AC/DT: ACTOR-CRITIC DECISION TRANSFORMER FOR OFFLINE-TO-ONLINE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce the Actor-Critic Decision Transformer (AC/DT), an offline-to-online (O2O) reinforcement learning transformer-based method that replaces the behavior-cloning objective of Decision-Transformer-style models with a value-based actor-critic optimization over a sequence-conditioned stochastic Transformer policy. AC/DT combines long-horizon sequence conditioning with explicit policy improvement through twin Markovian critics. To exploit the temporal information available within each sampled context, the critics are trained position-wise using context-truncated λ-returns, providing multi-step value targets while retaining lightweight Markovian value functions. During offline pretraining, we mitigate out-of-distribution overestimation through conservative value regularization. During online fine-tuning, conservative regularization is restricted to offline experience. On D4RL benchmarks, AC/DT moves beyond the offline data distribution and achieves competitive performance against action-likelihood-based Transformer baselines such as ODT, hybrid variants like ODT+TD3 and other state-of-the-art O2O methods. A state-matched distributional analysis further shows that AC/DT reduces adherence to the dataset action marginals relative to likelihood-based ODT, providing evidence of a consistent shift in the policy-induced action marginals toward an expert reference.

1 INTRODUCTION

The integration of transformers (Vaswani et al., 2017) into Reinforcement Learning (RL) has provided a novel perspective over RL control as a sequence-modeling problem. While Decision Transformer (DT) (Chen et al., 2021) demonstrated the power of conditioning actions on desired returns, it remains constrained by its reliance on maximizing the likelihood of actions within a fixed offline dataset. Even recent advancements, such as the Online Decision Transformer (ODT) (Zheng et al., 2022), which adds stochasticity and entropy regularization to encourage exploration, retain action-likelihood maximization as the primary training signal.

Such likelihood-based training matches the offline dataset's action distribution rather than directly optimizing expected return (Chen et al., 2021; Zheng et al., 2022). This can constrain the policy to the behavioral data support (Levine et al., 2020; Wu et al., 2019; Kumar et al., 2020), limiting the exploration of higher-return actions required for effective offline-to-online (O2O) improvement.

To overcome these limitations, we propose the Actor-Critic Decision Transformer (AC/DT), bridging between sequence modeling and value-based approaches in O2O RL. AC/DT retains the transformer's ability to model long-horizon dependencies via a non-Markovian actor, while optimizing the policy via a Markovian twin-critic architecture. Moving beyond hybrid architectures that treat value-based optimization as an auxiliary correction (Yan et al., 2024), AC/DT directly optimizes the Transformer policy against critic estimates rather than dataset-action likelihood. The actor objective is evaluated position-wise across the context, while the critics are trained at the same positions using context-truncated λ-returns. These targets propagate the rewards observed within each context through position-dependent multi-step backups and soft value bootstrapping from the sequence-conditioned policy.

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.

Furthermore, we investigate the dynamics of the offline-to-online transition. During offline pretraining, we integrate a conservative value regularization term (Kumar et al., 2020) to mitigate the overestimation of out-of-distribution actions. During online fine-tuning, the same position-wise critic objective is retained, but the conservative penalty applies only to offline-origin samples.

Our experimental results on D4RL benchmarks (Fu et al., 2020) demonstrate that AC/DT moves beyond the offline data distribution, achieving competitive performance against state-of-the-art transformer-based baselines and established O2O methods. Furthermore, our state-matched action-marginal analysis shows that, on HalfCheetah, AC/DT departs more strongly from the medium-data marginals and is closer to the expert-data marginals than likelihood-based ODT.

In summary, our main contributions are:

- A Unified Actor-Critic Transformer Architecture: We introduce AC/DT, which replaces the supervised action-likelihood objective of decision transformers with a sequence-conditioned soft actor-critic objective, enabling explicit policy improvement across all decision positions in the Transformer context. The same value-based framework is retained across offline pretraining and online fine-tuning, with conservative critic regularization restricted to offline-origin experience.
- Position-wise Context-Truncated λ-Return Critic Learning: We reuse each sampled Transformer context to train lightweight Markovian critics at every decision position. Each position receives a context-truncated λ-return whose maximum backup horizon is determined by the remaining suffix of the context, ranging from K steps at the earliest position to one step at the context boundary. This propagates the reward information already contained in the Transformer context while bootstrapping through the sequence-conditioned actor.
- Empirical and Distributional Analysis: We provide evidence on D4RL (Fu et al., 2020) benchmarks showing that AC/DT achieves strong offline and online performance, together with a state-matched analysis showing that AC/DT shifts its policy-induced action marginals toward the expert reference relative to likelihood-based ODT.

2 Related Works

Offline Reinforcement Learning. Offline RL (Levine et al., 2020) learns policies from fixed datasets without environmental interaction and must address distributional shift between learned policies and the behavioral data. Representative approaches constrain the learned policy to the data distribution (Wu et al., 2019), regularize out-of-distribution values as in CQL (Kumar et al., 2020), or avoid explicit OOD action evaluation through objectives such as IQL (Kostrikov et al., 2021). AC/DT builds on conservative value regularization for critic stability while targeting offline-to-online adaptation within a unified actor-critic Transformer framework.

Transformers in Reinforcement Learning. The Decision Transformer (DT) architecture (Chen et al., 2021) reformulates RL as a sequence modeling problem, generating actions conditioned on desired returns-to-go. Several offline extensions incorporate value information into this paradigm: Q-value Regularized Transformer (Hu et al., 2024) introduces action-value maximization, Q-Learning Decision Transformer (QDT) (Yamagata et al., 2023) uses Q-learning for return relabeling, and Critic-Guided Decision Transformer (CGDT) (Wang et al., 2024) uses critic guidance under stochastic dynamics. These methods remain offline and retain a supervised, return-conditioned training backbone, using value information primarily for auxiliary guidance or return relabeling. In a complementary purely online actor-critic setting, T-SAC (Tian et al., 2026) explores sequence-conditioned Transformer critics trained with multi-horizon N-step supervision. In contrast, AC/DT removes the supervised action-likelihood objective and directly optimizes the stochastic Transformer policy through an actor-critic objective across both offline pretraining and online fine-tuning.

Offline-to-Online (O2O) Reinforcement Learning. Bridging the gap between offline pretraining and online fine-tuning is critical for deploying RL agents. Naive fine-tuning often results in immediate performance collapse. Lee et al. (2022) address this via balanced replay and pessimistic Q-ensembles; Calibrated Q-Learning (Cal-QL) (Nakamoto et al., 2023) adapts CQL to O2O by calibrating Q-values. Kim & Oh (2026) introduce Conservative Peng's Q(λ) (CPQL), extending Peng-style multi-step value estimation to conservative offline RL and subsequently transitioning to PQL for online fine-tuning. In contrast, AC/DT couples multi-step value learning directly to the sampled

Transformer context, using the same context both to condition the sequence policy and to provide position-wise context-truncated λ-return supervision to lightweight Markovian critics. Adaptive Policy Learning (APL) (Zheng et al., 2023) distinguishes pessimistic updates for offline experience from optimistic/greedy updates through a two-level replay mechanism that separately samples near-on-policy and replay data. In the transformer setting, the Online Decision Transformer (ODT) (Zheng et al., 2022) introduces stochastic policies with entropy regularization to enable exploration during fine-tuning while maintaining likelihood-based training. Value-Guided Decision Transformer (VDT) (Zheng et al., 2025) incorporates value guidance into Decision Transformer training across offline and online settings while retaining behavior-cloning components in the policy objective. Yan et al. (2024) augment ODT with TD3-style gradients while retaining the supervised DT objective as the primary update. In contrast, AC/DT fully replaces action-likelihood training with a soft actor-critic objective, updating the transformer policy to maximize critic-estimated return, while training the critics position-wise across the sampled context.

3 PRELIMINARIES

A Markov Decision Process (MDP) is defined by the tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$, where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ the action space, P the transition dynamics, R the reward function, and $\gamma \in [0, 1]$ the discount factor. A policy π aims to maximize the expected cumulative discounted return $\mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_t]$. In Offline Reinforcement Learning (Levine et al., 2020), the agent must learn from a fixed dataset $\mathcal{D} = \{(s_t^i, a_t^i, r_t^i, s_{t+1}^i, d_t^i)\}$ of trajectories previously collected by a baseline policy π_b, where i represents a trajectory and t a time step and d_t^i indicates whether the transition is terminal. Unlike online RL, the agent cannot interact with the environment, making the quality of $\mathcal{D}$ critical.

The Decision Transformer (DT) (Chen et al., 2021) formulates offline reinforcement learning as a sequence modeling problem. Given an offline dataset of trajectories $\mathcal{D}$, Return-to-go (RTG) tokens are computed as $\hat{R}_t^i = \sum_{k=t}^{T} r_k^i$ and combined with states and actions into sequences $\tau^i = (\hat{R}_0^i, s_0^i, a_0^i, \ldots, \hat{R}_T^i, s_T^i, a_T^i)$. A deterministic policy $\pi_{\mathrm{DT}}(a_t \mid \hat{R}_{t-K+1:t}, s_{t-K+1:t}, a_{t-K+1:t-1})$ is then learned using a causal transformer (Radford et al., 2018) over a context window of K decision positions. A first O2O version of DT is the Online Decision Transformer (ODT) (Zheng et al., 2022) which extends the classic DT for online interaction by parameterizing a Gaussian stochastic policy $\pi_\theta(a_t \mid \tau_t)$ and incorporating entropy regularization. ODT pretrains this stochastic policy on $\mathcal{D}$ and fine-tunes it using a trajectory replay buffer $\mathcal{B}$ initialized with high-return offline trajectories and subsequently updated in a FIFO manner with online rollouts. Despite these changes, the core optimization objective of ODT remains behavior cloning-based. In particular, ODT maximizes the likelihood of observed actions while imposing a sequence-level entropy constraint,

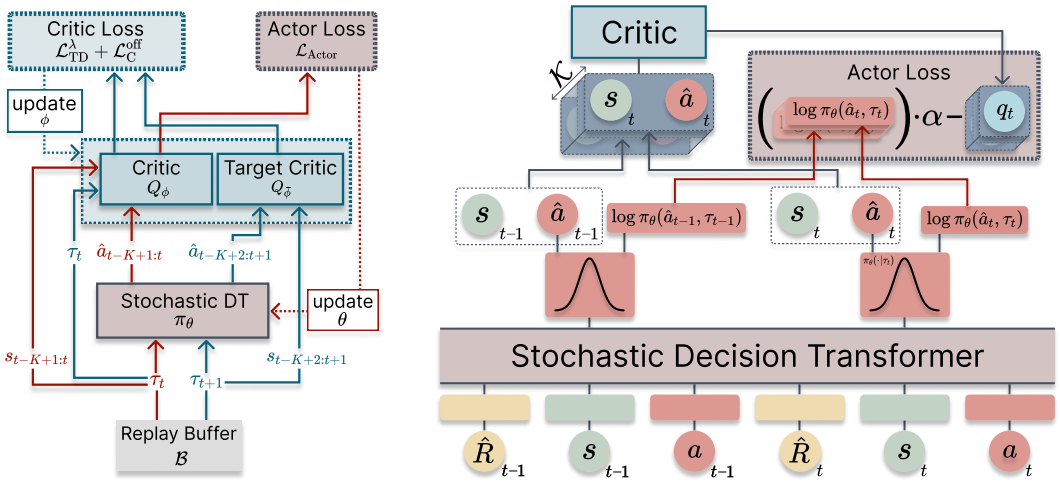
$$\mathcal{L}_{\mathrm{ODT}}(\theta) = \mathbb{E}_{\tau \sim \mathcal{B},\, t \sim \mathrm{Unif}(\tau)} \left[-\log \pi_\theta(a_t \mid \tau_t) - \alpha \mathcal{H}\big(\pi_\theta(\cdot \mid \tau_t)\big) \right], \tag{1}$$

where $\tau_t = (\hat{R}_{t-K+1:t}, s_{t-K+1:t}, a_{t-K+1:t-1})$ denotes the transformer context and α controls the strength of entropy regularization. [1]

4 ACTOR-CRITIC DECISION TRANSFORMER

The Actor-Critic Decision Transformer (AC/DT) is a novel O2O reinforcement learning method designed to pretrain (offline) and fine-tune (online) a stochastic decision transformer policy via value-based optimization. AC/DT employs the decision transformer exclusively as a policy network, enabling long-horizon sequence conditioning through attention. Unlike prior DT variants that optimize the policy via supervised action-likelihood maximization (i.e., behavioral cloning) over previously collected trajectories, AC/DT removes the behavioral cloning objective and instead trains the policy to maximize the expected return estimated by a learned critic (equation 5), following the actor-critic paradigm (Haarnoja et al., 2018). This design leverages the transformer for sequence conditioning while enabling explicit policy improvement through value learning.

[1] We remove the trajectory index i here to lighten the notation.



(a) AC/DT optimization workflow

(b) Zoom-in of the value-based DT optimization in (a)

Figure 1: AC/DT optimization workflow. (a) The stochastic Decision Transformer actor and twin critics are trained from context windows sampled from the replay buffer: the actor is optimized through the value-based objective, while the critics are trained using position-wise context-truncated λ-returns, with conservative regularization restricted to offline-origin data. (b) The Transformer actor generates sequence-conditioned actions that are evaluated by the Markovian critics to form the value-based actor objective in equation 5.

4.1 AC/DT Architecture Overview

The AC/DT training loop is summarized in Figure 1a. AC/DT maintains a replay buffer $\mathcal{B}$ that stores trajectory data used to construct fixed-length context windows of size K. In the offline phase, $\mathcal{B}$ is initialized with trajectories from the offline dataset $\mathcal{D}$. As detailed in Section 4.3, during training, context windows sampled from $\mathcal{B}$ are fed to a stochastic decision transformer actor, which produces an autoregressive distribution over actions (Figure 1b). At timestep t, the transformer conditions on the context

$$\tau_t := (\hat{R}_{t-K+1:t},\, s_{t-K+1:t},\, a_{t-K+1:t-1}), \tag{2}$$

where $\hat{R}$ denotes return-to-go tokens, s state tokens, and a past action tokens.

For each sampled window, the actor predicts actions autoregressively at the decision positions within the window. The critic evaluates the resulting state-action pairs and provides the value signal used in the actor update, together with the policy log-probability (equation 5).

As described in Section 4.4, the twin critics are trained at all K decision positions of the sampled context using context-truncated λ-returns with soft value bootstrapping, together with a conservative regularization term applied to offline data to mitigate overestimation of out-of-distribution actions (Figure 1a). Thus, both actor and critic learning exploit the full sampled context: the actor applies the value-based policy objective across decision positions (Section 4.3), while the critic propagates the observed reward sequence through multi-step value targets (Section 4.4).

In the online phase, the agent interacts with the environment using the current policy and appends newly collected trajectories to $\mathcal{B}$ (FIFO), yielding a mixture of offline and online data for subsequent updates. During online interaction, the Transformer is conditioned on a target return initialized at the start of each episode and updated by subtracting each observed scaled reward, while trajectories stored in $\mathcal{B}$ are hindsight-relabeled with the realized return-to-go before training.

To avoid conservative regularization on newly collected online data, we apply it only to offline-origin samples. Online samples are instead optimized without the conservative penalty.

4.2 Actor and Critic Parameterization

As illustrated in Figure 1b, we parameterize the actor as a stochastic DT that outputs a Gaussian policy conditioned on a context window of length K. At each timestep t, the actor conditions on the context τ_t defined in equation 2. The actor produces the parameters of a Gaussian distribution underlying the stochastic policy

$$\pi_\theta(a_t \mid \tau_t) = \mathcal{N}\big(\mu_\theta(\tau_t),\, \Sigma_\theta(\tau_t)\big). \tag{3}$$

This sequence-conditioned policy enables action selection by attending to dependencies across the sampled context.

Alongside the autoregressive actor, we employ a Markovian twin-critic architecture consisting of two networks, $Q_{\phi_1}(s,a)$ and $Q_{\phi_2}(s,a)$, and their corresponding target networks $Q_{\bar{\phi}_1}(s,a)$ and $Q_{\bar{\phi}_2}(s,a)$. Each critic maps a state-action pair to a scalar action-value estimate.

These value estimates are used for actor optimization and for constructing the multi-step critic targets introduced in the following sections. While the actor is sequence-conditioned, the critics remain Markovian and learn state-action value estimates by aggregating context-dependent learning signals across the replay contexts in which state-action pairs are observed; sequence dependence enters the critic update through the actions and soft bootstrap values induced by the Transformer policy and the position-wise multi-step targets introduced in Section 4.4. Accordingly, $Q_\phi(s,a)$ should be interpreted as a Markovian approximation that aggregates context-dependent targets induced by the sequence-conditioned policy over the replay distribution, rather than as an exact history-conditioned value $Q(\tau,a)$.

Why Markovian MLP critics? Our use of lightweight Markovian critics is consistent with Yan et al. (2024), who combine them with a sequence-conditioned Transformer policy and report improved stability over recurrent history-dependent critics. Extending this analysis to Transformer-based critics, we further assess the Markovian formulation against sequence-conditioned critics using the Transformer critic architecture proposed by Tian et al. (2026). The Markovian formulation consistently performs better in our setting, with detailed results in Appendix A.1.

4.3 Value-Based DT Optimization

Standard ODT trains the transformer policy by maximizing the likelihood of dataset actions with entropy regularization. In contrast, AC/DT, as shown in Figure 1b, optimizes the policy directly against the critic's estimate. The actor, as defined in equation 3, defines an autoregressive Gaussian policy over the decision positions within a context window. In particular, for any index $t' \in \{t-K+1,\dots,t\}$, the prediction at position t' is conditioned on the corresponding prefix context

$$\tau_{t'} \coloneqq (\hat{R}_{t-K+1:t'},\, s_{t-K+1:t'},\, a_{t-K+1:t'-1}), \tag{4}$$

and the policy is $\pi_\theta(\cdot \mid \tau_{t'}) = \mathcal{N}(\mu_\theta(\tau_{t'}),\, \Sigma_\theta(\tau_{t'}))$. This preserves the key strength of DTs: action distributions are conditioned on long-horizon trajectory information through attention, rather than solely on the Markovian state.

Application of the Markovian critic to the context window. Since each context contains K decision positions, we evaluate the soft actor-critic objective at every position and average the resulting terms. Concretely, for each $t' \in \{t-K+1,\dots,t\}$ we draw an action from the transformer-conditioned policy $\hat{a}_{t'} \sim \pi_\theta(\cdot \mid \tau_{t'})$, evaluate it with the twin Markovian critics $q_{t',j} = Q_{\phi_j}(s_{t'}, \hat{a}_{t'})$ with $j \in \{1,2\}$ and optimize the actor with the sequence-conditioned soft actor-critic objective

$$\mathcal{L}_{\text{Actor}}(\theta) = \mathbb{E}_{\tau_t \sim \mathcal{B}}\left[\frac{1}{K}\sum_{t'=t-K+1}^{t}\left(\alpha \log \pi_\theta(\hat{a}_{t'} \mid \tau_{t'}) - \min_{j\in\{1,2\}} q_{t',j}\right)\right], \tag{5}$$

where α is the entropy temperature tuned following the objective declared in (Haarnoja et al., 2018).

4.4 Critic Learning with Context-Truncated λ-Returns

We reuse the context windows sampled for policy optimization to train the critics at every decision position, as illustrated in Figure 2. For a context window of length K ending at timestep t, each

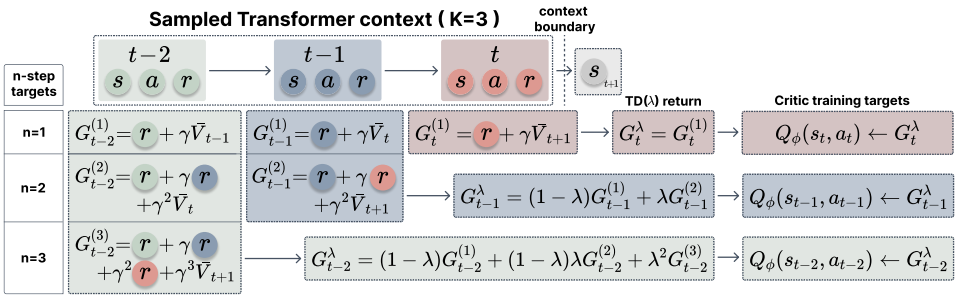


Figure 2: Position-wise critic learning with context-truncated λ-returns. For each decision position in a context of length K, AC/DT combines the available n-step targets up to the context boundary, yielding progressively shorter maximum backup horizons from the earliest to the latest position.

position $t' \in \{t-K+1, \ldots, t\}$ defines a reward suffix extending to the end of the context. We combine backups of different lengths through a context-truncated λ-return, with each backup terminated by a soft value estimate, to construct a multi-step target for the corresponding state-action pair. This allows the critic to exploit the temporal information already contained in the Transformer context while retaining lightweight Markovian state-action MLP critics.

Soft value bootstrapping. For each transition at decision position t', the remaining return is bootstrapped from the successor state $s_{t'+1}$. To construct the corresponding actor context, we shift the sampled context window one transition forward. We then sample the successor action $\hat{a}_{t'+1} \sim \pi_\theta(\cdot \mid \tau_{t'+1})$ and define the soft target value as

$$\bar{V}_{t'+1} = \min_{j \in \{1,2\}} Q_{\bar{\phi}_j}(s_{t'+1}, \hat{a}_{t'+1}) - \alpha \log \pi_\theta(\hat{a}_{t'+1} \mid \tau_{t'+1}). \tag{6}$$

where $\tau_{t'+1}$ denotes the corresponding actor context obtained from the one-step-shifted context window; $Q_{\bar{\phi}_1}$ and $Q_{\bar{\phi}_2}$ are the target critics and α is the entropy temperature. Thus, although the critics themselves remain Markovian, their bootstrap values are induced by the sequence-conditioned actor.

Context-truncated λ-returns For each decision position t', we consider n-step backups with $n \in \{1, \ldots, t-t'+1\}$, where n denotes the number of transitions included before bootstrapping. The corresponding target is

$$G_{t'}^{(n)} = \sum_{i=0}^{n-1} \gamma^i r_{t'+i} + \gamma^n (1 - d_{t'+n-1}) \bar{V}_{t'+n}. \tag{7}$$

where $d_{t'+n-1}$ indicates whether the final transition of the n-step backup is terminal. Thus, $G_{t'}^{(n)}$ propagates n observed rewards from the context before bootstrapping through the soft target value. 1-step target uses $r_{t'}$ and bootstraps at $s_{t'+1}$, while the longest target uses the complete reward suffix $(r_{t'}, \ldots, r_t)$ before bootstrapping at s_{t+1}. If the final transition is terminal, the bootstrap term is omitted through $(1 - d_t)$.

Rather than selecting a single backup length, we combine the multi-step targets associated with position t' as

$$G_{t'}^{\lambda} = (1-\lambda) \sum_{n=1}^{t-t'} \lambda^{n-1} G_{t'}^{(n)} + \lambda^{t-t'} G_{t'}^{(t-t'+1)}, \qquad \lambda \in [0,1]. \tag{8}$$

The final term absorbs the remaining weight, yielding a normalized finite-horizon mixture. The parameter λ controls the balance between early bootstrapping and the propagation of rewards observed within the context. For $\lambda = 0$, each decision position receives a one-step target; increasing λ assigns progressively more weight to longer backups, while $\lambda = 1$ selects the longest context-truncated return available from that position, maximally propagating the observed rewards within the available context while retaining soft value bootstrapping at its boundary. Importantly, the maximum backup horizon is position-dependent: the last decision position admits only a one-step target, while earlier positions combine progressively longer backups, up to $\mathcal{K}$ steps for the earliest position in the context.

Critic objective. The twin critics are trained against the corresponding context-truncated targets at every decision position:

$$\mathcal{L}_{\mathrm{TD}}^{\lambda}(\phi_j) = \mathbb{E}_{\tau_t \sim \mathcal{B}} \left[\frac{1}{K} \sum_{t'=t-K+1}^{t} \left(Q_{\phi_j}(s_{t'}, a_{t'}) - G_{t'}^{\lambda} \right)^2 \right]. \tag{9}$$

The TD objective sums the losses of the two critics, $\mathcal{L}_{\mathrm{TD}}^{\lambda} = \mathcal{L}_{\mathrm{TD}}^{\lambda}(\phi_1) + \mathcal{L}_{\mathrm{TD}}^{\lambda}(\phi_2)$.

This formulation makes the sampled Transformer context serve two complementary roles: it conditions policy optimization and simultaneously provides position-wise multi-step supervision for critic learning. The λ-weighted aggregation in equation 8 uses an uncorrected off-policy multi-step return structure of Peng's $Q(\lambda)$ (Peng & Williams, 1994; Kozuno et al., 2021), in which intermediate rewards are taken directly from replay trajectories without importance-sampling corrections. AC/DT couples this return construction to the Transformer training context and applies it position-wise: the available context suffix determines a different maximum backup horizon at each decision position, from K steps at the earliest position to one step at the context boundary, while the residual return is completed using the soft value induced by the sequence-conditioned policy. Section 4.5 complements this multi-step critic objective with conservative regularization restricted to offline data.

4.5 Source-Aware Conservative Regularization

While the context-truncated λ-return objective introduced in Section 4.4 provides position-wise multi-step value targets for critic learning, offline actor-critic optimization remains susceptible to overestimated Q-values for actions that are poorly supported by the offline data. We therefore complement the TD objective with a conservative regularizer inspired by Conservative Q-Learning (CQL). AC/DT applies conservative regularization according to whether samples originate from offline or online data within its unified replay buffer: offline-origin samples receive the conservative penalty, whereas online-origin samples are optimized through the context-truncated TD objective alone. Consistently with the position-wise critic training introduced in Section 4.4, the conservative penalty is evaluated at every decision position of each offline context window. Let $\mathcal{B}_o \subseteq \mathcal{B}$ denote the subset of replay data originating from the offline dataset. For each critic Q_{ϕ_j}, we define

$$\mathcal{L}_{\mathrm{C}}^{\mathrm{off}}(\phi_j) = \beta_C \cdot \mathbb{E}_{\tau_t \sim \mathcal{B}_o} \left[\frac{1}{K} \sum_{t'=t-K+1}^{t} \left(\log \sum_a \exp\left(Q_{\phi_j}(s_{t'}, a) \right) - Q_{\phi_j}(s_{t'}, a_{t'}) \right) \right], \qquad j \in \{1, 2\}. \tag{10}$$

where β_C controls the strength of the conservative regularization. Following the sampling strategy used in CQL(H) (Kumar et al., 2020), we approximate the $\log \sum_a \exp(Q(s_{t'}, a))$ term in equation 10 using sampled candidate actions, with policy-dependent candidates generated by the sequence-conditioned Transformer actor.

The resulting critic objective is conditioned on the source of the training data:

$$\mathcal{L}_{\mathrm{Critic}} = \begin{cases} \mathcal{L}_{\mathrm{TD}}^{\lambda} + \sum_{j=1}^{2} \mathcal{L}_{\mathrm{C}}^{\mathrm{off}}(\phi_j), & \text{offline-origin data,} \\ \mathcal{L}_{\mathrm{TD}}^{\lambda}, & \text{online-origin data.} \end{cases} \tag{11}$$

For mixed replay batches, the context-truncated TD component is evaluated on all sampled data, while the conservative component is evaluated only on the offline-origin subset $\mathcal{B}_o$. During offline pretraining, $\mathcal{B}_o = \mathcal{B}$, such that all critic updates combine context-truncated multi-step value learning with conservative regularization. During online fine-tuning, offline-origin samples retain the conservative term while online-origin samples are trained with the context-truncated TD objective alone. Conservative regularization is therefore restricted to the subset of offline-origin samples and disappears once offline experience has been fully evicted from replay, without requiring an explicit decay schedule.

5 Experiments

We designed our experiments to investigate the effectiveness of AC/DT in bridging the gap between offline pretraining and online fine-tuning. Our evaluation aims to answer the following re-

search questions: (Q1) Offline Performance (Sec. 5.2): Does shifting from a supervised sequence-modeling objective to a value-based RL objective enable AC/DT to learn higher-quality policies during offline pretraining? (Q2) Online Fine-tuning (Sec. 5.2): Can AC/DT achieve strong online fine-tuning performance compared with Transformer-based and value-based baselines under a common evaluation protocol? (Q3) Context-Truncated λ-Returns (Sec. 5.2): How does the degree of multi-step reward propagation controlled by λ affect offline pretraining and online fine-tuning compared to position-wise one-step critic targets? Beyond these research questions, we conduct additional ablation studies on other design choices in AC/DT, including the sequence-conditioned actor (Appendix A.2), the source-aware conservative regularization (Appendix B) and the context length (Appendix G).

5.1 Experimental Setup

Baselines. We compare AC/DT against transformer-based O2O methods, namely ODT (Zheng et al., 2022), ODT+TD3 (Yan et al., 2024). We also include representative offline RL methods that can be naturally continued online, including IQL (Kostrikov et al., 2021), AWAC (Nair et al., 2020) and CQL (Kumar et al., 2020). Published results for additional relevant methods, including Cal-QL (Nakamoto et al., 2023), VDT (Zheng et al., 2025), and CPQL/PQL (Kim & Oh, 2026), are reported separately in Appendix Table 8. Using the publicly available code and methodological details, we were unable to establish a reproducible evaluation of these methods under the common evaluation protocol adopted for Table 1. We therefore report their published results for completeness, while treating them separately from the main comparison.

Experimental setting and code availability. All agents are pretrained on the corresponding fixed offline dataset and subsequently fine-tuned for 10^6 environment interactions. Methods in Table 1 are evaluated under a common evaluation protocol using 5 random seeds; we report the mean normalized return and standard deviation across seeds, together with the interquartile mean (IQM), a robust aggregate performance metric for RL evaluation (Agarwal et al., 2021), computed separately over MuJoCo, AntMaze, and the complete benchmark. Hyperparameters and implementation details are provided in Appendix E. **The code will be released upon acceptance.**

Benchmarks. We evaluate on ten D4RL (Fu et al., 2020) tasks, comprising six MuJoCo locomotion tasks and four sparse-reward AntMaze tasks, and report the D4RL normalized score (0 for random and 100 for expert).

5.2 Performance comparison on the D4RL benchmark

Results are reported in Table 1.

(Q1) Offline performance. AC/DT achieves the highest mean offline normalized score on 9 out of 10 tasks and the highest aggregate IQM on MuJoCo (81.1), AntMaze (48.3), and across the complete benchmark (69.7), showing that AC/DT provides a stronger DT-style offline initialization than likelihood-based Transformer training.

(Q2) Online Fine-tuning: Across D4RL tasks, AC/DT after 10^6 environment interactions attains the highest mean normalized return on 9 out of 10 tasks and the highest aggregate IQM on MuJoCo (105.5), AntMaze (97.4), and across the complete benchmark (100.1). AC/DT outperforms ODT and ODT+TD3 on the aggregate metrics and achieves particularly strong improvements on AntMaze, suggesting that fully replacing action-likelihood updates with actor-critic optimization enables effective online improvement. Compared with the value-based baselines evaluated under the same protocol, AC/DT also achieves the highest aggregate online IQM, while CQL obtains a slightly higher final mean score on hopper-medium-replay (103.7 vs. 103.1). Overall, these results indicate that value-based policy improvement provides both a strong offline initialization and substantial subsequent online improvement.

(Q3) Context-Truncated λ-Returns: We evaluate the role of λ in multi-step context propagation using $\lambda \in \{0, 0.5, 0.9, 1\}$, where $\lambda = 0$ reduces the critic update to position-wise one-step targets and $\lambda = 1$ selects the longest context-truncated return available at each decision position. The results in Appendix C show that offline effects are task-dependent, while online differences are most pronounced on sparse-reward AntMaze tasks: $\lambda = 0$ is markedly less robust, whereas larger values maintain substantially stronger performance. The learning curves in Figure 7 further illustrate the

pronounced online collapse of $\lambda = 0$ on the reported large AntMaze tasks. No single λ dominates every environment; $\lambda = 0.9$, used in the main experiments, provides strong performance across both locomotion and AntMaze. Thus, propagating observed rewards across the available Transformer context is an important component of critic learning, while the precise weighting of shorter and longer backups is task-dependent.

Table 1: D4RL normalized scores under the common evaluation protocol. **Bold** denotes the best offline performance and **underlined bold** the best online performance after 1M environment steps; $\pm$ values denote the standard deviation over 5 random seeds.

Dataset	IQL		CQL		AWAC		ODT		ODT+TD3		AC/DT	
	Off.	On.	Off.	On.	Off.	On.	Off.	On.	Off.	On.	Off.	On.
hopper-medium	$59.6_{\pm 1.4}$	$69.7_{\pm 3.2}$	$56.9_{\pm 1.7}$	$64.8_{\pm 5.0}$	$67.4_{\pm 3.2}$	$103.2_{\pm 0.3}$	$46.3_{\pm 4.6}$	$99.3_{\pm 0.7}$	$51.3_{\pm 7.1}$	$83.4_{\pm 7.1}$	$\mathbf{86.4}_{\pm 4.0}$	$\underline{\mathbf{103.7}}_{\pm 3.2}$
hopper-medium-replay	$70.6_{\pm 3.2}$	$92.5_{\pm 10.2}$	$94.6_{\pm 0.3}$	$\underline{\mathbf{103.7}}_{\pm 0.0}$	$94.2_{\pm 2.8}$	$102.8_{\pm 0.4}$	$66.4_{\pm 24.7}$	$89.5_{\pm 5.2}$	$33.8_{\pm 27.0}$	$76.0_{\pm 17.5}$	$\mathbf{98.6}_{\pm 1.0}$	$103.1_{\pm 8.4}$
walker-medium	$77.5_{\pm 1.1}$	$89.7_{\pm 3.2}$	$82.1_{\pm 0.2}$	$84.9_{\pm 0.2}$	$75.5_{\pm 8.4}$	$91.6_{\pm 1.7}$	$59.5_{\pm 29.8}$	$75.1_{\pm 5.2}$	$66.5_{\pm 4.6}$	$93.6_{\pm 33.6}$	$\mathbf{91.2}_{\pm 1.2}$	$\underline{\mathbf{128.2}}_{\pm 4.7}$
walker-medium-replay	$73.5_{\pm 2.7}$	$101.7_{\pm 2.1}$	$87.9_{\pm 0.6}$	$112.1_{\pm 0.8}$	$79.7_{\pm 1.0}$	$106.2_{\pm 3.9}$	$58.7_{\pm 17.4}$	$76.4_{\pm 7.8}$	$66.3_{\pm 8.0}$	$101.3_{\pm 7.0}$	$\mathbf{92.7}_{\pm 1.9}$	$\underline{\mathbf{117.3}}_{\pm 4.1}$
halfcheetah-medium	$47.8_{\pm 0.1}$	$57.7_{\pm 0.8}$	$48.7_{\pm 0.1}$	$81.9_{\pm 0.5}$	$49.5_{\pm 0.1}$	$67.1_{\pm 0.8}$	$19.1_{\pm 9.6}$	$41.9_{\pm 1.0}$	$41.4_{\pm 1.7}$	$75.9_{\pm 4.8}$	$\mathbf{52.9}_{\pm 0.3}$	$\underline{\mathbf{99.5}}_{\pm 1.6}$
halfcheetah-medium-replay	$43.8_{\pm 0.3}$	$53.9_{\pm 0.6}$	$46.7_{\pm 0.0}$	$85.3_{\pm 0.8}$	$45.3_{\pm 0.3}$	$58.1_{\pm 1.7}$	$17.2_{\pm 9.2}$	$38.9_{\pm 1.7}$	$7.3_{\pm 6.0}$	$77.3_{\pm 1.9}$	$\mathbf{52.3}_{\pm 0.8}$	$\underline{\mathbf{97.2}}_{\pm 5.6}$
MuJoCo IQM	63.1	76.9	69.0	91.6	68.6	96.2	45.1	71.1	50.3	85.5	**81.1**	$\underline{\mathbf{105.5}}$
antmaze-medium-play	$59.0_{\pm 3.8}$	$89.8_{\pm 2.9}$	$53.7_{\pm 6.7}$	$97.0_{\pm 1.4}$	$0.7_{\pm 0.3}$	$0.0_{\pm 0.0}$	$1.0_{\pm 0.8}$	$5.7_{\pm 2.1}$	$51.6_{\pm 15.7}$	$89.0_{\pm 4.4}$	$\mathbf{73.0}_{\pm 10.2}$	$\underline{\mathbf{98.8}}_{\pm 1.0}$
antmaze-medium-diverse	$60.2_{\pm 1.1}$	$94.0_{\pm 2.8}$	$56.3_{\pm 1.8}$	$96.7_{\pm 0.5}$	$0.8_{\pm 0.8}$	$0.0_{\pm 0.0}$	$0.7_{\pm 0.5}$	$6.3_{\pm 1.7}$	$11.6_{\pm 10.4}$	$62.4_{\pm 16.1}$	$\mathbf{63.8}_{\pm 22.1}$	$\underline{\mathbf{98.2}}_{\pm 1.5}$
antmaze-large-play	$30.5_{\pm 3.3}$	$65.4_{\pm 2.4}$	$26.3_{\pm 7.6}$	$91.3_{\pm 2.4}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$\mathbf{42.6}_{\pm 11.0}$	$\underline{\mathbf{96.4}}_{\pm 1.7}$
antmaze-large-diverse	$23.5_{\pm 5.8}$	$67.0_{\pm 7.3}$	$\mathbf{28.5}_{\pm 5.9}$	$91.7_{\pm 4.9}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$1.3_{\pm 0.5}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$23.6_{\pm 6.9}$	$\underline{\mathbf{96.2}}_{\pm 1.3}$
AntMaze IQM	44.6	79.4	40.7	95.1	0.2	0.0	0.1	2.8	5.2	31.2	**48.3**	$\underline{\mathbf{97.4}}$
Global IQM	56.5	78.3	53.9	91.6	44.5	62.1	26.5	53.3	33.1	76.7	**69.7**	$\underline{\mathbf{100.1}}$

5.3 Going Beyond the Offline Data Distribution

To characterize the distributional differences between AC/DT and action-likelihood-based ODT, we perform a state-matched action-marginal analysis on halfcheetah-medium. After offline pretraining and 10^6 online environment steps on the halfcheetah-medium dataset, we query ODT and AC/DT on the same states drawn from (i) the medium dataset and (ii) the unseen halfcheetah-expert dataset, and compare the resulting policy-induced per-dimension action marginals with the corresponding dataset action marginals using average per-dimension KL divergence. The full protocol, per-action statistics, and KDE visualizations are provided in Appendix D.

On expert-dataset states, as shown in Table 6, AC/DT produces action marginals substantially closer to the expert reference than ODT, with an average KL divergence of 0.0920 versus 0.2700 across the six action dimensions. On medium-dataset states, ODT remains much closer to the behavioral action marginals (0.0018 versus 0.0242), whereas AC/DT departs more strongly from the offline action marginals. Together, these state-matched results show a consistent contrast across all six action dimensions: AC/DT's policy-induced marginals are closer to the expert-data reference, while ODT's remain closer to the medium-data reference.

6 Conclusions

We introduced the Actor-Critic Decision Transformer (AC/DT), an offline-to-online RL method that combines long-horizon sequence modeling with value-based policy optimization. By replacing the likelihood-based objective of Decision Transformers with a soft actor-critic formulation, AC/DT enables explicit return maximization beyond behavior cloning. Source-aware conservative regularization is restricted to offline-origin samples during online fine-tuning. Experiments on D4RL show strong offline and online performance, while the λ ablation supports multi-step reward propagation and the state-matched analysis shows a shift of the policy-induced action marginals toward the expert reference relative to ODT.

Limitations and Future Work: Transformer critics were less stable than Markovian MLP critics in our experiments (Appendix A.1), while conservative regularization introduces additional hyperparameters and can slow convergence. Future work will investigate more stable Transformer-based critics and lighter conservative regularization.

AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on the interpretation and presentation of experimental results, draft parts of the research paper, and edit the manuscript to improve readability. We have reviewed all AI-assisted work. All AI-assisted text was reviewed and revised by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The experimental protocol, evaluation metrics, and benchmark settings are described in Section 5.1. Implementation details and hyperparameters are provided in Appendix E, with additional ablations and analyses supporting the main results reported in the remaining appendices. The code will be released upon acceptance.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), Advances in Neural Information Processing Systems, volume 34, pp. 29304–29320. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/f514cec81cb148559cf475e7426eed5e-Paper.pdf.

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 M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), Advances in Neural Information Processing Systems, volume 34, pp. 15084–15097. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/7f489f642a0ddb10272b5c31057f0663-Paper.pdf.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. arXiv preprint arXiv:2004.07219, 2020.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In International conference on machine learning, pp. 1861–1870. Pmlr, 2018.

Shengchao Hu, Ziqing Fan, Chaoqin Huang, Li Shen, Ya Zhang, Yanfeng Wang, and Dacheng Tao. Q-value regularized transformer for offline reinforcement learning. In Proceedings of the 41st International Conference on Machine Learning, ICML'24. JMLR.org, 2024.

Byeongchan Kim and Min-hwan Oh. Peng's q(λ) for conservative value estimation in offline reinforcement learning. In The Fourteenth International Conference on Learning Representations, 2026.

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

Tadashi Kozuno, Yunhao Tang, Mark Rowland, Rémi Munos, Steven Kapturowski, Will Dabney, Michal Valko, and David Abel. Revisiting peng's q (λ) for modern reinforcement learning. In International Conference on Machine Learning, pp. 5794–5804. PMLR, 2021.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), Advances in Neural Information Processing Systems, volume 33, pp. 1179–1191. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/0d2b2061826a5df3221116a5085a6052-Paper.pdf.

Seunghyun Lee, Younggyo Seo, Kimin Lee, Pieter Abbeel, and Jinwoo Shin. Offline-to-online reinforcement learning via balanced replay and pessimistic q-ensemble. In Conference on Robot Learning, pp. 1702–1712. PMLR, 2022.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. arXiv preprint arXiv:2005.01643, 2020.

Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. Awac: Accelerating online reinforcement learning with offline datasets. arXiv preprint arXiv:2006.09359, 2020.

Mitsuhiko Nakamoto, Simon Zhai, Anikait Singh, Max Sobol Mark, Yi Ma, Chelsea Finn, Aviral Kumar, and Sergey Levine. Cal-ql: Calibrated offline rl pre-training for efficient online fine-tuning. Advances in Neural Information Processing Systems, 36:62244–62269, 2023.

Jing Peng and Ronald J Williams. Incremental multi-step q-learning. In Machine Learning Proceedings 1994, pp. 226–232. Elsevier, 1994.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Dong Tian, Onur Celik, and Gerhard Neumann. Chunking the critic: A transformer-based soft actor-critic with n-step returns. In International Conference on Learning Representations, volume 2026, pp. 120056–120094, 2026.

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

Yuanfu Wang, Chao Yang, Ying Wen, Yu Liu, and Yu Qiao. Critic-guided decision transformer for offline reinforcement learning. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 38, pp. 15706–15714, 2024.

Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. arXiv preprint arXiv:1911.11361, 2019.

Taku Yamagata, Ahmed Khalil, and Raul Santos-Rodriguez. Q-learning decision transformer: Leveraging dynamic programming for conditional sequence modelling in offline rl. In International Conference on Machine Learning, pp. 38989–39007. PMLR, 2023.

Kai Yan, Alex Schwing, and Yu-Xiong Wang. Reinforcement learning gradients as vitamin for online finetuning decision transformers. Advances in Neural Information Processing Systems, 37:38590–38628, 2024.

Han Zheng, Xufang Luo, Pengfei Wei, Xuan Song, Dongsheng Li, and Jing Jiang. Adaptive policy learning for offline-to-online reinforcement learning. In Proceedings of the AAAI conference on artificial intelligence, volume 37, pp. 11372–11380, 2023.

Hongling Zheng, Li Shen, Yong Luo, Deheng Ye, Shuhan Xu, Bo Du, Jialie Shen, and Dacheng Tao. Value-guided decision transformer: A unified reinforcement learning framework for online and offline settings. Advances in Neural Information Processing Systems, 38:71655–71684, 2025.

Qinqing Zheng, Amy Zhang, and Aditya Grover. Online decision transformer. In international conference on machine learning, pp. 27042–27059. PMLR, 2022.

A Role of Sequence Modeling in the Actor-Critic Architecture

A.1 Markovian vs Transformer Critics

AC/DT deliberately combines a sequence-conditioned Transformer actor with lightweight Markovian critics. While the actor can exploit trajectory context when selecting actions, the critics estimate state-action values from the current state and action. To investigate whether sequence conditioning is also beneficial for value estimation, we compare the Markovian twin-critic architecture used in the main experiments against an AC/DT variant employing sequence-conditioned Transformer critics, following the critic architecture of Tian et al. (2026). In both cases, the policy remains the stochastic Decision Transformer used by AC/DT; the comparison therefore focuses on the behavior of the two critic formulations within the same sequence-conditioned actor-critic framework.

Table 2: Starting from the default AC/DT architecture, which uses a sequence-conditioned Transformer actor and Markovian MLP critics, we independently replace either the Transformer actor with a Markovian MLP actor (Appendix A.2) or the Markovian MLP critics with sequence-conditioned Transformer critics (Appendix A.1), while keeping the remaining training components unchanged.

	AC/DT		**AC/DT (Actor MLP)**		**AC/DT (Critic Transformer)**	
Actor	Transformer		MLP		Transformer	
Critic	MLP		MLP		Transformer	
Dataset	**Off.**	**On.**	**Off.**	**On.**	**Off.**	**On.**
hopper-medium	$86.4_{\pm 4.0}$	$\underline{\mathbf{103.7}}_{\pm 3.2}$	$\mathbf{90.3}_{\pm 1.2}$	$99.6_{\pm 4.4}$	$1.56_{\pm 0.85}$	$2.59_{\pm 2.29}$
walker-medium	$\mathbf{91.2}_{\pm 1.2}$	$\underline{\mathbf{128.2}}_{\pm 4.7}$	$83.0_{\pm 1.8}$	$96.1_{\pm 26.6}$	$1.0_{\pm 1.7}$	$-0.1_{\pm 0.4}$
antmaze-large-play	$\mathbf{42.6}_{\pm 11.0}$	$\underline{\mathbf{96.4}}_{\pm 1.7}$	$41.7_{\pm 19.9}$	$95.3_{\pm 5.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$
antmaze-large-diverse	$\mathbf{23.6}_{\pm 6.9}$	$\underline{\mathbf{96.2}}_{\pm 1.3}$	$11.3_{\pm 1.5}$	$63.7_{\pm 55.1}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$

Figure 3 first examines the critic-value dynamics across offline pretraining and online fine-tuning. For each variant, we report the mean minimum critic estimate, $\min(Q_1, Q_2)$. Curves denote the mean across five random seeds, while shaded regions show the seed-wise minimum-to-maximum range. During offline pretraining, on both hopper-medium and walker-medium, the Transformer critic exhibits a pronounced growth in value estimates over training, reaching values several orders of magnitude larger than those produced by the Markovian critic. This behavior is also strongly seed-dependent, as indicated by the widening min-to-max range. In contrast, the Markovian critic remains bounded throughout offline pretraining. The transition to online fine-tuning does not consistently restore stable value estimation. On hopper-medium, the Transformer critic remains at extremely large magnitudes throughout the online phase. On walker-medium, its mean value estimates decrease after the transition, but remain substantially larger and more seed-dependent than those of the Markovian critic for most of online fine-tuning, followed by a sharp late-stage drop. Notably, the reported quantity is already the minimum of the two critic estimates; therefore, the observed growth cannot be attributed solely to an isolated overestimation by a single member of the twin critic. These results indicate substantially less stable value estimation for the sequence-conditioned critic in the evaluated AC/DT setting.

The difference in value-estimation behavior is accompanied by a substantial difference in policy performance. Figure 4 shows the offline-to-online learning curves, while Table 2 reports the corresponding endpoint performance. The Transformer-critic variant does not develop sustained performance during offline pretraining and does not recover during online fine-tuning. Its final offline normalized returns are 1.56, 1.0, 0.0, and 0.0 on hopper-medium, walker-medium, antmaze-large-play, and antmaze-large-diverse, respectively, and remain 2.59, -0.1, 0.0, and 0.0 after online fine-tuning. In contrast, the Markovian-critic AC/DT reaches offline scores of 86.4, 91.2, 42.6, and 23.6, and subsequently improves to 103.7, 128.2, 96.4, and 96.2, respectively. Together with the critic-value trajectories in Figure 3, these results show that the Transformer-critic instability persists beyond offline pretraining and is not resolved by online interaction.

Taken together, these experiments support the use of lightweight Markovian critics in AC/DT: in the considered offline-to-online setting, adding sequence conditioning to the critic does not improve value learning and is instead associated with unstable value estimates and poor policy performance. We do not interpret this result as evidence that Transformer critics are inherently unsuitable for reinforcement learning. Rather, it shows that, under the AC/DT training regime evaluated here, the Markovian formulation provides a substantially more stable and effective value-learning component, while sequence modeling can remain concentrated in the policy.

A.2 Transformer vs. MLP Actors

To isolate the contribution of the sequence-conditioned Transformer actor from the remaining components of AC/DT, we replace the Transformer policy with a Markovian MLP actor while leaving the rest of the method unchanged. In particular, both variants use the same Markovian twin critics, position-wise context-truncated λ-return critic training with $K = 5$ and $\lambda = 0.9$, source-aware

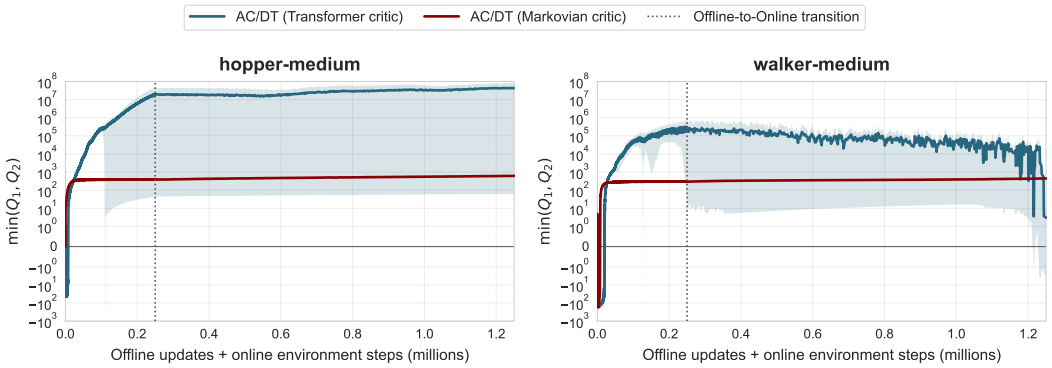


Figure 3: **Critic-value dynamics across offline pretraining and online fine-tuning.** We compare AC/DT with the Markovian twin critic used in the main experiments and a sequence-conditioned Transformer critic on hopper-medium (left) and walker-medium (right). We report the mean twin-critic minimum, $\min(Q_1, Q_2)$, at each training update. Curves denote the mean across five random seeds, while shaded regions show the seed-wise minimum-to-maximum range. The vertical dotted line marks the transition from offline pretraining to online fine-tuning. Both panels use the same symmetric-logarithmic scale. The Transformer critic develops large and strongly seed-dependent value estimates, whereas the Markovian critic remains bounded across both phases.

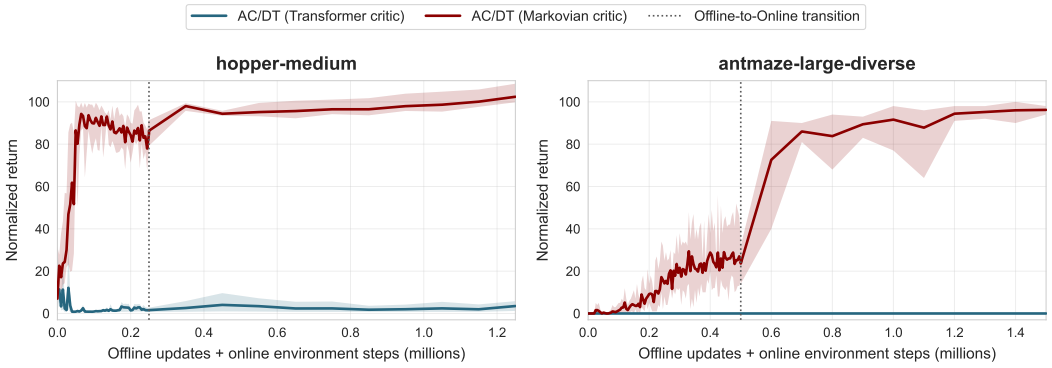


Figure 4: **Effect of critic architecture on offline-to-online performance.** Normalized return for AC/DT with sequence-conditioned Transformer critics and with the Markovian critics used in the main experiments on hopper-medium (left) and antmaze-large-diverse (right). Offline evaluations are followed by 10^6 online environment interactions; the vertical dotted line marks the transition between the two phases. Curves denote the mean across five random seeds, while shaded regions show the seed-wise minimum-to-maximum range. The Transformer-critic variant fails to develop sustained performance during offline pretraining and shows no meaningful recovery during online fine-tuning, whereas the Markovian-critic AC/DT learns a strong offline policy and improves further online.

conservative regularization, replay strategy, and offline-to-online training protocol. Therefore, the comparison directly evaluates the contribution of the sequence-conditioned Transformer actor within AC/DT, without changing the value-learning or regularization mechanisms of AC/DT. Table 2 reports the corresponding offline and final online performance, while Figure 5 shows the online fine-tuning dynamics.

Replacing the Transformer actor with an MLP results in a lower mean final online return across all four evaluated environments. On hopper-medium, the MLP actor starts from a slightly higher offline return (90.3 ± 1.2 versus 86.4 ± 4.0), yet the Transformer actor reaches the higher final online score (103.7 ± 3.2 versus 99.6 ± 4.4). This is particularly informative because, on this task, the online advantage of the Transformer cannot be explained by a stronger offline initialization. The difference

becomes substantially larger on walker-medium, where AC/DT with the Transformer actor improves from 91.2 ± 1.2 offline to 128.2 ± 4.7 after online fine-tuning, compared with 83.0 ± 1.8 and 96.1 ± 26.6, respectively, for the MLP variant. A similarly pronounced difference is observed on antmaze-large-diverse: the Transformer actor reaches 96.2 ± 1.3 after online fine-tuning, whereas the MLP actor reaches 63.7 ± 55.1. On antmaze-large-play, the two variants eventually attain similar final returns (96.4 ± 1.7 and 95.3 ± 5.0), but their online learning dynamics are markedly different.

Figure 5 shows that the advantage of the Transformer actor extends beyond the final evaluation point. On hopper-medium, both architectures exhibit relatively stable improvement, with the Transformer ultimately reaching the higher return. On walker-medium, the Transformer actor continues to improve throughout online interaction and reaches substantially higher performance, while the MLP actor exhibits a smaller improvement and considerably larger variability toward the end of fine-tuning. The benefit of the Transformer actor is especially evident on the sparse-reward AntMaze tasks. On antmaze-large-play, the MLP actor undergoes pronounced performance degradation during substantial portions of the online phase before recovering near the end of training, whereas the Transformer actor rapidly reaches high return and maintains it throughout most of fine-tuning. On antmaze-large-diverse, the Transformer actor similarly reaches and preserves high performance, while the MLP variant remains substantially less stable and finishes at a markedly lower return.

Overall, when the remaining components are held fixed, replacing the sequence-conditioned Transformer actor with an MLP weakens online adaptation: the Transformer variant attains a higher final online return in all four evaluated tasks and exhibits lower seed-wise variability. These results provide direct evidence that the sequence-conditioned Transformer actor contributes beyond the other elements of the actor-critic pipeline.

Together with the critic-architecture ablation in Table 2, these results also clarify where sequence modeling is most effective within AC/DT. While replacing the Transformer actor with an MLP degrades online fine-tuning, replacing the lightweight Markovian critics with Transformer critics leads to substantially poorer value learning and policy performance. The two ablations therefore support the architectural design adopted in AC/DT: the Transformer-based sequence model is beneficial when used as the policy, whereas lightweight Markovian critics provide substantially more effective value estimation in the evaluated setting.

B The Effect of Source-Aware Conservative Regularization

Effect of source-aware conservative regularization. We analyze the role of the proposed source-aware regularization strategy by comparing AC/DT against two alternative ways of handling conservative regularization across the offline-to-online transition. AC/DT (Pretrain-only) applies conservative regularization during offline pretraining and removes it completely when online fine-tuning begins. AC/DT (Always-regularized) instead retains conservative regularization throughout the entire online phase, independently of the origin of the replay samples. In contrast, AC/DT (Source-aware) applies the conservative penalty only to offline-origin samples, so that the penalty is progressively removed as offline experience is replaced by newly collected trajectories in the replay buffer. This behavior follows directly from the source-aware objective introduced in Section 4.5.

Figure 6 shows that the three variants exhibit behaviors consistent with the expected effect of conservative regularization across the offline-to-online transition. AC/DT (Pretrain-only) can experience a less stable transition to online learning: abruptly removing conservative regularization leads to an initial performance degradation in some environments and slower early improvement in others. As additional online experience is collected, however, the method recovers, in several cases reaching higher final returns.

A complementary behavior is observed for AC/DT (Always-regularized). Retaining conservative regularization throughout online fine-tuning generally preserves a more stable transition, but tends to limit subsequent policy improvement as the replay buffer becomes increasingly dominated by online experience. This is consistent with persistent conservatism becoming progressively less desirable once the agent moves away from the purely offline setting.

AC/DT (Source-aware) provides a gradual transition between these two regimes. It retains the stabilizing effect of offline regularization during the early stages of fine-tuning while progressively relaxing the penalty as online data replace offline experience. Consequently, the method substan-

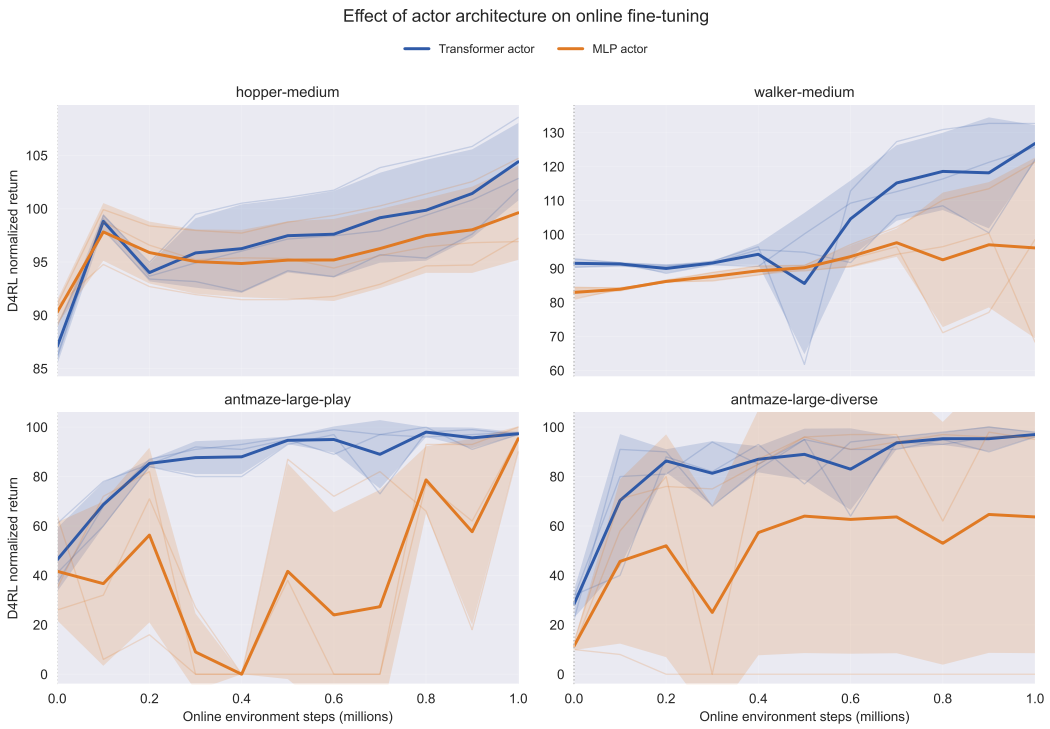


Figure 5: **Effect of actor architecture on online fine-tuning.** D4RL normalized return for AC/DT with the default sequence-conditioned Transformer actor and with the MLP-actor ablation, while keeping the remaining training components unchanged. The Transformer actor reaches a higher final online return across all four evaluated tasks and shows markedly stronger and more stable online adaptation on walker-medium and the AntMaze environments.

tially mitigates the pronounced early degradation observed with the Pretrain-only variant while still allowing substantial online improvement.

Source-aware regularization in CQL. To assess whether the source-aware treatment of conservative regularization is by itself sufficient to explain the online performance of AC/DT, we additionally apply the same source-dependent regularization rule to CQL. Standard CQL retains its conservative penalty during online fine-tuning, whereas in the source-aware variant the penalty is applied only to offline-origin samples and is omitted for newly collected online experience, matching the treatment used in AC/DT. The remaining CQL training procedure is left unchanged. Table 3 reports the corresponding offline and online performance.

The source-aware modification does not reproduce the performance of AC/DT. On the locomotion tasks, it provides only limited changes relative to standard CQL, while on the sparse-reward AntMaze tasks it leads to substantially weaker and more variable online performance. In contrast, AC/DT maintains strong online performance across all evaluated environments.

These results show that the source-aware regularization rule alone is not sufficient to account for the online gains of AC/DT. Rather, its effectiveness depends on the remaining components of the proposed actor-critic formulation.

C Role of λ in multi-step context propagation

To isolate the effect of the multi-step critic targets introduced in Section 4.4, we vary $\lambda \in \{0, 0.5, 0.9, 1\}$ while keeping the remaining AC/DT configuration unchanged. According to equation 8, $\lambda = 0$ reduces the update to a one-step soft target at every decision position in the sampled context. Intermediate values combine backups of different lengths, while $\lambda = 1$ selects the longest

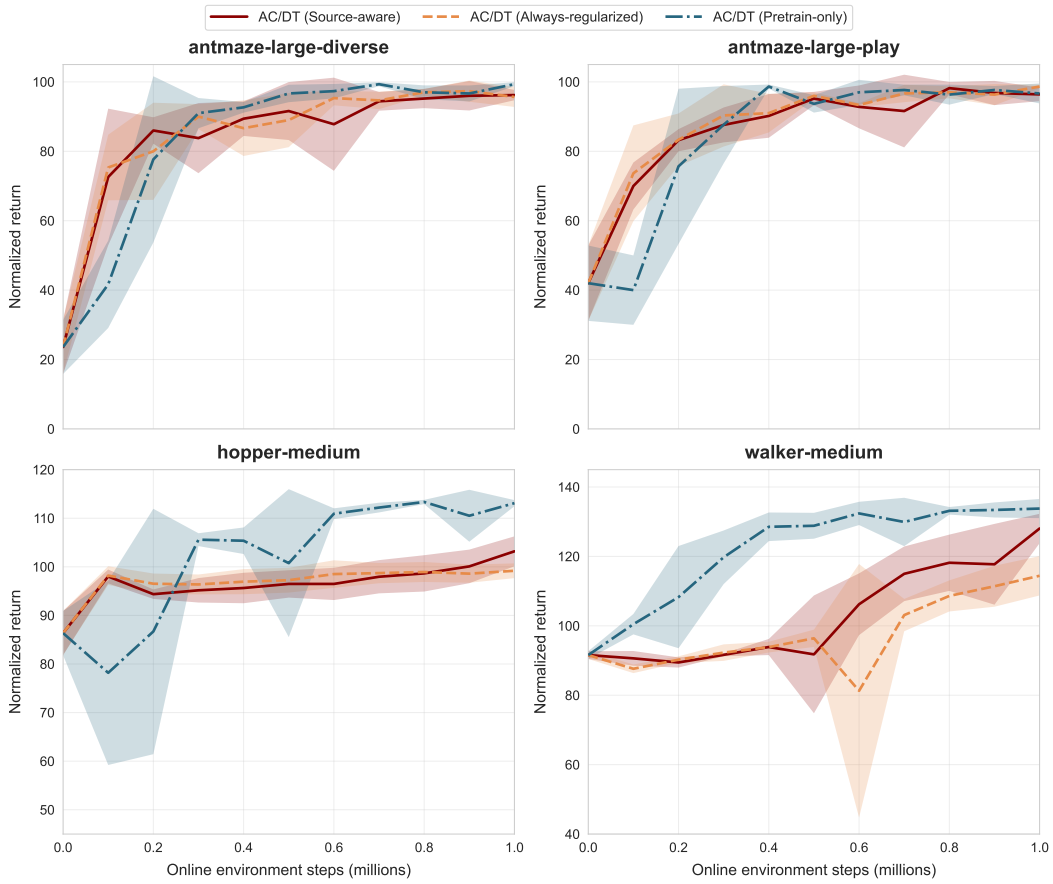


Figure 6: **Effect of conservative regularization across the offline-to-online transition.** Online normalized return for AC/DT (Source-aware), AC/DT (Always-regularized) and AC/DT (Pretrain-only). The variants respectively restrict the conservative penalty to offline-origin samples, retain it throughout the online phase or remove it at the start of online fine-tuning.

Table 3: Effect of source-aware conservative regularization on CQL. CQL (AC/DT-like) restricts the conservative penalty to offline-origin samples during online fine-tuning, matching AC/DT. We report D4RL normalized return (mean ± standard deviation) after offline pretraining (Off.) and after online fine-tuning (On.). Bold denotes the best offline score and bold-underlined the best online score.

Dataset	CQL		CQL (AC/DT-like)		AC/DT	
	Off.	On.	Off.	On.	Off.	On.
hopper-medium	$56.9_{\pm 1.7}$	$64.8_{\pm 5.0}$	$56.9_{\pm 1.7}$	$69.6_{\pm 8.9}$	$\mathbf{86.4}_{\pm 4.0}$	$\underline{\mathbf{103.7}}_{\pm 3.2}$
walker-medium	$82.1_{\pm 0.2}$	$84.9_{\pm 0.2}$	$82.1_{\pm 0.2}$	$86.6_{\pm 1.4}$	$\mathbf{91.2}_{\pm 1.2}$	$\underline{\mathbf{128.2}}_{\pm 4.7}$
antmaze-large-play	$26.3_{\pm 7.6}$	$91.3_{\pm 2.4}$	$26.3_{\pm 7.6}$	$27.0_{\pm 37.3}$	$\mathbf{42.6}_{\pm 11.0}$	$\underline{\mathbf{96.4}}_{\pm 1.7}$
antmaze-large-diverse	$\mathbf{28.5}_{\pm 5.9}$	$91.7_{\pm 4.9}$	$\mathbf{28.5}_{\pm 5.9}$	$45.7_{\pm 42.5}$	$23.6_{\pm 6.9}$	$\underline{\mathbf{96.2}}_{\pm 1.3}$

return available from each position, ranging from K steps at the earliest position to one step at the context boundary, maximally propagating the observed rewards over the available context while retaining soft value bootstrapping at the context boundary.

Table 4 reports offline and online performance for the different values of λ. Figure 7 complements these endpoint results by showing the corresponding online fine-tuning dynamics. In the sparse-reward AntMaze tasks, the $\lambda = 0$ variant undergoes a pronounced performance collapse during online fine-tuning, whereas larger values of λ maintain substantially stronger performance throughout the online phase.

The results show that the effect of λ is task dependent and that no single value dominates every environment. Locomotion tasks show smaller performance differences across λ values, with different non-zero values attaining the best performance depending on the environment.

The competitiveness of $\lambda = 1$ further highlights the benefit of exploiting the full reward suffix available within each Transformer context, while intermediate λ values retain the flexibility to combine shorter and longer backups. The same $\lambda = 0.9$ setting is used across all environments in the main experiments, providing a single robust configuration across both benchmark families.

Overall, these results indicate that propagating the rewards observed within the Transformer context is an important component of the critic update, particularly when reward information is sparse.

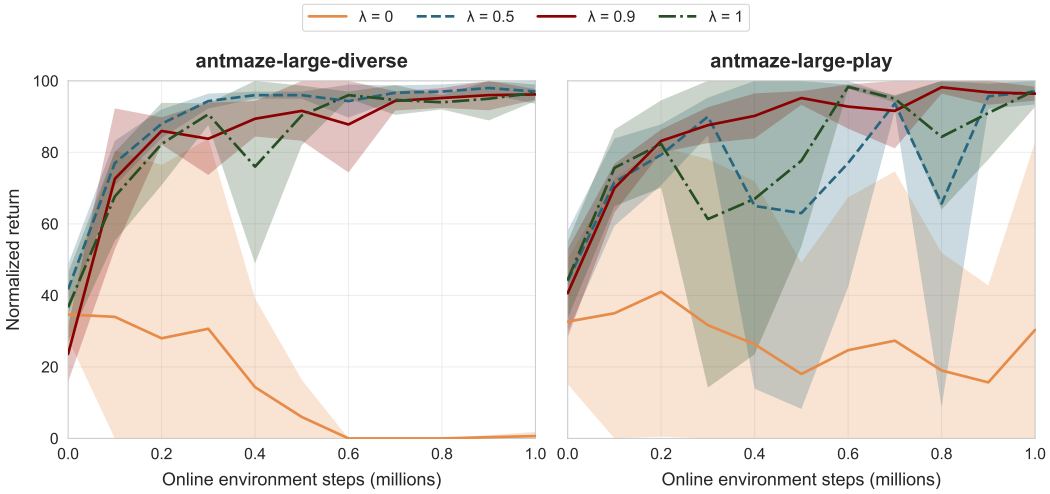


Figure 7: Effect of λ on online fine-tuning dynamics. Normalized return during online fine-tuning for AC/DT with $\lambda \in \{0, 0.5, 0.9, 1\}$. Curves report the mean across random seeds, with shaded regions denoting $\pm$ one standard deviation.

Table 4: Role of λ in multi-step context propagation. Offline and final online normalized returns for AC/DT with $\lambda \in \{0, 0.5, 0.9, 1\}$. $\lambda = 0$ uses position-wise one-step targets, whereas larger values progressively increase the contribution of longer context-truncated backups.

Dataset	$\lambda = 0$		$\lambda = 0.5$		$\lambda = 1$		$\lambda = 0.9$	
	Off.	**On.**	**Off.**	**On.**	**Off.**	**On.**	**Off.**	**On.**
hopper-medium	$77.2_{\pm 1.5}$	$90.1_{\pm 9.1}$	$74.5_{\pm 11.4}$	$94.0_{\pm 2.4}$	$\mathbf{92.5}_{\pm 5.1}$	$102.6_{\pm 1.7}$	$86.4_{\pm 4.0}$	$\underline{\mathbf{103.7}}_{\pm 3.2}$
hopper-medium-replay	$98.5_{\pm 0.6}$	$105.6_{\pm 0.5}$	$98.4_{\pm 1.2}$	$\underline{\mathbf{110.3}}_{\pm 0.6}$	$98.5_{\pm 1.1}$	$99.2_{\pm 23.3}$	$\mathbf{98.6}_{\pm 1.0}$	$103.1_{\pm 8.4}$
walker-medium	$56.9_{\pm 40.1}$	$94.5_{\pm 2.5}$	$85.8_{\pm 1.4}$	$76.2_{\pm 54.0}$	$84.8_{\pm 2.5}$	$95.3_{\pm 42.5}$	$\mathbf{91.2}_{\pm 1.2}$	$\underline{\mathbf{128.2}}_{\pm 4.7}$
walker-medium-replay	$92.2_{\pm 2.0}$	$112.5_{\pm 0.3}$	$\mathbf{93.6}_{\pm 0.7}$	$120.1_{\pm 3.1}$	$92.4_{\pm 2.4}$	$\underline{\mathbf{125.2}}_{\pm 6.1}$	$92.7_{\pm 1.9}$	$117.3_{\pm 4.1}$
halfcheetah-medium	$\mathbf{58.7}_{\pm 0.6}$	$96.8_{\pm 2.4}$	$56.8_{\pm 0.9}$	$\underline{\mathbf{100.4}}_{\pm 1.4}$	$56.4_{\pm 0.7}$	$98.6_{\pm 1.7}$	$52.9_{\pm 0.3}$	$99.5_{\pm 1.6}$
halfcheetah-medium-replay	$52.6_{\pm 0.1}$	$96.1_{\pm 1.3}$	$\mathbf{53.0}_{\pm 0.2}$	$\underline{\mathbf{97.4}}_{\pm 1.6}$	$52.5_{\pm 1.0}$	$96.5_{\pm 1.2}$	$52.3_{\pm 0.8}$	$97.2_{\pm 5.6}$
MuJoCo IQM	76.2	98.1	76.5	102.8	80.1	104.7	**81.1**	**105.5**
antmaze-medium-play	$28.7_{\pm 67.9}$	$71.3_{\pm 45.6}$	$31.7_{\pm 3.9}$	$97.5_{\pm 1.6}$	$41.0_{\pm 27.8}$	$98.0_{\pm 1.6}$	$\mathbf{73.0}_{\pm 10.2}$	$\underline{\mathbf{98.8}}_{\pm 1.0}$
antmaze-medium-diverse	$8.5_{\pm 8.5}$	$0.5_{\pm 0.5}$	$70.3_{\pm 5.9}$	$57.3_{\pm 41.1}$	$\mathbf{75.0}_{\pm 5.0}$	$87.3_{\pm 14.4}$	$63.8_{\pm 22.1}$	$\underline{\mathbf{98.2}}_{\pm 1.5}$
antmaze-large-play	$32.7_{\pm 14.3}$	$30.3_{\pm 42.9}$	$44.0_{\pm 11.5}$	$97.0_{\pm 2.9}$	$\mathbf{44.3}_{\pm 8.7}$	$\underline{\mathbf{97.3}}_{\pm 3.8}$	$42.6_{\pm 11.0}$	$96.4_{\pm 1.7}$
antmaze-large-diverse	$34.7_{\pm 5.6}$	$0.7_{\pm 0.9}$	$\mathbf{41.7}_{\pm 5.7}$	$\underline{\mathbf{97.0}}_{\pm 2.8}$	$36.7_{\pm 7.8}$	$96.7_{\pm 1.9}$	$23.6_{\pm 6.9}$	$96.2_{\pm 1.3}$
AntMaze IQM	28.5	0.3	47.7	96.2	47.0	94.7	**48.3**	**97.4**
Global IQM	60.4	85.9	66.9	99.1	67.8	98.2	**69.7**	**100.1**

D State Matched Analysis

State-matched action marginal analysis. We measure how far each policy's one-dimensional action marginals deviate from the corresponding dataset action marginals when evaluated on the same

states. After offline pretraining on *halfcheetah-medium* and online fine-tuning for 10^6 environment steps, we query ODT and AC/DT on states from (i) the medium dataset (in-distribution) and (ii) the unseen *halfcheetah-expert-v2* dataset (high-return), recording the deterministic action of each policy for all $N = 10^6$ states (d=6 action dimensions). Dataset statistics are reported in Table 5. This yields three aligned action matrices: $A^{\text{med}}_{\text{data}}, A^{\text{med}}_{\text{ODT}}, A^{\text{med}}_{\text{AC/DT}} \in \mathbb{R}^{N\times d}$ for the medium dataset, and analogously $A^{\text{exp}}_{\text{data}}, A^{\text{exp}}_{\text{ODT}}, A^{\text{exp}}_{\text{AC/DT}} \in \mathbb{R}^{N\times d}$ for the expert dataset.

KL divergence on action marginals. From the action matrices A, we estimate the marginal density of each action dimension. We then compare the policy-induced marginal P^{π}_j with the corresponding reference marginal P^{ref}_j obtained from the dataset actions, using the Kullback-Leibler (KL) divergence. We report the average across the $d = 6$ action dimensions:

$$\text{KL}_{\text{avg}}(\pi\|\text{ref}) = \frac{1}{d}\sum_{j=1}^{d} \text{KL}(P^{\pi}_j \| P^{\text{ref}}_j),$$

where P^{π}_j and P^{ref}_j denote the policy-induced and reference marginal densities of the j-th action dimension, respectively. Numerically, each density is estimated using Gaussian KDE with Scott's bandwidth rule and evaluated on a shared 1,000-point grid spanning the pooled range of the compared samples. After adding $\epsilon = 10^{-10}$, the densities are normalized and the KL divergence is computed by trapezoidal integration.

State-Matched Action Marginal Shift . Figure 8 shows kernel density estimates (KDEs) of per-joint action marginals. When queried on unseen expert-dataset states (top row of Fig 8), AC/DT produces action densities closely overlapping the expert reference marginals, while ODT shows substantially lower overlap with the expert reference. This is confirmed quantitatively in Table 6: AC/DT achieves consistently lower KL divergence to the expert reference than ODT (e.g., Joint 0: 0.1326 vs. 0.3768; Joint 3: 0.0334 vs. 0.1201), with an average KL of 0.0920 vs. 0.2700 across all 6 joints. On medium-dataset states (bottom row of Fig 8), ODT remains close to the behavioral action marginals (avg. KL 0.0018), while AC/DT exhibits larger deviations from the medium-data action marginals (avg. KL 0.0242), increasing probability mass near +1.0 on Joint 0, mirroring the expert-side KDE. In HalfCheetah, action dimensions 0 and 3 actuate the back and front thigh joints (often referred to as hip/thigh actuators), which are directly responsible for applying torque to the main leg segments that drive forward running. On these joints, the KDEs highlight a clear difference in how policies depart from the medium behavior at the level of their action marginals: for Joint 0, ODT concentrates mass near a single extreme action (close to -1), consistent with its closer agreement with the medium-data marginal , while AC/DT displays a broader (often bimodal) profile with mass near both extremes, consistent with its stronger departure from the medium-data action marginal after online fine-tuning.

Table 5: Statistics of the HalfCheetah datasets from D4RL

Dataset	**Max Return**	**Mean Return**
halfcheetah-medium	5309	4770
halfcheetah-expert	11252	10656

E Implementation Details

E.1 Hyperparameters

Unless otherwise specified, we use the same architecture and optimization settings across all D4RL tasks. The stochastic Decision Transformer actor uses a context length of $K = 5$, an embedding dimension of 256, 4 Transformer layers, 4 attention heads, ReLU activations, and a dropout probability of 0.1. The policy is parameterized as a Gaussian distribution. We use a batch size of 256 for all experiments.

The actor is optimized using LAMB, with a linear learning-rate warm-up over the first 10,000 gradient updates. The twin Markovian critics are independent three-layer MLPs with 256 hidden units per layer and orthogonal initialization, and are optimized using Adam. Target critics are updated

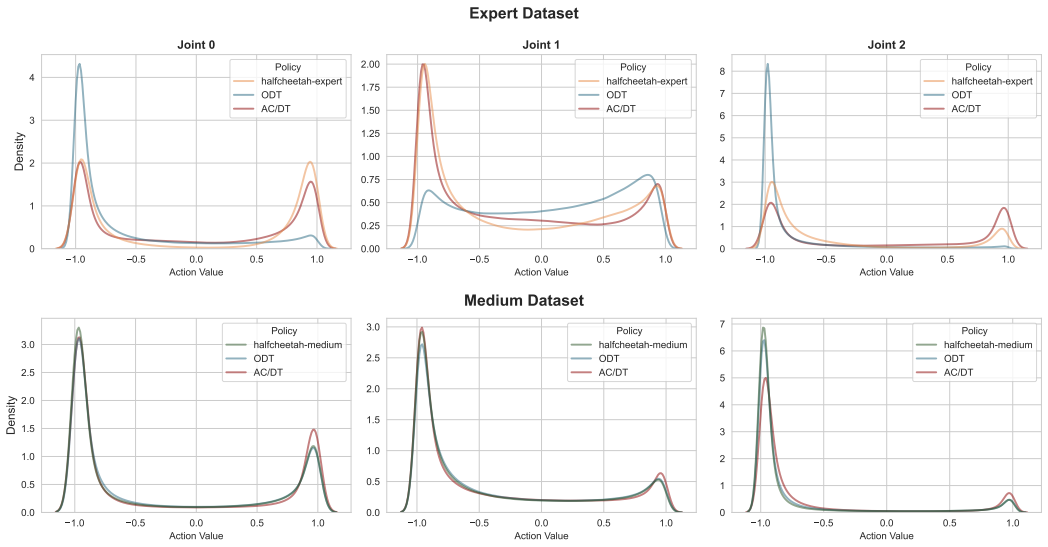


Figure 8: **Action density under state-matched evaluation.** KDEs of per-joint action marginals after 10^6 online steps, obtained by querying ODT and AC/DT on states from the expert dataset (top) and the medium dataset (bottom). AC/DT shows higher overlap with expert action marginals, while ODT remains closer to medium-dataset actions.

Table 6: Per-action-dimension KL divergence between the action marginals induced by AC/DT and ODT, both trained on the HalfCheetah-medium dataset, and the reference action marginals on states drawn from the Medium and Expert datasets. Lower values indicate greater similarity to the corresponding reference action distribution.

	Expert Dataset		Medium Dataset	
Joint	AC/DT	ODT	AC/DT	ODT
0	0.1326	0.3768	0.0078	0.0026
1	0.0202	0.1691	0.0128	0.0011
2	0.2201	0.3698	0.0644	0.0052
3	0.0334	0.1201	0.0174	0.0001
4	0.1293	0.3987	0.0157	0.0011
5	0.0164	0.1852	0.0271	0.0008
Average	0.0920	0.2700	0.0242	0.0018

after every gradient step using Polyak averaging with $\tau = 0.005$. Gradient norms are clipped to 0.25 for both actor and critic updates. During online fine-tuning, the agent collects one complete trajectory with the current policy, appends it to the shared FIFO replay buffer, and subsequently performs 300 gradient updates from replay. Online fine-tuning is terminated after 10^6 environment interactions. The replay-buffer capacity is 10^6 transitions for the MuJoCo tasks and 2×10^6 transitions for the AntMaze-medium and AntMaze-large tasks. Evaluation uses 10 episodes for MuJoCo and 100 episodes for AntMaze.

The entropy temperature α is automatically optimized following the SAC objective, with target entropy $-|\mathcal{A}|$. We use $\lambda = 0.9$ for the context-truncated λ-return objective in the main experiments.

States are normalized using the mean and standard deviation of the corresponding offline dataset. The *ordering* hyperparameter controls the use of the learned ordering embedding: ordering $= 0$ disables it, while ordering $= 1$ enables it. When enabled, the timestep-indexed embedding is added to the return-to-go, state, and action token embeddings.

Task-specific hyperparameters are reported in Table 7.

Table 7: Task-specific AC/DT hyperparameters

Dataset	Actor LR	Critic LR	γ	β_C	Ordering	Eval RTG	Online RTG
hopper-medium	3e-5	3e-4	0.990	2.5	0	3600	3600
hopper-medium-replay	3e-5	3e-4	0.995	2.5	0	3600	3600
walker-medium	3e-5	3e-4	0.990	2.5	0	5000	10000
walker-medium-replay	3e-5	1e-4	0.990	2.5	0	5000	10000
halfcheetah-medium	1e-4	3e-4	0.990	1.0	0	6000	12000
halfcheetah-medium-replay	1e-4	3e-4	0.990	1.0	0	6000	12000
antmaze-medium-play	3e-4	3e-4	0.998	2.5	1	-200	-200
antmaze-medium-diverse	3e-4	3e-4	0.998	2.5	1	-200	-200
antmaze-large-play	3e-4	3e-4	0.998	2.5	1	-400	-400
antmaze-large-diverse	3e-4	3e-4	0.998	2.5	1	-400	-400

E.2 Resources

All experiments were run on an NVIDIA RTX A6000 GPU with 48 GiB of memory.

F Additional Comparison with Published Results

Table 8 complements the main evaluation with published results from additional recent O2O methods that are particularly relevant to AC/DT, including Cal-QL, VDT, and CPQL/PQL. As discussed in Section 5.1, for Cal-QL and VDT, we attempted to reproduce their results under the common evaluation protocol adopted for Table 1, but the publicly available implementations and methodological details did not allow us to obtain a reliable evaluation under this protocol. For CPQL/PQL, the public repository available at the time of our experiments did not include the online PQL fine-tuning stage required for the O2O evaluation.

We therefore report published results collected from prior work for completeness. Since these results were obtained under different experimental protocols, numerical differences should not be interpreted as direct comparisons under a common evaluation protocol.

Despite the differences in evaluation protocols, the published results provide useful additional context for the performance of AC/DT relative to recent O2O approaches. AC/DT remains competitive across both locomotion and sparse-reward AntMaze tasks.

Table 8: Published offline and online normalized D4RL scores for additional relevant baselines. Results for external methods are reproduced from Nakamoto et al. (2023) for Cal-QL, IQL, AWAC, and CQL, from Yan et al. (2024) for ODT and ODT+TD3, and from the respective papers for VDT and CPQL/PQL. These results may rely on different training and evaluation protocols. AC/DT results are reported under the protocol described in Section 5.1. The table is provided for broader contextual comparison rather than as a controlled evaluation

Dataset	**AWAC**		**IQL**		**CQL**		**Cal-QL**		**ODT**		**ODT+TD3**		**VDT**		**CPQL/PQL**		**AC/DT**	
	Off.	**On.**	**Off.**	**On.**	**Off.**	**On.**	**Off.**	**On.**	**Off.**	**On.**	**Off.**	**On.**	**Off.**	**On.**	**Off.**	**On.**	**Off.**	**On.**
hopper-medium	58.0	90.0	57.0	26.0	78.0	99.0	89.0	98.0	49.1	97.8	63.1	89.1	98.3	108.1	99.7	111.5	86.4	103.7
hopper-medium-replay	70.0	108.0	76.0	86.0	69.0	111.0	76.0	110.0	20.1	83.3	29.8	95.7	96.0	119.2	103.0	112.1	98.6	103.1
walker-medium	75.0	98.0	80.0	111.0	81.0	104.0	75.0	103.0	57.1	75.6	70.8	97.9	81.6	89.8	90.0	127.8	91.2	128.2
walker-medium-replay	60.0	94.0	63.0	95.0	76.0	92.0	52.0	99.0	64.7	77.2	58.1	100.6	82.3	95.5	97.4	128.6	92.7	117.3
halfcheetah-medium	49.0	57.0	47.0	43.0	53.0	97.0	52.0	93.0	23.0	42.3	41.6	76.9	43.9	53.5	66.6	96.5	52.9	99.5
halfcheetah-medium-replay	43.0	47.0	43.0	36.0	51.0	95.0	52.0	93.0	14.7	41.5	29.3	73.3	39.4	49.2	60.3	95.8	52.3	97.2
antmaze-medium-play	0.0	0.0	72.0	94.0	62.0	98.0	54.0	97.0	0.0	0.0	54.0	96.8	–	–	72.4	93.4	73.0	98.8
antmaze-medium-diverse	0.0	0.0	70.0	92.0	65.0	98.0	75.0	97.0	0.0	0.0	60.0	96.3	20.0	75.0	71.7	98.2	63.8	98.2
antmaze-large-play	0.0	0.0	41.0	51.0	34.0	76.0	26.0	90.0	0.0	0.0	0.0	0.0	–	–	41.6	85.4	42.6	96.4
antmaze-large-diverse	0.0	0.0	40.0	59.0	25.0	87.0	33.0	95.0	0.0	0.0	0.0	0.0	–	–	46.6	82.0	23.6	96.2

G SENSITIVITY TO CONTEXT LENGTH

The context length K plays two coupled roles in AC/DT. It determines the amount of trajectory history available to the sequence-conditioned actor and, through the context-truncated critic objective, the maximum multi-step backup horizon available within each sampled context. We therefore evaluate the sensitivity of the complete AC/DT training mechanism to $K \in \{1, 5, 20\}$, with $K = 5$ corresponding to the default configuration used in the main experiments.

Figure 9 reports the online fine-tuning dynamics on two sparse-reward AntMaze tasks and two locomotion tasks. The effect of K is particularly pronounced on AntMaze. With $K = 1$, performance initially improves but subsequently deteriorates, eventually collapsing on both antmaze-large-play and antmaze-large-diverse. In contrast, the default $K = 5$ configuration maintains consistently high performance throughout online fine-tuning. Increasing the context to $K = 20$ enables substantial improvement over $K = 1$, but does not provide a consistent advantage over $K = 5$ and exhibits considerably larger variability across seeds.

A similar pattern emerges on the locomotion tasks, although sensitivity to K is weaker. On hopper-medium, all three context lengths attain strong performance over most of fine-tuning, indicating that the task is comparatively robust to the choice of K. On walker-medium, however, $K = 1$ largely plateaus around its offline performance, whereas both longer contexts enable substantial online improvement. The $K = 20$ configuration reaches strong intermediate performance but becomes highly variable near the end of fine-tuning, while $K = 5$ improves more consistently. Taken together, these results show that increasing context length is beneficial relative to the degenerate $K = 1$ setting on tasks that benefit from longer contexts, but that performance does not improve monotonically with longer contexts.

Longer contexts also incur a clear computational cost. Figure 10 reports the mean wall-clock duration of a complete online iteration, including trajectory collection and the associated training updates. The iteration time increases from approximately 15–16 seconds for $K = 1$ to about 20 seconds for $K = 5$, and to approximately 40–41 seconds for $K = 20$, with nearly identical trends across environments. Thus, increasing K from 5 to 20 roughly doubles the end-to-end online iteration time without yielding a corresponding systematic improvement in return.

Overall, the ablation supports $K = 5$ as a robust operating point for AC/DT rather than as a universally optimal context length.

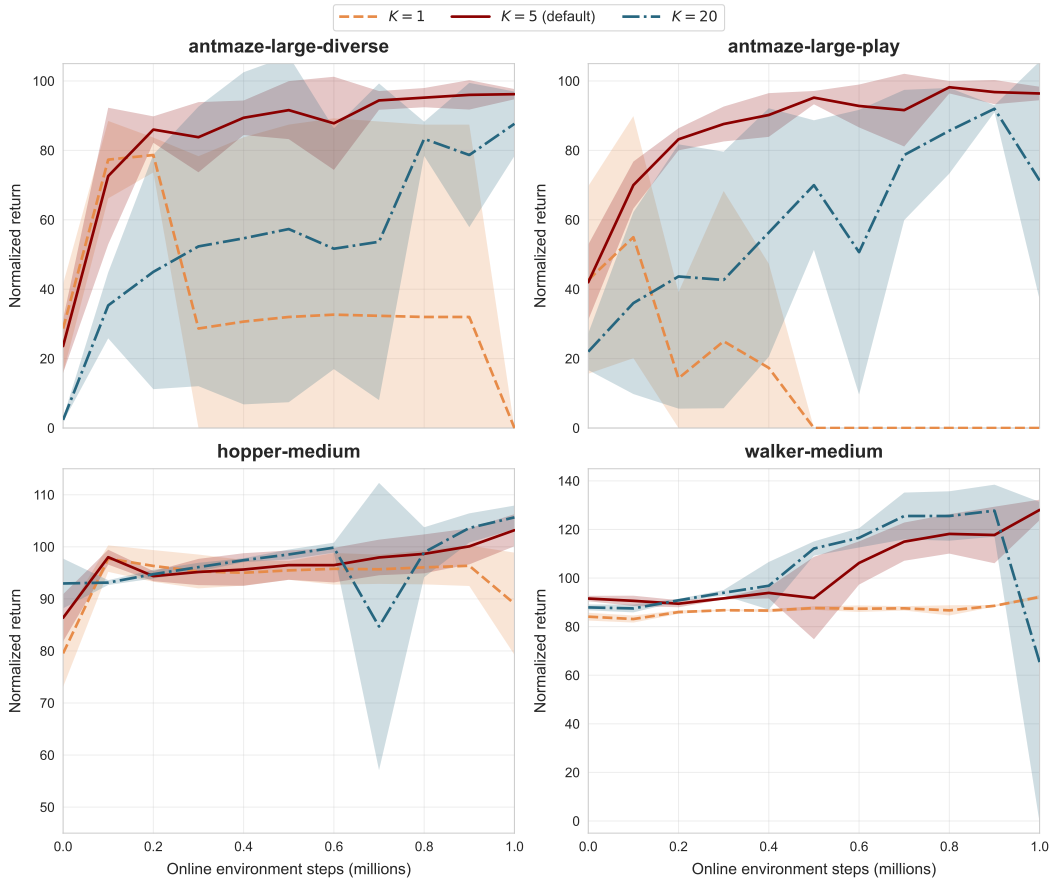


Figure 9: Sensitivity to context length during online fine-tuning. Normalized return for AC/DT with $K \in \{1, 5, 20\}$ on antmaze-large-diverse, antmaze-large-play, hopper-medium, and walker-medium. $K = 5$ denotes the default configuration used in the main experiments. The evaluation at zero online environment steps corresponds to the policy obtained after offline pretraining. Curves report the mean across five random seeds, while shaded regions indicate variability across seeds. The default $K = 5$ configuration provides consistently strong performance, whereas $K = 1$ is particularly brittle on AntMaze and $K = 20$ exhibits increased variability despite its longer context.

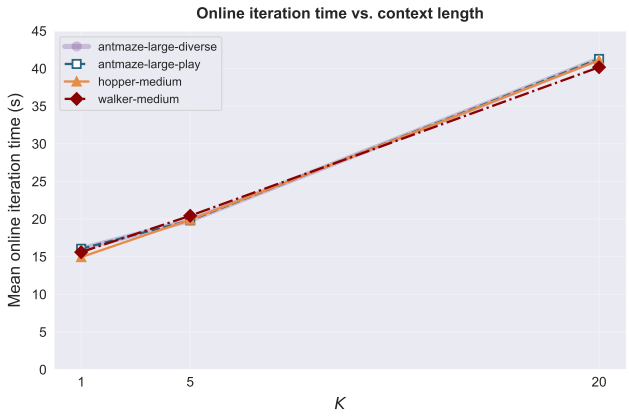


Figure 10: Computational cost as a function of context length. Mean wall-clock duration of a complete online iteration for $K \in \{1, 5, 20\}$ across the four evaluated environments. Increasing the context from $K = 5$ to $K = 20$ approximately doubles the online iteration time, while the timing trend is consistent across tasks.