

행동 청킹 기반 강화학습을 위한 부분 실행 재계획 기법

이지환⁰, 이호준, 조건우, 임재균, 박기승, 김선동

GIST, University of Toronto

{jihwan.lee,hojun172,joungju257,jaegyun2}@gm.gist.ac.kr, giseung.park@utoronto.ca,

sundong@gist.ac.kr

Receding-Horizon Execution for Action Chunking in

Reinforcement Learning

Jihwan Lee⁰, Hojun Yi, Geonwoo Cho, Jaegyun Im, Giseung Park, Sundong Kim

GIST, University of Toronto

{jihwan.lee,hojun172,joungju257,jaegyun2}@gm.gist.ac.kr, giseung.park@utoronto.ca,

sundong@gist.ac.kr

요약

행동 청킹(Action chunking)은 에이전트가 여러 시점의 행동열을 한 번에 예측하게 하여 장기 과제에서 강화학습 성능을 높일 수 있지만, 예측된 긴 행동열을 재계획 없이 끝까지 실행하면 새로운 관측을 제어에 반영하는 빈도가 낮아질 수 있다. 본 논문은 긴 예측 길이로 학습된 Q-청킹 정책에서 예측 길이 H 와 실제 실행 길이 k 를 분리하고, 행동열의 일부만 실행한 뒤 다시 계획을 세우는 부분 실행 후 재계획 방식을 검토한다. 이를 위해 서로 다른 실행 길이를 평가하는 다중 실행 길이 크리틱을 사용하여, 정책이 생성한 행동열을 여러 실행 길이에서 비교하였다. 실험 결과, 전체 청크를 끝까지 실행할 때 성능이 낮던 정책도 짧은 앞부분 행동만 실행하면 성공률을 크게 회복할 수 있었다. 다만 상태에 따라 실행 길이를 자동으로 선택하는 경우 짧은 실행 길이에 치우치는 경향이 나타났으며, 이를 안정적으로 보정하는 것은 향후 과제로 남는다.

1. 서론

오프라인-온라인 강화학습은 사전에 수집된 데이터로 초기 정책과 가치 함수를 학습한 뒤, 이후 온라인 상호작용을 통해 샘플 효율성과 최종 성능을 높이는 방법론이다 [1]. 이 방법론은 보상이 희소하고 과제 달성까지 여러 단계의 조작이 필요한 로봇 과제에서 특히 어렵다. 탐색이 조금만 어긋나도 보상에 도달하지 못할 수 있고, 드물게 관측된 보상 신호를 초반의 행동까지 안정적으로 전파해야 하기 때문이다. 행동 청킹(Action Chunking)은 이러한 난점을 완화하기 위해 정책이 단일

행동이 아니라 여러 시점의 미래 행동열을 한 번에 예측한다 [2].

행동 청킹은 모방 학습 분야에서 먼저 활용되어 행동 데이터의 시간적 구조를 포착한 행동을 생성하는 데 사용되었다 [2]. 이후 Q-chunking은 이를 강화학습으로 확장하여 행동열을 하나의 확장된 행동으로 보고 정책과 가치 함수(Critic)를 함께 최적화하였다 [3]. 이를 통해 청크 내부의 실제 보상 합을 이용한 편향 없는 n-step 시간차 백업과 시간적으로 일관된 탐색이 가능해진다.

그러나 가치 학습에 유리한 긴 행동 청크가 실제 실행에도 항상 유리한 것은 아니다. 정책이 긴 행동열을 예측하고 이를 재계획 없이 끝까지 실행하면 새로운 관측이 제어에 반영되는 빈도가 낮아진다. 이는 예상하지 못한 관측에 대해 반응성 저하로 이어질 수 있다 [4].

최근에는 이러한 문제를 완화하기 위해 다중 길이 타킷과 무작위 재계획을 결합한 실행 길이 학습 [5],

* 본 연구는 IITP 디지털혁신기술 국제공동연구(RS-2024-00445087), ETRI 연구개발지원사업(RS-2023-00216011)에 의해 수행되었습니다.

크리틱이 사용하는 청크 길이와 정책이 예측하는 청크 길이를 분리하는 시도 [6]가 제안되었다. 이들 연구는 긴 청크가 가치 학습에는 유리하지만, 실제 제어에서는 실행 길이를 짧게 가져가는 것이 중요할 수 있음을 보여준다.

본 논문은 이러한 관찰을 바탕으로, 긴 예측 길이로 학습된 정책 안에 실제 제어에 유용한 짧은 행동 구간이 남아 있는지를 분석한다. 실행 길이를 조절하여 유용한 행동을 회복할 수 있는지, 나아가 상태에 따라 적응적으로 실행 길이를 선택할 수 있는지를 검토한다.

실험 결과, 전체 청크를 끝까지 실행할 때 거의 실패하던 긴 정책도 짧은 앞부분 행동만 실행하면 기존보다 높은 성공률을 보였다.

2. 문제 설정

본 연구는 상태 s_t , 행동 a_t , 보상 r_t , 할인율 γ 로 정의되는 마르코프 의사결정과정(Markov Decision Process, MDP)을 고려한다. 여기서 마르코프 의사결정과정은 현재 상태와 행동이 다음 상태 및 보상의 분포를 결정하는 순차 의사결정 모델이다. Flow-matching 기반 액터(Actor) μ_ψ 는 현재 상태 s_t 와 표준정규분포에서 샘플링한 노이즈 z_t 를 입력으로 받아 길이 H 의 행동 청크 A_t 를 출력한다 [7].

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

여기서 d_a 는 행동 차원이고, H 는 정책이 한 번에 예측하는 미래 행동의 개수이다. Q-chunking에서는 예측 길이와 실행 길이가 동일하다. 즉, 정책이 한 번 생성한 행동 청크 A_t 의 H 개 행동을 모두 순서대로 실행한 뒤에야 다음 상태에서 다시 정책을 호출한다.

본 논문은 예측 길이 H 와 실행 길이 k 를 분리한다. 정책은 여전히 $H=50$ 길이의 행동 청크를 생성하지만, 실제 환경에는 그중 앞부분 k 개 행동만 적용한다.

이후 상태 s_{t+k} 를 새로 관측하고, 해당 상태에서 다시 행동 청크를 샘플링하여 재계획한다.

본 연구에서 사용하는 후보 실행 길이 집합은 다음과 같다.

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

이때 $k=50$ 은 예측된 전체 청크를 재계획 없이 실행하는 개방루프(Open-Loop) 실행에 해당하고, $k < 50$ 은 일부 행동만 실행한 뒤 다시 계획을 세우는 부분 실행 후 재계획(Receding-Horizon Execution)에 해당한다.

이 설정은 학습과 실행의 역할을 분리한다. 예측 길이 H 는 크리틱이 긴 구간의 보상 누적을 학습하고 정책이 장기 행동 구조를 표현하는 범위를 결정한다. 반면 실행 길이 k 는 실제 제어 과정에서 새로운 관측을 다시 반영하는 빈도를 결정한다.

3. 다중 실행 길이 기반 재계획

3.1 다중 길이 크리틱 및 액터 학습

본 연구의 다중 실행 길이 크리틱 학습은 SEAR [5]의 다중 길이 타깃과 무작위 실행 길이 샘플링 설정을 기반으로 한다. 다중 실행 길이 크리틱 $Q_{\text{exec}}(s_t, A_t, k)$ 는 상태 s_t 에서 예측된 행동 청크 A_t 의 앞 k 개 행동을 실행한 뒤 재계획할 때의 기대 반환을 추정한다. 크리틱에는 전체 행동 청크가 입력되지만, 특정 k 를 평가할 때 k 이후의 행동이 목표값에 영향을 주지 않도록 마스킹을 적용한다.

다음 재계획 상태의 부트스트랩 값은 목표 액터가 생성한 새 행동 청크에 대해 후보 실행 길이 중 가장 큰 크리틱 값을 사용하여 계산한다.

$$G_t^{(k)} = \sum_{i=0}^{k-1} \gamma^i r_{t+i} + \gamma^k V(s_{t+k}) \quad (2)$$

따라서 크리틱은 모든 후보 실행 길이 k 에 대한 시간차 오차를 평균한 손실을 최소화하도록 학습된다.

액터는 QC-FQL[3] 목적함수를 따른다. 즉, 오프라인 데이터에서 학습한 플로우 매칭 행동 사전을 증류하는 손실과, 크리틱 값을 높이는 Q-최대화를 함께 사용한다. 이때 사용하는 실행 길이는 학습 단계마다 K 에서 균등하게 샘플링한다. 온라인 데이터 수집에서도 실행 길이를 균등 샘플링하여, 액터와 크리틱이 특정 재계획 간격에만 편향되지 않도록 한다.

3.2 실행 길이 선택 규칙

학습된 다중 실행 길이 크리틱은 평가 시 세 가지 방식으로 사용한다. 1) 고정 길이 방식은 에피소드 전체에서 하나의 길이만 선택하며, 동일한 $H=50$ 액터에서 실행 길이만 바꾸었을 때의 길이별 성능을 제공한다.

2) 본 실험에서는 두 개의 크리틱을 사용하여, 두 크리틱 값의 평균을 Q-mean, 둘 중 더 작은 값을 Q-min으로 둔다. Q-greedy 선택기는 후보 집합 K 의 모든 k 에 대해 크리틱 값을 계산한 뒤, 그 값이 가장 큰 실행 길이를 선택한다. 이때 식 (3)의 Q_{exec} 을 평균으로 두면 Qmean greedy, 더 작은 값으로 두면 Q-min greedy로 표기한다.

$$k^*(s, A) = \operatorname{argmax}_{k \in K} Q_{\text{exec}}(s_t, A_t, k) \quad (3)$$

다음으로 3) 증분 가치 선택기(Marginal)는 인접한 실행 길이의 차이를 비교한다.

$$\Delta_i(s, A) = Q_{\text{exec}}(s_t, A_t, k_i) - Q_{\text{exec}}(s_t, A_t, k_{i-1}) \quad (4)$$

실행 길이를 비교하며 $\Delta_i \geq 0$ 인 동안 실행 길이를 늘리고, 처음으로 $\Delta_i < 0$ 이 되는 지점의 직전 길이를 선택한다.

4. 실험

실험은 OGBench [8]의 cube triple task3에서 수행하였다. 이 환경은 세 개의 큐브를 목표 배치로 옮기는 로봇 조작 과제로, 장기적인 행동 계획과 상황에 따른 조정이 요구된다. 비교 대상은 두 가지 학습 방식이다. 첫째, H=50 baseline은 표준 QC-FQL 방식으로 학습된 정책으로, 평가 시 예측된 50개의 행동을 실행하는 기존 방식을 사용한다. 둘째, 제안 방법은 동일한 H=50 액터를 유지하되, 여러 실행 길이에 대한 가치를 평가하는 다중 실행 길이 크리틱을 함께 학습한다.

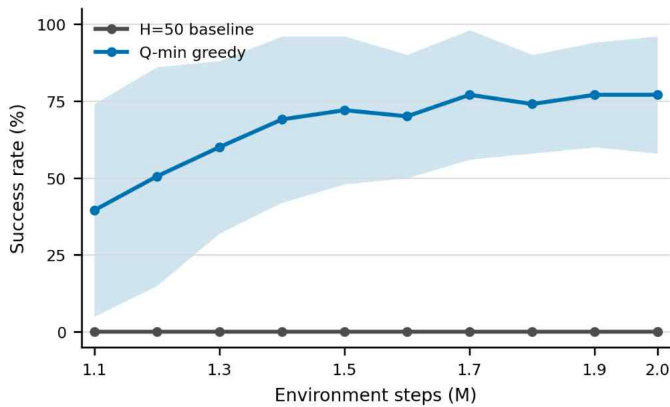


그림 1. 오프라인 사전학습 이후 온라인 학습 성능 비교

그림 1은 1M 오프라인 사전학습 이후 진행된 온라인 학습 1M 동안의 평가 성공률을 나타낸다. H=50 baseline은 예측된 50개 행동을 모두 재계획 없이 실행하는 방식으로, 온라인 학습 동안 거의 성공하지 못하였다. 반면 Q-min greedy는 학습이 진행되면서 성공률이 꾸준히 증가함을 확인할 수 있다. 이는 다중 실행 길이 크리틱을 통해 실행 길이를 조절하면 성능을 회복할 수 있음을 보여준다.

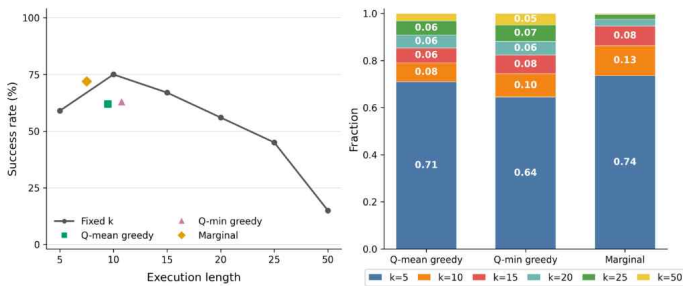


그림 2. 2M체크포인트에서 선택기에 따른 성능 비교

그림 2는 제안하는 방법으로 학습된 H=50 정책의 체크포인트를 고정한 뒤, 실행 길이 선택 방식만 바꾸어 재평가한 결과이다. Fixed k=10일 때 약 75%로 가장

높은 성공률을 보였고, 전체 청크를 실행하는 k=50은 약 15%였다. 그림 1의 H=50 baseline이 거의 0%였던 것과 비교하면, 다중 실행 길이 크리틱으로 학습한 액터는 동일한 k=50 평가에서도 일부 성능을 보임을 확인할 수 있다.

적응형 선택기들의 평균 선택 길이는 약 9로, 최적 고정 길이인 k=10에 가깝지만 일관되게 도달하지는 못한다. 오른쪽 그림은 적응형 선택기의 실행 길이 선택 비율을 나타낸다. 세 선택기 모두 k=5를 가장 자주 선택한다. 이는 가치 기반 선택이 짧은 실행 길이에 편향됨을 보여주며, 짧은 실행이 더 이른 재계획 기회를 제공하기 때문이다. 따라서 상태별 실행 길이 선택은 여전히 미해결 과제이다.

5. 논의 및 결론

본 논문은 긴 예측 길이로 학습된 Q-청킹 정책에서 예측 길이 H와 실행 길이 k를 분리하는 방식을 검토하였다. H=50 정책은 전체 청크를 한 번에 실행할 때는 거의 실패하였지만, 앞부분만 실행한 뒤 재계획하면 성능을 크게 회복할 수 있었다. 이는 긴 청크 정책의 실패가 행동열 자체의 문제만이 아니라, 실행 중 새로운 관측을 반영하지 못하는 문제와도 관련됨을 보여준다. 향후 과제로는 상태에 따라 적절한 k를 안정적으로 선택하는 방식과, H와 k의 적절한 조합을 찾는 문제가 남아있다.

참고문헌

- [1] T. Xie, N. Jiang, H. Wang, C. Xiong, Y. Bai, "Policy Finetuning: Bridging Sample-Efficient Offline and Online Reinforcement Learning," NeurIPS, 2021.
- [2] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware," arXiv preprint arXiv:2304.13705, 2023.
- [3] Q. Li, Z. Zhou, and S. Levine, "Reinforcement Learning with Action Chunking," NeurIPS, 2025.
- [4] Y. Liu, J. I. Hamid, A. Xie, Y. Lee, M. Du, and C. Finn, "Bidirectional Decoding: Improving Action Chunking via Guided Test-Time Sampling," ICLR, 2025.
- [5] C. F. M. Nagy, O. Celik, E. Gospodinov, F. Seligmann, W. Liao, A. Kaushik, and G. Neumann, "SEAR: Sample Efficient Action Chunking Reinforcement Learning," arXiv preprint arXiv:2603.01891, 2026.
- [6] Q. Li, S. Park, and S. Levine, "Decoupled Q-Chunking," ICLR, 2026.
- [7] S. Park, Q. Li, and S. Levine, "Flow Q-Learning," ICML, 2025.
- [8] S. Park, K. Frans, B. Eysenbach, and S. Levine, "OGBench: Benchmarking Offline Goal-Conditioned RL," ICLR, 2025.