프로젝트명 : Trajectory-level 우선순위 샘플링을 통한 Offline-to-Online 강화학습 안정화

팀원: 김지혜, 김정훈

요 약

본 프로젝트는 Flow Q-Learning(FQL)과 Action Chunking 을 결합한 offline-to-online 강화학습 구조에서, replay buffer 샘플링 전략이 학습 안정성과 성능에 미치는 영향을 분석한다. Offline 단계에서는 trajectory feature 기반 클러스터링을 통해 다양한 행동 모드를 보존하는 cluster-uniform 샘플링을 적용하고, online 단계에서는 trajectory-level TD-error 순위에 기반한 우선순위 샘플링으로 전환하는 단계별 replay 전략을 제안한다. 실험 결과, uniform sampling 만으로도 일정 성능은 가능하였으나, 장기·희소 보상(sparse reward) 과제에서는 성능 분산과 수렴 불안정이 관측되었다. 제안한 단계별 샘플링은 이러한 문제를 완화하고, 특히 가치 함수가 완전히 수렴되지 않은 과제에서 안정적인 value learning 과 정책 성능 향상에 기여함을 확인하였다.

1. 서론

강화학습(Reinforcement Learning, RL)은 환경과의 상호작용을 통해 누적 보상을 최대화하는 정책을 학습하는 일반적 프레임워크로, 로봇 조작과 같은 연속 제어 문제에서 강력한 성능을 보여왔다[6]. 그러나 실제 로봇 환경에서는 데이터 수집 비용과 안전 제약으로 인해 충분한 online 상호작용을 확보하기 어렵고, 이로 인해 이미 수집된 로그 데이터만으로 정책을 학습하는 Offline Reinforcement Learning 이 중요한 대안으로 부상하였다. Offline RL 은 고정된 데이터만을 이용하여 정책을 학습하므로 추가 상호작용 비용을 줄일 수 있으나, 데이터가 특정 행동 정책 π_β 하에서 수집되었다는 가정 때문에 학습 정책과 데이터 분포 간 불일치(distribution shift)가 구조적으로 발생하며, 이는 가치 함수의 과추정과 정책 붕괴로 이어질 수 있다[5].

이러한 한계를 완화하기 위해 최근에는 offline 데이터로 초기 정책·가치 함수를 안정적으로 형성한 뒤,

제한된 online 상호작용을 통해 성능을 향상시키는 Offline-to-Online RL 패러다임이 주목받고 있다. 특히 Flow Q-Learning(FQL)은 Flow Matching 을 활용해 multimodal 행동 분포를 모사하면서도 Q-learning 기반의 정책 개선을 수행하여 offline 및 online 구간을 연결하는 학습 동역학을 제시한다[1]. 또한 장기·다단계 조작 과제에서는 단일 step 행동 대신 길이 H의 행동 시퀀스를 하나의 의사결정 단위로 다루는 Action Chunking 이 정책 학습의 효율성과 가치 전파의 안정성을 개선할 수 있음이 보고되었다[2]. 본 프로젝트는 이러한 최근 연구 흐름을 기반으로, FQL 과 Action Chunking 을 결합한 학습 설정에서 replay buffer 의 샘플링 전략이 학습 안정성 및 성능에 미치는 영향을 체계적으로 분석하고, offline 단계와 online 단계에서 상이한 목적을 갖는 샘플링 설계를 제안한다.

2. 연구배경

2.1 Offline Reinforcement Learning 의 문제 설정과 핵심 난제

Offline RL 은 환경과의 추가 상호작용 없이, 미리 수집된 고정 데이터셋 $D = \{(s, a, r, s')\}$ 만을 이용해 정책 π 를 학습한다[5]. 이때 데이터는 행동 정책 π_β에 의해 생성되며, 학습 정책 π 가 π_β의 분포 밖 행동을 선택할 가능성이 높아진다. 이러한 분포 불일치는 Q-learning 계열의 bootstrapping 과 결합될 때 특히 치명적이며, 관측되지 않은 상태–행동 영역에서의 Q-value 과추정과 정책 붕괴(extrapolation error)로 이어질 수 있다[5]. 따라서 offline RL 의 핵심 과제는 고정 데이터에서의 학습 과정에서 발생하는 일반화 오류를 제어하면서 정책을 안정적으로 개선하는 데 있다[5,6].

또한 로봇 조작 데이터는 성공·실패, 서로 다른 전략, 부분적 목표 달성 등 다양한 궤적이 혼재된 mixture distribution 구조를 갖는 경우가 많다[5]. 이때 학습이 특정 모드에 과도하게 편향되면, 나머지 모드에 대한 가치 추정과 정책 일반화가 약화될 수 있으므로, offline RL 에서는 데이터가 포함한 behavior 다양성을 보존하는 설계가 중요한 전제 조건이 된다.

2.2 Offline-to-Online RL 의 동기: 데이터 효율성과 안전성의 절충

Offline-to-Online RL 은 offline 데이터로 초기 정책가치 함수를 학습하여 탐색 비용과 위험을 줄이고, 이후 제한된 online 상호작용으로 fine-tuning 을 수행해 성능을 향상시키는 접근이다[5]. Online 구간에서는 데이터 분포가 변화하며, 가치 함수는 정책 변화로 유도되는 새로운 상태–행동 분포에 적응해야 한다. 따라서 offline 학습으로 초기 근사를 확보한 뒤, online 단계에서 오류가 큰 영역을 교정하는 방식은 현실 로봇 문제에서 데이터 효율성과 안전성 측면에서 의미가 있다.

2.3 FQL 과 Action Chunking: 장기 조작 과제에서의 학습 구조

Flow Q-Learning(FQL)은 Flow matching 을 통해 행동 분포를 모델링하면서도 Q-learning 기반의 정책 개선을 수행하여, offline 및 online 구간을 연결하는 학습 동역학을 제시한다[1]. Action Chunking 은 길이 H의 행동 시퀀스를 하나의 의사결정 단위로 취급하여 장기 과제에서 신용할당 및 가치 전파를 개선하는 방향을 제시한다[2]. 이 설정에서 Q-함수는 chunk $a_{t:t+H-1}$ 에 대해 정의되며, 학습이 장기 행동 패턴 단위로 이루어진다[2].

2.4 Replay 우선순위의 위치: PER/PTR 관점에서의 재해석

Prioritized Experience Replay(PER)는 TD-error 가 큰 transition 을 더 자주 재사용함으로써 샘플 효율을 개선하는 기법으로, off-policy Q-learning 에서 널리 사용되어 왔다[4]. PER 의 핵심 가정은 (i) TD-error 가 "학습이 필요한 정도"를 근사하며, (ii) importance sampling 보정 등을 통해 transition 단위의 재가중이 minibatch SGD 기반 학습과 양립 가능하다는 점이다. 그러나 이러한 가정은 장기·희소 보상 환경이나, 단일 transition 이 아닌 행동 시퀀스를 의사결정 단위로 사용하는 학습 구조에서는 직접적으로 성립하기 어렵다.

Action Chunking 설정에서 Q-함수는 단일 상태–행동 쌍이 아닌 길이 (H)의 행동 시퀀스 $a_{\{t:t+H-1\}}$에 대해 정의되며[2], TD-error 역시 개별 step 이 아니라 chunk 단위의 예측 오차로 나타난다. 이로 인해 transition-level PER 는 장기 행동 구조를 충분히 반영하지 못하고, 실제로 학습이 필요한 trajectory segment 가 아니라 단기적 변동이 큰 일부 step 또는 chunk 에 과도하게 집중될 가능성이 높아진다. 특히 희소 보상 환경에서는 대부분의 transition 이 유사한 TD-error 를 갖거나, terminal 인접 구간에만 학습 신호가 집중되는 현상이 나타나 PER 의 효과가 제한될 수 있다.

Prioritized Trajectory Replay(PTR)는 이러한 한계를 보완하기 위해 sampling 단위를 transition 이 아닌 trajectory 로 확장하여 장기 구조를 직접 반영하려는 시도이다[3]. PTR 는 trajectory 단위로 데이터를 저장하고 trajectory 를 확률적으로 선택한 뒤, 선택된 trajectory 로부터 transition 들을 꺼내어 학습 배치를 구성한다. 이때 trajectory 우선순위는 trajectory-level TD-error 누적이 아니라, 평균 보상·상위 분위수 보상(UQM/UHM)·최소/최대 보상 등과 같은 trajectory quality 기반 통계 또는 Q-ensemble 분산으로 측정되는 trajectory uncertainty 기반 통계로 정의되며[3], 소수 trajectory 에 대한 과도한 집중을 방지하기 위해 priority 의 절대값 대신 rank 기반 확률 분포를 사용한다[3]. 이러한 trajectory-level 우선순위 관점은 장기·희소 보상 환경에서 "어떤 전이(transition)를 더 볼 것인가"가 아니라 "어떤 궤적(trajectory)을 더 자주 재사용할 것인가"를 통해 학습 신호의 구조를 제어한다는 점에서 PER 과 구별된다.

이러한 관점에서 replay 우선순위는 "학습을 가능하게 만드는 필수 요소"라기보다, offline–online 학습 단계에서 서로 다른 역할을 수행하는 보조적 제어 메커니즘으로 해석될 수 있다. Offline 단계에서는 데이터가 고정된 mixture distribution 을 이루고 TD-error 가 아직 신뢰할 수 있는 신호가 아니므로, PER/PTR 스타일의 우선순위는 오히려 behavior diversity 를 훼손할 위험이 있다. 반면 online 단계로 전환되면, 가치 함수가 일정 수준의 초기 근사를 확보한 상태에서 TD-error 는 "현재

가치 예측이 실패하고 있는 trajectory segment"를 식별하는 지표로 의미를 갖기 시작한다. 따라서 본 연구는 PER 의 "학습이 필요한 데이터의 재사용" 원리와 PTR 의 trajectory-level 관점을 계승하되, 이를 minibatch SGD 및 chunk-based Q-learning 구조에 정합적으로 재배치하여 단계별 목적(offline 다양성 보존, online 학습 필요 구간 강조)에 맞는 replay 샘플링 설계를 제안한다.

2.5 연구목표

본 프로젝트의 연구목표는 Flow Q-Learning(FQL) 과 Action Chunking 기반의 offline-to-online 강화학습 학습 구조에서, replay buffer 샘플링 전략(offline/online 단계별 샘플링) 이 학습 안정성과 성능에 미치는 영향을 분석하고, 이를 바탕으로 offline 단계(다양성 보존) 와 online 단계(학습 필요 구간 강조) 의 목적에 맞는 샘플링 설계를 제안·검증한다.

3. 배경지식 및 관련 연구

3.1) 기술 이론

MDP (markov decision process)

강화학습 문제는 일반적으로 마르코프 결정 과정(Markov Decision Process, MDP)으로 표현된다. MDP 는 상태 공간 S, 행동 공간 A, 초기 상태 분포 P0, 상태 전이 확률 $P(s'|s,a)$, 그리고 보상 함수 $R(s,a)$로 구성되며, 각 시점 t 에서 에이전트는 현재 상태 $s_t \in S$ 를 관측하고, 정책 $\pi(a_t \mid s_t)$ 에 따라 행동 $a_t \in A$를 선택한다. 이후 환경은 전이 확률 $P(s_{t+1}|s_t,a_t)$, 에 따라 다음 상태 s_{t+1} 로 이동시키고, 이에 대응하는 보상 $R(s_t,a_t)$ 를 에이전트에게 제공한다. 이러한 과정은 다음 상태의 분포가 과거 이력과 무관하게 현재 상태와 행동에 의해서만 결정된다는 마르코프 성질에 만족한다.

일반적으로 로봇조작 환경에서 MDP 가 주어졌을 때, 정책 π 는 초기 상태 분포 P_0와 전이 확률 P 에 의해 유도되는 상태–행동 쌍의 확률 분포를 정의한다. 강화학습의 목적은 할인 함수 $\gamma(\cdot)$에 의해 할인된 누적 보상의 기대값을 최대화하는 정책을 학습하는 것이며, 이는 다음과 같은 기대값 최적화 문제로 표현된다.

$$\mathcal{J}(\pi_\theta) = \mathbb{E}^{\pi_\theta, P_0}\left[\sum_{t\geq 0}\gamma(t)R(s_t,a_t)\right]$$

Offline reinforcement learning

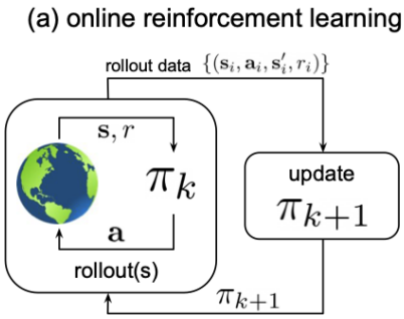


그림 1 online reinforcement learning Adapted from [8]

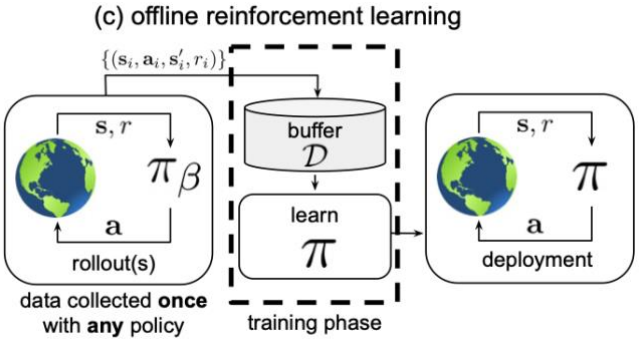
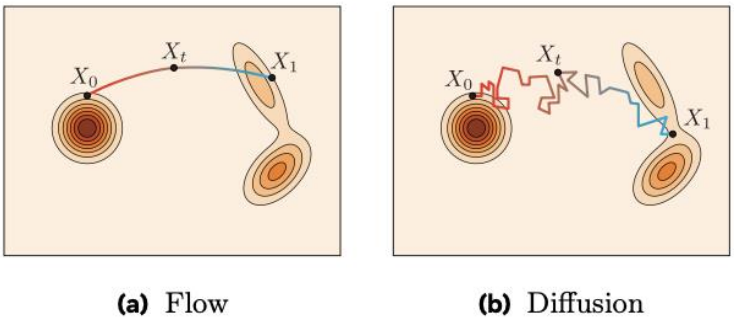


그림 2 offline reinforcement learning Adapted from [8]

일반적인 강화학습에서 사용하는 Online reinforcement learning 방식에서 정책 π_k 는 자신이 만들어낸 값을 통해 환경과 상호작용하여 그 다음 정책 π_{k+1} 를 만들어낸다. Offline reinforcement learning 은 환경과의 추가적인 상호작용 없이, 이미 수집되어 고정된 데이터만을 사용해 정책을 학습하는 방식이다. 이때 수집된 오프라인 데이터는 특정한 행동 정책 π_β 에 의해 수집되었다고 가정한다. 하지만 이로 인해 오프라인 강화학습은 학습하려는 정책과 데이터를 수집한 정책 사이의 분포 불일치 문제를 갖게 된다.

Flow-Matching



(a) Flow **(b)** Diffusion

그림 3. Flow 모델과 Diffusion 모델의 생성 과정 비교.

Diffusion 모델은 확률적 미분방정식(SDE)을 통해 점진적으로 노이즈를 제거하는 반면,

Flow 모델은 결정론적 ODE 를 통해 단일 경로 상에서 데이터 분포로 이동한다. Adapted from [7]

Flow matching 은 하나의 가우시안 분포를 실제 데이터 분포로 이동시키는 속도장(velocity field) 을 직접 학습시키는 모델링 기법이다. 속도장을 통해 값을 추정할때 확률적 미분방정식 (SDEs) 을 사용하는 diffusion 모델에 비해 flow 모델은 ordinary differential equation (ODEs) 방식을 사용하기 때문에 학습의 경로가 단순해지고 빠른 추론이 가능해진다. 또한, 데이터 분포 전반에 걸쳐 존재하는 여러 가능한 이동 경로와 방향을 속도장 이라는 연속적인 함수 형태로 학습하기 때문에 본질적으로 multimodal 한 분포를 학습할 수 있는 생성 방식이라는 점에서 표현력의 이점이 있다.

3.2) 기술 동향

Flow Q-learning (FQL)

Flow Q-Learning(FQL)은 flow-matching 을 이용해 오프라인 데이터에 포함된 다봉(multimodal) 행동 분포를 모델링하면서, 동시에 Q-learning 기반의 가치 극대화를 수행하는 offline-to-online 강화학습 알고리즘이다. 핵심적인 어려움은 flow policy 를 가치 함수로 직접 조정할 경우, ODE 기반의 반복적 생성 과정 전체에 대해 Q-loss 를 역전파해야 하므로 BPTT(backpropagation through time)가 필요해지고, 이는 계산 비용과 학습 불안정을 초래한다는 점이다.

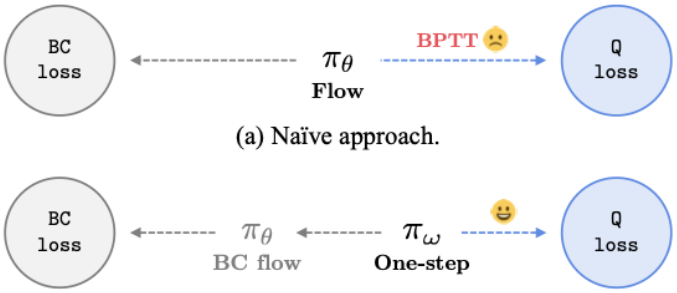


그림 3 Flow policy 를 Q-loss 로 직접 최적화할 경우 발생하는 BPTT 문제와,
이를 해결하기 위해 BC flow policy 와 one-step policy 를 분리한 FQL 의 아키텍처.
위: 단일 flow policy 에 Q-loss 를 적용할 경우 ODE 기반 생성 과정 전체에 대한
BPTT 가 필요함. 아래: one-step policy 를 통해 Q-loss 를 최적화하고,
BC flow policy 는 distillation 으로만 연결함으로써 BPTT 를 회피. Adapted from [1].

이를 해결하기 위해 FQL 은 정책을 두 개로 분리하는 아키텍처를 사용한다. 하나는 flow-matching 만으로 학습되는 BC flow policy 이고, 다른 하나는 단일 단계에서 행동을 생성하는 one-step policy 이다. One-step policy 는 Q 값을 최대화하도록 학습되며, 동시에 BC flow policy 의 출력을 따르도록 distillation loss 로 정규화된다. 이로써 가치 기반 정책 개선은 one-step policy 에서 수행되고, flow policy 는 Q-loss 와 분리되어 학습되므로 BPTT 없이 안정적인 최적화가 가능해진다.

이 distillation loss 는 단순한 모방 손실이 아니라, one-step policy 와 BC flow policy 가 유도하는 행동 분포 간의 2-Wasserstein 거리 상계로 해석될 수 있다. 이 Wasserstein regularization 의 핵심 역할은 flow policy 와 Q-loss 를 분리하여 BPTT 를 회피하는 데 있으며, 이를 통해 FQL 은 가치 기반 정책 개선을 one-step policy 에서 안정적으로 수행할 수 있다. 즉, Wasserstein regularization 은 FQL 에서 알고리즘적 안정성과 계산 가능성을 보장하는 구조적 요소로 작동한다.

$$L_{Distill}(\omega) = \mathbb{E}_{\substack{s\sim D \\ z\sim\mathcal{N}(0,I_d)}}\left[||\mu_\omega(s,z) - \mu_\theta(s,z)||_2^2\right]$$

$$L_\pi(\omega) = \mathbb{E}_{s\sim D, a^\pi\sim\pi_\omega}\left[-Q_\phi(s,a^\pi)\right] + \alpha L_{Distill}(\omega)$$

QC-FQL

QC-FQL 은 기존 Flow Q-Learning(FQL)에 action chunking 기법을 적용한 알고리즘으로, 매 시점마다 단일 행동을 예측하는 대신 길이 h의 행동 시퀀스를 하나의 temporally extended action 으로 예측하고 이를 Q-learning 의 기본 단위로 사용한다. 이를 통해 장기 시계열 의존성이 강하거나 희소 보상을 갖는 환경에서 보다 빠르고 안정적인 가치 전파가 가능해진다.

논문에서는 Q-learning 과 action chunking 을 결합한 이 학습 방식을 Q-chunking 이라 부르며, 이는 기존 Q-learning 이 갖는 구조적 한계를 완화하기 위한 설계로 제시된다. 표준 Q-learning 에서는 장기 보상 정보를 빠르게 전파하기 위해 n-step return 을 사용하지만, 이 경우 데이터가 현재 정책과

다를 때(off-policy) 가치 추정에 편향이 발생한다. 반면 Q-chunking 에서는 Q-함수를 단일 행동이 아닌 행동 시퀀스 전체에 대해 정의함으로써, n-step 보상을 생성한 동일한 행동 시퀀스를 가치 추정의 조건으로 포함시킨다. 그 결과, n-step return 을 사용하면서도 off-policy 편향이 발생하지 않는다는 장점을 갖는다.

QC-FQL 에서는 이러한 FQL 의 one-step/distillation 구조를 유지하되, Wasserstein regularization 의 역할이 action chunking 공간에서 재해석된다. Q-chunking 을 통해 n-step value backup 의 편향은 제거되지만, 길이 h의 행동 시퀀스 공간에서는 정책이 오프라인 데이터의 temporally coherent behavior 에서 이탈할 위험이 커진다. 이때 Wasserstein 거리 기반 distillation loss 는 chunked action distribution 이 오프라인 데이터 분포의 구조를 따르도록 제약하는 행동 정규화 항으로 작동한다. 결과적으로 QC-FQL 에서 Wasserstein regularization 은 BPTT 회피를 넘어, 장기 행동 시퀀스의 안정성과 일반화를 보장하는 학습적 제약으로 기능한다.

결과적으로 QC-FQL 은 (i) action chunking 을 통해 편향 없는 n-step value backup 을 수행하고, (ii) Wasserstein 기반 behavior constraint 를 통해 offline 데이터의 구조적 행동 분포를 보존하면서 online fine-tuning 을 가능하게 하는 통합적 학습 구조로 해석될 수 있다.

$$Q(s_t, a_t) \leftarrow r_t + \gamma Q(s_{t+1}, a_{t+1}) \quad \text{(standard 1-step TD)}$$

$$Q(s_t, a_t) \leftarrow \sum_{t'=t}^{t+h-1} \left[\gamma^{t'-t} r_{t'}\right] + \gamma^h Q(s_{t+h}, a_{t+h}) \quad \text{(n-step return, } n = h\text{)}$$

$$Q(s_t, a_{t:t+h}) \leftarrow \sum_{t'=t}^{t+h-1} \left[\gamma^{t'-t} r_{t'}\right] + \gamma^h Q(s_{t+h}, a_{t+h:t+2h}) \quad \text{(Q-chunking)}$$

QC-FQL Robomimic results

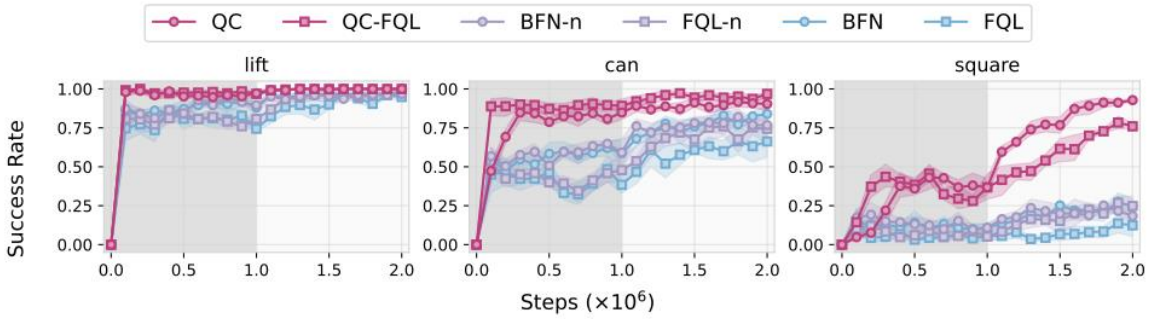


그림 4 Adapted from [2]

QC-FQL 을 사용하여 robomimic 환경에서 실험한 결과이다. Lift 와 can 은 7 DOF 를 가진 한개의 로봇팔만을 이용한 작업이고, square 는 두개의 로봇팔을 이용한 작업으로, lift 와 can 은 빠르게 성공률이 0.8 이상까지 올라가지만 square 는 느리게 성공률이 올라감을 볼 수 있다.

Prioritized Experience Replay

Prioritized experience replay 방법은 온라인 강화학습 에이전트가 데이터를 균일하게 샘플링하지 않고 중요도에 따라서 우선순위를 두는 방법이다. 기존 방법들은 경험을 통해 생성된 데이터들이 리플레이 메모리에 균일하게 저장되고, 다시 동일한 빈도로 사용되었다. 하지만 본 논문에서는 중요한 데이터들을 더 자주 사용함으로써 학습을 더 효율적으로 수행하는 방법을 제안한다. 해당 방식은 보상이 극도로 희소하여 대부분의 transition 이 실패인 환경에서 리플레이 데이터 선택의 순서만 바꿔도 학습 효율을 크게 증가시킬 수 있음을 보여준다.

우선순위 기반 재생의 핵심 요소는 각 데이터의 중요도를 어떻게 측정하는 것인가로, 이상적인 기준은 강화학습 에이전트가 현재 상태에서 해당 데이터로부터 얼마나 많이 배울 수 있는가를 측정하는 것이다. 이 기준으로 선택한 것은 현재 Q 추정값과 타깃값 사이의 TD 오류로, 해당 데이터가 얼마나 예상에서 벗어나는가를 측정한다. 이러한 TD 오류는 강화학습에서 기존에 계산되었기에 적용이 쉽고 적합하다.

하지만 우선순위 재생을 무조건 큰 값만을 사용하면 일부 데이터에만 과도하게 집중되고, 잡음과 근사 오차에 민감하며, 경험의 다양성이 부족해 과적합 위험이 있다는 문제가 있다. 이를 해결하기 위해 각 데이터가 우선순위에 비례하되 지정한 확률로 샘플링되도록 하는 확률적 우선순위 재생을 사용하여 학습 속도를 향상시킨다.

$$P(\tau_j) = \frac{p_{\tau_j}^{\alpha}}{\sum_k p_{\tau_k}^{\alpha}}$$

TD 오류가 클수록 샘플링 확률은 증가하며, 모든 Transition 은 0 이 아닌 확률로 선택된다. 파라미터 α를 통해서 greedy prioritization 과 uniform sampling 사이를 조절 가능하다.

ATARI Experiments

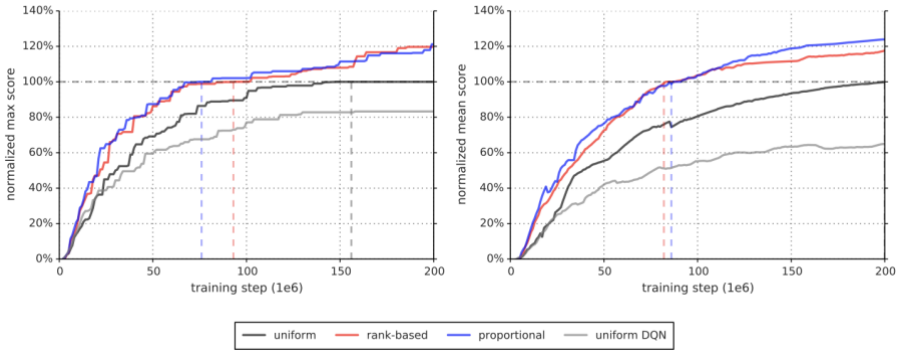


그림 5 Adapted from [4]

그림 5 는 Prioritized Experience Replay 를 DQN 모델을 적용했을때 Atari 에서 성능을 테스트 한 결과로, 전반적인 성능이 크게 향상되며 49 개 게임 중 41 개에서 균일 재생 대비 성능 개선을 보이고, 중앙값 기준 정규화 점수가 48% → 106%로 증가한다.

학습 속도 측면에서도 Prioritized Replay 는 동일 성능에 도달하는 데 필요한 학습 시간을 약 절반 수준으로 단축하며, 우선순위 재생이 보상이 희소한 게임에서 초기 성능 정체 구간을 줄이는 데 특히 효과적임을 관찰할 수 있다.

Prioritized Trajectory Replay

Prioritized Trajectory Replay 는 앞선 Prioritized Experience Replay 에서 한 발 더 나아가, 샘플링 관점을 개별 전이가 아닌 궤적(trajectory) 단위로 확장한 메모리 기법인 궤적 재생 방법 Trajectory Replay(TR/PTR)를 제안한다.

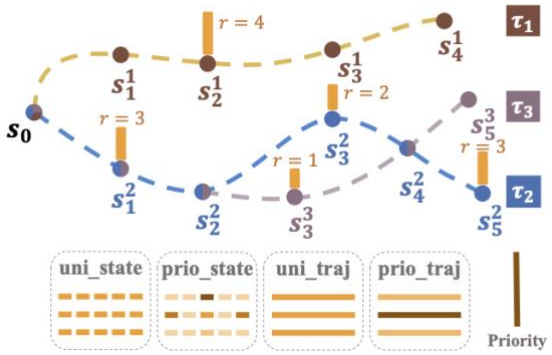


그림 6 Adapted from [5]

관찰된 바에 따르면 상태 전이를 직접 샘플링하는 방식은 균일 샘플링과 우선순위 샘플링 둘

모두에서 초기 상태 s_0 의 Q 값을 학습하는데 성능이 떨어지는 것으로 나타났다. 그 이유는 s_0 의 궤적에 연결된 중요한 상태 (예: s_1) 를 간과할 수 있기 때문이며, 궤적 단위로 샘플링을 해야지만 빠른 보상 신호 전파가 제대로 이루어질 수 있다는 점을 밝히고 있다.

오프라인 강화학습 알고리즘에서 궤적 관점의 데이터 샘플링 기법이 갖는 잠재력을 증명하기 위해 논문에서는 메모리 기법인 Prioritized Trajectory Replay (PTR) 과 이를 위해 궤적 단위로 데이터를 저장하고 샘플링하는 메모리 구조인 Trajectory Replay (TR) 를 구현한다.

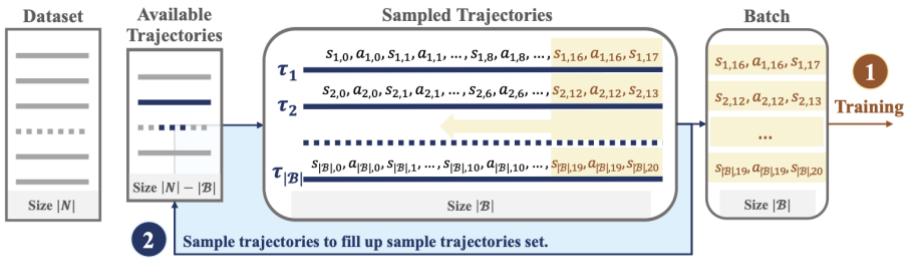


Figure 2. Overview of the process of data sampling based on Trajectory Replay.

그림 7 그림 8 Adapted from [5]

궤적 단위로 데이터를 저장하고 샘플링하는 Trajectory Replay(TR) 는 궤적 내 상태들을 역방향(backward) 으로 꺼내어 배치를 구성하고 학습에 사용한다. 이와 같은 역방향 샘플링은 후속 상태에서 얻은 정보와 장기 보상 신호를 보다 효율적으로 활용할 수 있도록 하여, 보상이 희소한 환경에서 보상 신호의 전달과 가치 함수 업데이트를 가속화한다.

$$P(\tau_j) = \frac{p_{\tau_j}^{\alpha}}{\sum_k p_{\tau_k}^{\alpha}}, s.t., p_{\tau_j} = \frac{1}{rank(pri(\tau_j))}$$

Prioritized Trajectory Replay(PTR) 은 이런 TR 의 궤적에 우선순위를 부여하여 샘플링한다. 궤적의 우선순위를 부여하는 기준은 두 가지인데, 궤적 품질과 궤적 불확실성이다. 궤적 품질은 해당 궤적이 의미 있는 보상 정보, 특히 높은 누적 보상이나 성공적인 행동 시퀀스를 포함하고 있는지를 반영하는 지표로, 희소 보상 환경에서 유용한 궤적을 선별할 수 있다. 반면 궤적 불확실성은 해당 궤적에 포함된 상태–행동 쌍에 대해 현재 Q 함수의 추정이 얼마나 신뢰 가능한지를 나타내는 척도로, 함수 근사

오차나 데이터 분포 외 영역에서 발생할 수 있는 잘못된 가치 추정을 완화하는 역할을 한다. 궤적 불확실성은 특히 밀집 보상 환경에서 과도한 추정 오류를 방지하고 안정적인 학습을 유도하는 데 중요하다. 또한 PTR 에서는 궤적 샘플링 확률을 계산할때 위의 수식과 같이 rank 를 기반으로 확률 분포를 선택한다. 이를 통해 일부 궤적의 과도한 사용을 배제하고 학습 안정성을 보장한다.

Task Name	TD3+BC	TD3+BC (TR)	EDAC	EDAC (TR)	IQL	IQL (TR)
Mujoco	**659.81**	630.54	**617.71**	546.44	**691.41**	669.99
Antmaze	98.36	**223.67**	-	-	329.90	**356.47**
Adroit	**540.41**	**545.95**	2.37	23.41	**545.70**	**543.52**

그림 9 Adapted from [5]

본 연구는 D4RL 벤치마크에서 기존 오프라인 강화학습 알고리즘에 TR 과 PTR 을 결합함으로써 얻을 수 있는 이점을 실험적으로 입증하였으며, 연구의 의의는 궤적 기반 데이터 샘플링 기법이 오프라인 강화학습 알고리즘의 효율성과 성능을 향상시키는 데 있어 중요한 역할을 한다는 점을 증명했다는 것이다.

3.2) 이슈 및 분석

본 연구의 학습 구조(FQL + Action Chunking + offline-to-online 전환)에서 관측된 핵심 이슈는 replay buffer 의 혼합 분포가 학습 신호를 불균형하게 만들 수 있다는 점이다. Offline 단계에서는 데이터가 고정되어 있고, 로봇 조작 데이터가 다양한 전략과 성공과 실패 패턴이 혼재된 mixture distribution 을 이루므로, 특정 성과 지표(보상 또는 초기 TD-error)에 기반한 우선순위 샘플링은 일부 trajectory subset 에 노출을 집중시켜 behavior diversity 를 훼손할 수 있다. 반면 online 단계에서는 replay buffer 가 offline 데이터와 online interaction 데이터의 혼합 분포가 되며, 이 시점의 병목은 탐색 자체보다 "현재 가치 함수가 크게 틀리는 trajectory segment 를 얼마나 효과적으로 식별·강조하는가"로 이동한다.

이러한 단계별 병목 차이로 인해, 동일한 replay 샘플링 규칙을 offline 과 online 에 일괄 적용하는 것은 비효율적일 수 있다. 특히 Action Chunking 설정에서는 TD-error 가 step 이 아니라 chunk 단위로

형성되며, minibatch 기반 학습에서는 terminal 인접 구간을 강제로 반복하는 backward consumption 이 batch 다양성을 감소시키고 학습 안정성을 저해할 수 있다. 또한 stored action 과 현재 정책 action 간의 분포 차이가 큰 설정에서는 SARSA-style target 이나 weighted target 이 요구하는 근접성 가정이 깨질 수 있어, 해당 구조를 그대로 채택하기 어렵다.

따라서 본 연구는 (i) offline 단계에서는 trajectory feature 기반 클러스터링을 통해 behavior diversity 를 보존하는 샘플링을 사용하고, (ii) online 단계에서는 trajectory-level TD-error 통계의 rank 기반 우선순위 샘플링으로 전환하되, backward consumption 과 SARSA-style target 은 배제하고 minibatch 기반 off-policy 업데이트와 정합적인 형태로 재구성한다. 또한 uniform sampling 만으로도 일정 성능은 가능하나, 장기·희소 보상 과제에서 성능 분산 및 수렴 불안정이 관측되어 이를 완화하는 방향으로 단계별 샘플링 전략을 도입하였다.

4. 프로젝트 내용

4-1. Trajectory-award Replay Sampling for Flow Q-Learning with Action Chunking

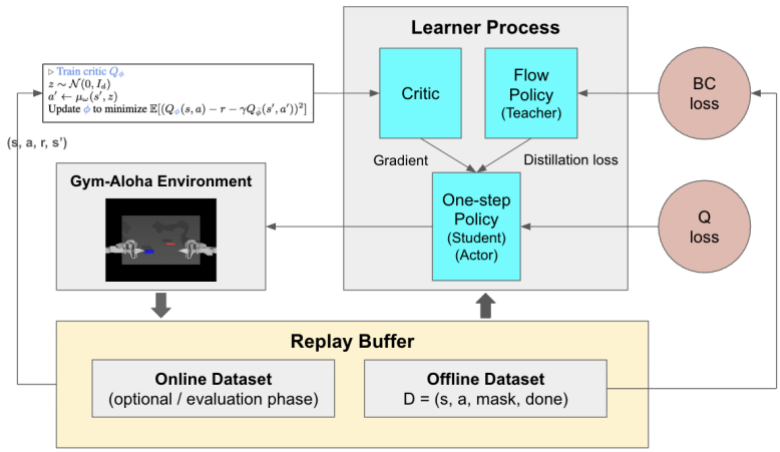


그림 10 FQL 아키텍처

본 연구는 FLow Q-Learning(FQL)과 Action Chunking 을 결합한 학습 프레임워크를 사용한다. 이 설정에서 정책과 가치 함수는 단일 행동이 아닌 길이 H의 행동 시퀀스에 대해 정의되며, Q함수는 다음과 같이 표현된다.

$$Q_\theta(s_t, a_{t:t+H-1})$$

이에 대응하는 H-step TD target 은

$$\hat{y} = \sum_{k=0}^{H-1} . \gamma^k Q_\theta(s_{t+H}, a_{t+H:t+2H-1})$$

로 주어진다. Action chunking 이론에 따르면, Q함수가 실제로 관측된 행동 시퀀스를 입력으로 받는 경우, 일반적인 n-step TD 와 달리 off-policy 편향이 발생하지 않는다. 이로 인해 replay buffer 로부터 어떤 시작 시점이 선택되는지가 곧 어떤 장기 행동 패턴과 TD 오차가 학습되는지를 직접적으로 결정하게 된다.

따라서 replay sampling 은 단순한 transition 문제가 아니라, 장기 행동 구조와 가치 전파 양상을 어떻게 제어할 것인가의 문제로 귀결된다.

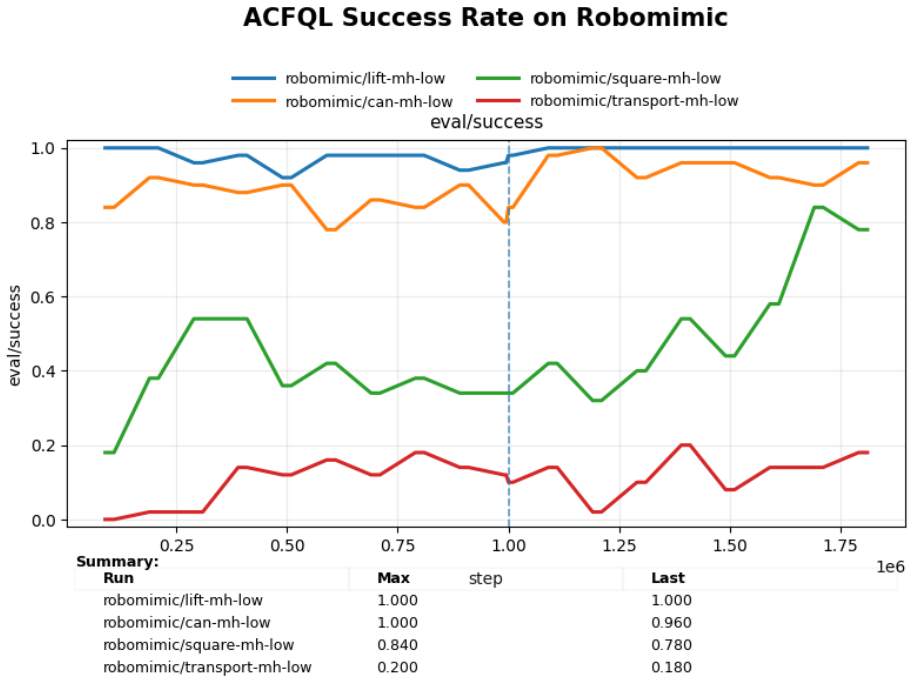


그림 11 Robomimic 에서의 ACFQL success rate

기존 ACFQL 알고리즘으로 robomimic 실험 결과는 위와 같다 task 별로 success rate 가 상이하지만 상대적으로 long horizon, sparse reward 환경을 갖춘 square, transport 환경에서 낮은 success rate 를 보인다. 각각 5 chunk 로 Offline 10^6 step, online 10^6 step 에서 실행되었으며, 최종적으로 replay 샘플링을 통하여 square 과 transport 의 성능을 높이는 것을 목표로 잡았다.

4-2. Offline sampling (cluster)

Offline 단계에서는 데이터가 고정되어 있으며, 정책은 환경과 상호작용하지 않는다. 이 단계에서의 목표는 다음과 같다.

1. offline 데이터에 존재하는 다양한 temporally coherent behavior mode 를 보존
2. flow 기반 정책이 다봉 행동 분포를 안정적으로 근사하도록 지원

Offline 데이터는 서로 다른 전략, 성공 및 실패 패턴, 서브태스크 조합이 혼재된 mixture distribution 을 이루는 경우가 많다. 이때 reward 또는 TD-error 기반 샘플링을 적용하면 특정 trajectory 모드가 과도하게 재사용되어 다른 모드의 학습이 저해될 수 있다.
이를 방지하기 위해 본 연구에서는 trajectory feature 공간에서 K-means clustering 을 적용하고, elbow method 를 통해 클러스터 수 K 를 선택한다. 각 trajectory τ는 클러스터 ID $c(\tau) \in \{1, \dots, K\}$에 할당된다.
Offline trajectory 샘플링 분포는 다음과 같이 정의된다.

$$p_{offline}(\tau) = \frac{1}{K} \cdot \frac{1}{|\tau_{c(\tau)}|}$$

Trajectory 가 선택된 이후, 해당 trajectory 내부에서 시작 시점은

$$s \sim Uniform\{b_\tau, \dots, e_\tau - H + 1\}$$

로 샘플링되며, 길이 H 의 행동 chunk 가 구성된다.
이 방식은 offline 단계에서 Q-chunking critic 이 다양한 장기 행동 시퀀스에 대해 가치를 학습하도록 유도하며, flow 정책이 소수의 고보상 trajectory 에 수렴하는 현상을 방지한다.

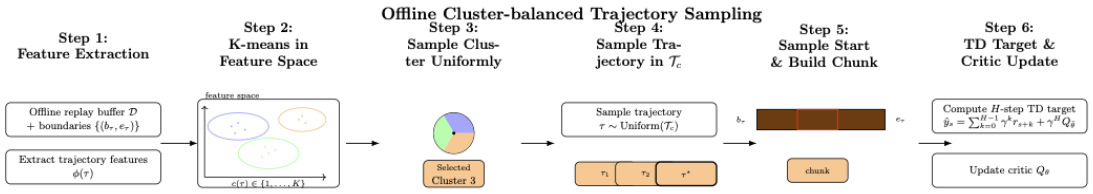


그림 12 Cluster sampling 순서

Offline step 에서 Cluster sampling 실험 결과 square 과 transport 의 success rate 의 max 값이 각각 0.06, 0.16 올랐으며 두 task 의 마지막 success rate 값도 기존 acfql 알고리즘보다 높아진 것을 확인할 수 있었다.

그림 13 cluster sampling success rate

4-3 .Online sampling (T-PER)

Online 단계에서는 replay buffer 가 offline 데이터와 online interaction 데이터의 혼합 분포를 갖는다. 이 시점에서 학습의 병목은 exploration 자체보다는, 현재 가치 함수가 가장 크게 틀리는 trajectory segment 를 식별하는 데 있다.
FQL 에서 critic 업데이트의 핵심 신호는 TD-error

$$\delta_t = Q_\theta(s_t, a_{t:t+H-1}) - \hat{y}_t$$

이며, 이는 장기 보상 예측이 잘못된 trajectory 에서 크게 나타난다. 이에 따라 본 연구에서는 trajectory 단위로 TD-error 통계를 집계한다.
각 trajectory τ에 대해 다음과 같은 지수 이동 평균을 유지한다.

$$\mu_\tau = EMA(|\delta|)$$

Trajectory-level priority 는 μ_τ의 순위에 기반하여 정의된다. Rank-based priority 분포는 다음과 같다.

$$p_{online}(\tau) = (1-\epsilon)\frac{(r_\tau + 1)^{-\alpha}}{\sum_{\tau'} \quad (r_{\tau'} + 1)^{-\alpha}} + \epsilon \frac{1}{|\tau|}$$

여기서 r_τ는 μ_τ의 내림차순 순위이다.
Trajectory 가 선택된 이후, offline 단계와 동일하게 trajectory 내부에서 시작 시점은 균등 샘플링되며 backward sampling 은 사용하지 않는다. Action chunking 설정에서는 이미 H-step value propagation 이 이루어지므로, terminal 인접 구간을 강제로 반복할 필요가 없으며, 이러한 편향은 오히려 minibatch 학습의 안정성을 저해한다.

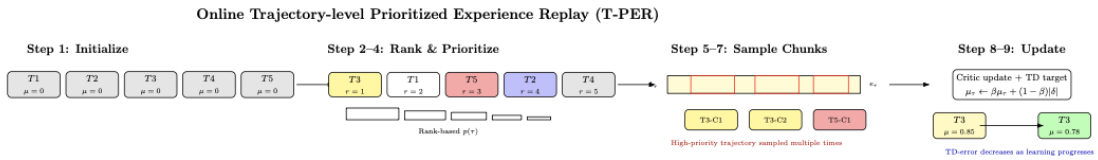


그림 14 TD-Error Sampling 순서

Online step 에서 TD-Error Sampling 실험 결과 square 과 transport 의 success rate 의 max 값이 각각 0.04, 0.22 올랐으며 두 task 의 마지막 success rate 값도 기존 acfql 알고리즘보다 높아진 것을 확인할 수 있었다.

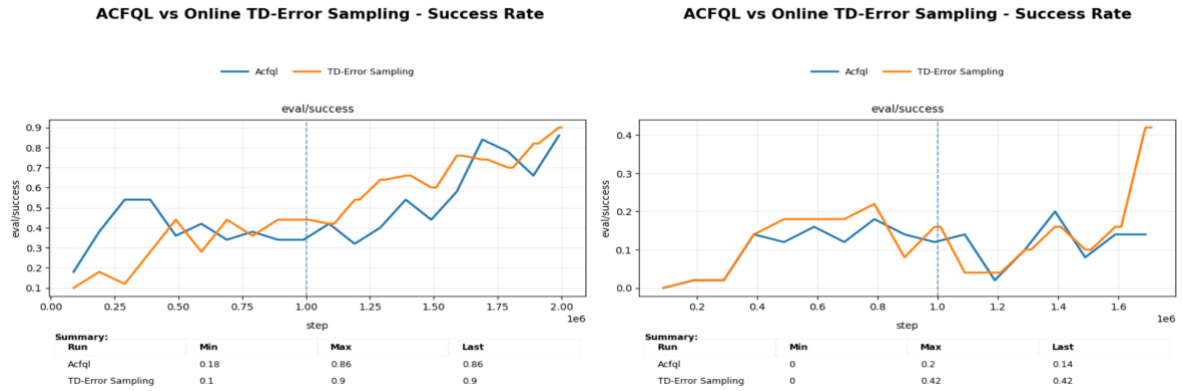


그림 15 TD-Error sampling success rate

4-4. 왜 Offline Sampler 와 Online Sampler 는 달라야 하는가

Offline 강화학습 단계에서 replay buffer 는 고정된 데이터 분포를 가지며, 정책은 환경과 상호작용하지 않는다. 이 설정에서 학습의 핵심 과제는 "어떤 trajectory 가 더 중요할 것인가"를 선택하는 것이 아니라, 데이터에 포함된 다양한 temporally coherent behavior mode 를 손실 없이 보존하는 것이다. 실제 로봇 조작 데이터는 단일 정책으로 생성된 분포가 아니라, 성공과 실패, 서로 다른 전략, 부분적 목표 달성 등으로 구성된 mixture distribution 의 형태를 갖는다. 이러한 상황에서 reward 또는 TD-error 기반 우선순위 샘플링을 offline 단계에 적용할 경우, 특정 trajectory subset 이 과도하게 반복되면서 다른 behavior mode 에 대한 노출이 급격히 감소하는 문제가 발생한다.

이에 본 연구에서는 trajectory feature 공간에서의 클러스터링을 통해 offline 데이터의 구조적 다양성을 명시적으로 모델링하고, 각 클러스터로부터 trajectory 를 균등하게 샘플링하는 cluster-uniform 전략을 채택하였다. 이 방식은 trajectory 의 성과나 길이에 관계없이 각 behavior mode 가 critic 학습에 동등하게 기여하도록 보장하며, action chunking 설정에서 다양한 장기 행동 시퀀스에 대한 Q-value 를 안정적으로 학습할 수 있게 한다. 특히 offline 단계에서는 TD-error 가 아직 신뢰할 수 있는 학습 신호가 아니며, reward 역시 희소하고 편향된 경우가 많기 때문에, cluster-uniform sampling 은 가장 정보 손실이 적고 가정이 최소화된 선택이 된다.

반면 online 단계로 전환되면 replay buffer 는 offline 데이터와 online interaction 데이터가 혼합된 분포를 갖게 되며, critic 은 이미 다양한 behavior mode 에 대한 초기 근사를 확보한 상태이다. 이 시점에서 학습의 병목은 더 이상 다양성 보존이 아니라, 현재 가치 함수가 어떤 trajectory segment 에서

가장 크게 틀리고 있는지를 식별하는 것으로 이동한다. Online 단계에서는 TD-error 가 실질적인 학습 신호로 작동하기 시작하며, trajectory-level TD-error 통계는 장기 행동 시퀀스 중 가치 전파가 실패한 구간을 효과적으로 드러낸다.

따라서 본 연구는 offline 단계에서는 cluster-uniform sampling 을 통해 behavior diversity 를 보존하고, online 단계에서는 TD-error rank 기반 우선순위 샘플링으로 전환하는 이원적 sampling 전략을 채택한다. 이러한 전환은 임의적인 설계 선택이 아니라, offline 과 online 단계에서 replay sampling 이 수행해야 할 역할이 근본적으로 다르다는 점에서 필연적이다. 실험 결과에서도 이와 같은 단계별 sampling 분리는 장기·희소 보상 환경에서 안정적인 value learning 과 정책 성능 향상으로 이어짐을 확인하였다.

4-5. Robomimic Square/Transport 과제에 따른 결과의 차이

Cluster-uniform offline sampling 과 TD-error–based online prioritization 을 결합했을 때의 효과는 과제의 학습 난이도와 value 수렴 특성에 따라 상이하게 나타난다. 본 연구에서는 동일한 sampling 전략을 적용했음에도 불구하고, Transport 와 Square 환경에서 offline 및 online 학습 동역학이 본질적으로 다름을 관측하였다.

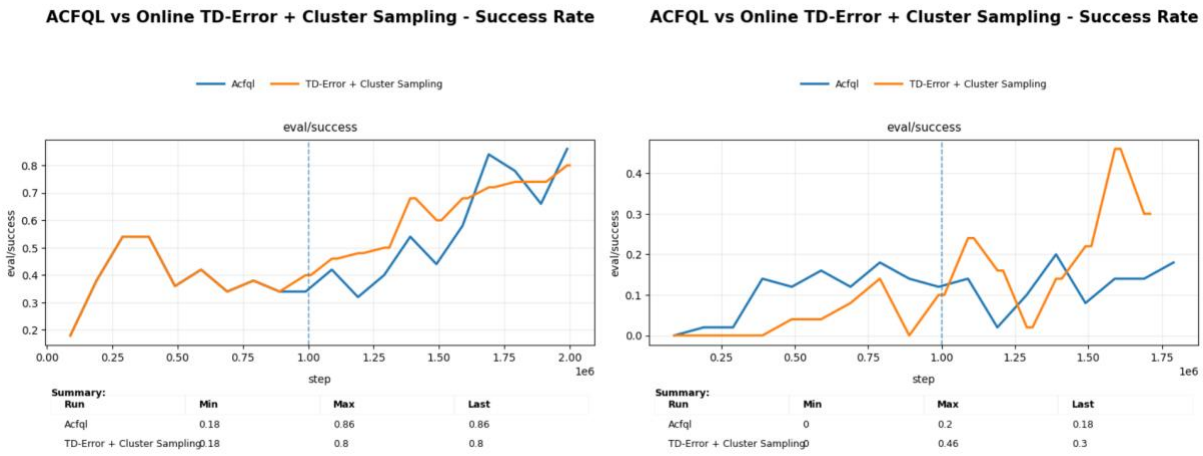


그림 16 Cluster Sampling 과 TD-Error Sampling 결합 Success rate

Offline 단계에서 TD-error 평균을 비교하면, Square 환경에서는 학습 초반 이후 TD-error 가 빠르게 0 근처로 수렴하며 이후 단계에서도 매우 낮은 수준을 유지한다. 이는 Square 과제가 비교적 짧은

horizon 과 밀집된 보상 구조를 가지며, offline 데이터만으로도 가치 함수가 조기에 안정화됨을 의미한다. 반면 Transport 환경에서는 offline 학습 전반에 걸쳐 TD-error 가 유의미하게 유지되며, 가치 전파가 완전히 수렴되지 않은 상태가 지속된다. 이는 Transport 가 장기·희소 보상 구조를 가지며, 동일한 offline 데이터에서도 value prediction 의 불확실성이 크게 남아 있음을 시사한다.

이러한 차이는 actor 가 평가한 Q-value 평균에서도 명확히 나타난다. Square 환경에서는 Q-value 가 학습 초기에 급격히 감소한 이후 완만하게 회복되지만, 전체적으로 낮은 값에서 정체된다. 이는 critic 이 이미 안정적인 value landscape 를 형성한 상태에서, 추가적인 sampling 제약이 정책 개선으로 이어지지 않음을 의미한다. 반면 Transport 환경에서는 Q-value 가 비교적 높은 수준을 유지하며 안정적으로 수렴하는데, 이는 critic 이 지속적으로 value 를 수정할 여지가 존재하고, actor 가 이를 활용해 정책을 개선할 수 있음을 나타낸다.

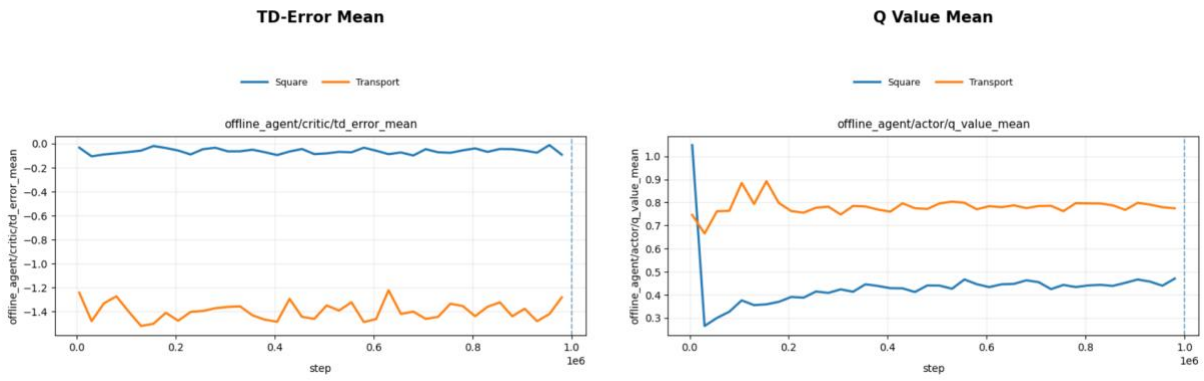


그림 17 Cluster Sampling 과 TD-Error Sampling 결합 결과 좌: TD-Error Mean, 우: Q Value Mean

이러한 상황에서 cluster-uniform offline sampling 과 TD-error rank 기반 online sampling 을 동시에 적용하면, Square 와 Transport 에서 상반된 효과가 나타난다. Square 에서는 offline 단계에서 이미 TD-error 가 거의 소실된 상태이므로, online 단계에서 TD-error 기반 우선순위 샘플링이 실질적인 학습 신호를 제공하지 못한다. 여기에 cluster-uniform sampling 까지 결합될 경우, replay distribution 이 불필요하게 제한되어 actor 가 관측할 수 있는 유효한 학습 신호가 감소하며, 결과적으로 성능이 소폭 저하된다. 이는 critic 의 안정성 자체가 문제가 아니라, 정책 업데이트에 필요한 정보량이 과도하게 줄어든 결과이다.

반면 Transport 환경에서는 offline 단계 이후에도 TD-error 가 유지되므로, online 단계에서 TD-error rank 기반 샘플링이 여전히 의미 있는 trajectory segment 를 강조한다. 이때 cluster-uniform sampling 은 replay buffer 가 특정 subset 에 과도하게 집중되는 것을 방지하는 역할을 하며, 두 sampling 전략은 상호 보완적으로 작용한다. 그 결과 value function 은 점진적으로 정교화되고, actor 는 안정적인 Q-value 증가를 기반으로 정책 성능을 향상시킨다.

요약하면, combined offline–online sampling 전략의 효과는 trajectory 의 구조가 아니라, offline 단계에서 TD-error 가 얼마나 빠르게 소실되는지, 그리고 value landscape 가 얼마나 조기에 안정화되는지에 의해 결정된다. 이는 replay sampling 전략이 task 의 학습 난이도와 value 수렴 특성에 따라 조절되어야 함을 시사한다.

5. Ablation

5-1. cluster vs uniform

Offline 단계에서 본 연구는 replay buffer 에 포함된 trajectory 들의 구조적 다양성(behavior diversity)을 보존하기 위해 trajectory feature 공간에서 클러스터링 기반 샘플링을 적용하였다. 각 trajectory 는 관측·행동 시퀀스로부터 추출된 feature 로 표현되며, PCA 이후 K-means clustering 을 수행하였다. 클러스터 수 K는 elbow method 로 결정하였다.

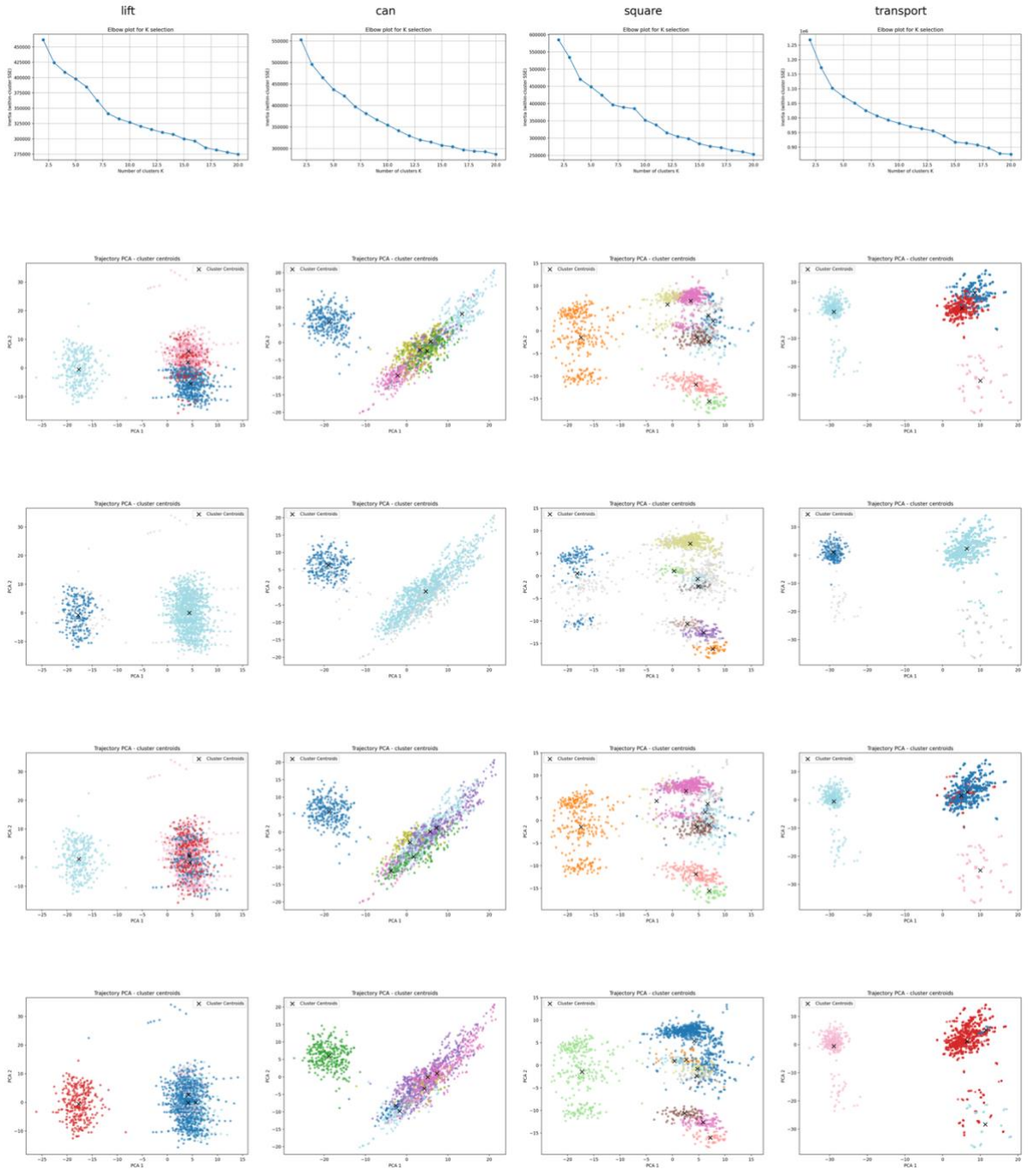


그림 18 Robomimic task(lift, can, square, transport)별 trajectory feature 를 PCA 로 2 차원으로 축소한 뒤 수행한 클러스터링 결과 시각화. 각 열은 task 를, 각 행은 서로 다른 클러스터링 방법을 나타내며, 위에서부터 순서대로 K-means, HDBSCAN, Gaussian Mixture Model(GMM), Spectral Clustering 의 결과를 보여준다.

Offline 학습에서는 각 클러스터로부터 trajectory 를 균등 샘플링(cluster-uniform) 하며, trajectory τ의 샘플링 확률은 다음과 같이 정의된다.

$$p_{offline}(\tau) = \frac{1}{K} \cdot \frac{1}{|\tau_{c(\tau)}|}$$

여기서 $c(\tau)$)는 trajectory τ가 속한 클러스터이며, $|\tau_{c(\tau)}|$는 해당 클러스터의 trajectory 개수이다. 이 방식은 reward 크기나 trajectory 길이에 의해 특정 behavior mode 가 과도하게 반복되는 현상을 방지하고, action chunking 설정에서 다양한 장기 행동 시퀀스가 critic 학습에 고르게 노출되도록 한다.

실험 결과, cluster-uniform sampling 은 offline 단계에서 Q-value 추정을 안정화하고, 이후 online fine-tuning 에서의 성능 향상으로 이어짐을 확인하였다.

표 1 Notation and Configuration

Symbol	Description
$\mathcal{D}$	Replay buffer containing offline and online trajectories
τ	Trajectory (episode) index
(b_τ, e_τ)	Start and end indices of trajectory τ
H	Action chunk length
γ	Discount factor
$a_{t:t+H-1}$	H-step action chunk
Q_θ	Flow-based Q-function over action chunks
$\hat{y}_t$	H-step TD target for chunk starting at t
δ_t	TD-error at chunk start index t
μ_τ	EMA of absolute TD-error for trajectory τ
r_τ	Rank of μ_τ among all trajectories
α	Rank-based priority exponent
ε	Uniform exploration mixing coefficient
K	Number of trajectory clusters (offline phase)
$c(\tau)$	Cluster assignment of trajectory τ

Algorithm 1 Offline Cluster-balanced Trajectory Sampling

Require: Offline replay buffer $\mathcal{D}$
Require: Trajectory boundaries $\{(b_\tau, e_\tau)\}$
Require: Chunk length H, discount factor γ
Require: Number of clusters K

Extract trajectory-level features
Perform K-means clustering with K clusters
Assign each trajectory τ to a cluster $c(\tau) \in \{1, \ldots, K\}$
for each offline training iteration **do**
    Sample cluster

$$c \sim \text{Uniform}(\{1, \ldots, K\})$$

    Sample trajectory

$$\tau \sim \text{Uniform}(\mathcal{T}_c)$$

    Sample start index

$$s \sim \text{Uniform}\{b_\tau, \ldots, e_\tau - H + 1\}$$

    Construct action chunk $a_{s:s+H-1}$
    Compute H-step TD target

$$\hat{y}_s = \sum_{k=0}^{H-1} \gamma^k r_{s+k} + \gamma^H Q_{\bar{\theta}}(s_{s+H}, a_{s+H:s+2H-1})$$

    Update critic Q_θ via minibatch SGD
end for

알고리즘 1 Offline Cluster-balanced Trajectory Sampling

5-2. Reward based vs ranked based

본 연구에서는 trajectory 수준의 보상 통계를 이용한 우선순위 리플레이 샘플링의 한계를 분석한다. 각 trajectory τ에 대해 평균 보상(AvgReturn) 또는 상위 분위 평균 보상(UQMReturn)을 계산하고, 이를 정규화하여 샘플링 확률로 직접 사용하는 방식을 고려한다.

$$p_{reward}(\tau) \propto Normalize(R(\tau))$$

여기서 $R(\tau)$는 trajectory 전체에 대한 스칼라 보상 통계량이다.

그러나 장기·희소 보상 환경에서는 대부분의 trajectory 가 거의 동일한 낮은 보상을 가지며, 극히 일부만 큰 보상을 갖는다. 이때 정규화는 스케일만 조정할 뿐 trajectory 간 확률 비율의 폭주(skewness) 를 해소하지 못하므로, 샘플링 확률이 소수의 trajectory 에 과도하게 집중된다. 또한 reward 기반 우선순위는 trajectory 내부의 시간적 구조(중간 실패 구간, 부분적 오류 패턴)를 제거한 채 전체를 단일 스칼라로 압축하므로, replay buffer 가 이미 충분히 학습된 trajectory 를 반복적으로 재사용하게 된다.

이 과정에서 critic 의 TD-error 는 빠르게 0 으로 수렴할 수 있으나, 실제로 학습이 필요한 구간에 대한

신호는 소실된다. 더 나아가 편향된 trajectory 집합에 대해 학습된 critic 은 해당 행동 시퀀스에 과도하게 높은 Q 값을 추정할 수 있고, actor 는 이를 신뢰하여 정책을 극단적으로 업데이트하게 된다. 실제 실험에서 reward 기반 정규화 우선순위는 손실이 낮게 유지되더라도 online 단계에서 성공률이 0 으로 수렴하는 현상을 보였다.

그림 19 AvgReward/UQMReward 기반 우선순위와 rank-based reward 우선순위의 online 학습 성능 비교.

Reward 기반 정규화 우선순위는 성공률 붕괴를 유발하는 반면, rank-based 방식은 안정적인 학습을 유지함.

5-3. reward-ranked-based vs TD-error-ranked-based

Reward 기반 우선순위의 한계는 trajectory 의 성과와 학습 필요성이 반드시 일치하지 않는다는 점에서

기인한다. Action chunking 설정에서 Q-함수는 길이 H의 행동 시퀀스에 대해 다음과 같이 정의된다.

$$Q_\theta(s_t, a_{t:t+H-1})$$

이때 핵심은 "보상이 컸던 trajectory"가 아니라, 현재 가치 함수가 장기 행동 시퀀스 중 어느 구간에서 잘못된 예측을 하고 있는가를 식별하는 것이다.

이에 본 연구에서는 TD-error 를 trajectory-level 학습 신호로 사용한다. chunk-level TD-error 는

$$\delta_t = Q_\theta(s_t, a_{t:t+H-1}) - \hat{y}_t$$

로 계산되며, 각 trajectory 에 대해 $|\delta|$의 지수 이동 평균을 유지한다.

$$\mu_\tau = EMA(|\delta|)$$

Trajectory-level priority 는 μ_τ의 순위(rank) 에 기반하여 정의되며, 최종 샘플링 분포는 다음과 같다.

$$p_{online}(\tau) = (1-\epsilon)\frac{(r_\tau + 1)^{-\alpha}}{\sum_{\tau'} \quad (r_{\tau'} + 1)^{-\alpha}} + \epsilon\frac{1}{|\tau|}$$

여기서 r_τ 는 μ_τ의 내림차순 순위이다. Rank-based 정의는 TD-error 의 절대 크기에 의존하지 않으면서도 학습이 필요한 trajectory 를 안정적으로 강조하며, 우선순위 분포의 과도한 집중을 방지한다.

실험 결과에서 TD-error rank 는 trajectory 성공 비율을 점진적으로 증가시키는 동시에 priority 분포의 분산을 유지하여 critic 과 actor 의 안정적인 동시 수렴을 가능하게 하였다.

Algorithm 2 Online Trajectory-level Prioritized Experience Replay (T-PER)

Require: Replay buffer $\mathcal{D}$
Require: Trajectory boundaries $\{(b_\tau, e_\tau)\}$
Require: Chunk length H, discount factor γ
Require: Priority parameters α, ε

Initialize trajectory TD-error statistics

$$\mu_\tau \leftarrow 0 \quad \forall \tau$$

for each online training iteration **do**
  Rank trajectories by μ_τ to obtain r_τ
  Define trajectory sampling distribution

$$p(\tau) = (1-\varepsilon)\frac{(r_\tau + 1)^{-\alpha}}{\sum_{\tau'}(r_{\tau'} + 1)^{-\alpha}} + \varepsilon\frac{1}{|\mathcal{T}|}$$

  Sample minibatch of trajectories

$$\{\tau_i\} \sim p(\tau)$$

  for each sampled trajectory τ_i **do**
    Sample start index

$$s_i \sim \text{Uniform}\{b_{\tau_i}, \ldots, e_{\tau_i} - H + 1\}$$

    Construct action chunk $a_{s_i:s_i+H-1}$
  end for
  Compute H-step TD targets $\hat{y}_{s_i}$
  Update critic Q_θ via minibatch SGD
  Update trajectory TD-error statistics

$$\mu_{\tau_i} \leftarrow \text{EMA}(|\delta_{s_i}|)$$

end for

알고리즘 2 Online Trajectory-level PER

6. Discussion

6-1. PTR 와의 차이

본 연구는 Prioritized Trajectory Replay(PTR)의 trajectory-level 우선순위 개념을 참고하였으나, PTR 의 backward trajectory consumption 및 trajectory 소모 구조는 채택하지 않았다.
PTR 는 trajectory 를 terminal 상태부터 순차적으로 소비하는 trajectory-centric 학습 구조를 전제로 한다. 반면 본 연구는 여러 trajectory 로부터 샘플링된 chunk 들을 확률적으로 결합하는 minibatch 기반 SGD 구조를 사용한다. 이러한 설정에서 backward trajectory consumption 을 적용할 경우, terminal 인접 상태에 대한 샘플링 편향이 심화되고 batch 다양성이 급격히 감소한다.
또한 PTR 에서 사용되는 SARSA 또는 weighted target 은 stored action 이 현재 정책과 충분히 근접하다는 가정을 필요로 한다. 그러나 본 연구에서는 action 이 flow 기반 정책에 의해 생성되는 H-step chunk 로 표현되며, stored action 과 현재 정책 action 간의 분포 차이가 크다. 이로 인해 SARSA-style target 은 편향을 완화하기보다는 오히려 증폭시킬 가능성이 있다.
이러한 이유로 본 연구에서는 backward consumption 과 SARSA-style target 을 제외하고, minibatch 기반 off-policy 학습과 구조적으로 정합적인 trajectory-level prioritized replay 를 채택하였다.

표 2 Comparison of Replay Sampling Methods

Category	PER	PTR (Original)	Trajectory-level PER (Ours)
Priority unit	Transition	Trajectory	Trajectory
Priority signal	TD-error	Return / Q-based	Trajectory-agg TD-error
Sampling order	i.i.d minibatch	Sequential (backward)	i.i.d minibatch
Backward consumption	No	Yes	No
Trajectory exhaustion	No	Yes	No
Action granularity	1-step	1-step	H-step chunked actions
Update style	SGD minibatch	Trajectory-centric	SGD minibatch
Target formulation	Policy target	SARSA / weighted target	Policy target
Long-horizon structure	No	Yes	Yes (via trajectory agg)
Chunking compatibility	Yes	No	Yes
Stability in online RL	High	Medium	High

7. 결론

본 프로젝트는 Flow Q-Learning(FQL)과 Action Chunking(QC-FQL)을 결합한 offline-to-online 강화학습 구조에서, replay buffer 샘플링 전략이 학습 안정성과 성능에 미치는 영향을 체계적으로 분석하였다. 특히 replay 우선순위를 단순한 성능 향상 기법이 아닌, offline 단계와 online 단계에서 서로 다른 역할을 수행하는 제어 메커니즘으로 재해석하고, 각 단계의 목적에 정합적인 샘플링 설계를 제안·검증하였다.

Offline 단계에서는 데이터가 고정된 mixture distribution 을 이루고 TD-error 나 reward 가 신뢰할 수 있는 학습 신호로 작동하지 않는다는 점에 주목하여, trajectory feature 기반 클러스터링을 통한 cluster-uniform sampling 을 적용하였다. 이를 통해 offline 데이터에 포함된 다양한 temporally coherent behavior mode 를 보존하고, flow 기반 정책과 Q-chunking critic 이 특정 고보상 trajectory 에 과도하게 수렴하는 현상을 방지하였다.

반면 online 단계에서는 critic 이 일정 수준의 초기 근사를 확보한 이후, 학습의 병목이 "어디에서 가치 예측이 실패하고 있는가"로 이동함을 관측하였다. 이에 따라 trajectory-level TD-error 통계의 rank 기반 우선순위 샘플링을 도입하여, 절대값 기반 우선순위에서 발생하는 분포 붕괴와 불안정을 완화하면서도 학습이 필요한 trajectory segment 를 효과적으로 강조하였다. 실험 결과, reward 기반 우선순위는 손실 감소에도 불구하고 성공률이 붕괴되는 반면, TD-error rank 기반 샘플링은 안정적인 value learning 과 정책 성능 향상으로 이어짐을 확인하였다.

또한 본 연구는 Prioritized Trajectory Replay(PTR)를 기존 구조 그대로 차용하는 대신, minibatch 기반 off-policy SGD 및 action chunking 설정에 맞게 재구성하였다. PTR 에서 가정하는 backward trajectory consumption 과 SARSA-style target 은 배제하고, trajectory-level 우선순위 개념만을 선택적으로 활용함으로써, batch 다양성을 유지하면서도 장기 행동 시퀀스 단위의 학습 신호를 강조하는 구조를 제시하였다.

종합적으로, 본 프로젝트는 offline 단계에서는 다양성 보존, online 단계에서는 학습 필요 구간 강조라는 단계별 목표에 따라 replay sampling 전략을 분리 설계하는 것이 장기·희소 보상 환경에서의 학습 안정성과 성능 향상에 핵심적임을 실험적으로 입증하였다.

5.1) 향후 연구방향

향후 연구에서는 다음과 같은 확장이 가능하다.

첫째, trajectory feature 클러스터링 방식의 고도화이다. 본 연구에서는 PCA 기반 feature 와 K-means 중심의 클러스터링을 사용하였으나, representation learning 을 통해 task-adaptive trajectory embedding 을 학습하고 이를 기반으로 한 동적 클러스터링 전략을 도입할 수 있다.

둘째, trajectory-level TD-error 외에도 value uncertainty, ensemble disagreement, 또는 world model 기반 prediction error 를 결합한 복합 우선순위 지표를 설계함으로써, online 단계에서의 샘플링 효율을 더욱 향상시킬 수 있다.

셋째, 본 연구의 sampling 전략을 FQL/QC-FQL 외의 offline-to-online 알고리즘(CQL, IQL, SAC-flow 등)에 적용하여, 알고리즘 독립적인 일반성을 검증하는 방향의 연구가 가능하다.

마지막으로, 실제 로봇 환경에서의 온라인 fine-tuning 실험을 통해, 제안한 단계별 replay sampling 전략이 시뮬레이션을 넘어 현실 환경에서도 안정성과 데이터 효율성을 제공하는지를 검증하는 것이 중요한 향후 과제로 남아 있다.

참고문헌

[1] S. Park, Q. Li, and S. Levine, "**Flow Q-Learning**," *arXiv preprint arXiv:2502.02538*, 2025. [Online]. Available: https://arxiv.org/abs/2502.02538

[2] Q. Li, Z. Zhou, and S. Levine, "**Reinforcement Learning with Action Chunking**," in *Proceedings of the 39th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2507.07969. [Online]. Available: https://arxiv.org/abs/2507.07969

[3] J. Liu, Y. Ma, J. Hao, Y. Hu, Y. Zheng, T. Lv, and C. Fan, "**Prioritized Trajectory Replay: A Replay Memory for Data-driven Reinforcement Learning**," in *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2024. arXiv:2306.15503. [Online].

Available: https://doi.org/10.48550/arXiv.2306.15503

[4] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "**Prioritized Experience Replay**," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016. arXiv:1511.05952. [Online]. Available: https://doi.org/10.48550/arXiv.1511.05952

[5] S. Levine, A. Kumar, G. Tucker, and J. Fu, "**Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems**," *arXiv preprint arXiv:2005.01643*, 2020. [Online]. Available: https://arxiv.org/abs/2005.01643

[6] R. S. Sutton and A. G. Barto, **Reinforcement Learning: An Introduction**, 2nd ed. Cambridge, MA, USA: MIT Press, 2018. [Online]. Available: https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf

[7] Y. Lipman, M. Havasi, P. Holderrieth, N. Shaul, M. Le, B. Karrer, R. T. Q. Chen, D. Lopez-Paz, H. Ben-Hamu, and I. Gat, "Flow Matching Guide and Code," arXiv preprint arXiv:2412.06264, 2024. [Online]. Available: https://arxiv.org/abs/2412.06264

[8] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative Q-Learning for Offline Reinforcement Learning," arXiv preprint arXiv:2006.04779, 2020.
[Online]. Available: https://doi.org/10.48550/arXiv.2006.04779

[프로젝트 GitHub 링크]

https://github.com/Revivekirin/qc-flow-priority-sampling