RESTITCH: OFFLINE TRAJECTORY STITCHING VIA RETRIEVAL-BASED POLICY IMPROVEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Return-Conditioned Supervised Learning (RCSL), notably Decision Transformers (DT), has emerged as a stable paradigm for offline reinforcement learning. However, these methods can struggle to stitch suboptimal trajectories, limiting their performance in sparse-reward environments. While recent methods incorporate value-based guidance into DTs, they typically represent policy improvement through conditioning signals or optimization objectives rather than explicit trajectory-level continuations. In this work, we propose ReStitch, which formulates retrieval as an instance-based mechanism for offline policy improvement. ReStitch retrieves trajectory segments that are relevant, feasible, and advantageous, and injects them as explicit state-action guidance through cross-attention. Experiments on D4RL show substantial gains on sparse-reward AntMaze tasks, while improvements are more limited on dense-reward MuJoCo control. Mechanism analyses show that these gains require an informative value representation and the joint use of relevance, feasibility, and advantage in retrieval. These results highlight both the effectiveness and the empirical capability boundary of retrieval-based trajectory stitching.

1 INTRODUCTION

Offline reinforcement learning (RL) promises to extract effective policies from static datasets without environment interaction. Recently, Return-Conditioned Supervised Learning (RCSL)(Chen et al., 2021b; Emmons et al., 2022; Kim et al., 2024a; Ding et al., 2019; Reed et al., 2022), particularly Decision Transformer (DT) (Chen et al., 2021b), has demonstrated that offline RL can be cast as a sequence modeling problem. Recent work has further explored trajectory-level generative modeling for offline decision-making (Lin et al., 2026; Duan et al., 2026). Despite their stability and simplicity, RCSL methods remain limited by behavior cloning and can struggle with *trajectory stitching*, i.e., composing suboptimal trajectory segments into improved behavior (Liu et al., 2025; Brandfonbrener et al., 2022; Wu et al., 2023).

Existing approaches address trajectory stitching through several complementary paradigms. Value-based offline RL methods, such as CQL, TD3+BC, and IQL (Kumar et al., 2020; Fujimoto & Gu, 2021; Kostrikov et al., 2022), achieve stitching implicitly through temporal-difference learning and Bellman-style value propagation. Value-guided sequence models instead incorporate dynamic-programming information into RCSL through return relabeling, advantage conditioning, Q-regularization, or policy guidance (Kim et al., 2024b; Hu et al., 2024; Wang et al., 2024; Yamagata et al., 2023; Gao et al., 2024). Hierarchical approaches provide another route by decomposing long-horizon decisions through intermediate goals or high-value states (Park et al., 2023). Although these methods differ substantially in implementation, policy-improvement information is typically represented through learned value functions, conditioning variables, optimization objectives, or intermediate goals. They do not explicitly provide the sequence model with retrieved trajectory-level continuations that demonstrate how an improvement can be realized from offline experience.

This motivates a complementary perspective: *Can policy improvement be instantiated non-parametrically by retrieving improving continuations directly from the offline dataset?*

To this end, we propose **ReStitch**, a retrieval-based approach to offline trajectory stitching. Given the current state, ReStitch retrieves trajectory continuations according to three criteria: they should be *relevant* to the current state, *feasible* to follow, and *advantageous* relative to the current behavior.

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.

The retrieved continuations are then provided to the sequence model as explicit trajectory-level guidance for action selection. From this perspective, retrieval serves as an instance-based source of policy-improvement information: rather than only indicating that a state or action is desirable, it provides concrete offline experience illustrating how improved behavior may be realized.

Realizing this idea requires addressing two challenges. First, similarity alone is insufficient for trajectory stitching: a nearby trajectory may be irrelevant due to environment topology, while a high-value trajectory may be difficult to reach from the current state. ReStitch therefore combines similarity in a learned value representation with a feasibility constraint and an advantage-weighted term motivated by the policy-extraction mechanism of IQL (Kostrikov et al., 2022). Second, retrieved continuations can be beneficial in long-horizon navigation but potentially interfere with local action consistency in continuous control. We therefore use a dual-stream cross-attention architecture to preserve the original behavioral-cloning pathway while incorporating retrieved context, together with distance-aware selective stitching that reduces the influence of mismatched retrievals.

We evaluate ReStitch on D4RL navigation and continuous-control benchmarks (Fu et al., 2020). On sparse-reward AntMaze tasks, ReStitch substantially improves over standard sequence models and strong value-guided and hierarchical baselines, with the largest gains appearing in challenging medium and large mazes. Controlled ablations show that these gains depend on jointly retrieving relevant, feasible, and advantageous continuations. Additional analyses of the learned retrieval representation, IQL checkpoint quality, and cross-attention allocation provide evidence for the mechanism underlying these improvements. In contrast, the gains are more limited on dense-reward MuJoCo locomotion, where local kinematic consistency plays a larger role. We use this contrast to characterize an *empirical capability boundary* of retrieval-based trajectory stitching rather than claiming general superiority across continuous-control domains.

Our contributions are threefold:

- **Retrieval-Based Policy Improvement.** We formulate offline trajectory stitching from an instance-based perspective, in which policy-improvement guidance is provided by retrieving feasible and advantageous trajectory continuations from offline experience.
- **ReStitch.** We instantiate this perspective with relevance-, feasibility-, and advantage-aware trajectory retrieval, combined with trajectory-level cross-attention and distance-aware selective stitching to incorporate retrieved continuations while preserving the base sequence-modeling pathway.
- **Empirical Characterization.** Across D4RL benchmarks, we demonstrate substantial improvements on sparse-reward AntMaze tasks and systematically analyze when and why retrieval-based stitching helps, including its dependence on value-representation quality and its empirical limitations in dense-reward locomotion.

2 PRELIMINARIES

We consider a Markov Decision Process (MDP) (Bellman, 1957) defined by $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$. In the offline RL setting, the agent learns from a static dataset $\mathcal{D}$ collected by a behavior policy π_β.

Decision Transformer. Return-Conditioned Supervised Learning (RCSL) reframes offline policy optimization as a conditional sequence modeling problem. The Decision Transformer (DT) represents a trajectory as a temporal sequence of returns-to-go (RTG), states, and actions: $\tau = (\hat{R}_1, s_1, a_1, \hat{R}_2, s_2, a_2, \ldots, \hat{R}_T, s_T, a_T)$, where $\hat{R}_t = \sum_{t'=t}^{T} r_{t'}$. DT employs a causal Transformer to predict actions by minimizing the mean squared error (MSE): $\mathcal{L}_{DT}(\theta) = \mathbb{E}_{\tau\sim\mathcal{D}}\left[\sum_{t=1}^{T} \left\| a_t - \pi_\theta\left(a_t \mid \hat{R}_t, s_t, \ldots, \hat{R}_1, s_1\right)\right\|^2\right]$.

Implicit Q-Learning. While dynamic programming methods enable trajectory stitching, they often suffer from out-of-distribution (OOD) overestimation bias when querying unseen actions. Implicit Q-Learning (IQL) circumvents this by treating value estimation as an asymmetric expectile regression problem (Jiang et al., 2017). By approximating the maximum Q-value only over actions present in the dataset, IQL learns a value function $V(s)$ without ever querying OOD actions. Upon convergence, the optimal policy is extracted via advantage-weighted regression (AWR):

$$\pi^*(a \mid s) \propto \pi_\beta(a \mid s) \exp(\beta \cdot A(s, a)) \tag{1}$$

where $A(s, a) = Q(s, a) - V(s)$ is the advantage function and β is the inverse temperature parameter. In this work, we use this advantage-weighting principle to motivate the trajectory-level advantage component of our retrieval utility.

3 Related Work

3.1 Trajectory Stitching in Offline Sequence Modeling

Return-conditioned sequence modeling methods can struggle to compose suboptimal trajectory segments into improved behavior when high-return trajectories are insufficiently represented in the offline dataset (Brandfonbrener et al., 2022; Wang et al., 2025b). A growing body of work incorporates value-based policy-improvement signals into sequence models to address this limitation. ACT (Gao et al., 2024) and QDT (Yamagata et al., 2023) relabel or condition trajectories using value-based information, while CGDT (Wang et al., 2024), QCS (Kim et al., 2024b), and QT (Hu et al., 2024) use learned critics to guide training or action selection. POR (Xu et al., 2022) introduces policy-guided optimization, while hierarchical approaches such as HIQL decompose long-horizon decision making through intermediate subgoals. Other extensions improve sequence modeling through adaptive context selection (Wu et al., 2023), data fusion (Wang et al., 2025a), return conditioning (Zhuang et al., 2024), and RTG decomposition (Wang et al., 2026). ReStitch instead provides policy-improvement guidance by retrieving explicit trajectory continuations from offline experience, using relevance, feasibility, and advantage to select continuations for trajectory stitching.

3.2 Retrieval-Augmented Reinforcement Learning

Retrieval and memory mechanisms have been explored in reinforcement learning as a means of reusing past experience and augmenting parametric policies with non-parametric context (Azarafrooz, 2022; Graves et al., 2014; Wu et al., 2026). Prior work has studied learned retrieval for value-based RL (Goyal et al., 2022), large-scale memory and cross-attention for decision making (Humphreys et al., 2022), and retrieved context for in-context policy inference (Lee et al., 2023). Most closely related to our work, Retrieval-Augmented Decision Transformer (RA-DT) (Schmied et al., 2025) equips Decision Transformers with an external trajectory memory and retrieves relevant experience to support in-context adaptation and generalization. ReStitch assigns retrieval a different role: an instance-based mechanism for offline policy improvement and trajectory stitching. Rather than selecting experience by relevance alone, it combines similarity in a learned value representation with feasibility and advantage, retrieving continuations intended to demonstrate how improved behavior can proceed from the current state. A direct numerical comparison with RA-DT is complicated by a mismatch in evaluation settings. RA-DT is designed for multi-task in-context adaptation, where trajectories are associated with task identities and retrieval operates over experience collected across tasks. In contrast, our D4RL evaluation trains a separate policy for each fixed task and provides no corresponding multi-task task-ID structure. We therefore do not treat RA-DT as a directly comparable baseline in this setting; additional discussion and fixed-task experiments are provided in Appendix H

4 Methodology: ReStitch

We present ReStitch, a retrieval-based framework designed to enhance the trajectory stitching capabilities of Decision Transformers. ReStitch integrates three key components: (i) a value-guided retrieval mechanism for identifying high-utility continuations, (ii) a multi-layer cross-attention module for incorporating retrieved trajectory context, and (iii) a distance-gated loss that reduces the influence of mismatched retrieved targets. Further analysis of the retrieval utility and distance gating is provided in Appendices A and C.

4.1 Value-Guided Feasible Retrieval

Following QCS (Kim et al., 2024b), we pre-train IQL to obtain fixed Q- and V-functions for trajectory filtering. We pre-compute value features for all states and construct a FAISS-gpu index (Johnson et al., 2019) for efficient training-time and inference-time retrieval.

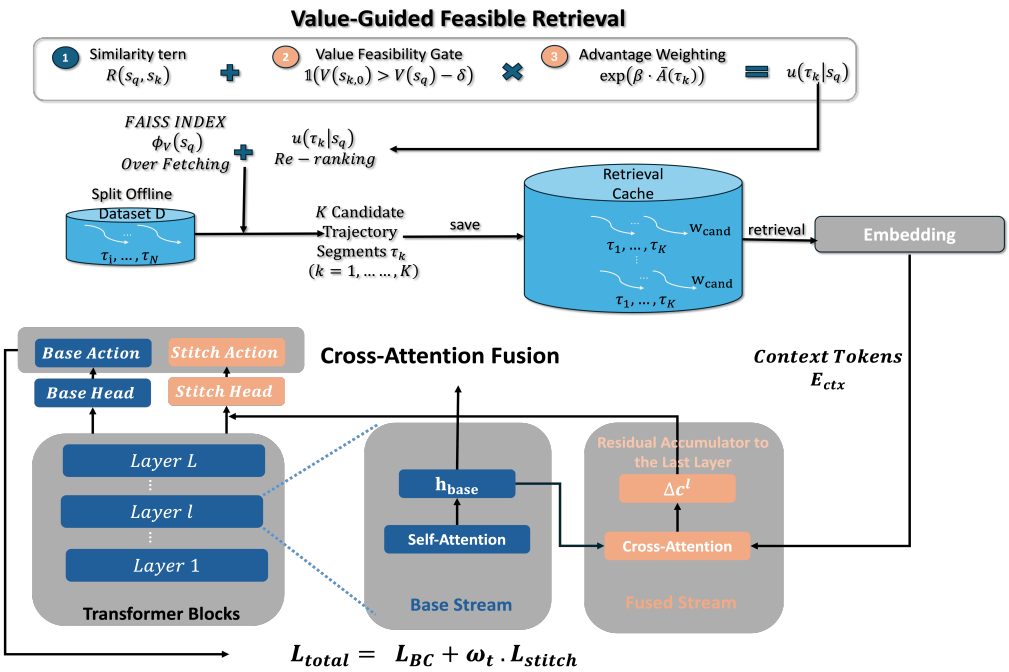


Figure 1: **Overview of ReStitch.** Given a query state, ReStitch retrieves trajectory continuations using relevance, feasibility, and advantage signals. The selected continuations are encoded as trajectory-level context and integrated through a parallel cross-attention stream. Separate action heads preserve a retrieval-free prediction pathway while enabling retrieval-conditioned action selection.

The primary challenge in retrieval-augmented RL is defining a metric that captures both the topological reachability and the dynamic quality of a candidate trajectory. To address this, we define a retrieval utility that combines relevance, feasibility, and advantage. Appendix A provides a motivation for its advantage-weighted component. We maintain an offline index $\mathcal{D}$ where each candidate state s_k is associated with its succeeding trajectory segment $\tau_k = \{(s_{k,j}, a_{k,j})\}_{j=0}^{H-1}$. Given the current online query state s_q, the utility of a candidate trajectory τ_k is evaluated as:

$$u(\tau_k|s_q) = \underbrace{R(s_q, s_{k,0})}_{\text{Similarity Relevance}} + \alpha \cdot \underbrace{\mathbb{1}(V(s_{k,0}) > V(s_q) - \delta)}_{\text{Value Feasibility Gate}} \cdot \underbrace{\exp(\beta \cdot \bar{A}(\tau_k))}_{\text{Advantage Weighting}} \tag{2}$$

where $\bar{A}(\tau_k) = \frac{1}{H} \sum_{j=0}^{H-1} A(s_{k,j}, a_{k,j})$ denotes the average advantage of the trajectory segment, and α is a balancing coefficient. $\phi_V(s_q)$, $\phi_V(s_{k,0})$ denote the penultimate-layer output of the pre-trained value network evaluated at the current state and the initial state of each candidate segment. β follows the setting in IQL policy extraction, while δ is chosen as a small tolerance proportional to the standard deviation of the value function (e.g.,$0.05 \times \text{std}(V)$). This formulation decomposes the retrieval process into three criteria:

- **Relevance** (R)**:** Measures spatial proximity via cosine similarity in the pre-trained IQL latent space, $R = \cos(\phi_V(s_q), \phi_V(s_{k,0}))$, intended to reflect connectivity beyond raw geometric distance.
- **Feasibility**($\mathbb{I}$)**:** A hard indicator $\mathbb{I}(V(s_{k,0}) > V(s_q) - \delta)$ that prevents stitching into inferior trajectories while allowing a margin δ for navigating local bottlenecks.
- **Advantage**(exp)**:** Weights candidates according to their mean advantage, motivated by the exponential advantage weighting used in AWR/IQL-style policy extraction.

The last term is motivated by the exponential advantage weighting used for policy extraction in AWR-style methods and IQL. Here, we extend the weighting from individual actions to trajectory segments by aggregating advantage over the retrieval horizon. Appendix A discusses this connection.

Importantly, we use this connection as motivation for the advantage-weighted component; the complete retrieval utility additionally incorporates relevance and feasibility criteria.

Together, these components rank continuations by relevance, value compatibility, and behavioral advantage. Candidates are first retrieved via approximate nearest-neighbor search and then reranked using the utility above.

Evaluating the utility over the entire offline dataset at each decision step is expensive. We therefore use a two-stage retrieval pipeline—FAISS over-fetching followed by utility reranking—together with offline caching for efficient training and inference. Implementation details are provided in Appendix F.

4.2 Cross-Attention Fusion and Dimensional Representation

To inject the retrieved topological knowledge into the decision-making process without overwhelming the computational budget, we design a highly efficient feature encoding and fusion module. Unlike Large Language Models, the standard Decision Transformer employs a shallow architecture for continuous control tasks. We leverage this shallow backbone to permeate the stitching guidance through a compact, quality-aware cross-attention mechanism (Chen et al., 2021a).

Compact Context Encoding. At each decision step t, the retrieval module provides K candidate segments, each spanning a horizon of H timesteps. The raw retrieved features consist of states $S_{cand} \in \mathbb{R}^{B\times K\times H\times d_{obs}}$ and actions $A_{cand} \in \mathbb{R}^{B\times K\times H\times d_{act}}$.

To construct a dimensionally efficient memory bank, we first independently embed the states and actions into the hidden dimension d, and then project the state-action transitions at each timestep into a unified latent token:

$$E_{seg} = \text{MLP}\big(\text{Concat}(\text{Embed}_s(S_{cand}), \text{Embed}_a(A_{cand}))\big) \in \mathbb{R}^{B\times K\times H\times d} \tag{3}$$

This projection compresses the context into exactly H tokens per candidate segment. We then flatten these tokens to form the global context memory bank $E_{ctx} \in \mathbb{R}^{B\times M\times d}$, where $M = K \times H$ is the total number of context tokens. This compact representation effectively mitigates the sequence length explosion typical in multi-token per-step setups.

Quality-Aware Cross-Attention (CCA). The core of our fusion module is an External Query Cross-Attention mechanism that integrates the current intent with retrieved trajectory context.

For the current timestep t, the base DT computes the causal hidden state $h_{base} \in \mathbb{R}^{B\times d}$. We treat the layer-normalized h_{base} as a single query token: $Q = \text{LayerNorm}(h_{base}) \in \mathbb{R}^{B\times 1\times d}$. The keys and values are derived from the layer-normalized context memory bank E_{ctx}.

We apply precomputed retrieval scores to context tokens prior to attention (Eq. 2). We apply these scores $w_{cand} \in \mathbb{R}^{B\times M}$ as a soft weighting mechanism directly to the context tokens prior to the attention operation:

$$\tilde{E}_{ctx} = \text{LayerNorm}(E_{ctx}) \odot w_{cand} \tag{4}$$

where $\odot$ denotes element-wise multiplication broadcasted over the hidden dimension. This *Quality-Aware Weighting* ensures that the attention heads inherently prioritize tokens from higher-utility retrieved continuations.

To preserve a retrieval-free autoregressive pathway, we maintain the base stream independently from the retrieval-conditioned side stream. Following the parallel-stream design of Zhang et al. (2024), we adopt a Parallel Dual-Stream Architecture. Specifically, in each layer l, the base hidden state $h_{base}^{(l)}$ preserves its causal self-attention pathway without integrating external context as $h_{\text{base}}^{(l)} = \text{DT_Block}\left(h_{\text{base}}^{(l-1)}\right)$.

Simultaneously, an independent fused side-stream $z^{(l)}$ acts as a residual accumulator. At each layer, $h_{base}^{(l)}$ serves as the query to extract topological guidance via cross-attention, and the resulting delta $\Delta c^{(l)}$ is added exclusively to the side-stream:

$$z^{(l)} = z^{(l-1)} + \Delta c^{(l)}, \qquad \text{where } \Delta c^{(l)} = \text{CrossAttn}\left(Q = h_{base}^{(l)}, K = V = E_{ctx}\right) \tag{5}$$

The final representation at layer L is evaluated as: $h_{fused}^{(L)} = h_{base}^{(L)} + z^{(L)}$.

This unidirectional information flow preserves a retrieval-free base representation prior to action decoding. Instead of introducing complex layer-wise gating, we accumulate the cross-attention residuals linearly in the parallel side-stream. The stitching head Φ_{stitch} decodes the fused representation into a retrieval-conditioned action prediction: $\hat{a}_{\text{stitch}} = \Phi_{\text{stitch}}(h_{\text{fused}}^{(L)})$.

4.3 Decoupled Action Decoding and Selective Stitching Loss

Following the cross-attention fusion, the network possesses two distinct latent representations at the final layer L: the base representation $h_{\text{base}}^{(L)}$ and the retrieval-conditioned representation $h_{\text{fused}}^{(L)}$. A naive approach would route both representations through a shared action prediction head. However, a shared head would be optimized toward two distinct targets: the observed dataset action a_t and the retrieved target action a_{stitch}^*.

Decoupled Action Heads. To resolve this representation interference, we introduce a decoupled decoding architecture. We employ two structurally identical but independently parameterized multi-layer perceptrons (MLPs) to predict the respective actions:

$$\hat{a}_{base} = \Phi_{base}(h_{base}^{(L)}), \qquad \hat{a}_{stitch} = \Phi_{stitch}(h_{fused}^{(L)}) \tag{6}$$

This explicit separation ensures that the base head Φ_{base} focuses solely on estimating the behavioral cloning (BC) distribution of the offline dataset, while the specialized stitching head Φ_{stitch} is dedicated to interpreting the cross-attention prompt and outputting the retrieval-conditioned action.

We use separate action heads rather than blending the base and retrieval-conditioned action predictions. During training, the contribution of the stitching objective is modulated by a distance-aware weight, reducing the influence of retrieved targets whose anchor states are far from the current state.

Dual-Objective Soft-Gated Loss. During training, the ground-truth target for the base head is the actual action a_t executed in the dataset, whereas the target for the stitching head is the first action of the highest-utility retrieved continuation, denoted by a_{stitch}^*.

To reduce the influence of retrieved targets associated with distant anchor states, we define

$$d_t = \|s_t - s_t^{\text{anc}}\|_2, \qquad w_t = \exp(-d_t/\sigma) \in (0, 1],$$

where s_t^{anc} denotes the anchor state associated with the retrieved context. During training, $s_t^{\text{anc}} = s_{\text{stitch}}^*$; during inference, $s_t^{\text{anc}} = s_{\text{NN}}$. See Appendix C for the full analysis. Only valid time steps (indicated by the padding mask) participate in the loss calculation. The final optimization objective is formulated as a dynamic dual-loss:

$$\mathcal{L}_{total}(\theta) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\underbrace{\|\hat{a}_{base} - a_t\|^2}_{\text{Base BC Loss}} + \lambda \cdot \text{sg}(w_t) \cdot \underbrace{\|\hat{a}_{stitch} - a_{stitch}^*\|^2}_{\text{Guided Stitching Loss}}\right] \tag{7}$$

where $\text{sg}(\cdot)$ denotes the stop-gradient operator, and λ is a global weighting coefficient to balance the two objectives.

Training Dynamics. The distance weight controls the contribution of the stitching objective. When the retrieved anchor is close to the current state, the stitching loss receives a larger weight; as the distance increases, its contribution decreases. The base BC objective is optimized independently of this weight, preserving a retrieval-free training pathway.

To reduce online retrieval overhead, we decouple retrieval preprocessing from policy learning through offline caching and latent anchor matching. The complete system implementation details are provided in Appendix F, and the corresponding pseudocode is outlined in Appendix D.

At inference, the stitching head is selected when $w_t \geq \tau_{\text{safe}}$, where τ_{safe} is fixed across all tasks and specified in Appendix E.

5 EXPERIMENTS

Our experiments address three questions: (1) Does retrieval-based policy improvement improve offline trajectory stitching? (2) Under what task characteristics is retrieval-based stitching most effective? (3) Which components of ReStitch account for its empirical behavior?

5.1 EXPERIMENTAL SETUP

Benchmarks. We evaluate our method on two distinct domains from the D4RL benchmark (Fu et al., 2020): navigation and control. For navigation, we utilize the challenging **AntMaze** domain (`umaze`, `medium`, and `large`), which features sparse rewards and heavily relies on long-horizon trajectory stitching to bypass dead-ends. For continuous control, we evaluate on the **MuJoCo** domain (`HalfCheetah`, `Hopper`, and `Walker2d`) across three dataset qualities: `medium`, `medium-replay`, and `medium-expert`.

Baselines. We compare ReStitch against several representative offline RL baselines, grouped into three categories:

1. **Value-Based Methods**:The standard dynamic programming approach IQL (Kostrikov et al., 2022), CQL (Kumar et al., 2020), and TD3+BC (Fujimoto & Gu, 2021).
2. **Standard Sequence Models (RCSL)**: Decision Transformer (DT) (Chen et al., 2021b), Decision convformer (DC) (Kim et al., 2024a) and RvS (Emmons et al., 2022). These represent pure behavioral cloning without policy improvement methods.
3. **Stitching-Enhanced Sequence Models**: This is the most critical comparison group, representing recent efforts to inject dynamic programming into sequence models. We compare against **QDT** (Yamagata et al., 2023), **CGDT** (Wang et al., 2024), **ACT** (Gao et al., 2024), **EDT** (Wu et al., 2023), and notably, **POR** (Xu et al., 2022) (which uses imitation learning conditioned on high-value states) and **QCS** (Kim et al., 2024b) (integrates Q-aid into the RCSL loss function). We also reference **QT** (Hu et al., 2024)(Q-Transformer variants) to represent generalized Q-guided sequence generation and **HIQL** (Park et al., 2023)(Hierarchical Implicit Q-Learning), a goal-conditioned hierarchical offline baseline that shares IQL foundation and tackles long-horizon navigation tasks by extracting intermediate subgoals.

Evaluation Protocol and Metrics. For the baselines that we reproduce, we use a unified evaluation protocol following QCS (Kim et al., 2024b). Specifically, we evaluate expert-normalized returns over 10 episodes at each checkpoint, recorded every 10^3 gradient steps, and compute a running average over 10 consecutive checkpoints. We report the mean and standard deviation of these temporally smoothed scores across five independent random seeds. This protocol is applied to ReStitch and all newly reproduced baselines, including QT, POR, QCS, and HIQL.

5.2 MAIN RESULTS

Navigation Tasks. Table 1 summarizes performance on the six AntMaze tasks. ReStitch achieves an average normalized return of 83.5, compared with 79.3 for HIQL and 77.3 for QCS-RTG. The improvements are particularly visible on the large-maze tasks: ReStitch obtains 74.5 and 80.9 on large-play and large-diverse, respectively, compared with 71.4 and 78.5 for HIQL and 69.2 and 70.3 for QCS-RTG. On medium-diverse, ReStitch performs similarly to QCS-RTG (78.6 vs. 79.1).

These results show that ReStitch is effective in sparse-reward navigation settings that require long-horizon composition of offline experience. We further investigate the mechanisms underlying these improvements in Section 5.4.

Control Tasks. As shown in Table 2, ReStitch improves modestly over vanilla DT on average (78.4 vs. 74.7), but remains below several strong dense-reward baselines, including QT (82.2) and QCS (90.1). Its behavior also varies across datasets: improvements over DT are more visible on some medium and medium-replay tasks, while the gains are small or absent on several medium-expert tasks. This contrast with AntMaze suggests that the benefit of retrieval-based trajectory stitching is task dependent. In sparse-reward navigation, retrieved continuations can provide useful long-horizon

guidance, whereas dense-reward locomotion places greater emphasis on locally consistent actions and dynamics. We examine this contrast in Section 5.3.

Table 1: Normalized returns on the D4RL AntMaze-v2 benchmarks. ReStitch is compared with value-based, return-conditioned, and stitching-oriented offline RL baselines. Higher values indicate better performance.

Dataset	TD3+BC	IQL	CQL	DT	DC	RvS-R	RvS-G	QT	HIQL	POR	QCS-RTG	ReStitch
antmaze-u	78.6	87.5	74.0	65.6	85.0	64.4	65.4	58.0 ± 7.2	82.1 ± 7.3	67.0 ± 9.0	92.4 ± 4.1	**92.7 ± 0.4**
antmaze-u-d	71.4	62.2	84.0	51.2	78.5	70.1	60.9	51.8 ± 9.6	81.5 ± 4.2	33.6 ± 18.4	71.8 ± 12.1	**84.6 ± 0.8**
antmaze-m-p	10.6	71.2	61.2	4.3	33.2	4.5	58.1	18.0 ± 7.0	85.3 ± 3.8	84.0 ± 5.0	81.2 ± 7.3	**88.2 ± 1.2**
antmaze-m-d	3.0	70.0	53.7	1.2	27.5	7.7	67.3	13.6 ± 10.5	76.8 ± 5.4	75.2 ± 8.4	**79.1 ± 6.0**	78.6 ± 1.5
antmaze-l-p	0.2	39.6	15.8	0.0	4.8	3.5	32.4	3.0 ± 2.0	71.4 ± 6.2	51.6 ± 5.9	69.2 ± 7.4	**74.5 ± 1.8**
antmaze-l-d	0.0	47.5	14.9	0.5	12.3	3.7	36.9	1.6 ± 2.3	78.5 ± 4.7	55.6 ± 8.6	70.3 ± 5.9	**80.9 ± 1.6**
Average	27.3	63.0	50.6	20.5	40.2	25.6	53.5	24.3	79.3	61.2	77.3	**83.5**

Note: ± denotes standard deviation over 5 random seeds. Results for QT, HIQL, POR, and QCS-RTG are reproduced under the same unified evaluation protocol on AntMaze-v2 environments. Bold indicates the highest mean performance.

Table 2: Normalized returns on the D4RL MuJoCo-v2 locomotion benchmarks. ReStitch is compared with value-based, return-conditioned, and stitching-oriented offline RL baselines. Higher values indicate better performance.

Dataset	TD3	IQL	CQL	DT	DC	RvS	QDT	EDT	CGDT	ACT	QT	POR	QCS	ReStitch
hc-m	48.3	47.4	44.0	42.6	43.0	41.6	42.3	42.5	43.0	49.1	50.6 ± 0.2	46.7 ± 0.1	**58.7 ± 0.6**	44.5 ± 0.3
hop-m	59.3	66.3	58.5	67.6	92.5	60.2	66.5	63.5	**96.9**	67.8	67.6 ± 3.9	46.7 ± 7.0	96.1 ± 3.9	72.3 ± 1.1
walk-m	83.7	78.3	72.5	74.0	79.2	71.7	67.1	72.8	79.1	80.9	85.6 ± 0.7	80.3 ± 2.6	**87.8 ± 1.4**	76.5 ± 0.8
hc-m-r	44.6	44.2	45.5	36.6	41.3	38.0	35.6	37.8	40.4	43.0	48.3 ± 0.4	43.3 ± 0.4	**54.4 ± 0.6**	42.8 ± 0.4
hop-m-r	60.9	94.7	95.0	82.7	94.2	73.5	52.1	89.0	93.4	98.4	89.3 ± 4.7	36.5 ± 13.7	**100.1 ± 1.3**	88.5 ± 1.5
walk-m-r	81.8	73.9	77.2	66.6	76.6	60.6	58.2	74.8	78.1	56.1	90.9 ± 4.1	38.8 ± 16.2	**93.8 ± 2.2**	75.2 ± 0.9
hc-m-e	90.7	86.7	91.6	86.8	93.0	92.2	-	-	93.6	**96.1**	93.6 ± 1.1	89.9 ± 2.0	92.9 ± 2.1	90.5 ± 0.6
hop-m-e	98.0	91.5	105.4	107.6	110.4	101.7	-	-	107.6	**111.5**	104.8 ± 3.9	66.6 ± 9.4	110.5 ± 2.7	106.2 ± 1.3
walk-m-e	110.1	109.6	108.8	108.1	109.6	106.0	-	-	109.3	113.3	109.5 ± 0.2	86.4 ± 10.8	**116.3 ± 2.6**	109.5 ± 0.8
Average	75.3	77.0	77.6	74.7	82.2	71.7	-	-	82.4	79.6	82.2	59.5	**90.1**	78.4

Note: ± denotes standard deviation over 5 random seeds. Results for QT, POR, and QCS are reproduced under the same unified evaluation protocol on MuJoCo-v2 environments. Bold indicates the highest mean performance. Shortened names: hc (halfcheetah), hop (hopper), walk (walker2d).

5.3 Empirical Capability Boundary of Retrieval-Based Stitching

The contrast between AntMaze and MuJoCo reveals an empirical capability boundary of ReStitch. AntMaze requires long-horizon navigation, where useful behavior can be composed by connecting trajectory segments across the maze. Retrieved continuations can therefore provide guidance beyond the local trajectory.

In contrast, MuJoCo has dense rewards and requires locally consistent actions and high-frequency control, making retrieved continuations from nearby states less reliable. This helps explain why ReStitch yields larger gains on AntMaze than on MuJoCo. Overall, retrieval-based stitching is most useful when trajectory-level continuations address a long-horizon composition problem; extending it to dynamics-sensitive control remains an important direction for future work.

5.4 Mechanism and Ablation Analysis

We analyze ReStitch from three perspectives: retrieval utility, value representation (its topological structure and quality), and retrieved-context integration. Additional hyperparameter sensitivity and cross-attention visualizations are provided in Appendix E.5 and G respectively.

1. Utility Function Design. We ablate the three components of the retrieval utility in Eq. 2 using relevance-only, advantage-only, and no-value-gate variants. As in Figure 3a, the full utility performs best, indicating that relevance, feasibility, and advantage provide complementary signals for retrieval.

2. Topological Awareness of Retrieval. To examine whether the IQL representation contains information beyond raw geometric proximity, Figure 2 shows a matched-distance example near an

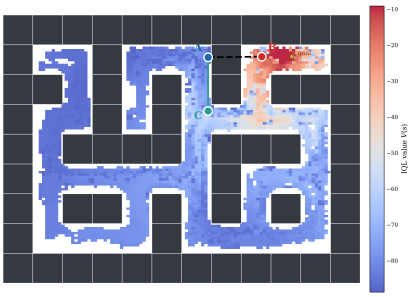


Figure 2: Comparison of learned representation similarity for geometrically matched states with different connectivity.

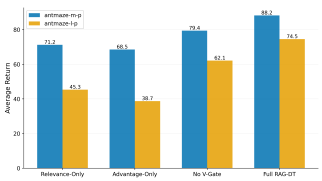
Table 3: Effect of IQL checkpoint quality on ReStitch performance.

Steps	Quality	IQL	ReStitch
0	Random	0.0	18.4 ± 12.7
500k	Mid	32.0	75.2 ± 3.8
1M	Final	47.2	88.2 ± 1.2

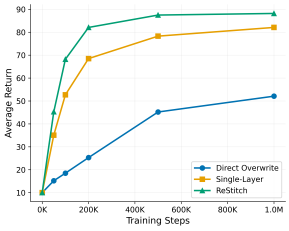
AntMaze bottleneck. Although s_B and s_C are approximately equidistant from s_A ($d_{\text{pos}} \approx 7.20$), the same-corridor state s_C has higher similarity than the across-wall state s_B (0.978 vs. 0.831). This provides qualitative evidence that the learned representation captures connectivity information beyond raw geometric proximity.

3. Effect of IQL Representation Quality. To investigate the extent to which ReStitch relies on the quality of the pretrained IQL value network, we evaluated performance using retrieval indices built from IQL checkpoints at different training stages (0, 500k, and 1M steps) on the **antmaze-medium-play-v2** dataset. Table 3 shows that retrieval with an untrained representation performs poorly, whereas the intermediate checkpoint already recovers most of the final performance. Thus, useful retrieval requires a meaningful learned representation, though not a fully trained one, suggesting that simply adding retrieved context is insufficient.

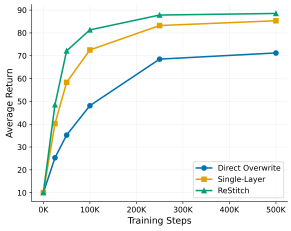
4. Architectural Design: Multi-Layer vs. Single-Layer Injection. We compare the full dual-stream architecture with single-layer injection and direct overwrite. As in Figure 3b and Figure 3c, the dual-stream design achieves the strongest performance on both AntMaze-medium-play and Hopper-medium-replay. Direct overwrite exhibits less stable optimization, while single-layer injection underperforms the multi-layer design, supporting the use of a separate retrieval-conditioned pathway.



(a) Utility Function Design Ablation



(b) Antmaze-medium-play



(c) Hopper-medium-replay

Figure 3: Mechanism analysis. (a) Retrieval-utility ablation on AntMaze-medium-play. (b)–(c) Architecture ablations on AntMaze-medium-play and Hopper-medium-replay.

6 Conclusion

We studied offline trajectory stitching through retrieval-based policy improvement. ReStitch retrieves relevant, feasible, and advantageous trajectory continuations as explicit guidance for action selection. Across D4RL benchmarks, ReStitch provides substantial improvements on sparse-reward AntMaze tasks, while its gains are more limited on dense-reward MuJoCo locomotion. Our analyses further show that performance depends on the quality of the learned value representation and on how retrieved context is incorporated into the sequence model. These results suggest that retrieval-based policy improvement is particularly useful when trajectory-level continuation information addresses a genuine long-horizon composition problem, while extending this approach to dynamics-sensitive control remains an important direction for future work.

AI USE STATEMENT

Generative AI tools were used to assist with research methodology and experimental design discussions, interpretation and presentation of results, code generation and debugging, and drafting, editing, and polishing portions of the manuscript. All AI-assisted technical content and generated code were reviewed, tested, and verified by the authors. The authors take full responsibility for the methodology, theoretical claims, experimental results, analyses, and final content of the manuscript.

ETHICS STATEMENT

This work studies offline reinforcement learning algorithms and evaluates them exclusively on publicly available simulated benchmarks (D4RL). It involves no human subjects, personal or sensitive data, or real-world deployment, and we are not aware of any direct ethical concerns or conflicts of interest arising from this research.

REPRODUCIBILITY STATEMENT

We have made efforts to ensure the reproducibility of our results. The full method is described in Section 4, with pseudocode in Appendix D and system implementation details, including the retrieval index construction, in Appendix F. All hyperparameters, task-specific configurations, and tuning guidelines are reported in Appendix E, together with sensitivity analyses. The theoretical results (Propositions 1 and 2) are stated with their assumptions and proofs in Appendices A and C. All experiments use the public D4RL benchmark, and results are averaged over five random seeds as described in Section 5.2.

REFERENCES

Ari Azarafrooz. Differentiable neural computers with memory demon. *arXiv preprint arXiv:2211.02987*, 2022.

Richard Bellman. A Markovian decision process. *Journal of Mathematics and Mechanics*, 6(5): 679–684, 1957.

David Brandfonbrener, Alberto Bietti, Jacob Buckman, Romain Laroche, and Joan Bruna. When does return-conditioned supervised learning work for offline reinforcement learning? In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. CrossViT: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 357–366, 2021a.

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. *Advances in neural information processing systems*, 34:15084–15097, 2021b.

Yiming Ding, Carlos Florensa, Mariano Phielipp, and Pieter Abbeel. Goal-conditioned imitation learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Xintong Duan, Yutong He, Fahim Tajwar, Ruslan Salakhutdinov, J. Zico Kolter, and Jeff Schneider. Accelerating diffusion planners in offline RL via reward-aware consistency trajectory distillation. In *International Conference on Learning Representations*, 2026.

Scott Emmons, Benjamin Eysenbach, Ilya Kostrikov, and Sergey Levine. RvS: What is essential for offline RL via supervised learning? In *International Conference on Learning Representations*, 2022.

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.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

Chen-Xiao Gao, Chenyang Wu, Mingjun Cao, Rui Kong, Zongzhang Zhang, and Yang Yu. ACT: Empowering decision transformer with dynamic programming via advantage conditioning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 12127–12135, 2024.

Anirudh Goyal, Abram L. Friesen, Andrea Banino, Theophane Weber, Nan Rosemary Ke, Adrià Puigdomènech Badia, Arthur Guez, Mehdi Mirza, Peter C. Humphreys, Ksenia Konyushkova, et al. Retrieval-augmented reinforcement learning. In *International Conference on Machine Learning*, pp. 7740–7765. PMLR, 2022.

Alex Graves, Greg Wayne, and Ivo Danihelka. Neural Turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

Shengchao Hu, Ziqing Fan, Chaoqin Huang, Li Shen, Ya Zhang, Yanfeng Wang, and Dacheng Tao. Q-value regularized transformer for offline reinforcement learning. In *International Conference on Machine Learning*, 2024.

Peter Humphreys, Arthur Guez, Olivier Tieleman, Laurent Sifre, Théophane Weber, and Timothy Lillicrap. Large-scale retrieval for reinforcement learning. *Advances in Neural Information Processing Systems*, 35:20092–20104, 2022.

Cuixia Jiang, Ming Jiang, Qifa Xu, and Xue Huang. Expectile regression neural network model with applications. *Neurocomputing*, 247:73–86, 2017.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

Jeonghye Kim, Suyoung Lee, Woojun Kim, and Youngchul Sung. Decision ConvFormer: Local filtering in MetaFormer is sufficient for decision making. In *International Conference on Learning Representations*, 2024a.

Jeonghye Kim, Suyoung Lee, Woojun Kim, and Youngchul Sung. Adaptive Q-aid for conditional supervised learning in offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 37, 2024b.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations*, 2022.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

Jonathan Lee, Annie Xie, Aldo Pacchiano, Yash Chandak, Chelsea Finn, Ofir Nachum, and Emma Brunskill. Supervised pretraining can learn in-context reinforcement learning. *Advances in Neural Information Processing Systems*, 36:43057–43083, 2023.

Chenxing Lin, Xinhui Gao, Haipeng Zhang, Xinran Li, Haitao Wang, Songzhu Mei, Chenglu Wen, Weiquan Liu, Siqi Shen, and Cheng Wang. MAGE: Multi-scale autoregressive generation for offline reinforcement learning. In *International Conference on Learning Representations*, 2026.

Zhishuai Liu, Yu Yang, Ruhan Wang, Pan Xu, and Dongruo Zhou. How to provably improve return conditioned supervised learning? *arXiv preprint arXiv:2506.08463*, 2025.

Seohong Park, Dibya Ghosh, Benjamin Eysenbach, and Sergey Levine. HIQL: Offline goal-conditioned RL with latent states as actions. *Advances in Neural Information Processing Systems*, 36:34866–34891, 2023.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *Transactions on Machine Learning Research*, 2022.

Thomas Schmied, Fabian Paischer, Vihang Patil, Markus Hofmarcher, Razvan Pascanu, and Sepp Hochreiter. Retrieval-augmented decision transformer: External memory for in-context RL. In *Conference on Lifelong Learning Agents (CoLLAs)*, 2025.

Guojian Wang, Quinson Hon, Xuyang Chen, and Lin Zhao. Data fusion-enhanced decision transformer for stable cross-domain generalization. *arXiv preprint arXiv:2511.09173*, 2025a.

Jiuqi Wang, Ethan Blaser, Hadi Daneshmand, and Shangtong Zhang. Transformers can learn temporal difference methods for in-context reinforcement learning. In *International Conference on Learning Representations*, 2025b.

Yongyi Wang, Hanyu Liu, Lingfeng Li, Bozhou Chen, Ang Li, Qirui Zheng, Xionghui Yang, and Wenxin Li. Decoupling return-to-go for efficient decision transformer. *arXiv preprint arXiv:2601.15953*, 2026.

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, 2024.

Peiran Wu, Zhuorui Yu, Yunze Liu, Chi-Hao Wu, Enmin Zhou, and Junxiao Shen. MARC: Memory-augmented RL token compression for efficient video understanding. In *International Conference on Learning Representations*, 2026.

Yueh-Hua Wu, Xiaolong Wang, and Masashi Hamaya. Elastic decision transformer. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Haoran Xu, Li Jiang, Jianxiong Li, and Xianyuan Zhan. A policy-guided imitation approach for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

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*, 2023.

Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. LLaMA-Adapter: Efficient fine-tuning of language models with zero-init attention. In *International Conference on Learning Representations*, 2024.

Zifeng Zhuang, Dengyun Peng, Jinxin Liu, Ziqi Zhang, and Donglin Wang. Reinformer: Max-return sequence modeling for offline RL. In *International Conference on Machine Learning*, 2024.

A MOTIVATION FOR THE ADVANTAGE-WEIGHTED COMPONENT

This appendix motivates the advantage-weighted term in Eq. 2 through the KL-regularized policy improvement step used in AWR and IQL.

Proposition 1 (Trajectory-Level Advantage Weighting). *The problem* $\max_\pi \mathbb{E}_{a\sim\pi(\cdot|s)}[A(s,a)] - \frac{1}{\beta} D_{\mathrm{KL}}(\pi(\cdot|s) \,\|\, \pi_\beta(\cdot|s))$ *has the solution* $\pi^*(a|s) = \pi_\beta(a|s)\exp(\beta A(s,a))/Z(s)$. *For a segment* $\tau = (s_0, a_0, \ldots, s_{H-1}, a_{H-1})$ *generated under the same dynamics,*

$$\log \frac{P^*(\tau)}{P_\beta(\tau)} = \beta \sum_{j=0}^{H-1} A(s_j, a_j) - \sum_{j=0}^{H-1} \log Z(s_j).$$

If $\log Z(s)$ *varies little across the states visited by candidate segments, then for a fixed horizon* H*, ranking candidates by* $P^*(\tau)/P_\beta(\tau)$ *is approximately equivalent to ranking them by* $\exp(\beta \bar{A}(\tau))$*, where* $\bar{A}(\tau) = \frac{1}{H}\sum_{j=0}^{H-1} A(s_j, a_j)$.

Proof. **1. Single-step MDP Constraint:** Following the AWR framework, the constrained policy improvement objective for a single step is defined as:

$$\max_\pi \mathbb{E}_{a\sim\pi(\cdot|s)}[A(s,a)] \quad \text{s.t.} \quad D_{\mathrm{KL}}(\pi(\cdot|s)||\pi_\beta(\cdot|s)) \le \epsilon \tag{8}$$

Using the method of Lagrange multipliers with the inverse temperature parameter $1/\beta$, the unconstrained Lagrangian is:

$$\mathcal{L}(\pi) = \mathbb{E}_{a\sim\pi}[A(s,a)] - \frac{1}{\beta} D_{\mathrm{KL}}(\pi||\pi_\beta) \tag{9}$$

The closed-form optimal solution for the single-step policy is:

$$\pi^*(a|s) = \frac{1}{Z(s)} \pi_\beta(a|s)\exp(\beta A(s,a)) \tag{10}$$

2. Extension to Trajectory Space: In ReStitch, we retrieve trajectory segments $\tau = (s_0, a_0, \ldots, s_{H-1}, a_{H-1})$ of horizon H. Assuming Markovian transition dynamics, the ratio between the optimal trajectory likelihood P^* and the behavior trajectory likelihood P_β is the product of individual step ratios:

$$\frac{P^*(\tau)}{P_\beta(\tau)} = \prod_{j=0}^{H-1} \frac{\pi^*(a_j|s_j)}{\pi_\beta(a_j|s_j)} = \prod_{j=0}^{H-1} \frac{\exp(\beta A(s_j,a_j))}{Z(s_j)} \tag{11}$$

Taking the logarithm of Eq. (11) gives

$$\log \frac{P^*(\tau)}{P_\beta(\tau)} = \beta \sum_{j=0}^{H-1} A(s_j, a_j) \; - \; \sum_{j=0}^{H-1} \log Z(s_j). \tag{12}$$

Under the approximation that $\log Z(s_j)$ is approximately constant across candidate segments, and dividing by the fixed horizon H, we obtain

$$\frac{1}{H} \log \frac{P^*(\tau)}{P_\beta(\tau)} = \beta\, \bar{A}(\tau) + \text{const}, \qquad \bar{A}(\tau) = \frac{1}{H} \sum_{j=0}^{H-1} A(s_j, a_j). \tag{13}$$

Since the constant does not depend on τ and $\exp(\cdot)$ is monotone, ranking candidates by $P^*(\tau)/P_\beta(\tau)$ is approximately equivalent to ranking them by the trajectory-level advantage weighting

$$w_{\mathrm{adv}}(\tau) \propto \exp\big(\beta\, \bar{A}(\tau)\big),$$

which motivates the advantage component of Eq. 2. □

Scope of the connection. This derivation motivates only the advantage-weighted component of ReStitch. The complete retrieval utility additionally incorporates relevance and feasibility criteria.

B Discussion on Local Transition Retrieval

Standard return-conditioned sequence modeling methods rely on observing sufficiently high-return trajectories within the offline dataset. When optimal long-horizon trajectories are absent, the learned conditional distribution may suffer from poor generalization due to insufficient dataset support.

ReStitch partially alleviates this issue by decomposing long-horizon decision making into locally reachable retrieval-conditioned transitions. Instead of directly modeling:

$$P(a_t \mid s_t, \hat{R}_t = g_{\max}),$$

our method conditions action generation on retrieved anchor trajectories:

$$P(a_t \mid s_t, \tau_{\text{ret}}),$$

where the retrieved trajectory segment τ_{ret} is selected from nearby states in the latent value space.

Intuitively, this retrieval mechanism shifts the learning problem from extrapolating globally optimal trajectories toward leveraging locally supported transition patterns already present in the dataset. While this does not eliminate the fundamental coverage limitations of offline RL, it empirically reduces the dependence on globally optimal demonstrations in sparse-reward environments such as AntMaze.

C Analysis of Distance Gating

This appendix characterizes the locality induced by the distance gate. Let

$$d_t = \|s_t - s_t^{\text{anc}}\|_2$$

denote the distance between the current state s_t and the anchor state s_t^{anc} associated with the retrieved context, where s_t^{anc} corresponds to the retrieval anchor s^*_{stitch} during training and the nearest-neighbor anchor s_{NN} at inference. Consistent with the distance gate used in the main method, we define

$$w_t = \exp(-d_t/\sigma), \qquad \sigma > 0.$$

At inference, the stitching head is selected if and only if $w_t \geq \tau_{\text{safe}}$, where $\tau_{\text{safe}} \in (0, 1)$.

Proposition 2 (Gate-induced locality)**.** *Let*

$$r = \sigma \ln(1/\tau_{\text{safe}}).$$

Then:

- *(i)* ***Inference.*** *The stitching head is selected only when $d_t \leq r$; otherwise, the base head is selected.*
- *(ii)* ***Training.*** *For any sample with $d_t > r$, the stitching term in Eq. 7 is weighted by $\lambda w_t < \lambda\tau_{\text{safe}}$, and its weight decays exponentially with d_t.*

Proof. Since the exponential function is monotone,

$$w_t \geq \tau_{\text{safe}} \iff \exp(-d_t/\sigma) \geq \tau_{\text{safe}} \iff d_t \leq \sigma \ln(1/\tau_{\text{safe}}) = r,$$

which proves (i).

For $d_t > r$,

$$w_t = \exp(-d_t/\sigma) < \exp(-r/\sigma) = \tau_{\text{safe}}.$$

Therefore, the corresponding stitching term is weighted by $\lambda w_t < \lambda\tau_{\text{safe}}$, and this weight decreases exponentially as d_t increases, proving (ii). □

Remark 1 (Scope of the analysis)**.** *Proposition* **??** *establishes only a locality property of the distance gate. It does not imply a safety or performance guarantee. Retrieval quality still depends on the relevance, feasibility, and advantage criteria in Eq. 2. Moreover, because the base and stitching streams share the Transformer backbone, the stitching objective can influence the learned representation even when the base action head is selected. Consequently, the analysis does not imply that ReStitch performs at least as well as DT. The empirical effect of distance gating is evaluated in Section 5.*

D PSEUDOCODE

Explanation of Algorithm: The training procedure (Line 1-10) emphasizes the dual-stream fusion. Specifically, Line 4-5 compute the feasibility-gated utility u, ensuring that the cross-attention module (Line 7) only prioritizes trajectories favored by both the value-feasibility and advantage criteria. During inference (Line 11-16), the process is highly efficient: the FAISS query (Line 12) operates in sub-millisecond time, and the cross-attention mechanism acts as a soft-switching gate, enabling the model to dynamically follow the "high-value shortcut" provided by the retrieved instance without iterative optimization.

Algorithm 1 ReStitch

Phase 1: Offline Pre-computation

Input: Offline dataset $\mathcal{D}$, trained IQL networks V_ψ and Q_θ, hyperparameters $\alpha, \beta, \delta, K, H$.
Compute advantage $A(s,a) = Q_\theta(s,a) - V_\psi(s)$ for all $(s,a) \in \mathcal{D}$.
Extract latent topological features $\phi_V(s)$ for all states and build FAISS index.
for each state $s_q \in \mathcal{D}$ **do**
  Retrieve top candidate anchor states $s_{k,0}$ using FAISS over-fetching.
  for each candidate trajectory τ_k **do**
    Compute similarity $R(s_q, s_{k,0}) = \cos(\phi_V(s_q), \phi_V(s_{k,0}))$.
    Compute feasibility gate $\mathbb{I}_{gate} = \mathbb{I}(V(s_{k,0}) > V(s_q) - \delta)$.
    Compute utility: $u(\tau_k|s_q) = R(s_q, s_{k,0}) + \alpha \cdot \mathbb{I}_{gate} \cdot \exp(\beta \cdot \bar{A}(\tau_k))$.
  end for
  Rank and select top-K trajectories. Normalize utility scores to get w_{cand}.
  Save candidate indices and w_{cand} to disk (*e.g.,* `.npz` cache).
end for

Phase 2: Autoregressive Training

Input: Initialized ReStitch parameters, cached `.npz` arrays, distance gate σ, loss weight λ.
while not converged **do**
  Sample a batch of sequences τ_{batch} from $\mathcal{D}$.
  Perform $\mathcal{O}(1)$ cache lookup to fetch top-K context tokens E_{ctx} and weights w_{cand}.
  Apply quality-aware weighting: $E_{ctx} \leftarrow \text{LayerNorm}(E_{ctx}) \odot w_{cand}$.
  Initialize base hidden state $h_{base}^{(0)}$ and side-stream $z^{(0)} = 0$.
  for layer $l = 1$ **to** L **do**
    Base stream: $h_{base}^{(l)} = \text{DT_Block}(h_{base}^{(l-1)})$.
    Fused stream cross-attention: $\Delta c^{(l)} = \text{CrossAttn}(Q = h_{base}^{(l)}, K = V = E_{ctx})$.
    Accumulate side-stream: $z^{(l)} = z^{(l-1)} + \Delta c^{(l)}$.
  end for
  Final fusion: $h_{fused}^{(L)} = h_{base}^{(L)} + z^{(L)}$.
  Decode actions: $\hat{a}_{base} = \Phi_{base}(h_{base}^{(L)})$ and $\hat{a}_{stitch} = \Phi_{stitch}(h_{fused}^{(L)})$.
  Compute distance gate $w_t = \exp(-||s_t - s^*_{stitch}||_2/\sigma)$.
  Compute dual-loss: $\mathcal{L} = ||\hat{a}_{base} - a_t||^2 + \lambda \cdot \text{sg}(w_t) \cdot ||\hat{a}_{stitch} - a^*_{stitch}||^2$.
  Update ReStitch parameters using gradient descent.
end while

Phase 3: Fast Online Inference with Distance-Gated Fallback

Input: Initial state s_1^{env}, target return $\hat{R}_1$, frozen ReStitch, FAISS index, `.npz` cache, safety threshold τ_{safe}.
for $t = 1$ **to** T_{max} **do**
  Compute latent feature $\phi_V(s_t^{env})$ using frozen V_ψ.
  Query FAISS for the 1-Nearest Neighbor historical anchor s_{NN}.
  Zero-Shot Cache Inheritance: Perform $\mathcal{O}(1)$ lookup using s_{NN}'s index to load context E_{ctx}.
  Pass sequence through ReStitch to obtain both $\hat{a}_{base}$ and $\hat{a}_{stitch}$.
  Distance-Gated Head Selection: Compute $d_t = \|s_t - s_{\text{NN}}\|_2$ and $w_t = \exp(-d_t/\sigma)$.
  if $w_t \geq \tau_{safe}$ **then**
    $a_t = \hat{a}_{stitch}$ {Select stitching head}
  else
    $a_t = \hat{a}_{base}$ {Select base head}
  end if
  Execute a_t in the environment, observe r_t, s_{t+1}^{env}.
  Update sequence and return-to-go $\hat{R}_{t+1} = \hat{R}_t - r_t$.
end for

E HYPERPARAMETER SELECTION

E.1 CRITICAL HYPERPARAMETERS

ReStitch introduces eight key hyperparameters controlling retrieval utility, distance gating, inference head selection, and loss weighting. We provide recommended ranges and task-specific settings:

Table 4: Critical Hyperparameters of ReStitch

Parameter	Effect	Recommended		Searched Range
		AntMaze	MuJoCo	
α (utility balance)	Relevance vs. advantage tradeoff	1.5	0.8	$[0.0, 2.0]$
β (inverse temperature)	Advantage upweighting sharpness	0.25	0.25	$[0.1, 0.5]$
δ (value feasibility)	Value drop tolerance	$0.05 \times \text{std}(V)$	$0.1 \times \text{std}(V)$	$[0, 0.2 \times \text{std}(V)]$
K (retrieved neighbors)	Diversity vs. cost	5	5	$[1, 11]$
H (trajectory horizon)	Lookahead window	5	10	$[1, 30]$
σ (distance gate temp.)	Distance sensitivity	0.8	0.3	$[0.3, 4.0]$
τ_{safe} (inference gate)	Stitch-head selection threshold	0.01	0.01	–
λ (stitching loss weight)	BC vs. stitching importance	1.5	0.8	$[0.2, 2.5]$

Note: $\text{std}(V)$ denotes the standard deviation of the value function.

E.2 TASK-SPECIFIC CONFIGURATIONS

AntMaze Navigation Tasks (Sparse Rewards):

$$\alpha = 1.5 \quad \text{(emphasize advantage quality)}$$
$$\delta = 0.05 \times \text{std}(V) \quad \text{(strict value gate)}$$
$$\sigma = 0.8 \quad \text{(relaxed distance gating, leverage stitching)}$$
$$\lambda = 1.5 \quad \text{(strong stitching signal for sparse rewards)}$$
$$H = 5 \quad \text{(short horizon for agile local topological routing)}$$

MuJoCo Continuous Control (Dense Rewards):

$$\alpha = 0.8 \quad \text{(emphasize topological proximity)}$$
$$\delta = 0.1 \times \text{std}(V) \quad \text{(more tolerant value gate)}$$
$$\sigma = 0.3 \quad \text{(stricter distance gating, prevent jitter)}$$
$$\lambda = 0.8 \quad \text{(emphasize BC stability over stitching)}$$
$$H = 10 \quad \text{(long horizon for kinematic smoothness and gait consistency)}$$

E.3 DISTANCE GATE MECHANISM

The distance gate $w_t = \exp(-d_t/\sigma)$ modulates the contribution of the stitching loss. When the query state is close to the retrieved anchor, the stitching objective receives a larger weight; as the distance increases, its contribution is suppressed. The base BC loss remains active independently of the distance gate.

The combined loss is:

$$L_{\text{total}} = \underbrace{\|\hat{a}_{\text{base}} - a_t\|^2}_{\text{BC loss}} + \lambda \cdot \underbrace{\text{sg}(w_t)}_{\text{stop-gradient}} \cdot \underbrace{\|\hat{a}_{\text{stitch}} - a^*_{\text{stitch}}\|^2}_{\text{stitching loss}} \tag{14}$$

where $\text{sg}(\cdot)$ denotes stop-gradient; the distance weight is treated as a fixed coefficient during optimization.

E.4 HYPERPARAMETER TUNING GUIDELINES

When to Increase α (Relevance Balance):

- If stitching frequently fails due to kinematically infeasible candidates, increase α to emphasize topological proximity over advantage.
- Range: $\alpha \in [0.5, 2.0]$ for balanced task complexity.

When to Decrease σ (Distance Gate Temperature):

- If the agent exhibits high-frequency action jitter (especially in locomotion), decrease σ to apply stricter distance gating.
- For navigation tasks, relaxed $\sigma = 0.8$ is preferred; for continuous control, we use the more conservative setting $\sigma = 0.3$.

When to Adjust λ (Stitching Loss Weight):

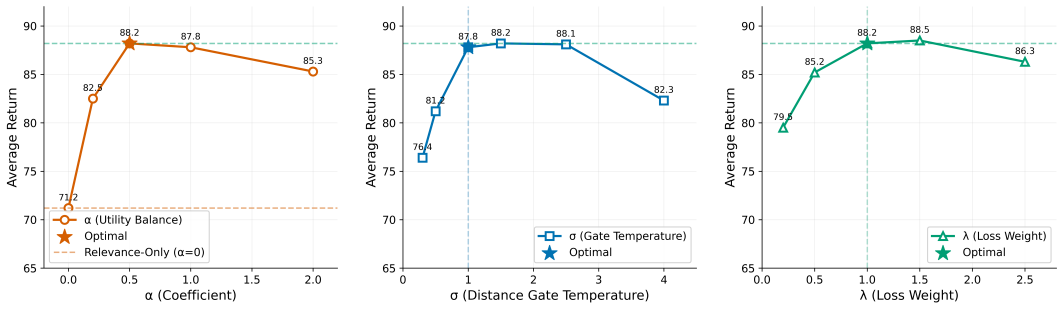
- If stitching does not improve performance, increase λ (more emphasis on stitching loss).
- If performance oscillates or becomes unstable, decrease λ (emphasize BC stability).
- Conservative default: a smaller $\lambda = 0.8$ places greater emphasis on the BC objective.

Fixed Parameters (from IQL):

- $\beta = 0.25$: fixed across tasks.
- $\delta = 0.05 \times \text{std}(V)$: Automatically set based on value function statistics.

E.5 Hyperparameter Robustness and Sensitivity Analysis.

To evaluate the stability of our method, we systematically sweep three critical hyperparameters individually on the `antmaze-medium-play` dataset: the utility balance coefficient α, the distance gate temperature σ, and the dual-loss weighting factor λ. Specifically, we vary α across a full spectrum of $[0.0, 2.0]$ to assess the exact trade-off between the similarity relevance term and the value feasibility gate in our utility function. The distance gate temperature σ is evaluated within a range of $[0.3, 4.0]$ aligned with the environment's spatial scale, testing the sensitivity of the distance-gating mechanism. Finally, the loss weighting coefficient λ, which dictates the balance between the base behavioral cloning objective and the guided stitching loss, is swept across a focused, meaningful range of $[0.2, 2.5]$.



(a) Sensitivity to α (b) Sensitivity to σ (c) Sensitivity to λ

Figure 4: Hyperparameter sensitivity analysis on the `antmaze-medium-play` dataset. The performance remains robust across a wide range of values for utility balance α, distance gate temperature σ, and loss weighting λ.

E.6 Sensitivity Analysis of Retrieval Scale (K and H)

To systematically evaluate the impact of the retrieval scale hyperparameters, we conduct our sensitivity analysis on two representative tasks: `antmaze-medium-play-v2` (representing sparse-reward topological navigation) and `hopper-medium-replay-v2` (representing dense-reward continuous control). We employ a control variable method for this analysis. When evaluating the sensitivity of K, we search across the range $K \in \{1, 3, 5, 7, 9, 11\}$ while fixing the horizon to its task-specific

optimal value ($H = 5$ for AntMaze and $H = 10$ for MuJoCo). Conversely, when evaluating the trajectory horizon H, we adopt an exponentially scaled search space $H \in \{1, 2, 5, 10, 20, 30\}$ to rigorously test both short-term lookahead capabilities and robustness to extreme long-horizon extraction, while fixing $K = 5$. All reported metrics are normalized average returns aggregated over 5 independent random seeds, consistent with the evaluation protocol in our main experiments. To further validate our design choices in Table 4, we conduct a comprehensive sensitivity analysis on the retrieval scale hyperparameters: the number of retrieved segments (K) and the trajectory horizon (H). The results are presented in Figure 5.

Sensitivity to Retrieval Scale (K). The top row of Figure 5 demonstrates the impact of K. Across both sparse-reward navigation (AntMaze) and continuous control (MuJoCo) tasks, performance follows an inverted U-shape, peaking consistently at $K = 5$. When K is too small (e.g., $K = 1$), the model lacks sufficient candidate diversity to discover optimal stitching paths. Conversely, overly large K values introduce irrelevant offline trajectories, causing attention dilution and context overload in the cross-attention module.

Robustness to Trajectory Horizon (H). The bottom row of Figure 5 illustrates the sensitivity to the horizon length H. As expected, the environments exhibit distinct optimal requirements: AntMaze peaks at a shorter horizon ($H = 5$) to facilitate agile local topological routing, whereas MuJoCo requires a longer horizon ($H = 10$) to maintain kinematic smoothness and gait consistency. Crucially, while extremely short horizons severely degrade performance due to insufficient lookahead information, **the model exhibits remarkable robustness when H exceeds the optimal length** (e.g., $H \geq 20$). We attribute this qualitative stability to our value feasibility gate based on $V(s)$. Because the retrieval utility is heavily anchored by the accurate value estimation at the initial state of the segment, the quality of the selected candidate remains topologically correct regardless of the extraction length, thereby preventing catastrophic performance drops.

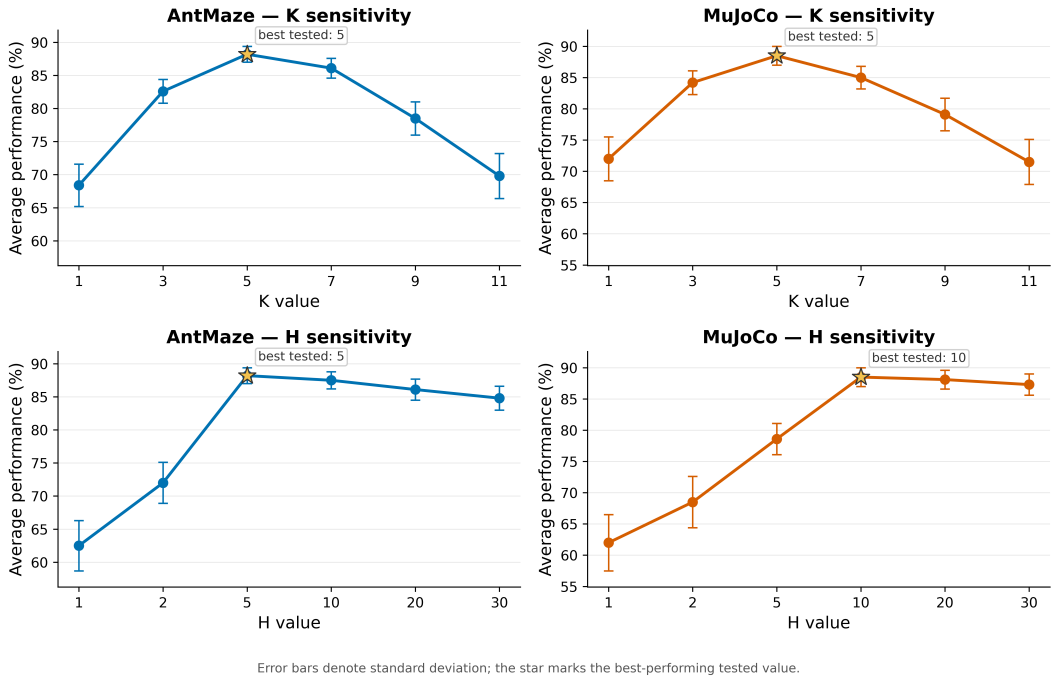


Figure 5: Sensitivity analysis of retrieval hyperparameters (K and H) across AntMaze and MuJoCo environments. Error bars denote the standard deviation over 5 seeds, and the star marks the optimal recommended setting.

F SYSTEM IMPLEMENTATION

A challenge in retrieval-augmented sequence modeling is the stark discrepancy between the training phase (which processes large batches of historical data) and the inference phase (which processes single, unobserved states in real-time). To achieve strict $O(1)$ scaling during training while maintaining ultra-low latency during deployment, ReStitch is implemented using a decoupled, *asymmetric* architecture.

Training Implementation: $O(1)$ Offline Caching. To circumvent the computational bottleneck of performing high-dimensional dense search within the autoregressive training loop, we decouple the retrieval pipeline from the representation learning. Prior to training, we execute a globally batched precomputation over the offline dataset $\mathcal{D}$. For every historical transition, we extract the latent feature $\phi_V(s)$, perform heuristic over-fetching, and apply the utility re-ranking (Eq. 2) to isolate the top-K candidates. These results are statically serialized to disk as memory-mapped arrays (e.g., `.npz` files), storing `candidate_indices` $\in \mathbb{R}^{N\times K}$ and their corresponding quality scores. During the actual training loop, the dataloader merely performs an $O(1)$ index array slicing to fetch the pre-assembled context tokens.

Inference Implementation: Fast Latent Anchor Matching. During online inference, the agent continuously encounters novel, out-of-distribution (OOD) states s_t^{env}. Re-evaluating the complex utility function (which requires computing advantage values $\bar{A}$ via the offline Q-network) for every novel state is not only computationally prohibitive but also prone to OOD value overestimation.

To achieve real-time continuous control, ReStitch bypasses online utility scoring entirely by employing a *Fast Latent Anchor Matching* strategy:

1. **State Projection:** The novel observation s_t^{env} is passed through the frozen IQL V-network to yield the topological feature $\phi_V(s_t^{env})$.
2. **1-NN Anchor Search:** We query the frozen FAISS index to find the single nearest neighbor anchor state s_{NN} from the offline dataset in the latent space.
3. **Zero-Shot Cache Inheritance:** Relying on the continuity of the ϕ_V topological space, we assume the optimal topological guidance for s_t^{env} closely matches that of its nearest historical anchor s_{NN}. We directly map to the index of s_{NN} and perform an $O(1)$ lookup in the precomputed `.npz` cache to inherit its top-K candidate trajectories and normalized scores.
4. **Context Assembly:** These inherited segments are embedded and injected into the DT's Cross-Attention modules, exactly mimicking the training behavior.

This asymmetric design provides value-gated topological guidance during training, while the online inference is reduced to a sub-millisecond FAISS query and a direct cache lookup, satisfying the latency requirements of real-world robotic control.

Computing Resources: All experiments were conducted on a high-performance computing cluster equipped with 8 NVIDIA RTX A6000 GPUs (48GB VRAM each).

End-to-end inference latency We profile `antmaze-medium-diverse-v2` with seed 0, batch size 1, evaluation context length 8, $K = 5$, retrieved horizon $H = 10$, and 998,998 cached segments on one NVIDIA RTX 6000 Ada 48GB GPU. After warm-up, latency is averaged over 300 repeated forward calls using the same implementation and device for all methods. The retrieval-path measurement includes value projection, GPU retrieval/FAISS operations, and cached-context access rather than only the nearest-neighbor search.

Table 5: System and Computational Overhead Profiling

Method	Latency / step	Retrieval / FAISS overhead	Offline cache construction	Cache / extra memory
DT	2.94 ms	0	0	0
IQL Q/V	0.24 ms	0	0	1.62 MB Q/V checkpoint
ReStitch	4.48 ms	2.76 ms GPU retrieval path	9.81 s stitch-memory initialization	86.47 MB disk cache / 2.61 GB GPU resident retrieval tensors

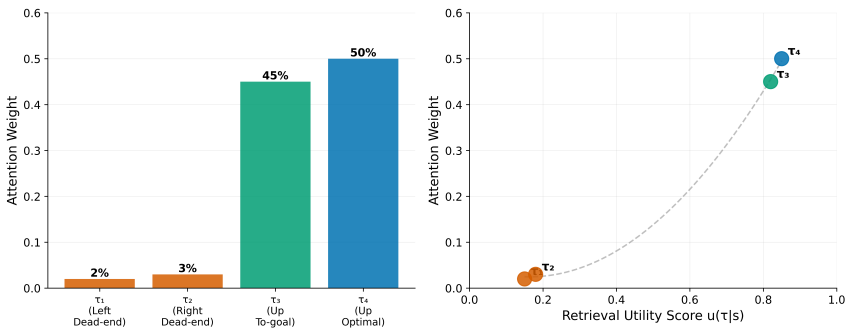


Figure 6: Attention Weight in Antmaze-Large

ReStitch increases end-to-end latency by 1.54 ms relative to DT (4.48 versus 2.94 ms), corresponding to approximately 223 model steps per second on the tested GPU. The 9.81 s stitch-memory initialization is a one-time cost and is not incurred per action. The main systems trade-off is therefore memory rather than latency: the cached index is only 86.47 MB on disk, while keeping the retrieval tensors resident requires 2.61 GB of GPU memory. These measurements show that ReStitch adds a measurable but modest per-step latency while avoiding online utility reranking.

G Mechanism Visualization: Attention Allocation during Stitching

To demystify the internal dynamics of the Cross-Attention fusion module, we visualize the attention weights during a critical decision point in the `antmaze-large` environment.

Figure 6 captures the agent at a T-junction. The retrieval module fetches 4 candidate trajectories: two leading to a dead-end and two leading towards the goal. The corresponding attention heatmap reveals that the model does not distribute its attention uniformly. Instead, guided by the Quality-Aware Initial Bias (w_{cand}), over 95% of the attention mass is precisely concentrated on the specific tokens of the single highest-value, physically feasible candidate. This visualization suggests that ReStitch learns to actively ignore suboptimal or unreachable context, successfully executing the intended "high-value shortcut."

H RA-DT under the Fixed-Task D4RL Setting.

RA-DT was originally developed for multi-task in-context adaptation, where training data from multiple tasks are combined and trajectories are associated with task identities. Our evaluation instead follows the standard fixed-task D4RL setting, in which each dataset is used to train a separate policy and no corresponding multi-task task identity is available.

To examine whether the released RA-DT implementation can be directly transferred to this setting, we assigned a constant task ID of 0 to all trajectories. This allows the implementation to run, but removes the task-dependent structure used in its original setting: the task embedding becomes constant and no cross-task information is available for adaptation. Under this modified setting, RA-DT obtains normalized returns of 3.3 ± 5.8 on antmaze-medium-play-v2 and 13.3 ± 5.8 on antmaze-medium-diverse-v2.

We emphasize that these results should not be interpreted as a faithful evaluation of RA-DT under its intended multi-task setting. Rather, they illustrate the difficulty of directly transferring its original task-conditioned retrieval mechanism to our fixed-task D4RL evaluation. Consequently, we do not include these numbers in the main comparison tables.