Data-Centric Robot Learning:
From Data Curation to Scalable VLA Pretraining



2026년 11월 13일

장성호

발표자 소개



❖ **장성호 (Seongho Jang)**
- 고려대학교 산업경영공학과 박사과정 재학
- Data Mining & Quality Analytics Lab. (김성범 교수님)

❖ **Research Interest**
- Reinforcement learning
- Physics-informed machine learning
- Physical AI

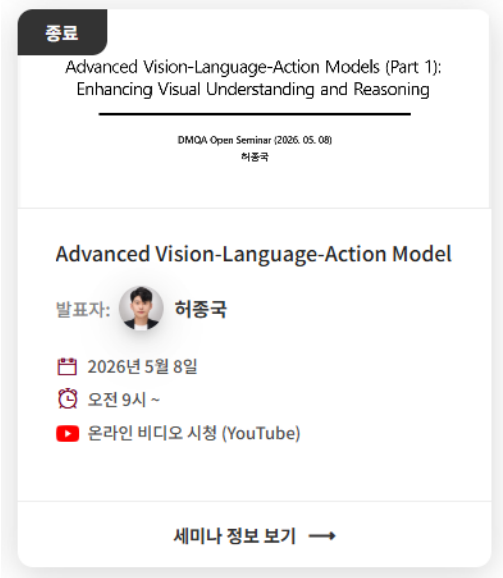
❖ **Contact**
- jsh01@korea.ac.kr

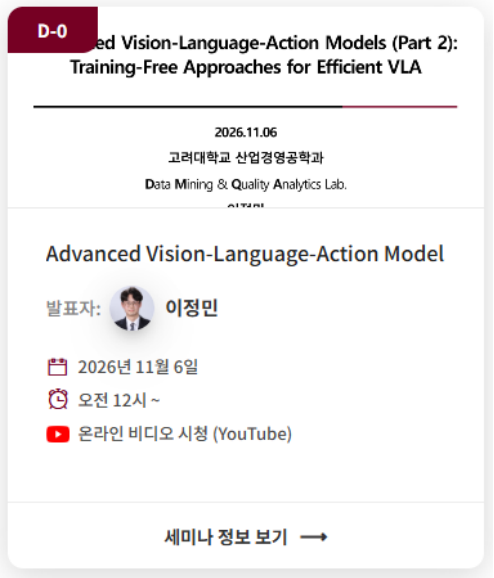
Introduction

VLA는 어떤 경험으로 행동을 배우는가?

❖ 사전학습된 시각·언어 지식을 로봇 행동으로 연결하는 VLA

- **VLA (Vision-Language-Action)**: 카메라 영상과 언어 지시를 입력받아 로봇 행동을 직접 출력하는 모델
- CUPID (CoRL, 2025): 남은 데이터 중 정책 성능에 실질적으로 도움이 되는 경험을 판단



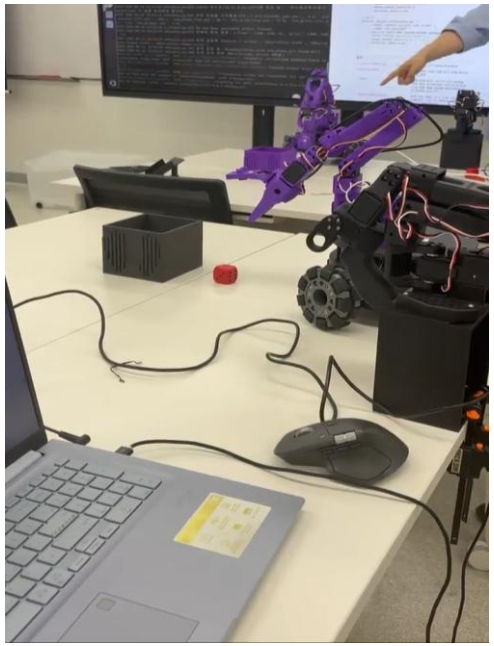


Introduction

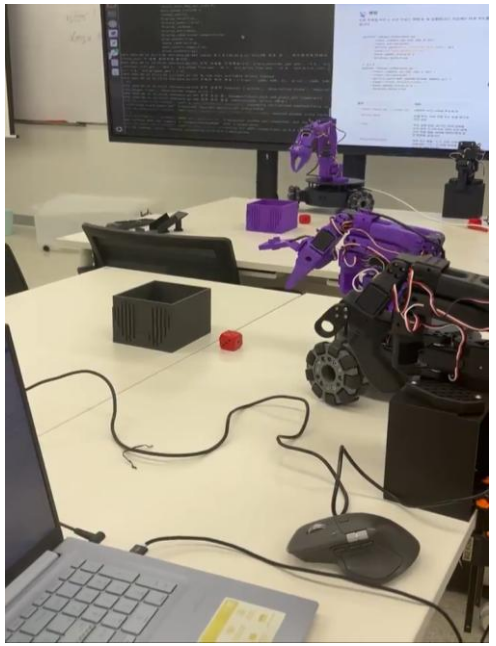
데이터를 많이 수집하면 로봇의 학습 성능도 계속 좋아질까?

❖ 데이터를 더 수집하기 전에, 무엇이 학습을 방해하는지 살펴봐야 함

- 기존 데이터: 잘못된 행동이나 중복된 경험이 포함되어 학습 성능을 저하시킬 수 있음
- 추가 데이터: 수집 비용이 크므로 정책 성능 향상에 필요한 경험을 우선적으로 확보해야 함



데이터 50개로 학습한 모델



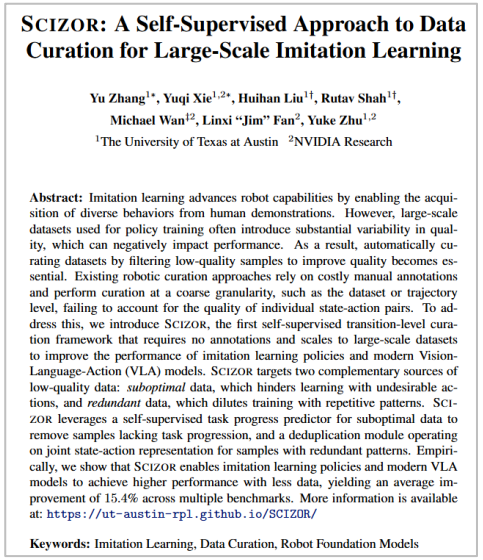
데이터 100개로 학습한 모델

Introduction

VLA 학습에 필요한 데이터를 어떤 기준으로 선별할 수 있을까?

❖ **데이터 자체의 품질과 정책 성능에 대한 기여도라는 관점에서 접근**

- SCIZOR (2026, ICRA, Oral): Task progress와 State-Action 유사도를 활용하여 비효율적이거나 중복된 경험을 제거
- CUPID (2025, CoRL): 각 시연이 Policy의 실제 수행 성능에 미치는 영향을 추정하여 학습에 유용한 데이터를 선별



SCIZOR: A Self-Supervised Approach to Data Curation for Large-Scale Imitation Learning

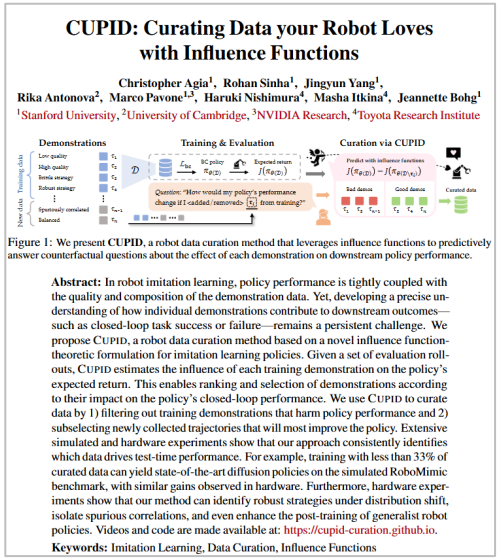
Yu Zhang[1*], Yuqi Xie[1,2*], Huihan Liu[1†], Rutav Shah[1†], Michael Wan[12], Linxi "Jim" Fan[2], Yuke Zhu[1,2]

[1]The University of Texas at Austin [2]NVIDIA Research

Abstract: Imitation learning advances robot capabilities by enabling the acquisition of diverse behaviors from human demonstrations. However, large-scale datasets used for policy training often introduce substantial variability in quality, which can negatively impact performance. As a result, automatically curating datasets by filtering low-quality samples to improve quality becomes essential. Existing robotic curation approaches rely on costly manual annotations and perform curation at a coarse granularity, such as the dataset or trajectory level, failing to account for the quality of individual state-action pairs. To address this, we introduce SCIZOR, the first self-supervised transition-level curation framework that requires no annotations and scales to large-scale datasets to improve the performance of imitation learning policies and modern Vision-Language-Action (VLA) models. SCIZOR targets two complementary sources of low-quality data: *suboptimal* data, which hinders learning with undesirable actions, and *redundant* data, which dilutes training with repetitive patterns. SCIZOR leverages a self-supervised task progress predictor for suboptimal data to remove samples lacking task progression, and a deduplication module operating on joint state-action representation for samples with redundant patterns. Empirically, we show that SCIZOR enables imitation learning policies and modern VLA models to achieve higher performance with less data, yielding an average improvement of 15.4% across multiple benchmarks. More information is available at: https://ut-austin-rpl.github.io/SCIZOR/

Keywords: Imitation Learning, Data Curation, Robot Foundation Models

SCIZOR
(2026, ICRA)



CUPID: Curating Data your Robot Loves with Influence Functions

Christopher Agia[1], Rohan Sinha[1], Jingyun Yang[1], Rika Antonova[2], Marco Pavone[1,3], Haruki Nishimura[4], Masha Itkina[4], Jeannette Bohg[1]

[1]Stanford University, [2]University of Cambridge, [3]NVIDIA Research, [4]Toyota Research Institute

Figure 1: We present **CUPID**, a robot data curation method that leverages influence functions to predictively answer counterfactual questions about the effect of each demonstration on downstream policy performance.

Abstract: In robot imitation learning, policy performance is tightly coupled with the quality and composition of the demonstration data. Yet, developing a precise understanding of how individual demonstrations contribute to downstream outcomes—such as closed-loop task success or failure—remains a persistent challenge. We propose CUPID, a robot data curation method based on a novel influence function-theoretic formulation for imitation learning policies. Given a set of evaluation rollouts, CUPID estimates the influence of each training demonstration on the policy's expected return. This enables ranking and selection of demonstrations according to their impact on the policy's closed-loop performance. We use CUPID to curate data by 1) filtering out training demonstrations that harm policy performance and 2) subselecting newly collected trajectories that will most improve the policy. Extensive simulated and hardware experiments show that our approach consistently identifies which data drives test-time performance. For example, training with less than 33% of curated data can yield state-of-the-art diffusion policies on the simulated RoboMimic benchmark, with similar gains observed in hardware. Furthermore, hardware experiments show that our method can identify robust strategies under distribution shift, isolate spurious correlations, and even enhance the post-training of generalist robot policies. Videos and code are made available at: https://cupid-curation.github.io.

Keywords: Imitation Learning, Data Curation, Influence Functions

CUPID
(2025, CoRL)

Introduction

어떤 데이터를 제거할 것인가?

❖ **SCIZOR: A Self-Supervised Approach to Data Curation for Large-Scale Imitation Learning (2026, ICRA, Oral)**

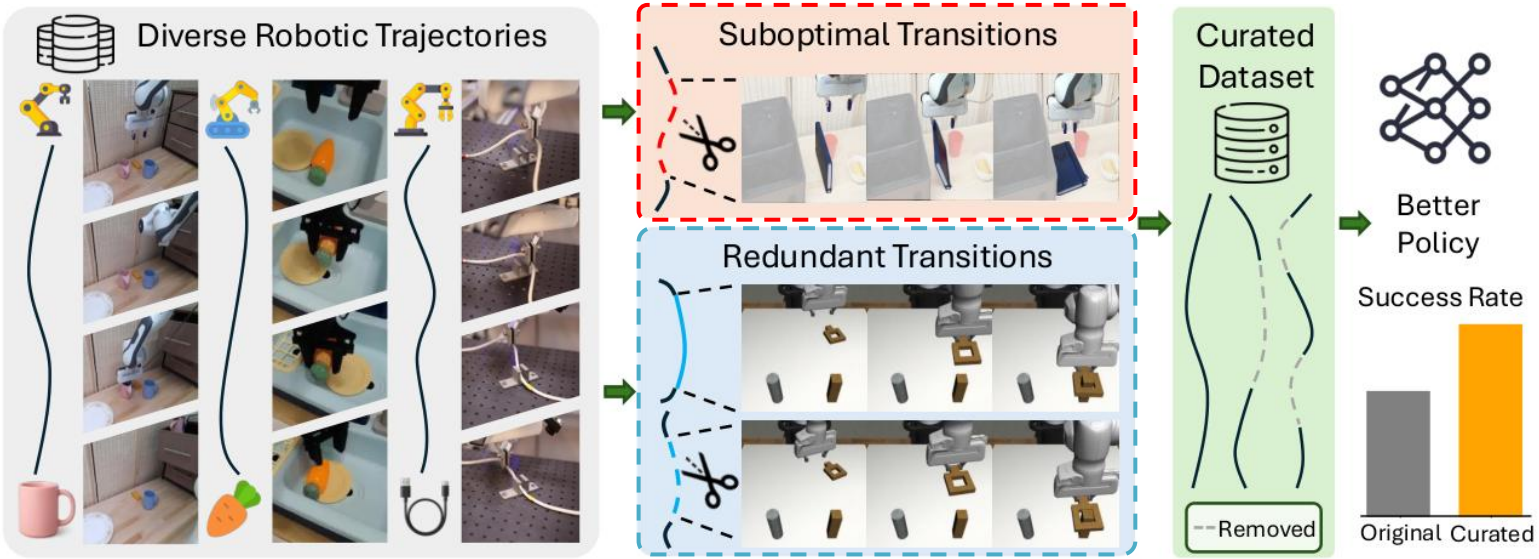
- 기존 방법은 데이터셋 또는 시연 전체를 기준으로 선별
- 시연 전체를 제거하면 유용한 행동까지 손실될 수 있음
- SCIZOR는 개별 transition의 품질을 고려하여 데이터를 정제

SCIZOR: A Self-Supervised Approach to Data Curation for Large-Scale Imitation Learning

Yu Zhang[1*], Yuqi Xie[1,2*], Huihan Liu[1†], Rutav Shah[1†],
Michael Wan[‡2], Linxi "Jim" Fan[2], Yuke Zhu[1,2]
[1]The University of Texas at Austin [2]NVIDIA Research

Abstract: Imitation learning advances robot capabilities by enabling the acquisition of diverse behaviors from human demonstrations. However, large-scale datasets used for policy training often introduce substantial variability in quality, which can negatively impact performance. As a result, automatically curating datasets by filtering low-quality samples to improve quality becomes essential. Existing robotic curation approaches rely on costly manual annotations and perform curation at a coarse granularity, such as the dataset or trajectory level, failing to account for the quality of individual state-action pairs. To address this, we introduce SCIZOR, the first self-supervised transition-level curation framework that requires no annotations and scales to large-scale datasets to improve the performance of imitation learning policies and modern Vision-Language-Action (VLA) models. SCIZOR targets two complementary sources of low-quality data: *suboptimal* data, which hinders learning with undesirable actions, and *redundant* data, which dilutes training with repetitive patterns. SCIZOR leverages a self-supervised task progress predictor for suboptimal data to remove samples lacking task progression, and a deduplication module operating on joint state-action representation for samples with redundant patterns. Empirically, we show that SCIZOR enables imitation learning policies and modern VLA models to achieve higher performance with less data, yielding an average improvement of 15.4% across multiple benchmarks. More information is available at: https://ut-austin-rpl.github.io/SCIZOR/

Keywords: Imitation Learning, Data Curation, Robot Foundation Models

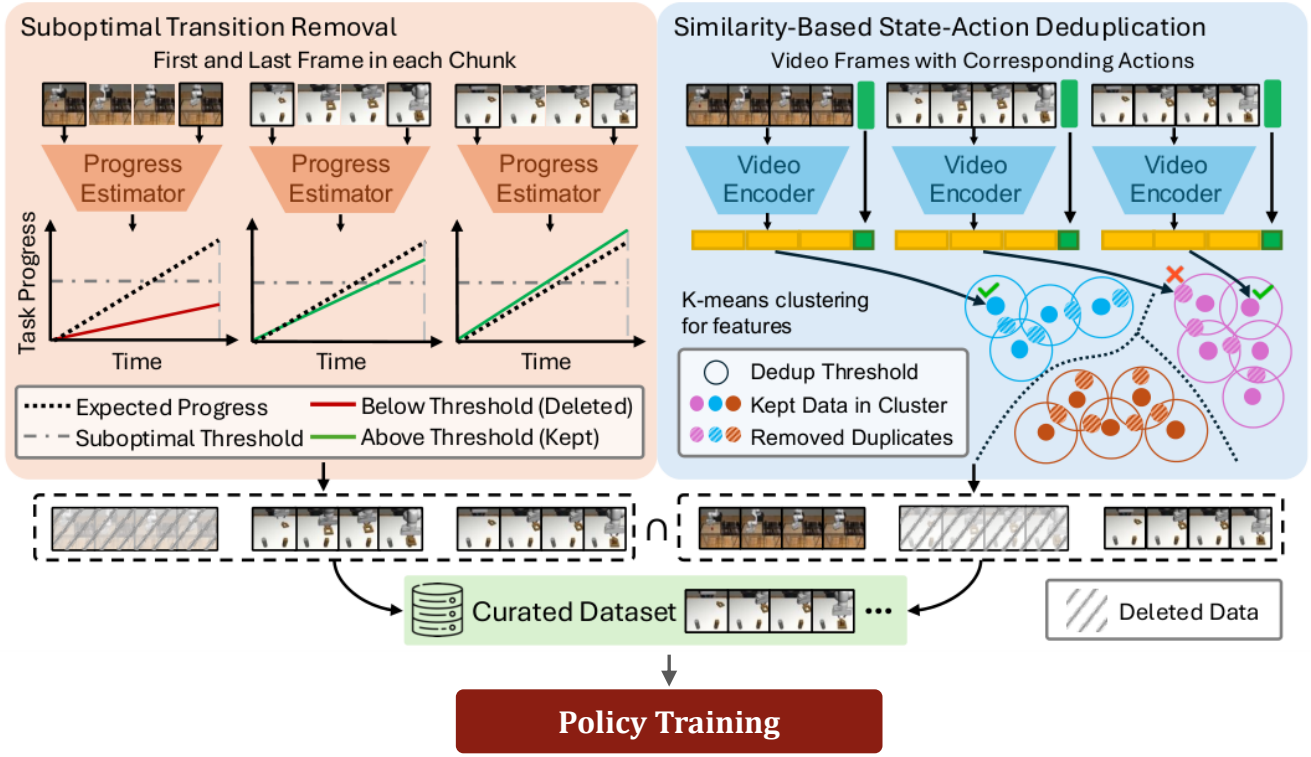


Proposed method

SCIZOR는 어떤 과정을 통해 데이터를 정제할까?

❖ 두 모듈을 결합하여 저품질 transition과 중복 구간을 함께 제거

- **Suboptimal transition removal**: Task progress를 추정하여 작업 진척도가 부족한 구간을 식별
- **State-action deduplication**: 시각 정보와 행동 정보를 결합하여 중복되는 구간을 식별
- 두 모듈 중 하나라도 제거 대상으로 판단한 구간을 제외한 curated dataset으로 최종 policy를 학습

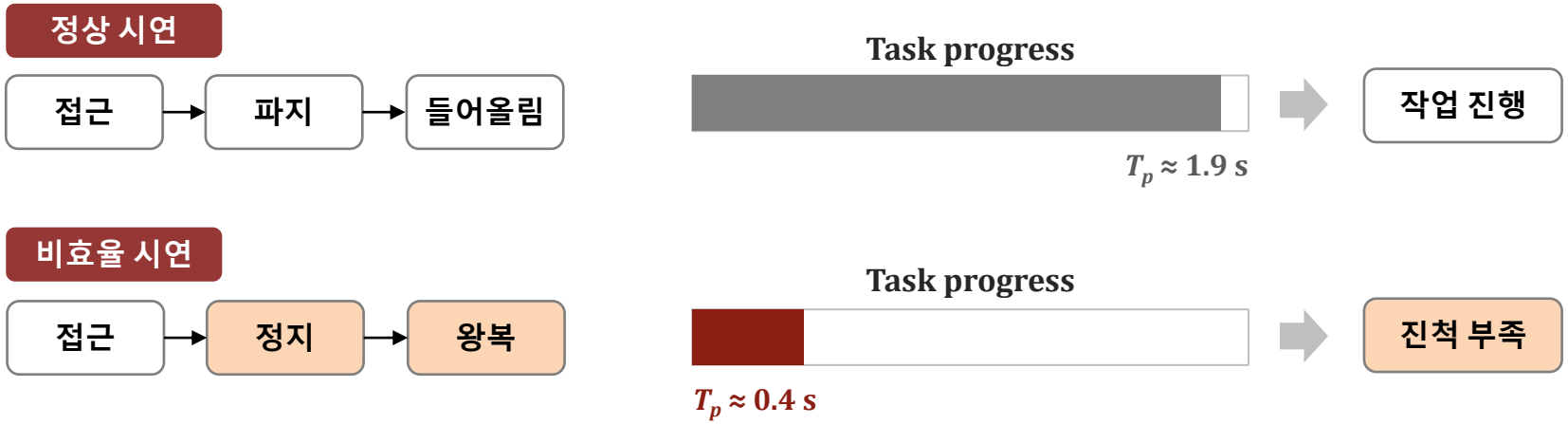


Proposed method

시간이 흘렀다고 작업도 진행된 것일까?

❖ 실제로 흐른 시간과 작업이 진척된 정도는 다를 수 있음

- **Task progress**: 관측된 상태 변화로 추정한 작업 진척도를 시간(T_p)에 대응시킨 값
- 정답으로 주어지는 값이 아니라, **모델이 관측으로부터 추정**하는 값
- 정지·왕복 같은 비효율적 행동은 실제 경과 시간에 비해 **추정된 progress가 작음**

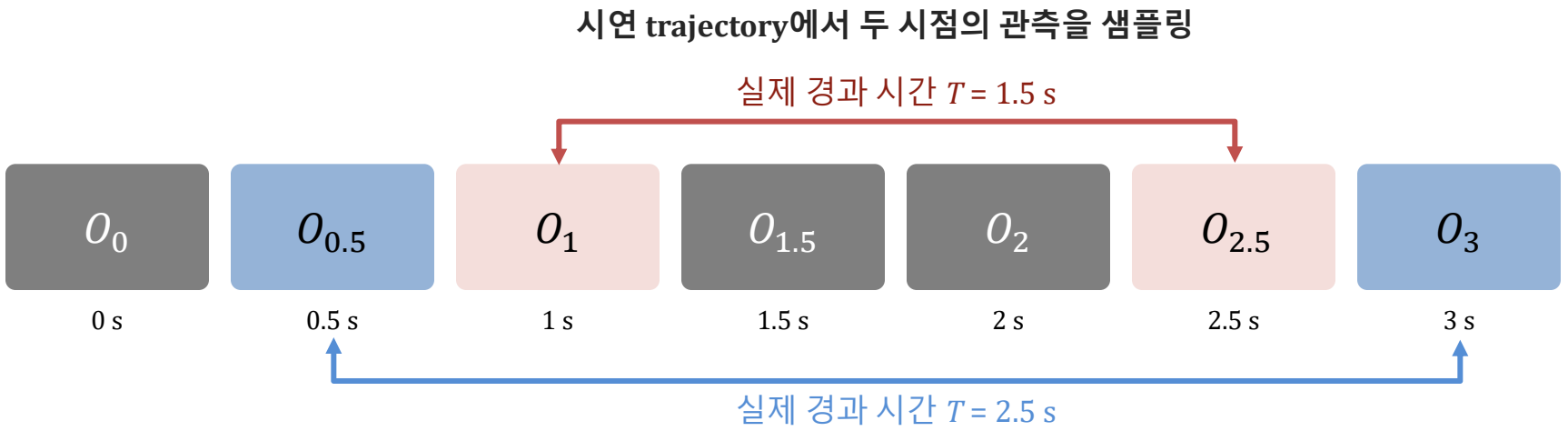


Proposed method

진행 정도를 표시한 정답 없이 어떻게 학습할까?

❖ 관측 사이의 실제 시간 간격을 task progress의 pseudo-label로 활용

- 시연 데이터에는 각 구간의 task progress를 나타내는 정답 레이블이 없음
- 동일 trajectory에서 두 시점의 이미지를 추출하고, **실제 시간 간격(T)**을 학습 정답으로 사용
- 시간 간격을 예측하도록 학습한 뒤, 예측 간격(T_p)이 실제 간격보다 **짧은 구간**을 진행 부족으로 판단

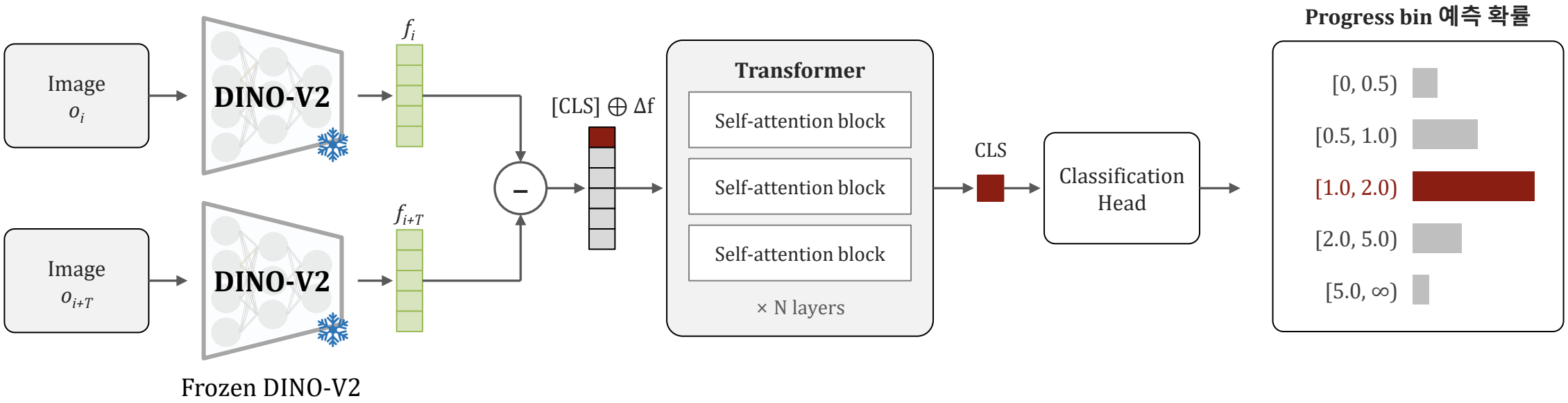


Proposed method

두 이미지로부터 무엇을 예측할까?

❖ 두 관측의 시각적 변화로부터 시간 간격이 속하는 구간을 분류

- 두 관측 이미지를 **Frozen DINO-V2**로 인코딩하고, 특징 차이(Δf)를 [CLS] 토큰과 함께 Transformer에 입력
- Classification head가 **5개 time bin** $[0, 0.5), [0.5, 1.0), [1.0, 2.0), [2.0, 5.0), [5.0, \infty)$ 중 하나를 예측
- 실수 값 회귀보다 이산 구간 분류가 경험적으로 더 robust하여 classification을 사용

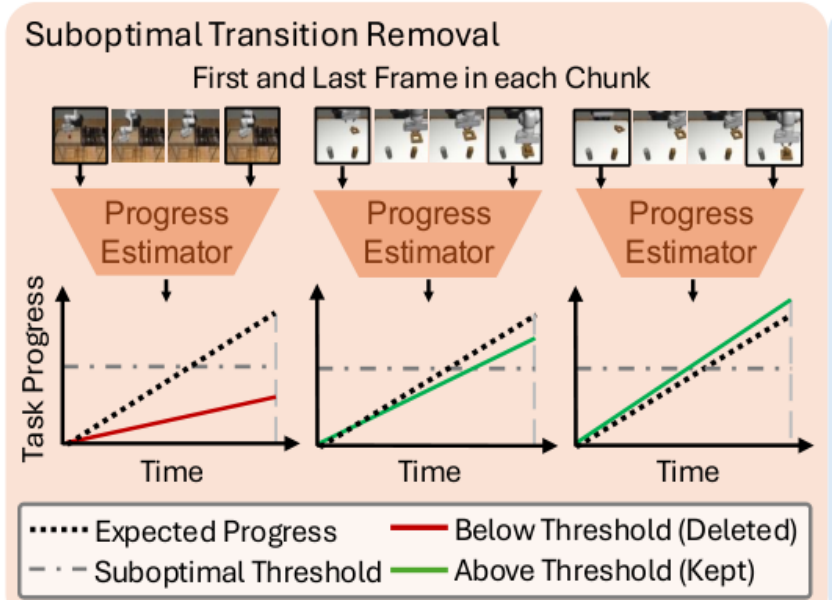


Proposed method

예측 결과를 어떻게 데이터 제거에 활용할까?

❖ 실제 경과 시간에 비해 예측된 task progress가 부족한 구간을 식별

- $V_{i:i+T} = T - T_P$
- T: 실제 경과 시간, T_P: 관측 변화로부터 예측한 진척도에 해당하는 시간
- 값이 클수록 시간에 비해 진척이 부족한 구간으로 판단



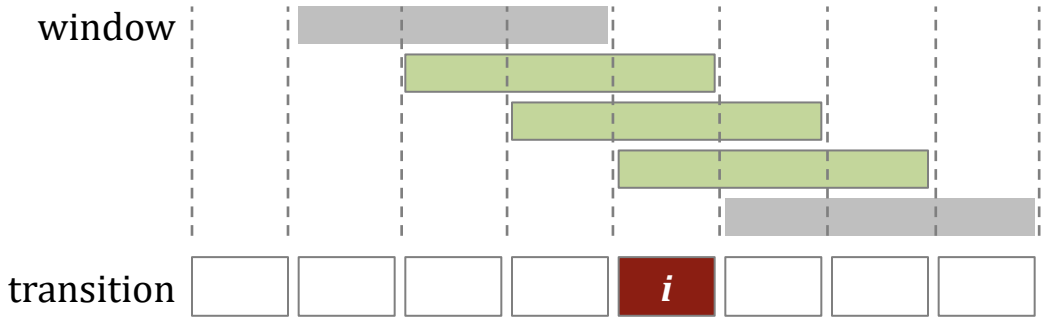
Proposed method

현재 구간의 task progress만으로 행동의 품질을 판단할 수 있을까?

❖ **구간 점수를 transition에 배분하고, 이후 구간과 trajectory 전체의 품질을 함께 반영**

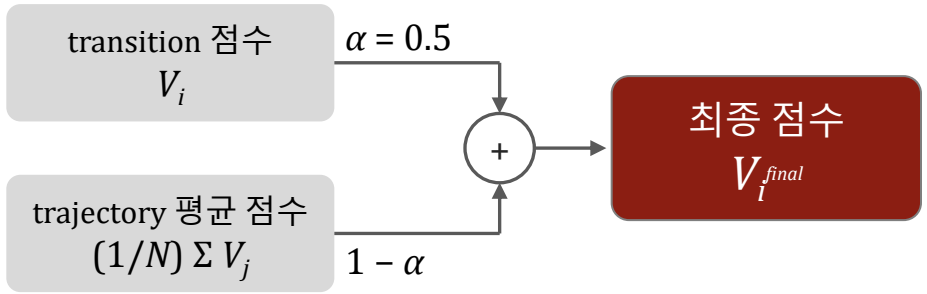
- **Overlapping windows**: 한 transition을 포함하는 모든 구간의 점수를 1/T씩 나누어 합산
- **Temporal discounting:** 이후 구간의 suboptimality를 현재 transition에 반영하되, 시간적으로 먼 구간의 영향은 작게 반영
- **Score mixture**: transition 점수와 trajectory 평균 점수를 0.5 : 0.5로 결합

Overlapping windows



transition *i*의 점수 = i를 포함한 모든 window 점수 × $1/T$ 의 합

Score mixture



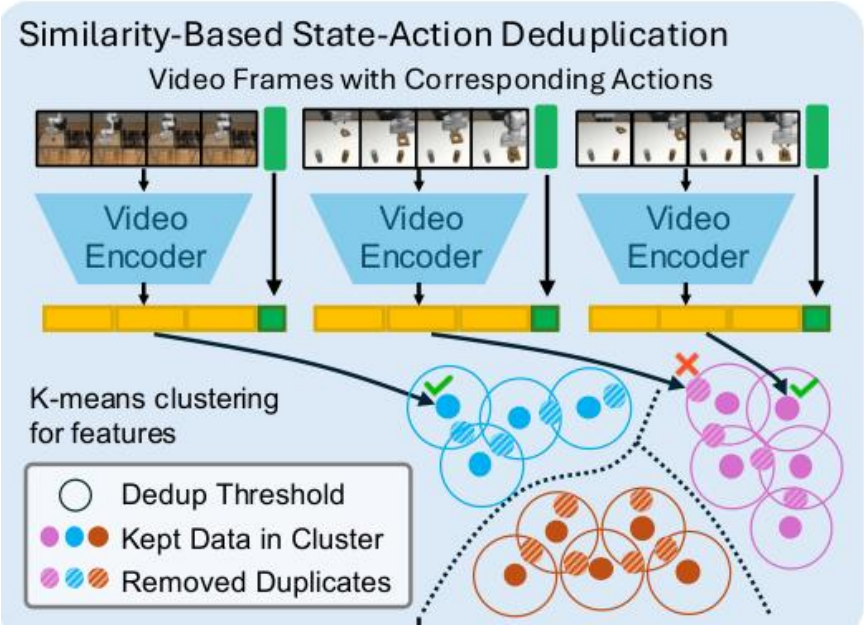
$$V_i^{final} = \alpha \cdot V_i + (1-\alpha) \cdot (1/N)\sum V_j$$

Proposed method

비슷하게 보이는 시연은 같은 경험일까?

❖ **시각적으로 유사한 상태에서도 다른 행동이 수행될 수 있으므로 state와 action을 함께 고려**

- 시각 정보만으로는 비슷한 장면에서 수행된 서로 다른 행동을 구분하기 어려움
- 의미 있는 행동 차이는 보존하면서 반복적인 경험만 제거해야 함
- 이를 위해 영상과 행동을 결합한 joint state-action representation으로 구간을 비교



State
영상 chunk의 특징

Action
해당 구간의 로봇 행동

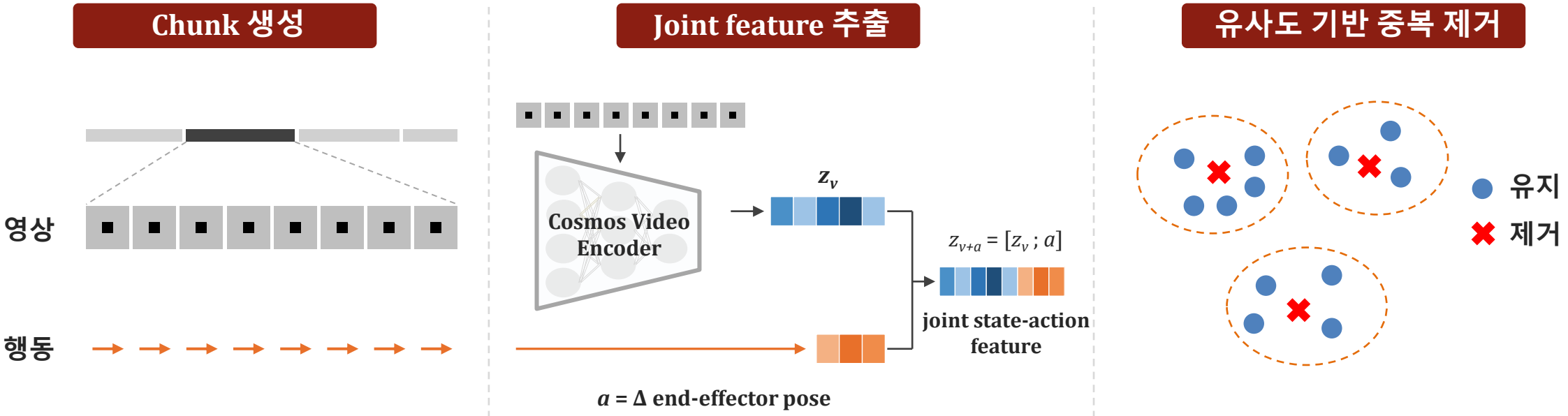
두 특징을 이어 붙인 joint feature로
K-means 군집 내 유사도를 비교

Proposed method

SCIZOR는 어떤 기준으로 중복되는 state-action 구간을 식별할까?

❖ 영상의 시간적 특징과 로봇 행동을 결합한 뒤, 특징 유사도로 중복 구간을 제거

- Trajectory를 2 s 비중첩 chunk로 나누고, chunk마다 RGB 프레임 8장을 균등 샘플링
- Cosmos video encoder 특징(z_v)에 delta end-effector pose를 이어 붙여 joint feature(z_{v+a}) 생성
- K-means 군집 내 cosine similarity가 threshold($\varepsilon_d = 0.99$)를 넘는 chunk를 중복으로 제거



Experimental setup

SCIZOR의 데이터 정제 효과를 어떤 조건에서 검증했을까?

❖ 서로 다른 데이터 품질과 규모를 가진 세 벤치마크에서 평가

- RoboMimic, Sirius-Fleet, OXE에서 서로 다른 Policy를 학습
- 전체 데이터, 무작위 제거 및 기존 Data Curation 방법과 비교

Dataset	특징	Policy
RoboMimic	서로 다른 숙련도의 시연	BC
Sirius-Fleet	사람의 개입이 포함된 실제 로봇 데이터	BC-Transformer
OXE	대규모 다중 로봇 데이터	Octo

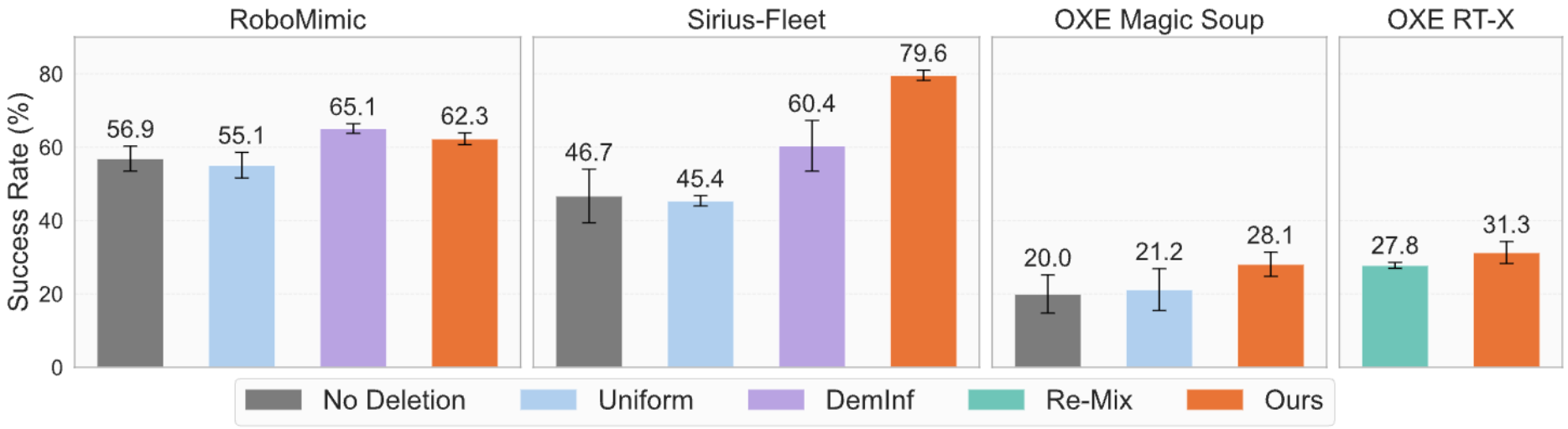
Method	비교 기준
No Deletion	전체 데이터 사용
Uniform	동일한 비율로 무작위 제거
DemInf	Trajectory-level 정제
Re-Mix	Dataset-level 데이터 비중 조정

Experimental results

데이터를 줄여도 로봇의 성능이 높아질까?

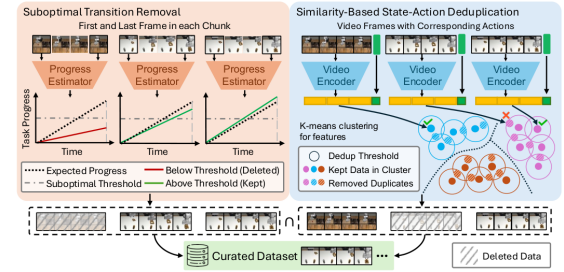
❖ **데이터를 일부 제거한 SCIZOR가 전체 데이터로 학습한 policy보다 높은 성공률을 기록**

- 같은 비율을 무작위로 제거한 Uniform은 No Deletion과 비슷하거나 낮아, 무엇을 제거하는지가 성능을 좌우
- Sirius-Fleet에서 SCIZOR는 79.6%로 No Deletion(46.7%)보다 32.9%p, DemInf(60.4%)보다 19.2%p 높음
- RoboMimic에서는 DemInf(65.1%)가 더 높았지만, OXE에서는 SCIZOR가 No Deletion·Re-Mix보다 높은 성공률을 기록



Experimental results

비효율 구간 제거와 중복 제거가 모두 필요할까?



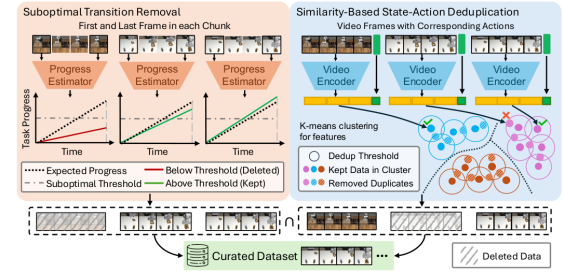
❖ **두 모듈을 함께 적용할 때 세 데이터셋 모두에서 가장 높은 성공률**

- Transition removal 제외 시 RoboMimic 62.3% → 48.3%로 가장 크게 하락
- Deduplication 제외 시 Sirius-Fleet 79.6% → 64.2%로 하락
- 한 모듈만 적용한 구성은 모든 데이터셋에서 두 모듈을 결합한 SCIZOR보다 낮아, 두 모듈이 상호 보완적

	RoboMimic	**Sirius-Fleet**	**$\text{OXE}_{\text{Magic}}$**
SCIZOR (ours)	62.3 ± 1.6	79.6 ± 1.4	28.1 ± 3.3
w/o deduplication	60.9 ± 1.8	64.2 ± 2.6	25.3 ± 2.9
w/o transition removal	48.3 ± 0.8	63.3 ± 6.9	22.1 ± 0.9

Experimental results

Temporal discounting과 score mixture가 정책 성능에 도움이 될까?



❖ 두 요소를 함께 적용한 SCIZOR가 네 작업 모두에서 최고 성공률

- Score mixture 제외 시 전 작업 하락, RT-1 Pick은 30.9% → 21.8%
- Temporal discounting 제외 시 네 작업 모두 가장 낮은 성공률
- 이후 구간의 suboptimality와 trajectory 전체의 품질을 반영할 때 성능이 개선됨

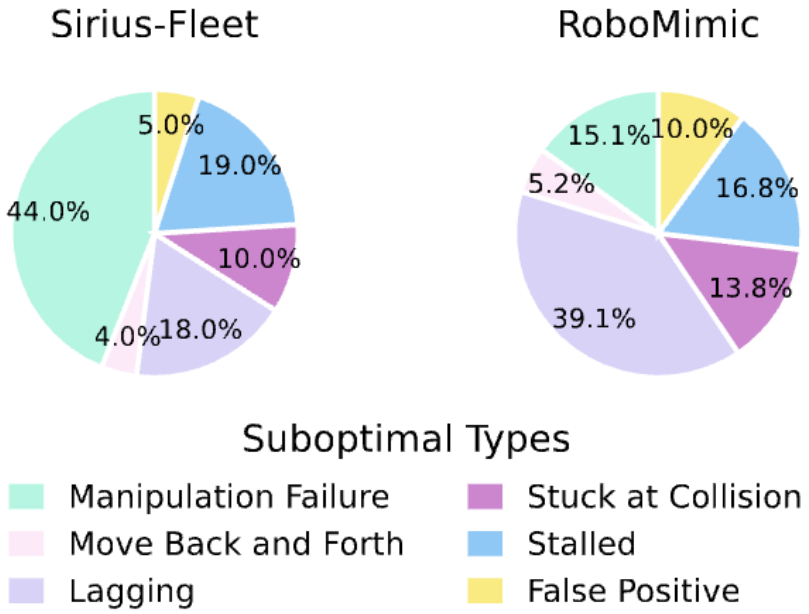
	RoboMimic Can	RoboMimic Square	$\text{OXE}_{\text{RT-1}}$ Pick	$\text{OXE}_{\text{RT-1}}$ Move
SCIZOR (ours)	87.3 ± 0.7	37.2 ± 2.5	30.9 ± 8.4	17.5 ± 1.0
w/o score mixture	81.3 ± 0.6	36.0 ± 1.4	21.8 ± 7.9	12.4 ± 4.6
w/o temporal discounting	79.6 ± 1.4	31.5 ± 5.5	20.7 ± 6.4	9.4 ± 1.4

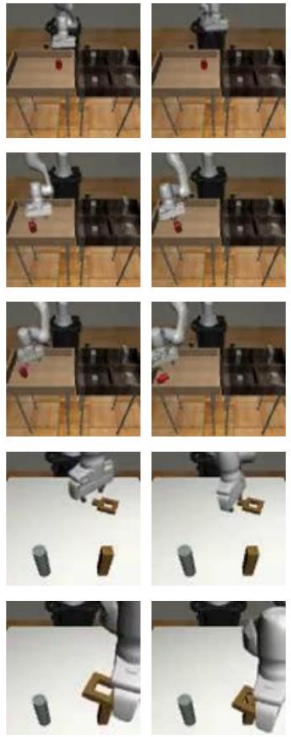
Experimental results

SCIZOR는 실제로 어떤 구간을 저품질로 판단했을까?

❖ **식별된 구간에는 조작 실패, 지연된 움직임, 정체 등 작업 수행을 방해하는 행동이 포함**

- 저품질 구간을 하나 이상 포함한다고 예측한 시연을 데이터셋별로 100개씩 선정
- 선정한 시연 내 예측 구간을 사람이 확인하고 행동 유형별로 분류
- Sirius-Fleet에서는 manipulation failure, RoboMimic에서는 lagging이 가장 큰 비중을 차지





Move Back and Forth: The robot arm moved aimlessly and didn't contribute to the task progress.

Manipulation Failure: The robot gripper missed the can when trying to pick it up.

Manipulation Failure: The robot gripper knocked over the can when trying to pick it up.

Slow Motion: The robot gripper move towards the square rod slowly.

Stuck at Collision: The square rod held by the robot collided with the column when trying to insert.

Conclusion

SCIZOR는 무엇을 개선했을까?

❖ 시연 전체를 남기거나 버려야 할까?

- 시연 내부의 품질 차이를 고려하여, 저품질 구간을 transition 단위로 선별해 제거

❖ 사람의 품질 평가 없이 비효율적인 구간을 찾을 수 있을까?

- 시연의 시간 정보로 학습한 Task Progress Predictor를 활용해 진행이 부족한 구간을 탐지

❖ 비슷하게 보이는 경험을 모두 중복으로 판단해도 될까?

- 영상의 시간적 특징과 행동을 함께 비교하여 반복되는 state-action 구간을 제거

❖ 데이터를 줄여도 policy 성능이 좋아질 수 있을까?

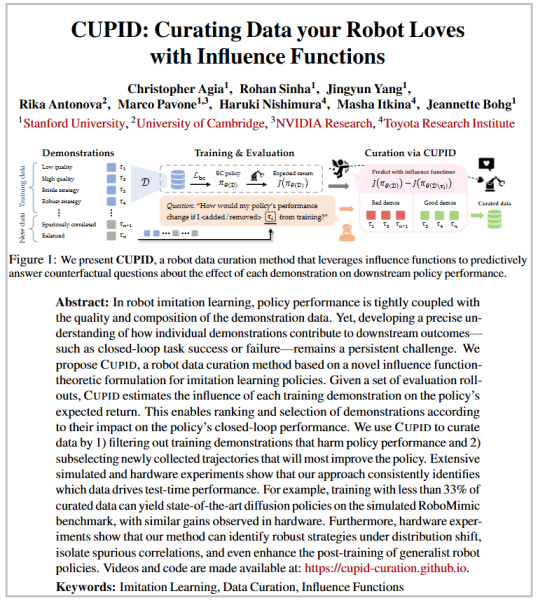
- OXE-Magic Soup에서 데이터를 15.8% 줄이면서, 전체 데이터로 학습한 Octo 대비 평균 성공률 8.1%p 향상

Introduction

잘 수행된 시연은 모두 policy 학습에 도움이 될까?

❖ CUPID: Curating Data your Robot Loves with Influence Functions (2025, CoRL)

- 성공한 시연에도 비효율적인 동작이나 새로운 환경에서 실패하기 쉬운 전략이 포함될 수 있음
- Policy는 시연의 행동을 모방하여 학습하지만, 실제 성능은 환경에서 작업을 끝까지 수행한 결과로 평가
- CUPID는 각 demonstration이 현재 policy의 closed-loop performance에 미치는 영향을 기준으로 데이터를 선별



CUPID: Curating Data your Robot Loves with Influence Functions

Christopher Agia[1], Rohan Sinha[1], Jingyun Yang[1], Rika Antonova[2], Marco Pavone[1,3], Haruki Nishimura[4], Masha Itkina[4], Jeannette Bohg[1]
[1]Stanford University, [2]University of Cambridge, [3]NVIDIA Research, [4]Toyota Research Institute

Figure 1: We present **CUPID**, a robot data curation method that leverages influence functions to predictively answer counterfactual questions about the effect of each demonstration on downstream policy performance.

Abstract: In robot imitation learning, policy performance is tightly coupled with the quality and composition of the demonstration data. Yet, developing a precise understanding of how individual demonstrations contribute to downstream outcomes—such as closed-loop task success or failure—remains a persistent challenge. We propose CUPID, a robot data curation method based on a novel influence function-theoretic formulation for imitation learning policies. Given a set of evaluation rollouts, CUPID estimates the influence of each training demonstration on the policy's expected return. This enables ranking and selection of demonstrations according to their impact on the policy's closed-loop performance. We use CUPID to curate data by 1) filtering out training demonstrations that harm policy performance and 2) subselecting newly collected trajectories that will most improve the policy. Extensive simulated and hardware experiments show that our approach consistently identifies which data drives test-time performance. For example, training with less than 33% of curated data can yield state-of-the-art diffusion policies on the simulated RoboMimic benchmark, with similar gains observed in hardware. Furthermore, hardware experiments show that our method can identify robust strategies under distribution shift, isolate spurious correlations, and even enhance the post-training of generalist robot policies. Videos and code are made available at: https://cupid-curation.github.io.

Keywords: Imitation Learning, Data Curation, Influence Functions

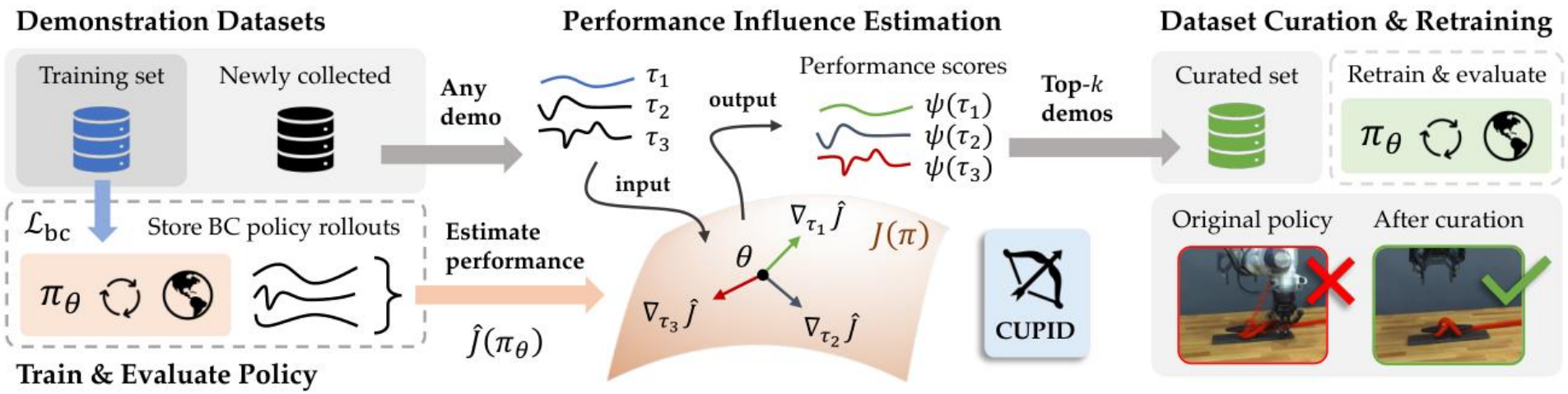
구분	Demonstration	Policy rollout
의미	학습을 위해 제공된 시연 기록	학습된 policy를 환경에서 실행한 기록
역할	행동을 학습하는 데이터	실제 수행 결과를 확인하는 데이터
논문의 설정	작업에 성공한 시연	성공과 실패 모두 발생 가능

Introduction

시연마다 policy를 다시 학습해야 기여도를 알 수 있을까?

❖ Influence function으로 학습 비중에 따른 성능 변화를 근사

- 특정 demonstration을 학습에 더 반영했다면 policy의 평가 결과가 어떻게 달라졌을지 추정
- 현재 모델의 미분 정보와 evaluation rollout을 활용하므로, 시연마다 policy를 다시 학습할 필요는 없음
- 추정한 영향도로 데이터를 선별한 뒤, 최종 policy는 재구성한 데이터로 다시 학습하고 평가



Proposed method

시연의 학습 기여도를 수학적으로 어떻게 정의할까?

❖ **Performance influence는 학습 비중에 따른 expected return의 변화율**

- 시연 ξ의 평균 학습 손실 ℓ_{traj}에 가중치 ε을 추가하여, 해당 시연을 조금 더 강조하는 상황을 정의
- 학습 비중 변화가 policy parameters와 expected return J에 미치는 영향을 연결하여 기여도를 계산
- 양의 값은 해당 시연의 비중을 높이면 성능이 좋아질 것으로, 음의 값은 나빠질 것으로 추정함을 의미

$$\theta_\varepsilon = \arg\min_\theta \left[\mathcal{L}_{\text{BC}}(\theta) + \varepsilon \ell_{\text{traj}}(\xi;\theta)\right]$$

$$\Psi_{\pi-inf}(\xi) = \left.\frac{dJ(\pi_{\theta_\varepsilon})}{d\varepsilon}\right|_{\varepsilon=0}$$

Proposed method

시연이 행동에 미친 영향을 어떻게 작업 성능에 대한 기여도로 바꿀까?

❖ Action influence에 rollout의 성공과 실패를 반영

- 학습 데이터의 (s, a)가 rollout에서 관측된 행동 (s', a')의 log-likelihood에 미치는 영향을 action influence로 계산
- 각 demonstration의 action influence를 rollout 전체에 걸쳐 합산하고, 성공 +1·실패 −1의 return으로 가중하여 평균
- 개별 행동의 정답을 제공하지 않고, 실행 전체의 결과를 활용하여 demonstration의 performance influence를 추정

① 행동 단위의 영향 계산

$$\Psi_{a\text{-inf}}\big((s', a'), (s, a)\big) = -\nabla_\theta \log \pi_\theta(a'|s')^\top H_{\mathrm{BC}}^{-1} \nabla_\theta \ell(s, a; \theta)$$

② 행동에 미친 영향을 합산하고, rollout의 return으로 가중

$$\hat{\Psi}_{\pi\text{-inf}}(\xi) = \frac{1}{m}\sum_{i=1}^{m} R(\tau_i)\left[\frac{1}{H}\sum_{(s\prime,a\prime)\in\tau_i}\sum_{(s,a)\in\xi}\Psi_{a\text{-inf}}((s', a'), (s, a))\right]$$

Proposed method

계산한 영향도로 어떤 demonstration을 제거하거나 추가할까?

❖ 제거·추가에 따른 성능 변화를 근사하여 k개의 시연을 선별

- Performance influence를 이용해 demonstration 하나를 제거하거나 추가한 뒤 재학습했을 때의 expected return 변화를 1차 근사
- Filter-k는 기여도가 낮거나 해로운 시연을 제거하고, Select-k는 성능 개선에 유리한 시연을 추가
- 선별 단위는 demonstration 전체이며, 재구성된 데이터로 최종 policy를 다시 학습

구분	Task 1: Filter-k	Task 2: Select-k
평가 대상	학습 데이터 $\mathcal{D}$	추가 학습 후보 시연 집합 $\mathcal{H}$
Leave-one-out / Add-one-in 영향도	$\Psi^{\text{out}}_{\pi\text{-inf}}(\xi) := -\dfrac{\widehat{\Psi}_{\pi\text{-inf}}(\xi)}{\lvert\mathcal{D}\rvert}$ $\approx J\left(\pi_{\theta(\mathcal{D}\setminus\{\xi\})}\right) - J\left(\pi_{\theta(\mathcal{D})}\right)$	$\Psi^{\text{in}}_{\pi\text{-inf}}(\xi) := \dfrac{\widehat{\Psi}_{\pi\text{-inf}}(\xi)}{\lvert\mathcal{D}\rvert}$ $\approx J\left(\pi_{\theta(\mathcal{D}\cup\{\xi\})}\right) - J\left(\pi_{\theta(\mathcal{D})}\right)$
k개 선별	$S^{\star}_{\text{out}} = \arg\text{top-}k\left(\{\Psi^{\text{out}}_{\pi\text{-inf}}(\xi^i) : \xi^i \in \mathcal{D}\}\right)$	$S^{\star}_{\text{in}} = \arg\text{top-}k\left(\{\Psi^{\text{in}}_{\pi\text{-inf}}(\xi^i) : \xi^i \in \mathcal{H}\}\right)$
재구성된 데이터	$\mathcal{D} \setminus S^{\star}_{\text{out}}$	$\mathcal{D} \cup S^{\star}_{\text{in}}$

Proposed method

Rollout이 적어 영향도 추정이 불안정하면 어떻게 보완할까?

❖ CUPID-Quality는 performance influence에 품질 기준을 결합

- Rollout 수가 적으면 performance influence 추정의 변동성이 커질 수 있으므로, quality score의 결합 효과를 검토
- Quality score는 reward를 사용하지 않고 action influence의 극단적이거나 잡음이 큰 값을 불리하게 평가하는 heuristic
- 두 점수를 정규화한 뒤 결합하며, 품질 기준의 추가가 실제 성능에 도움이 되는지는 작업별로 확인

$$\Psi(\xi) = \alpha\tilde{\Psi}_{\pi\text{-inf}}(\xi) + (1-\alpha)\tilde{\Psi}_{\text{qual}}(\xi)$$

$$\alpha = \begin{cases} 1, & \text{CUPID} \\ 0.5, & \text{CUPID-Quality} \end{cases}$$

Experimental setup

어떤 조건과 비교 방법으로 데이터 선별의 효과를 검증했을까?

❖ 시뮬레이션과 실제 로봇에서 서로 다른 선별 기준을 비교

- RoboMimic의 Lift·Square·Transport에서는 state-based Diffusion Policy를 평가
- Franka의 Figure-8·TuckBox·Bookshelf에서는 image-based Diffusion Policy를 평가
- Curation rollout 예산은 시뮬레이션 100회, 실제 로봇 25회로 제한하며, 재학습한 policy의 작업 성공률을 비교

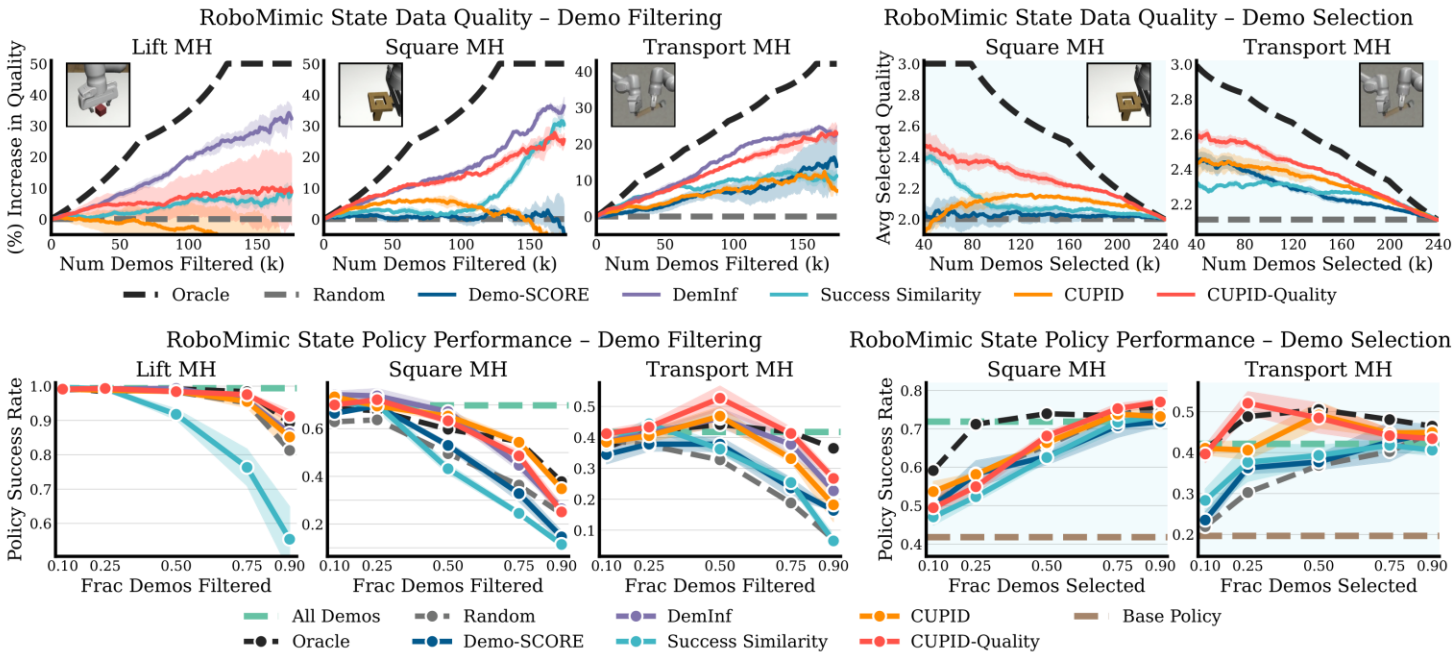
비교 방법	데이터 선별 기준
DemInf	다양성과 예측 가능성을 활용한 offline 정제 (filtering만 적용)
Demo-SCORE	성공·실패 rollout의 상태를 구분하는 classifier
Success similarity	성공 rollout과의 상태 유사성
Random	무작위 선별
Oracle	품질·전략 등의 사전 라벨을 활용한 선별

Experimental results

데이터 품질을 높이거나 제거량을 늘리면 성능도 좋아질까?

❖ 데이터 품질과 제거량만으로 policy 성능을 설명하기 어려움

- DemInf가 높은 품질의 시연을 남기더라도, CUPID 계열로 학습한 policy가 더 높은 성공률을 보이는 경우가 존재
- Transport에서는 CUPID-Quality가 원래 300개 시연의 33% 미만으로 공식 Diffusion Policy보다 높은 성공률을 달성
- Lift는 일부 데이터만으로 높은 성능을 유지하지만, Square는 시연 제거에 따라 성능이 하락하여 적절한 선별량이 작업마다 다름



Experimental results

실제 로봇에서도 quality score를 추가하면 항상 유리할까?

❖ 품질 기준의 결합 효과는 작업에 따라 달라짐

- 품질이 혼재된 Figure-8에서는 데이터 선별이 정밀한 로프 조작 성능을 높이며, CUPID-Quality가 추가적인 개선을 보임
- 반면 전략 변화와 spurious correlation이 중요한 TuckBox·Bookshelf에서는 CUPID-Quality의 성능이 크게 낮아짐
- 변동성이 크더라도 유용한 전략을 제공하는 시연을 품질 heuristic이 불리하게 평가할 수 있음

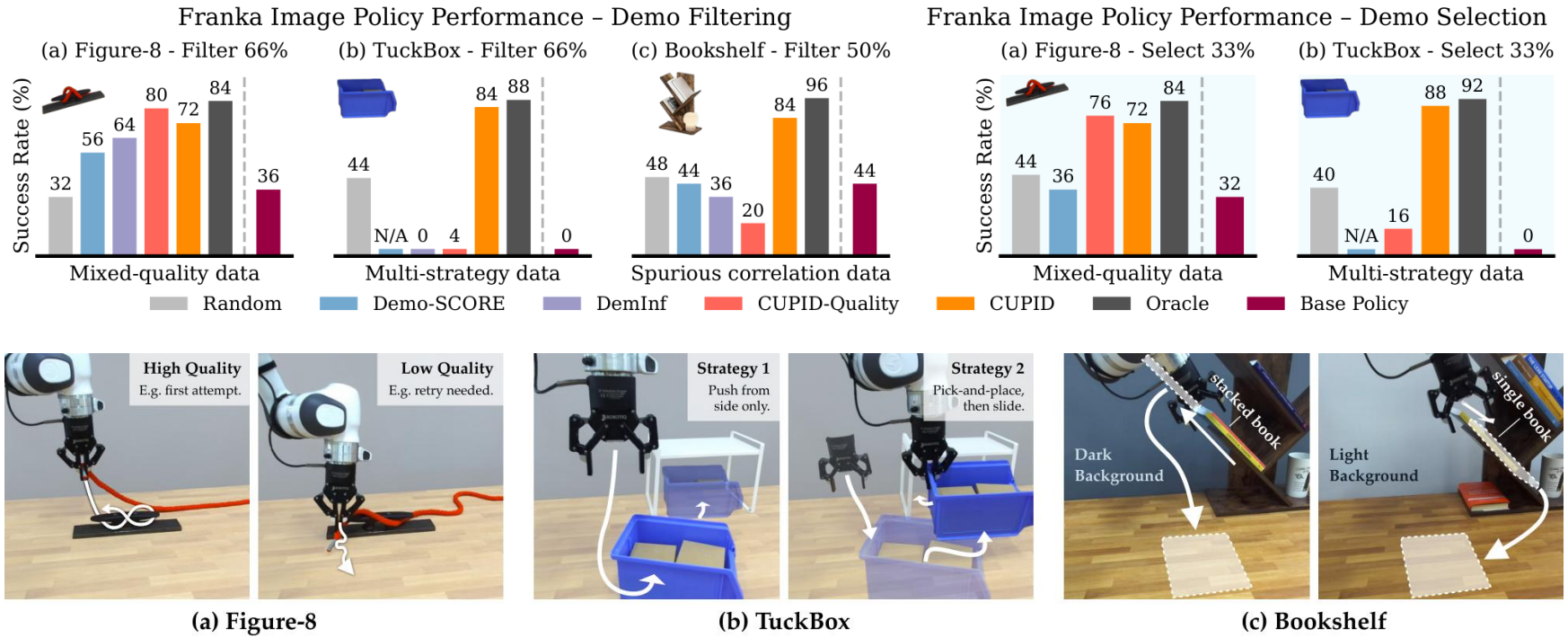
작업 · 주요 조건	제거 비율	Base policy	CUPID	CUPID-Quality
Figure-8 · 품질 혼재	66%	36%	72%	**80%**
TuckBox · 전략 변화	66%	0%	**84%**	4%
Bookshelf · spurious correlation	50%	44%	**84%**	20%

Experimental results

성공한 rollout이 없어도 새로운 환경에 유용한 전략을 고를 수 있을까?

❖ 실패에 기여한 시연을 찾아 기존 데이터의 전략 비중을 조정

- TuckBox는 sliding과 pick-and-place 기반 시연이 2:1로 구성되며, 평가 시 상자의 질량 분포 변화로 sliding이 불안정해짐
- Base policy의 rollout이 모두 실패한 상황에서도, CUPID는 기존 데이터에 포함된 대안 전략의 활용을 높임
- 성공률 0%에서 filtering은 84%, selection은 88%로 개선하며, 선별 데이터에서 robust한 전략의 비중이 증가



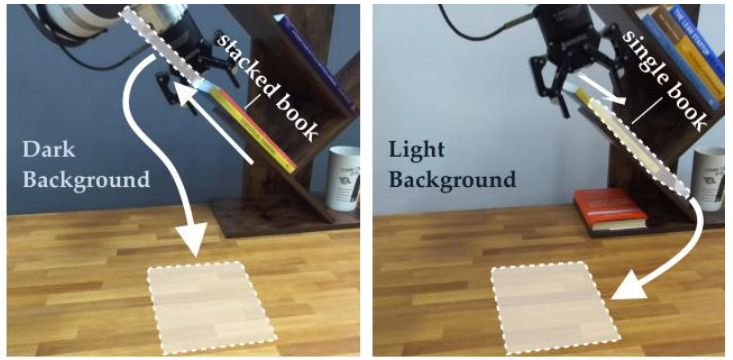
Experimental results

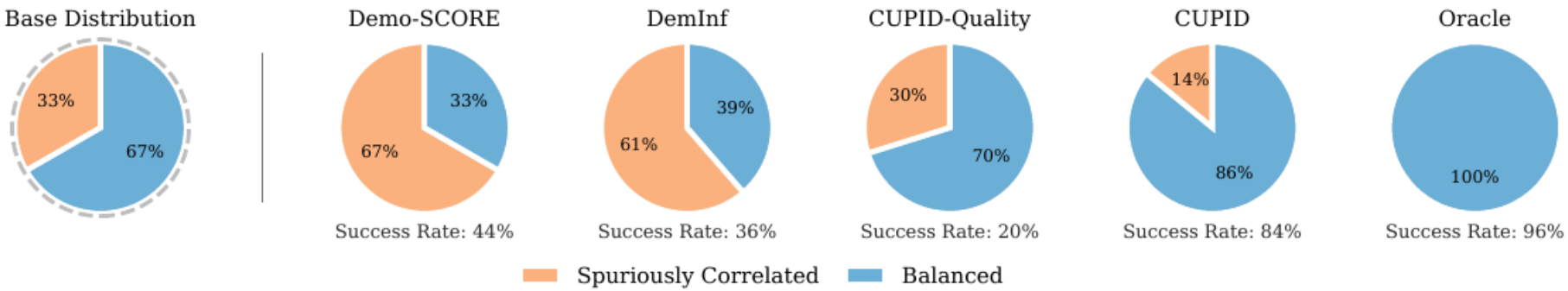
작업과 무관한 배경에 의존하는 문제도 데이터 선별로 줄일 수 있을까?

❖ 작업과 무관한 배경-행동 간 편향을 데이터 선별로 완화

- Bookshelf에서는 배경색과 책을 꺼내는 방향 사이에 허위 상관관계가 존재
- CUPID는 실패에 영향을 주는 시연을 식별하고 제거하여 데이터 구성을 개선
- 데이터 50% 제거 후 성공률 44% → 84% 향상

Bookshelf





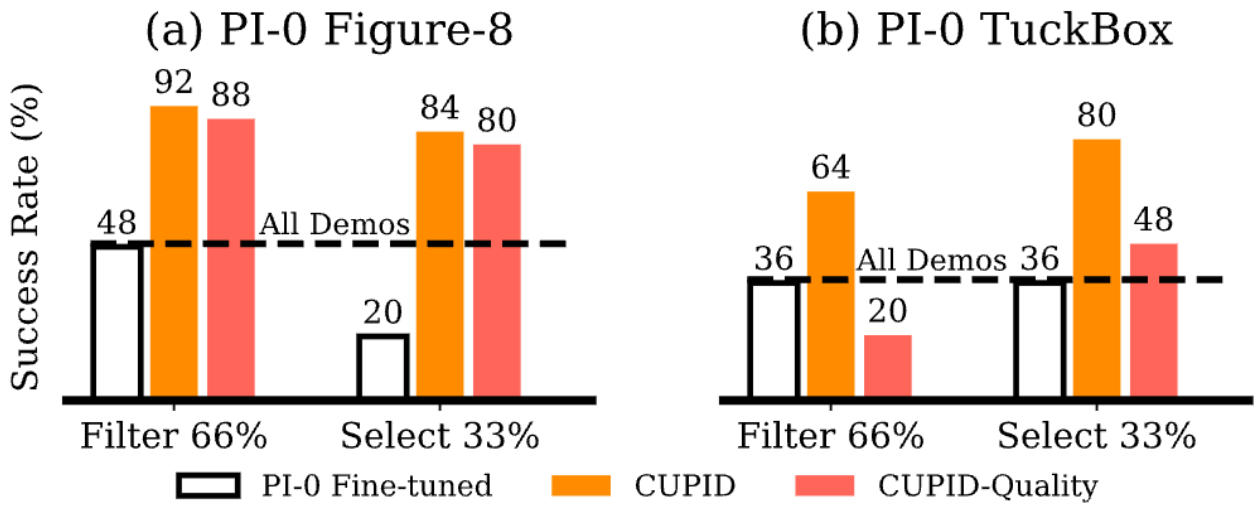
(c) **Bookshelf:** Distribution of curated demonstrations after *filtering* 50%. Balanced data is better.

Experimental results

작은 policy로 선별한 데이터가 VLA 학습에도 도움이 될까?

❖ Diffusion Policy로 선별한 데이터가 π_0의 fine-tuning 성능도 개선

- 단일 작업 Diffusion Policy를 기준으로 선별한 데이터를 π_0의 LoRA fine-tuning에 활용
- Figure-8은 filtering에서 48%에서 92%, selection에서 20%에서 84%로 성공률 향상
- TuckBox는 filtering에서 36%에서 64%, selection에서 36%에서 80%로 성공률 향상

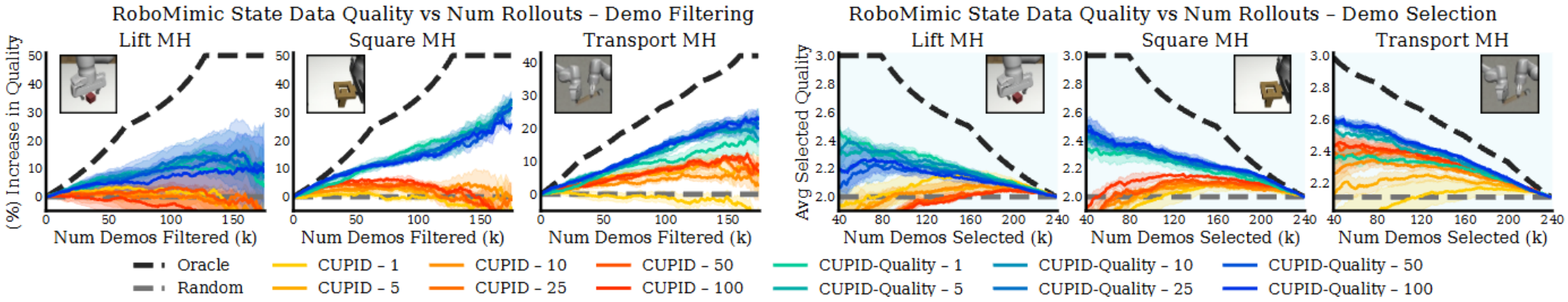


Experimental results

유용한 시연을 선별하려면 policy를 얼마나 실행해야 할까?

❖ 검증한 작업에서는 수십 회의 rollout으로 선별 경향이 안정화

- Lift·Square에서는 약 25–50회, Transport에서는 약 50–100회에서 선별 경향이 안정화
- Curation에서는 영향도의 정확한 절댓값보다 유용한 시연을 상대적으로 높은 순위에 배치하는 것이 중요
- 실제 Franka 실험에서는 25회 rollout을 사용하며, 더 복잡한 작업은 더 많은 평가 경험이 필요할 수 있음



Conclusion

CUPID는 무엇을 개선했을까?

❖ 잘 수행된 시연이면 policy 학습에도 도움이 될까?

- 시연의 품질과 학습 기여도가 다를 수 있음을 확인하고, closed-loop 성능에 미치는 영향으로 시연을 평가

❖ 기여도를 알려면 시연마다 policy를 다시 학습해야 할까?

- 학습된 policy와 평가 rollout을 활용한 Influence Function으로 각 시연의 성능 기여도를 근사

❖ 계산한 기여도를 데이터 선별에 어떻게 활용할까?

- 기여도가 낮은 기존 시연은 Filter-k로 제거하고, 높은 후보 시연은 Select-k로 추가

❖ 이렇게 선별한 데이터가 실제 작업 성능도 개선할까?

- 실제 로봇에서 취약한 전략과 데이터 편향의 영향을 완화하고, π_0 미세조정에서도 선별 데이터의 성능 개선 효과를 확인

Conclusion

결국 어떤 데이터를 남겨야 할까?

❖ SCIZOR: 데이터의 품질과 중복성을 고려한 선별

- Task progress를 이용해 비효율적인 행동 구간을 찾아 제거
- 영상과 행동의 유사도를 함께 고려하여 중복 경험을 제거
- 별도의 품질 레이블 없이 데이터를 정제하여, 더 적은 데이터로 정책 성능을 높일 수 있음을 확인

❖ CUPID: 정책 성능에 대한 기여도를 고려한 선별

- 정책의 rollout 결과를 바탕으로 각 시연의 성능 기여도를 추정
- 성능에 해로운 시연을 제거하거나, 추가 후보에서 유용한 시연을 선택
- 시연 자체의 품질이 높더라도 현재 정책의 성능 향상에 항상 도움이 되는 것은 아님을 확인

Thank you