{S}_u^{(k)} \subset \mathcal{S}$ denote the measurable subset of states updated at iteration k, where $\mu(\mathcal{S}_u^{(k)}) = 0$ under the Lebesgue measure μ [54]. By the countable additivity of measure theory, the probability of nonoverlapping states between $\tilde{\pi}_\beta$ and π_β satisfies

$$\mathbb{P}\left(\mu\left(\mathcal{S} \setminus \bigcup_{k=1}^{\infty} \mathcal{S}_u^{(k)}\right) > 0\right) = 0, \tag{A7}$$

implying $d_{\pi_\beta^*}$ and d_{π_β} coincide almost surely. Consequently, the state visitation probabilities remain approximately invariant under policy updates, that is,

$$\mathbb{E}_{\boldsymbol{s}\sim d_{\tilde{\pi}}}[f(\boldsymbol{s})] \approx \mathbb{E}_{\boldsymbol{s}\sim d_{\pi_\beta}}[f(\boldsymbol{s})], \tag{A8}$$

for any bounded measurable function $f : \mathcal{S} \to \mathbb{R}$. This property ensures the stability of policy evaluation in continuous domains.

Then, we have

$$J(\tilde{\pi}) - J(\pi_\beta) = \int_s d_{\pi_\beta}(s)\big(Q(s, \tilde{\pi}(s)) - Q(s, \pi_\beta(s))\big)\mathrm{d}s, \tag{A9}$$

In Equation (10), we have

$$J(\tilde{\pi}) - J(\pi_\beta) = \int_s d_{\pi_\beta}(s)\big(A(s, \tilde{\pi}(s)) - A(s, \pi_\beta(s))\big)\mathrm{d}s, \tag{A10}$$

Combining Equations (16) and (17), we have

$$A(s, \tilde{\pi}(s)) - A(s, \pi_\beta(s)) \geq 0. \tag{A11}$$

Substituting this into Equation (A11), the proof is completed.

Theorem A2. *(Convergence to near-optimal policies):*

Proof. We extend the construction of the conditioning function f from Theorem A2. Let $f(s_1)$ represent the cumulative return of the deterministic optimal policy π^* under nominal dynamics:

$$f(s_1) = \sum_{t=1}^{H} r\big(T^{\pi^*}(s_1, t), \pi^*\big(T^{\pi^*}(s_1, t)\big)\big), \tag{A12}$$

where $T^{\pi^*}(s_1, t)$ denotes the state reached after t-step transitions under π^*. □

From Lemma 1, the discrepancy between the expected value of f and the optimal policy satisfies

$$\mathbb{E}_{s_1}[f(s_1)] - J(\pi^*) \leq \epsilon H^2. \tag{A13}$$

For the policy π_f, the performance gap can be decomposed as

$$J(\pi^*) - J(\pi_f) = \underbrace{J(\pi^*) - \mathbb{E}[f(s_1)]}_{\text{(I)}} + \underbrace{\mathbb{E}[f(s_1)] - J(\pi_f)}_{\text{(II)}}. \tag{A14}$$

Term (I) is bounded by Equation (A13). For term (II), we apply the return coverage assumption $P_\beta(\hat{r}(\tau) = f(s_1)|s_1) \geq \alpha_f$, yielding

$$\mathbb{E}[f(s_1)] - J(\pi_f) \leq \epsilon\left(\frac{1}{\alpha_f} + 2\right)H^2. \tag{A15}$$

Combining both terms, let $\eta \geq 3$, we have

$$J(\pi^*) - J(\pi_f) \leq \epsilon H^2 + \epsilon\left(\frac{1}{\alpha_f} + 2\right)H^2 \leq \epsilon\left(\frac{1}{\alpha_f} + \eta\right)\mathcal{H}^2. \tag{A16}$$

if $\epsilon = 0$, $\alpha_f \geq 0$, we have

$$J(\pi^*) - J(\pi_f) \leq 0. \tag{A17}$$

if $f(s_1) = A^{\pi^*}(s_1, a_1)$, combining Equation (10), we have

$$\begin{aligned} f(s_1) &= A^{\pi^*}(s_1, a_1) \\ &= Q^{\pi^*}(s_1, a_1) - V^{\pi^*}(s_1). \end{aligned}$$

Under the optimal policy π^*, the state-action value function satisfies

$$Q^{\pi^*}(s, a) = V^{\pi^*}(s). \tag{A18}$$

This implies $A^{\pi^*}(s_1, a_1) = 0$ for optimal actions, leading to

$$\mathbb{E}[f(s_1)] = \mathbb{E}\big[V^{\pi^*}(s_1)\big] = J(\pi^*). \tag{A19}$$

Combining this equality with the original bound from Lemma 1, we have the following equation:

$$J(\pi_f) \leq \mathbb{E}\big[V^{\pi^*}(s_1)\big] + \epsilon\left(\frac{1}{\alpha_f} + 2\right)H^2. \tag{A20}$$

Substituting $\mathbb{E}[f(s_1)] = J(\pi^*)$ in Equation (A20), we have

$$J(\pi^*) - J(\pi_f) \geq -\epsilon\left(\frac{1}{\alpha_f} + 2\right)H^2 \tag{A21}$$

Combining Equations (A16) and (A21), we have

$$-\epsilon\left(\frac{1}{\alpha_f} + 2\right)H^2 \leq J(\pi^*) - J(\pi_f) \leq \epsilon\left(\frac{1}{\alpha_f} + \eta\right)H^2$$

$$\Rightarrow 0 \leq J(\pi^*) - J(\pi_f) \leq 0$$

When $\epsilon \to 0$, $J(\pi_f) \to J(\pi^*)$.