SAC-AE

김동영

무엇을 다루는가?

- 이미지와 같은 고차원 입력을 받는 이미지 컨트롤 테스크 환경

Model-free + policy-off + latent space 를 통합한 프레임 워크 제공

기존의 문제점

기존 : 이미지 입력을 받아서 학습 시키는 경우 그라디언트 신호가 보상에 관련된 약한 신호만 존재 => 학습 효율이 매우 떨어짐

기존 : representation과 control task를 분리해서 학습

둘을 잘 통합하는 방법이 많이 어렵다.

기존의 문제점

- 둘을 통합하는 형태는 많이 있어왔다.

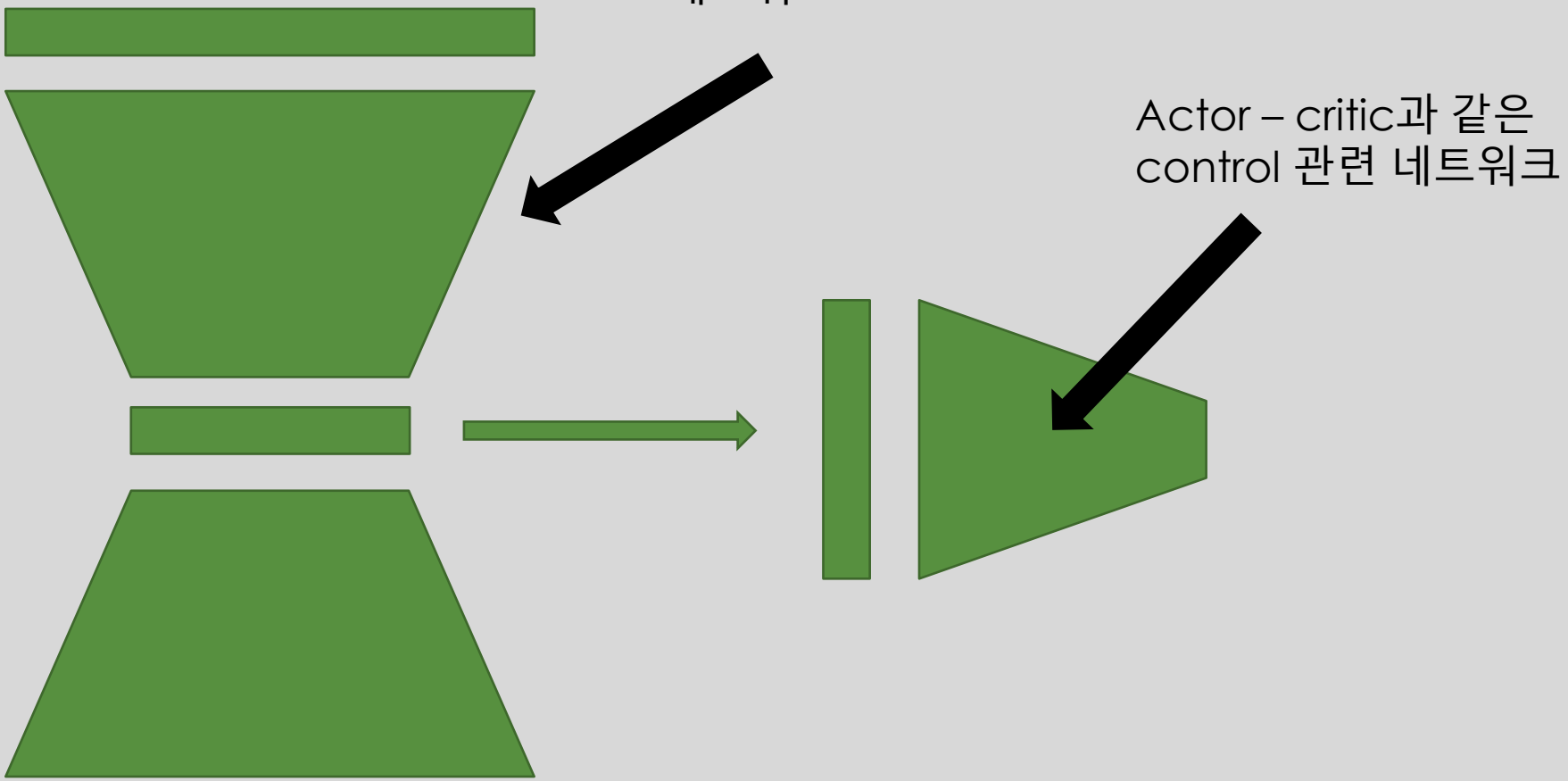
- SLAC
- PlaNet
- Dreamer
- Simple
- 기타 등등

- 각자 느낌이 다르고 각 아키텍처 방식에 대한 설명은 생략

기존의 문제점

VAE 와 같은 representation 추출 관련 네트워크

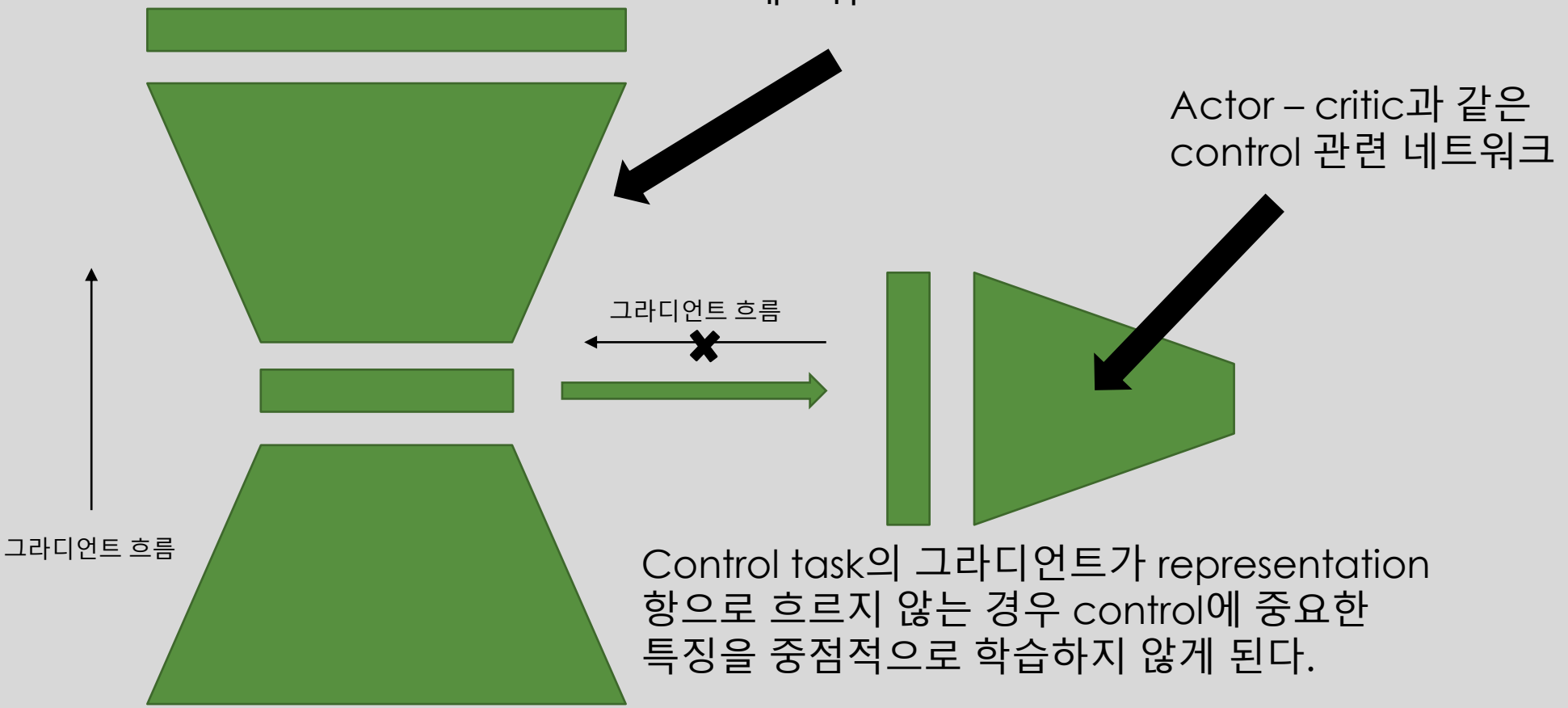
Actor – critic과 같은 control 관련 네트워크



기존의 문제점

VAE 와 같은 representation 추출 관련 네트워크

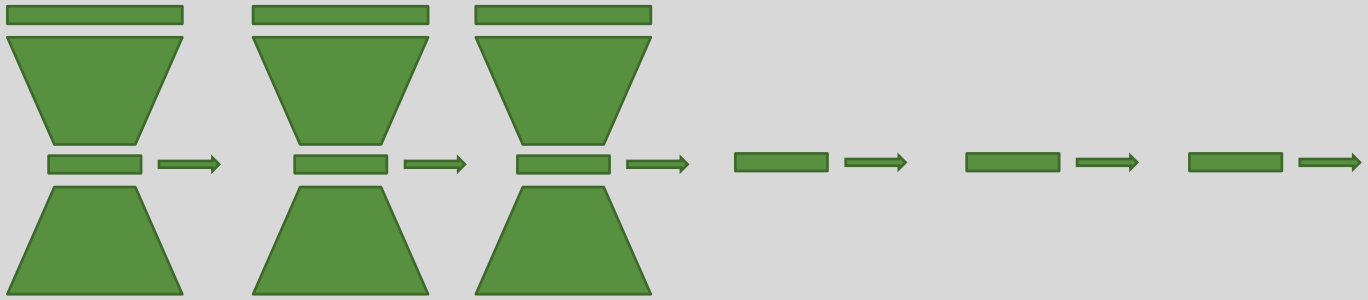
Actor – critic과 같은 control 관련 네트워크



Control task의 그라디언트가 representation 항으로 흐르지 않는 경우 control에 중요한 특징을 중점적으로 학습하지 않게 된다.

기존의 문제점

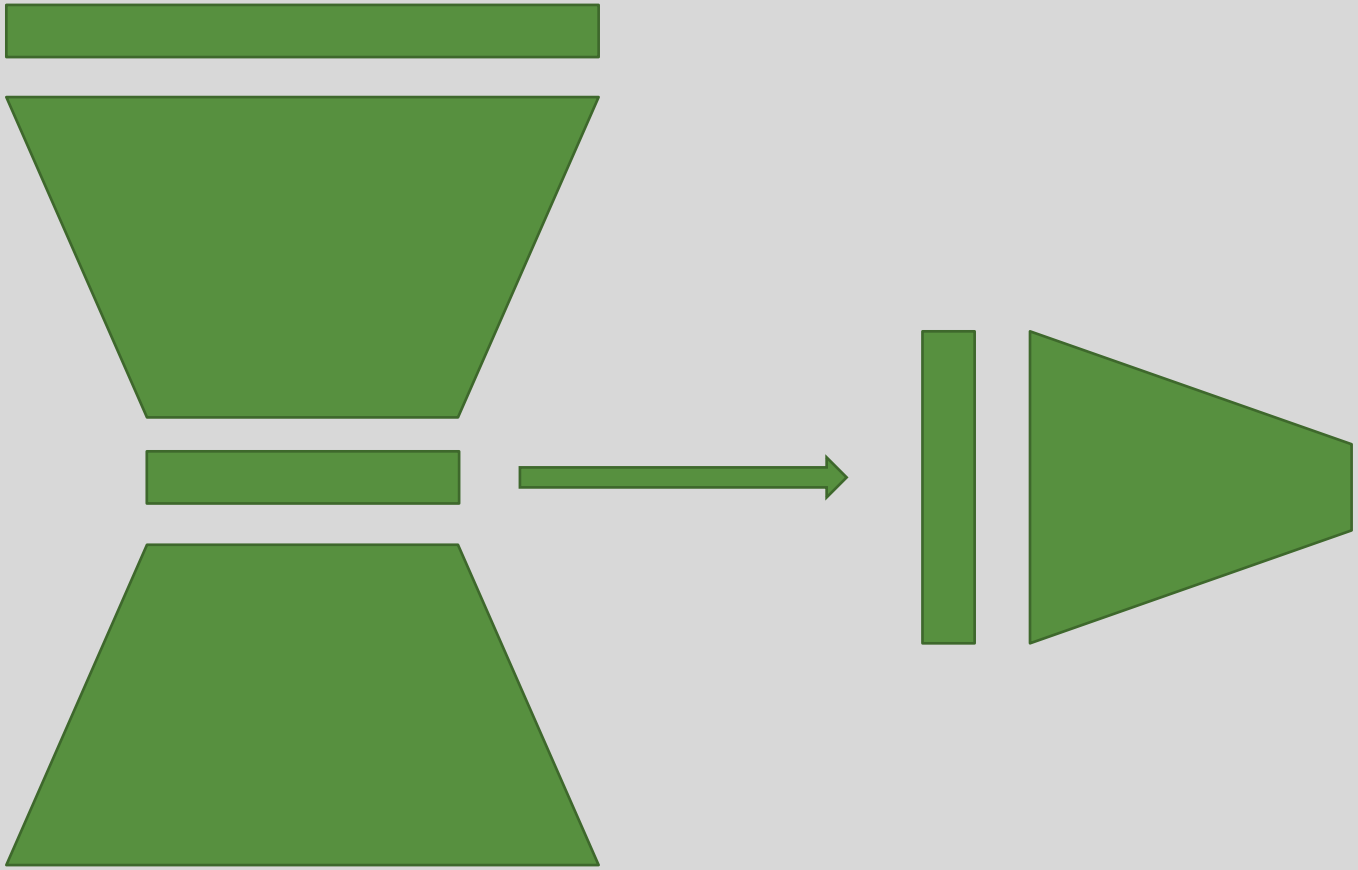
Model base 방식에 항상 들어가는 형태인
transition model



노이즈에 민감

기존의 문제점

On policy -> 약간의 robust 부족



그래서 SAC-AE는 뭘 다루는가

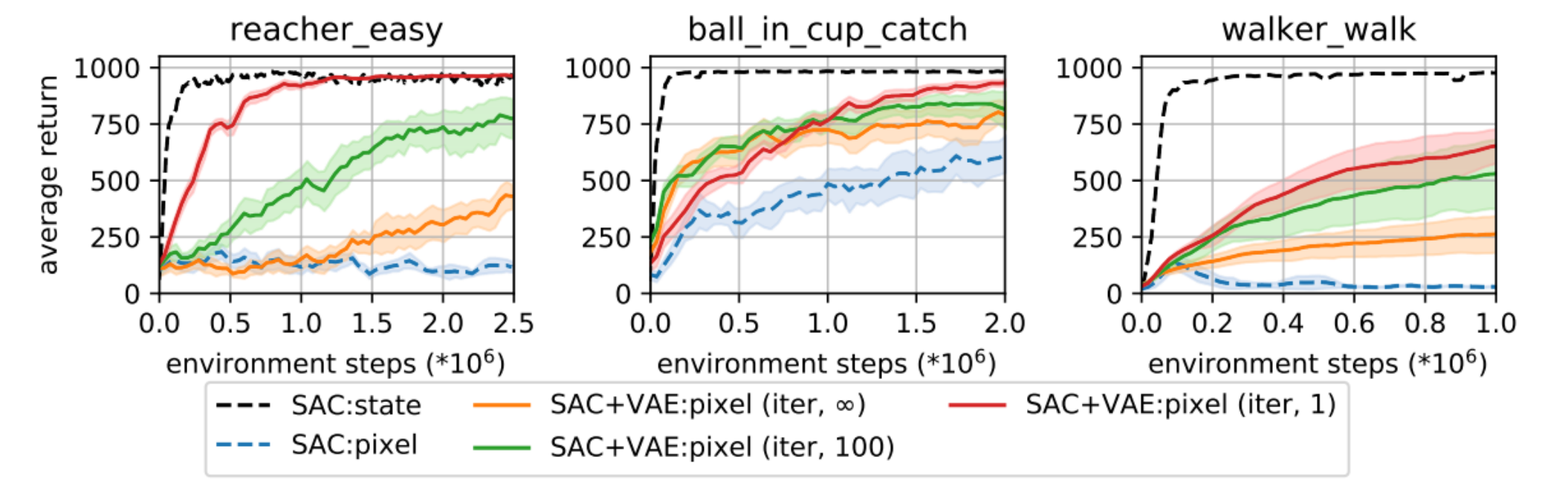
Model free RL + AutoEncoder + off-policy

Beta – VAE / AE

$$J(\mathrm{VAE}) = \mathbb{E}_{\mathbf{o}_t \sim \mathcal{D}}\Big[\mathbb{E}_{\mathbf{z}_t \sim q_\phi(\mathbf{z}_t|\mathbf{o}_t)}[\log p_\theta(\mathbf{o}_t|\mathbf{z}_t)] - \beta D_{\mathrm{KL}}(q_\phi(\mathbf{z}_t|\mathbf{o}_t)||p(\mathbf{z}_t))\Big],$$

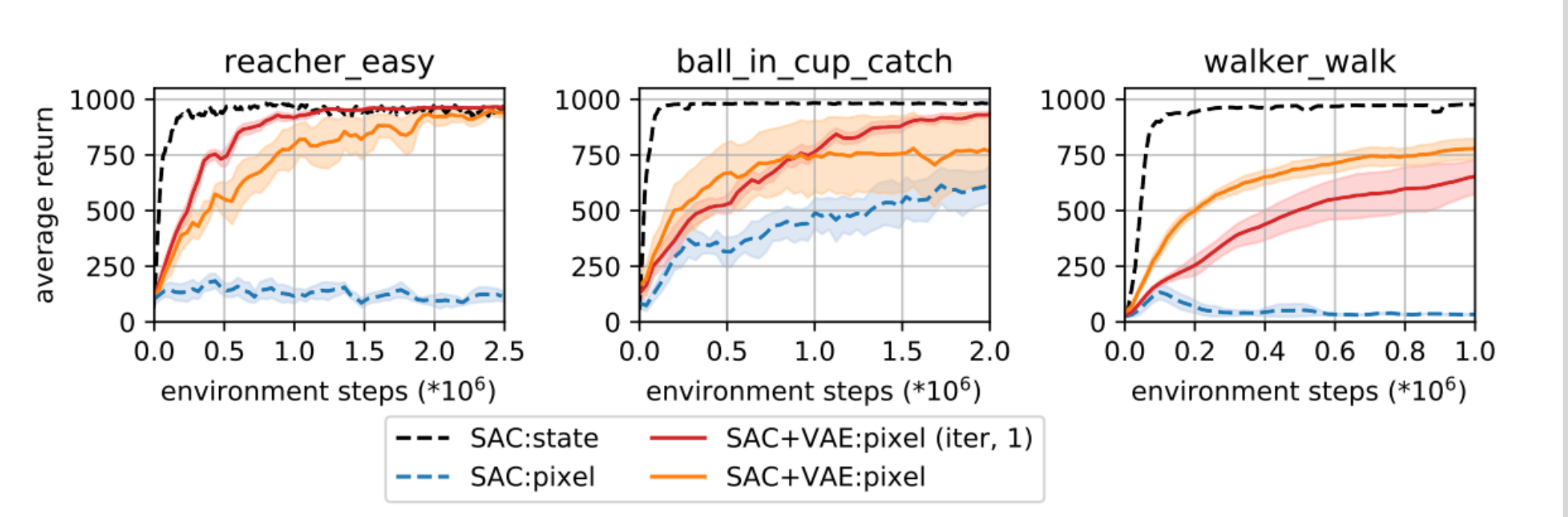
$$J(\mathrm{AE}) = \mathbb{E}_{\mathbf{o}_t \sim \mathcal{D}}\big[\log p_\theta(\mathbf{o}_t|\mathbf{z}_t)\big]$$

VAE 추가



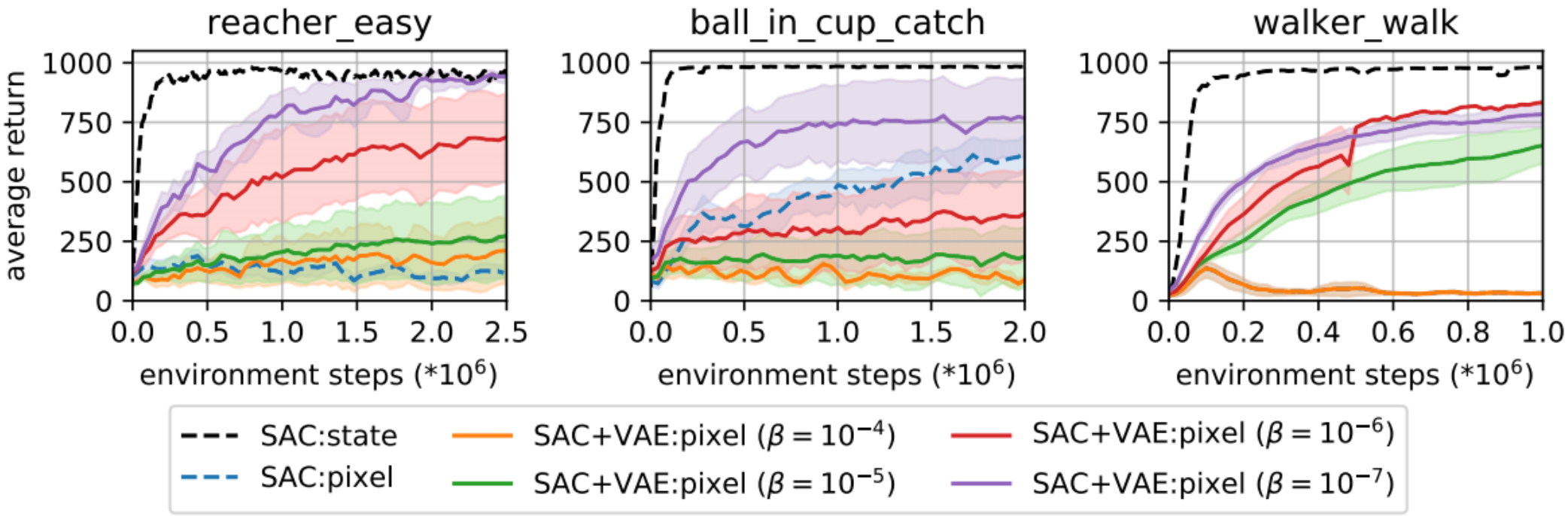
(Iter, x) : x-> 오토인코더와 강화학습
모델 자체의 학습 반복 주기

Actor – Critic 그라디언트 흐름 추가



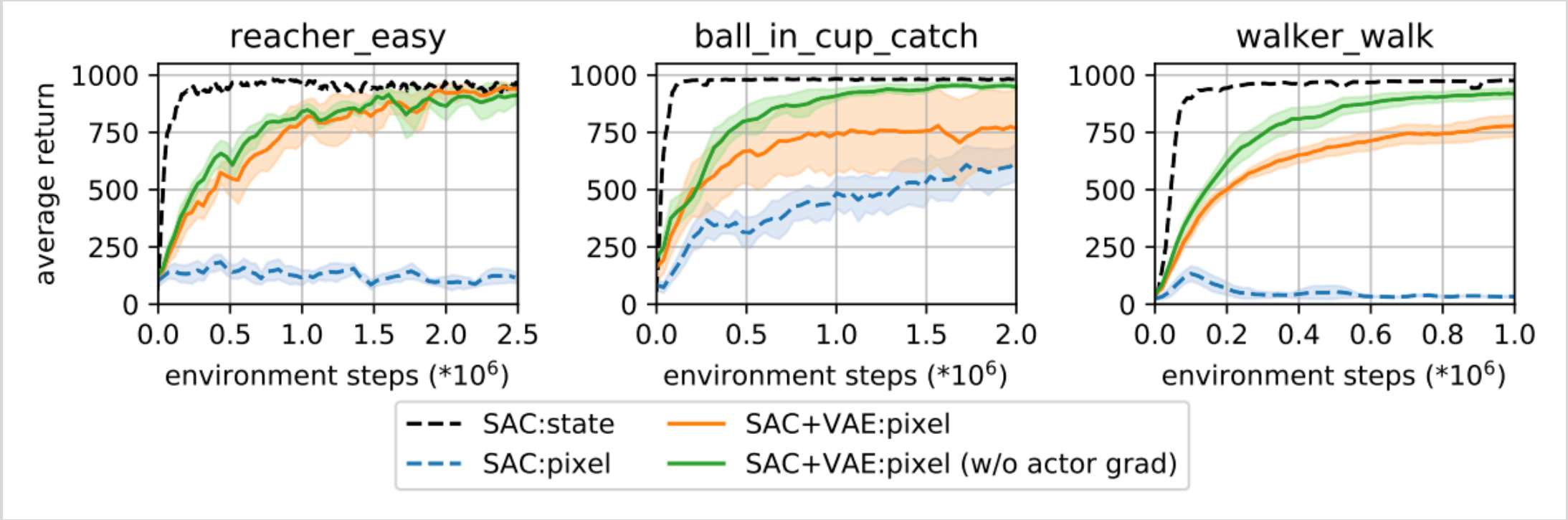
성능은 증가하는데 항상 증가하는 것도 아니다.

Beta 값 변경

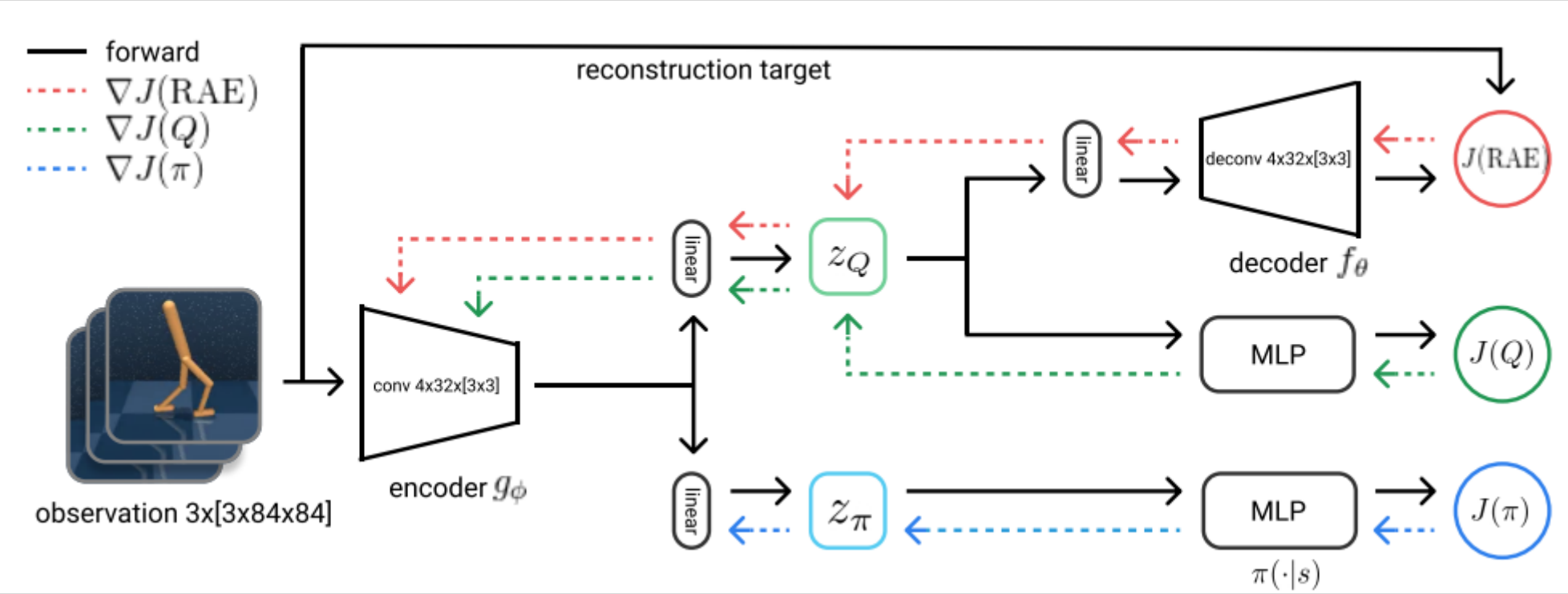


Beta 가 작을수록 => AE 에 가까울 수록 성능 증가

Actor 그라디언트 제거



아키텍처


forward
$\nabla J(\mathrm{RAE})$
$\nabla J(Q)$
$\nabla J(\pi)$
reconstruction target
observation 3x[3x84x84]
conv 4x32x[3x3]
encoder g_ϕ
linear
z_Q
linear
deconv 4x32x[3x3]
decoder f_θ
$J(\mathrm{RAE})$
MLP
$J(Q)$
linear
z_π
MLP
$\pi(\cdot|s)$
$J(\pi)$

Actor 그라디언트

$$J(\pi) = \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}}\left[D_{\mathrm{KL}}(\pi(\cdot|\mathbf{s}_t)||\mathcal{Q}(\mathbf{s}_t, \cdot))\right],$$

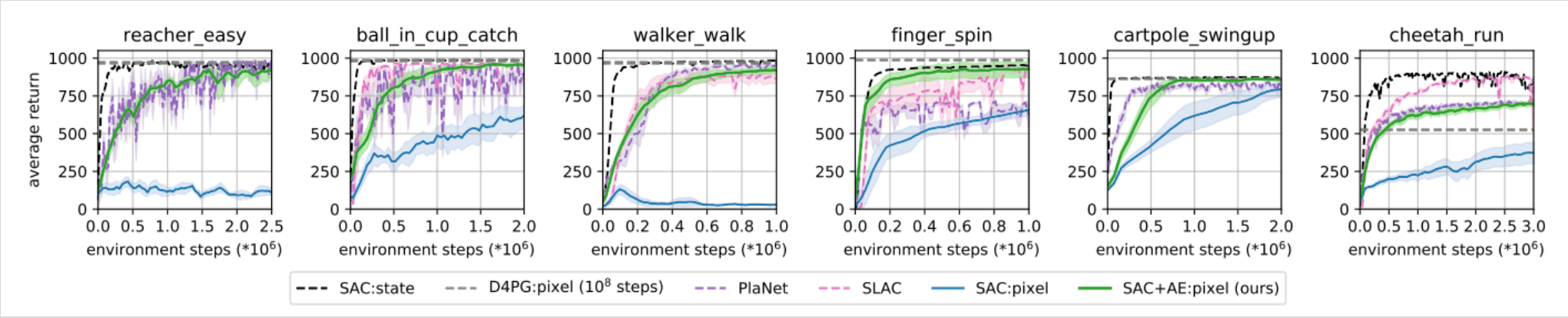
where $\mathcal{Q}(\mathbf{s}_t, \cdot) \propto \exp\{\frac{1}{\alpha}Q(\mathbf{s}_t, \cdot)\}$.

- Actor 안의 그라디언트에 Q와 관련된 항이 있는데 이것 때문에 타겟이 흔들리는 상황이 발생해서이지 않을까 라고 분석한다.

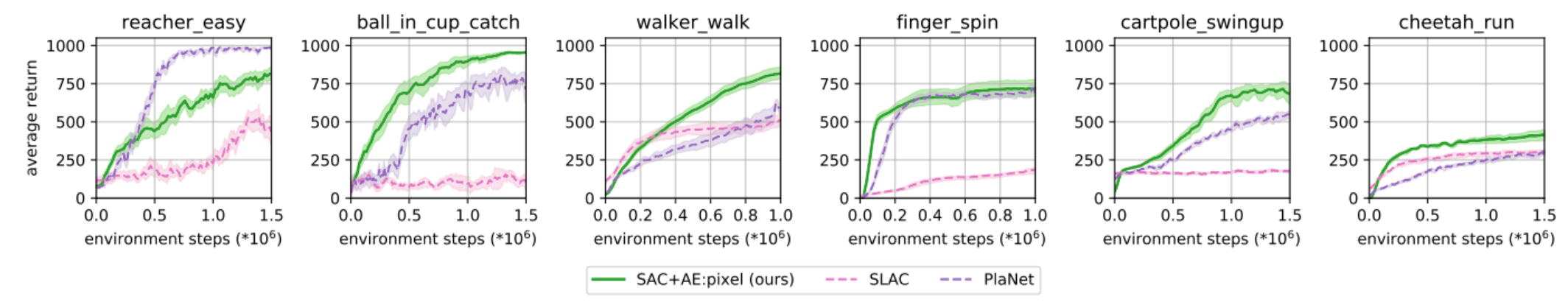
구현

- AE 사용
- L1, L2 정규화 이용
- Actor의 그라디언트를 사용 안하고
- Critic이 중요하기에 Critic의 학습 속도를 Actor 보다 빠르게 진행 <= 안정성 + representation 항 그라디언트 제공

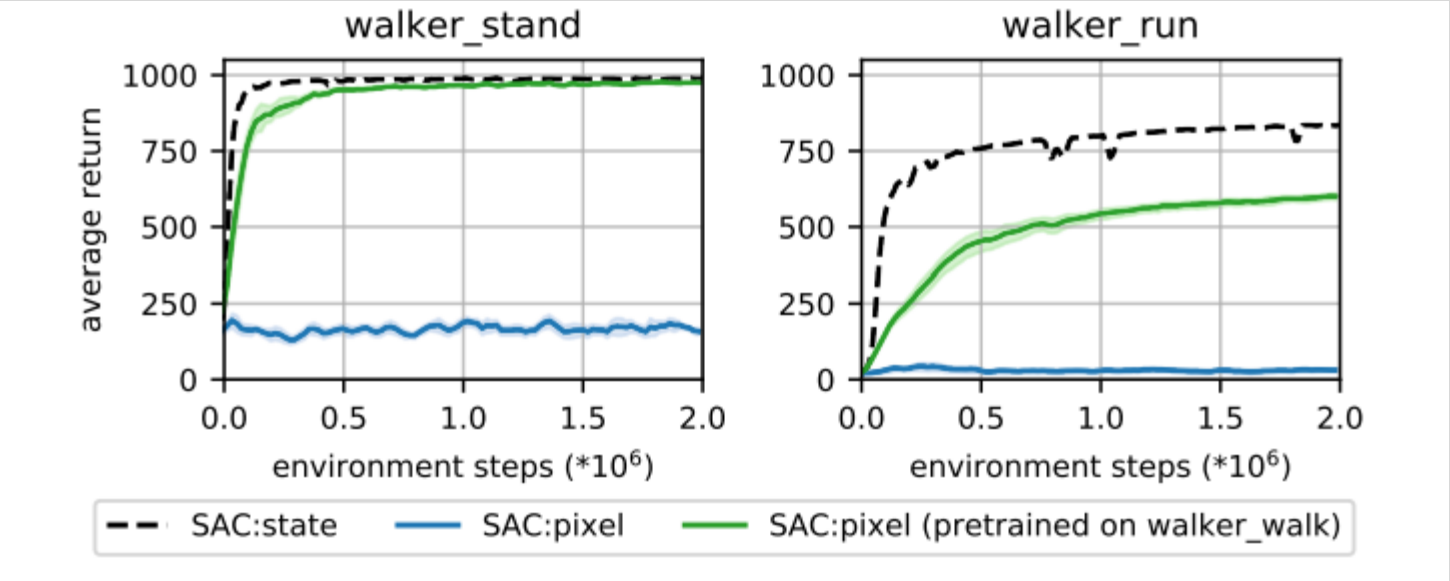
성능


reacher_easy
ball_in_cup_catch
walker_walk
finger_spin
cartpole_swingup
cheetah_run
average return
environment steps (*10^6)
SAC:state
D4PG:pixel (10^8 steps)
PlaNet
SLAC
SAC:pixel
SAC+AE:pixel (ours)

성능 - 노이즈


reacher_easy
ball_in_cup_catch
walker_walk
finger_spin
cartpole_swingup
cheetah_run
average return
environment steps (*10^6)
SAC+AE:pixel (ours)
SLAC
PlaNet

성능 – multi task



끝