Receding-Horizon Execution for Action Chunking in Offline-to-Online Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Action chunking improves offline-to-online reinforcement learning (RL) by using temporally extended actions for exploration and multi-step temporal-difference (TD) backups. However, a chunk length that accelerates value propagation can be too long for open-loop execution, since executing the full predicted chunk before replanning reduces reactivity and degrades performance. We study receding-horizon execution for Q-chunking policies with a fixed prediction horizon. We train a horizon-conditioned execution critic that evaluates prefixes of a predicted chunk under fixed and value-based execution rules. On a long-horizon manipulation task, receding-horizon execution substantially improves policies that fail under full open-loop execution but does not match a fixed short-horizon-trained actor or the best fixed execution horizon. Value-based horizon selection remains biased toward short horizons, and state-dependent selection remains an open problem.

1 Introduction

Offline-to-online reinforcement learning (RL) uses previously collected data to improve sample efficiency, but remains challenging in long-horizon, sparse-reward manipulation tasks where exploration is difficult [Li et al., 2023, Rengarajan et al., 2022]. In imitation learning, action-chunking policies predict multi-step action sequences rather than single actions, which helps capture temporally coherent and potentially non-Markovian behavior in demonstration data [Zhao et al., 2023, Black et al., 2026]. Q-chunking (QC-FQL) [Li et al., 2025] extends action chunking to RL by optimizing the policy and critic over the action sequences. This enables unbiased multi-step temporal-difference backups and temporally coherent exploration, leading to strong performance on long-horizon tasks.

However, executing a larger portion of a predicted chunk before replanning reduces the frequency with which new observations influence control [Zhao et al., 2023, Liu et al., 2025, So et al., 2026]. In Q-chunking, the chunk length also determines the backup horizon used by the critic. Thus, a horizon that improves value propagation can exceed the horizon that remains reliable under open-loop execution [Li et al., 2025, 2026, Song et al., 2026].

Recent methods address this trade-off in different ways. SEAR [Nagy et al., 2026] combines receding-horizon execution with multi-horizon critic targets and random replanning, but its evaluation policy still uses a fixed replanning interval. DQC [Li et al., 2026] instead decouples the critic chunk length from the policy chunk length, so that the critic can use long-horizon backups while the policy predicts shorter chunks. However, whether the execution horizon can be selected adaptively at run time rather than fixed remains an open question.

We study this question in the offline-to-online QC-FQL setting [Li et al., 2025]. We use multi-horizon targets and random replanning from SEAR [Nagy et al., 2026], with a selector-consistent bootstrap that uses the same selector at training and evaluation. We use this critic to study state-dependent execution-horizon selection rather than a fixed replanning interval. Empirically, receding-horizon

execution substantially improves a long prediction-horizon policy that fails under full open-loop execution, but does not reach the best short-horizon-trained actor. Value-based horizon selection remains biased toward short executions, and a simple marginal selector that compares neighboring horizons does not consistently outperform the best fixed horizon. Our results suggest that multi-horizon critics partially recover long-horizon Q-chunking actors, while reliable state-dependent horizon selection remains open.

2 Preliminaries

We consider the offline-to-online QC-FQL setting [Li et al., 2025]. The actor μ_ψ with parameters ψ is a flow-matching policy that maps a state s_t and a noise variable z_t to an action chunk,

$$A_t = \mu_\psi(s_t, z_t) \in \mathbb{R}^{H \times d_a}, \qquad z_t \sim \mathcal{N}(0, I), \tag{1}$$

where H is the prediction horizon and d_a is the action dimension. Following QC-FQL, the actor is trained by distilling a flow-matching behavior prior learned from the offline dataset, together with a Q-maximization term using the chunked critic.

Standard Q-chunking executes all H actions of A_t before querying the actor again, so the prediction horizon and the execution horizon coincide. We instead distinguish the prediction horizon H from the execution horizon $k \leq H$. Given a predicted action chunk A_t, the agent executes the first k actions, observes the state s_{t+k}, and replans by sampling a new chunk from the actor. In our experiments, $H = 50$ and we consider a sparse set of candidate execution horizons,

$$\mathcal{K} = \{5, 10, 15, 20, 25, 50\}. \tag{2}$$

For the $H = 50$ actor, full open-loop execution corresponds to $k = 50$, while receding-horizon execution uses $k < 50$.

3 Method

3.1 Multi-horizon execution critic

Inspired by SEAR [Nagy et al., 2026], we train a horizon-conditioned critic $Q_\phi(s_t, A_t, k)$ that estimates the return of executing the first k actions of A_t and then replanning. Unlike SEAR, our bootstrap is selector-consistent and our critic is an MLP masking actions after step k rather than a causal transformer. Q_ϕ is the ensemble-min over $M = 2$ critics (similarly $Q_{\bar{\phi}}$ for the target).

For the bootstrap value at the next replanning state, we sample $A' \sim \mu_\psi(\cdot \mid s_{t+k})$ from the current actor and apply the Q-min greedy selector (Section 3.2) at training time. The bootstrap value is

$$V(s_{t+k}) = Q_{\bar{\phi}}(s_{t+k}, A', k'), \tag{3}$$

where k' is the selected horizon and $\bar{\phi}$ denotes the target critic. We do not maintain a target actor. The multi-horizon target for horizon k is

$$G_t^{(k)} = \sum_{i=0}^{k-1} \gamma^i r_{t+i} + \gamma^k V(s_{t+k}), \tag{4}$$

and the critic is trained by averaging the squared temporal-difference error over candidate horizons,

$$\mathcal{L}_Q(\phi) = \mathbb{E}\left[\frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} \left(Q_\phi(s_t, A_t, k) - G_t^{(k)}\right)^2\right]. \tag{5}$$

The actor follows the QC-FQL objective, with the Q-maximization term using $Q_\phi(s, \mu_\psi(s, z), k_a)$ for k_a sampled uniformly from $\mathcal{K}$ at each gradient step. Algorithm 1 summarizes the procedure.

3.2 Selectors

Fixed-horizon evaluation chooses a constant k throughout the rollout and provides a strong reference within the same actor family. The Q-min greedy selector chooses

$$k(s, A) = \arg\max_{k' \in \mathcal{K}} Q_\phi(s, A, k'). \tag{6}$$

Algorithm 1 Adaptive-horizon QC-FQL training

Input: offline dataset $\mathcal{D}_{\text{off}}$, candidate horizons $\mathcal{K}$
Initialize actor μ_ψ, critic Q_ϕ, target critic $Q_{\bar{\phi}}$
for phase $\in$ {offline, online} **do**
  if phase is online **then**
    $\mathcal{D} \leftarrow \mathcal{D}_{\text{off}} \cup \mathcal{D}_{\text{buf}}$
  else $\mathcal{D} \leftarrow \mathcal{D}_{\text{off}}$
  end if
  for each gradient step **do**
    Sample $(s_t, A_t, r_{t:t+H-1}, \{s_{t+k}\}_{k\in\mathcal{K}})$ from $\mathcal{D}$
    Construct $G_t^{(k)}$ for each $k \in \mathcal{K}$ and update critic on $\mathcal{L}_Q(\phi)$
    Update actor with QC-FQL objective using $k_a \sim \text{Unif}(\mathcal{K})$
  end for
  if phase is online **then**
    Collect a rollout with $k \sim \text{Unif}(\mathcal{K})$ at each replanning step, appending to $\mathcal{D}_{\text{buf}}$
  end if
end for

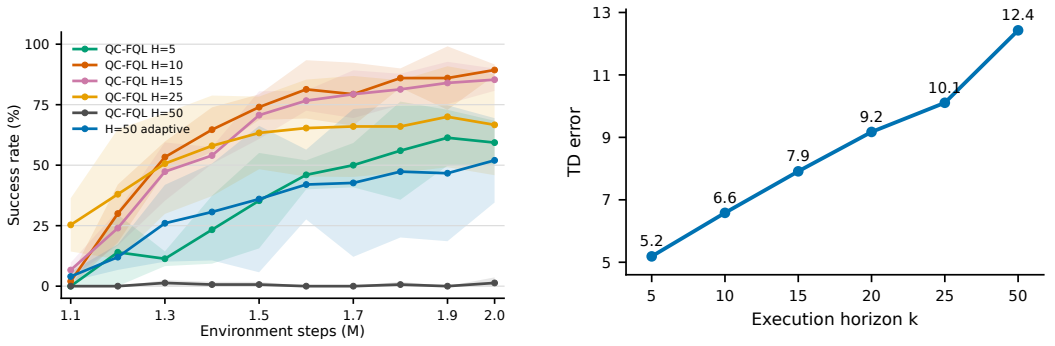
We also consider a marginal selector that compares neighboring horizons. Let k_i and k_{i-1} be consecutive elements of $\mathcal{K}$. The marginal improvement of extending from k_{i-1} to k_i is

$$\Delta_i(s, A) = Q_\phi(s, A, k_i) - Q_\phi(s, A, k_{i-1}). \tag{7}$$

The forward variant starts from the smallest candidate horizon $k = 5$ and extends to the next horizon as long as $\Delta_i \geq 0$, returning the previous horizon at the first negative gap. The reverse variant starts from $k = 50$ and shrinks to the previous horizon as long as $\Delta_i \leq 0$. We use the forward and reverse variants to test how the scan order affects the selected horizon.

4 Experiments

We evaluate on cube-triple task3 from OGBench [Park et al., 2025]. We first compare actors trained with different prediction horizons, and then freeze the final $H = 50$ multi-horizon checkpoint to compare execution rules with the same actor and critic. This separates prediction-horizon training from evaluation-time execution choices.



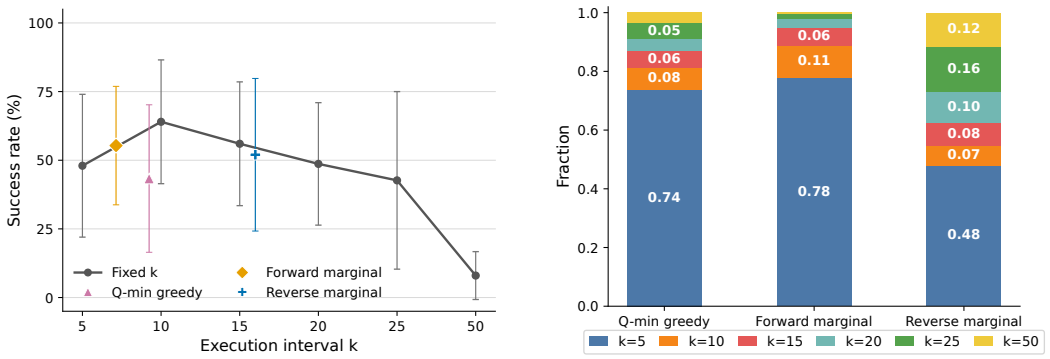
(a) (b)

Figure 1: (a) Online evaluation curves between 1.1M and 2M environment steps for fixed-H QC-FQL actors and the $H = 50$ multi-horizon actor evaluated with adaptive execution (over 3 seeds). (b) TD error $|Q_\phi(s_t, A_t, k) - G_t^{(k)}|$, averaged over rollout states.

Online learning across prediction horizons. The $H = 50$ actor under full open-loop execution stays near zero throughout online training (Figure 1a). The $H = 10$ fixed actor is the strongest baseline at about 89%, while our $H = 50$ adaptive policy reaches about 52%. Receding-horizon

execution improves the long-horizon actor relative to full open-loop execution, but does not match the best short-horizon-trained actor.

Horizon-dependent critic error. To test whether short-horizon selection reflects only execution reliability, we compute a post-hoc bootstrap TD error on fresh rollouts from the final $H = 50$ checkpoint, measuring the absolute difference between $Q_\phi(s_t, A_t, k)$ and the bootstrap target $G_t^{(k)}$ in Eq. 5. The TD error increases monotonically with k and more than doubles from $k = 5$ to $k = 50$ (Figure 1b). This rollout-based diagnostic is not replay-buffer validation, but it suggests that long-horizon targets are harder to fit and may confound an interpretation of short-horizon selection as a reliable execution signal.



(a) (b)

Figure 2: Selector evaluation at the final $H = 50$ multi-horizon checkpoint (mean over 3 seeds, 50 episodes per seed). (a) Success rate under fixed execution horizons (line) and adaptive selectors (markers). (b) Distribution of selected k for each adaptive selector.

Selector evaluation under frozen actor and critic. We freeze the final $H = 50$ multi-horizon checkpoint and reevaluate it under fixed-k execution and three adaptive selectors (Q-min greedy, forward marginal, reverse marginal), holding the actor and critic fixed. The best fixed horizon is $k = 10$ at about 64%, while $k = 50$ collapses to about 8% (Figure 2a), showing that execution length critically determines whether the same actor succeeds or fails. Adaptive selectors improve over $k = 50$ but consistently fall below the best fixed $k = 10$, with the marginal selectors closer to $k = 10$ than Q-min greedy.

Selected horizon distribution. Q-min greedy and forward marginal mostly select $k = 5$. Reverse marginal selects longer horizons much more often, including $k \geq 20$ (Figure 2b). However, this longer selection does not improve success and is slightly below forward marginal. The short-horizon bias of value-based selectors can be partially shifted by changing the comparison rule, but this shift alone does not close the gap to the best fixed horizon.

5 Discussion and Limitations

Our results show that the execution horizon has a large effect for action-chunking policies. Receding-horizon execution improves a long-horizon actor relative to full open-loop execution, but value-based selectors remain below both the best fixed execution horizon for the same checkpoint and the best short-horizon-trained actor.

The execute-then-replan critic estimates the return of executing k actions and replanning, so its value reflects both the quality of the current prefix and the benefit of an earlier decision point. We also observe that the post-hoc TD error grows with k, suggesting that long-horizon targets are harder to fit and may bias selection toward shorter horizons. Better calibrated horizon-conditioned critics and losses that balance candidate horizons are natural directions for future work.

References

Kevin Black, Manuel Y Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=UkR2zO5uww.

Qiyang Li, Jason Zhang, Dibya Ghosh, Amy Zhang, and Sergey Levine. Accelerating exploration with unlabeled prior data. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 67434–67458. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/d53d51e88d92d3723755f6d425bc513b-Paper-Conference.pdf.

Qiyang Li, Zhiyuan Zhou, and Sergey Levine. Reinforcement learning with action chunking. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=XUks1Y96NR.

Qiyang Li, Seohong Park, and Sergey Levine. Decoupled q-chunking. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=aqGNdZQL9l.

Yuejiang Liu, Jubayer Ibn Hamid, Annie Xie, Yoonho Lee, Maximilian Du, and Chelsea Finn. Bidirectional decoding: Improving action chunking via guided test-time sampling. In *International Conference on Learning Representations*, 2025.

C. F. Maximilian Nagy, Onur Celik, Emiliyan Gospodinov, Florian Seligmann, Weiran Liao, Aryan Kaushik, and Gerhard Neumann. Sear: Sample efficient action chunking reinforcement learning, 2026.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. In *International Conference on Learning Representations (ICLR)*, 2025.

Desik Rengarajan, Gargi Vaidya, Akshay Sarvesh, Dileep Kalathil, and Srinivas Shakkottai. Reinforcement learning with sparse rewards using guidance from offline demonstration. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=YJ1WzgMVsMt.

Junhyuk So, Chiwoong Lee, Shinyoung Lee, Jungseul Ok, and Eunhyeok Park. Improving generative behavior cloning via self-guidance and adaptive chunking. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=GctsZXLCpl.

Gwanwoo Song, Kwanyoung Park, and Youngwoon Lee. Chunk-guided q-learning, 2026. URL https://arxiv.org/abs/2603.13971.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware, 2023. URL https://arxiv.org/abs/2304.13705.