FORTY-THIRD INTERNATIONAL CONFERENCE ON MACHINE LEARNING · ICML 2026

QHyer: Q-conditioned Hybrid Attention-Mamba Transformer for Offline Goal-conditioned RL

Xing Lei[1] **Jincheng Wang**[2] **Xuetao Zhang**[1]* **Donglin Wang**[3]

[1]Xi'an Jiaotong University [2]University College London [3]Westlake University

Correspondence: xuetaozhang@xjtu.edu.cn

人工智能学院 College of Artificial Intelligence, XJTU

1 Background & Motivation

Offline GCRL. Learn a goal-reaching policy π(a | s, g) purely from a static dataset, with no environment interaction — a practical paradigm when online exploration is costly or unsafe.

The non-Markovian challenge. Many real datasets are gathered by *history-dependent* behavior policies $\beta(a_t \mid s_t, h_{t0})$. Value-based methods that assume Markovian transitions break down, while Decision Transformers (DT) fit such data via sequence modeling.

Yet DT has two limitations for GCRL:

1. **Return-to-go (RTG) is trajectory-dependent** — under sparse rewards it gives no signal for stitching promising segments out of failed trajectories.
2. **Fixed-window local extractors** (convolution in LSDT / DMixer) cannot adapt the effective memory to variable-length temporal dependencies.

2 Limitation 1 — RTG Fails Under Sparse Rewards

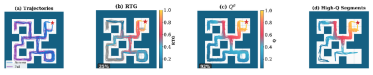

Fig. 2 RTG vs. NFs-based Q-value conditioning on non-Markovian AntMaze-medium. RTG leaves failed trajectories signal-less (25% coverage); Q^β colors every trajectory (92%), and high-Q segments form goal-reaching paths.

Root cause. RTG answers *"did this trajectory succeed?"* not *"how good is this state for the goal?"* The same state gets RTG=1 on a success and RTG=0 on a failure, making cross-trajectory comparison — and stitching — impossible.

Our fix. Replace RTG with a goal-reaching Q-value $Q^\beta(s,a,g)=p_+^\beta(g \mid s,a)$, measured *per state-action* and independent of which trajectory it came from — so high-value segments from failed demos can be identified and composed toward the goal.

3 QHyer Architecture

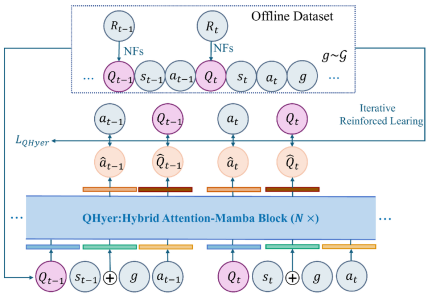

Fig. 1 QHyer replaces RTG tokens with NFs-estimated Q-tokens and processes the (Q, [cg], a) sequence with a gated Hybrid Attention-Mamba backbone (N×), trained by iterative reinforced learning.

Two *coupled* innovations: **(i)** a Normalizing-Flows (NFs) Q-estimator used **directly as conditioning tokens** (not an auxiliary loss), and **(ii)** a gated **Hybrid Attention-Mamba** backbone with content-adaptive history compression.

4 NFs-based Q-Value Conditioning

The goal-conditioned Q equals the discounted state-occupancy — the probability of reaching g:

$$Q^\pi(s,a,g) = p_+^\pi(s_+ = g \mid s_0 = s, a_0 = a)$$

Coupling-based NFs estimate it with an **exact, properly-normalized log-density**:

$$Q_\theta^\beta(s,a,g) = \log p_0(f_\theta(g;z)) + \log \left| \det \frac{\partial f_\theta(g;z)}{\partial g} \right|$$

Why NFs? The transformer reads Q-tokens spanning *multiple goals*, so goal-independent normalization is essential. CVAE (ELBO gap), contrastive RL (goal-dependent offset) and diffusion (ODE / trace-estimator variance) cannot provide it — coupling NFs uniquely can.

Expectile regression then extracts the in-distribution **maximum** Q (τ > 0.5 penalizes underestimation; τ → 1 → optimal Q):

$$\mathcal{L}_Q = \mathbb{E}_{(s,a,g)\sim\mathcal{D}}\left[L_\tau^2\big(Q_\theta^\beta(s_t,a_t,g) - \hat{Q}_\phi(s_t,g)\big)\right]$$

5 Hybrid Attention-Mamba Block

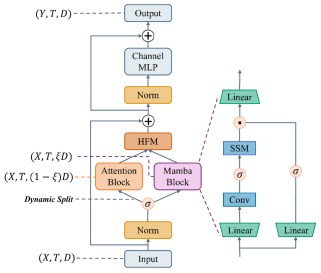

Fig. 3 A dynamic split routes capacity to an Attention branch (global goal-directed planning) and a Mamba branch (selective-SSM temporal dynamics), fused by a learnable gate (HFM).

Effective memory is set by **input-dependent** SSM dynamics, not a fixed kernel:

$$\bar{A}_t = \exp(\Delta_t \cdot A), \quad \Delta_t = \mathrm{softplus}(\mathrm{Linear}_\Delta(x'_t))$$

Small $\Delta_t \to \bar{A}_t \approx 1$ (long memory, *play*); large $\Delta_t \to \bar{A}_t \approx 0$ (short memory, *noisy*). A learnable gate fuses both branches: $y = \alpha \cdot y_{Attn} + (1-\alpha) \cdot y_{Mamba}$.

6 Content-Adaptive Memory

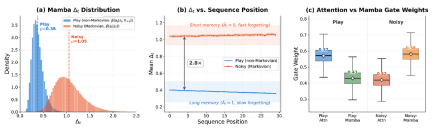

Fig. 4 On cube-single, the selective step Δ_t differs 2.8× between regimes and the gate flips accordingly — adaptation a fixed convolution cannot achieve.

2.8×	~12 → 3	0.57 / 0.58
Δ_t ratio (play vs. noisy)	effective memory (steps) play → noisy	gate — attention (play) gate — Mamba (noisy)

7 State-of-the-Art Results

SOTA on **both** non-Markovian (OGBench *play*, D4RL Maze) and Markovian (OGBench *noisy*) data.

Benchmark / setting	Metric	Best prior	QHyer
OGBench cube-play (total)	SR%	111 GCIQL	**152**
OGBench puzzle-play (total)	SR%	147 GCIQL	**169**
OGBench visual-scene-play	SR%	62 DTA	**96**
D4RL AntMaze-v2 (total)	score	378 IQL	**483**
D4RL Maze2D (total)	score	277 QT	**292**

Averaged over 8 seeds (OGBench) / 5 seeds (D4RL). Largest gains on big mazes that demand extensive stitching.

8 Ablations: Both Innovations Matter

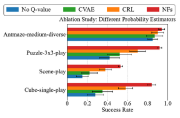

Fig. 5 Q-estimator

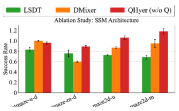

Fig. 6 Backbone (RTG-fixed)

Q-estimator: No-Q < CVAE < CRL < **NFs** — density-estimation accuracy tracks policy quality. **Backbone:** LSDT < DMixer < **QHyer**. The two contributions are **necessary and complementary** (additive gains).

9 Conclusion

- First sequence-modeling framework to **jointly** resolve trajectory-dependent conditioning and fixed-window memory for non-Markovian offline GCRL.
- **NFs Q-tokens** (exact density + expectile regression) enable trajectory stitching from failed demonstrations — no bootstrapping, no policy projection.
- **Hybrid Attention-Mamba** delivers content-adaptive memory, giving SOTA across OGBench manipulation and D4RL navigation.