TAAC: Temporally Abstract Actor-Critic for Continuous Control

Haonan Yu, *Wei Xu*, and *Haichao Zhang*
Horizon Robotics

Horizon Robotics

Motivation: temporal abstraction (TA)

Temporally correlated actions

Motivation: temporal abstraction (TA)

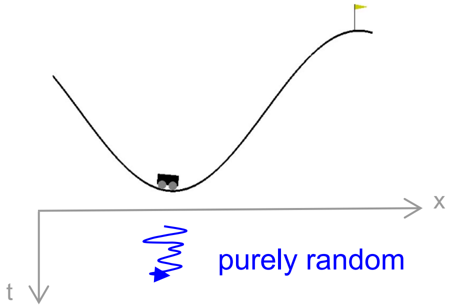
Temporally correlated actions

1. Temporally persistent exploration

Motivation: temporal abstraction (TA)

Temporally correlated actions

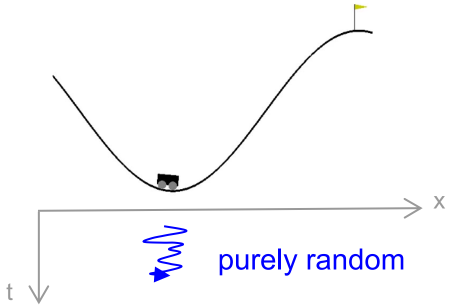
1. Temporally persistent exploration

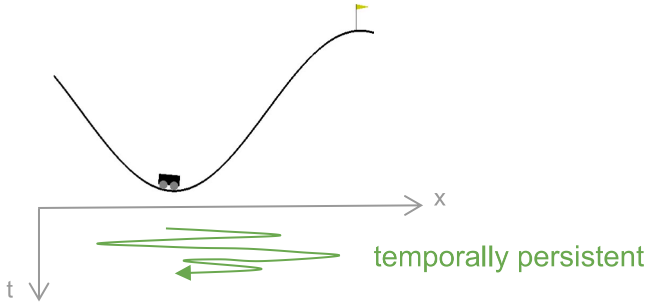


Motivation: temporal abstraction (TA)

Temporally correlated actions

1. Temporally persistent exploration





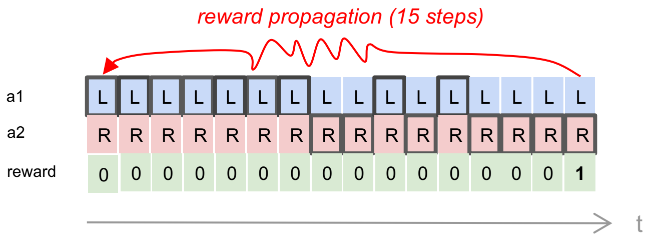
Moving out of the valley more easily!

Motivation: temporal abstraction (TA)

2. Better credit assignment with delayed reward (shorter horizon)

Motivation: temporal abstraction (TA)

2. Better credit assignment with delayed reward (shorter horizon)

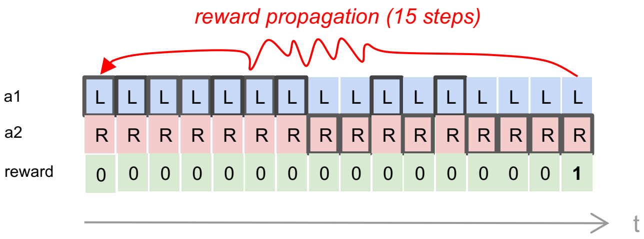


a1	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
a2	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
reward	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**

Take actions at every step

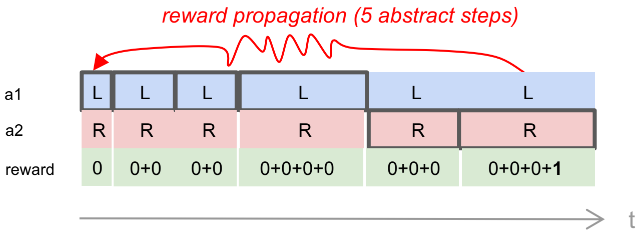
Motivation: temporal abstraction (TA)

2. Better credit assignment with delayed reward (shorter horizon)



a1	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
a2	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
reward	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**

Take actions at every step



a1	L	L	L	L	L	L
a2	R	R	R	R	R	R
reward	0	0+0	0+0	0+0+0+0	0+0+0	0+0+0+**1**

Take actions only at each abstract step

Open-loop *vs.* closed-loop repetition

Action repetition: perhaps the simplest temporal abstraction technique.

Open-loop *vs.* closed-loop repetition

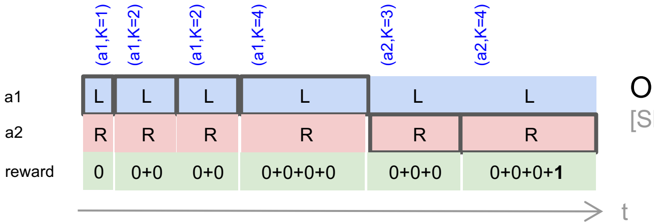
Action repetition: perhaps the simplest temporal abstraction technique.

Key questions: *what action to repeat & how long to repeat it?*

Open-loop *vs.* closed-loop repetition

Action repetition: perhaps the simplest temporal abstraction technique.

Key questions: *what action to repeat & how long to repeat it?*

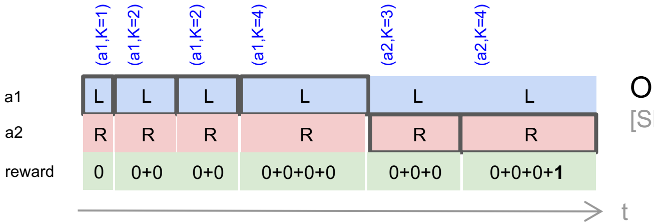


Open-loop methods output an action and its duration at once
[Sharma et al., 2017] [Biedenkapp et al., 2021] [Dabney et al., 2021]

Open-loop *vs.* closed-loop repetition

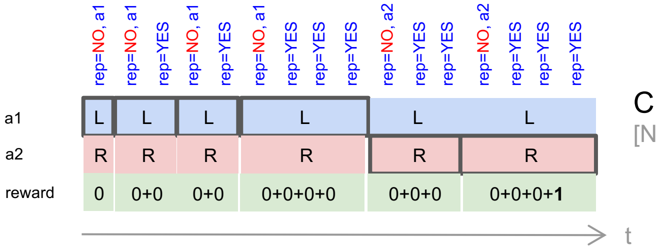
Action repetition: perhaps the simplest temporal abstraction technique.

***Key questions:** what action to repeat & how long to repeat it?*



	(a1,K=1)	(a1,K=2)	(a1,K=2)	(a1,K=4)	(a2,K=3)	(a2,K=4)
a1	L	L	L	L	L	L
a2	R	R	R	R	R	R
reward	0	0+0	0+0	0+0+0+0	0+0+0	0+0+0+**1**

Open-loop methods output an action and its duration at once
[Sharma et al., 2017] [Biedenkapp et al., 2021] [Dabney et al., 2021]



	rep=NO, a1	rep=NO, a1 rep=YES	rep=NO, a1 rep=YES	rep=NO, a1 rep=YES rep=YES rep=YES	rep=NO, a2 rep=YES rep=YES	rep=NO, a2 rep=YES rep=YES rep=YES
a1	L	L	L	L	L	L
a2	R	R	R	R	R	R
reward	0	0+0	0+0	0+0+0+0	0+0+0	0+0+0+**1**

Closed-loop methods decide “act-or-repeat” at every step
[Neunert et al., 2020] [Chen et al., 2021]

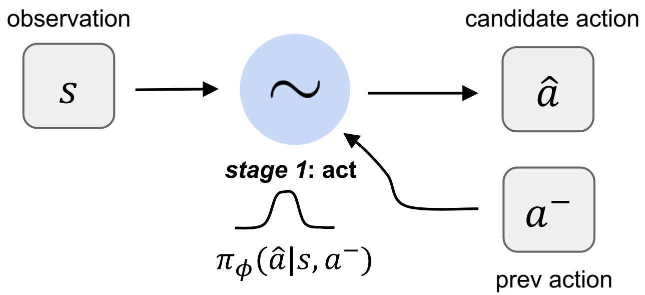
Temporally abstract actor-critic (TAAC)

TAAC incorporates closed-loop action repetition into off-policy actor-critic for continuous control

Temporally abstract actor-critic (TAAC)

TAAC incorporates closed-loop action repetition into off-policy actor-critic for continuous control

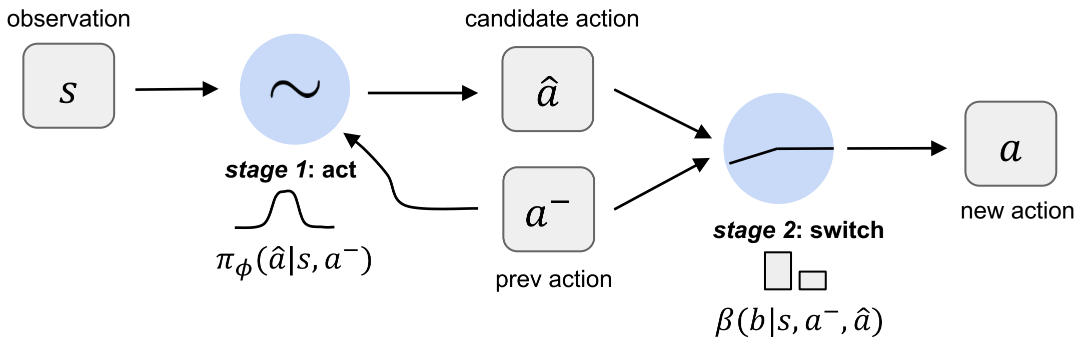
TAAC's two-stage policy



Temporally abstract actor-critic (TAAC)

TAAC incorporates closed-loop action repetition into off-policy actor-critic for continuous control

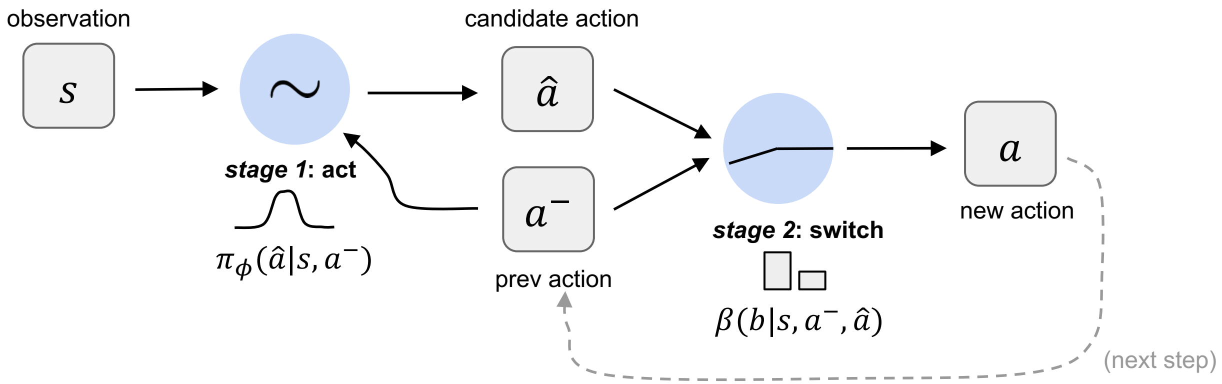
TAAC's two-stage policy



Temporally abstract actor-critic (TAAC)

TAAC incorporates closed-loop action repetition into off-policy actor-critic for continuous control

TAAC's two-stage policy

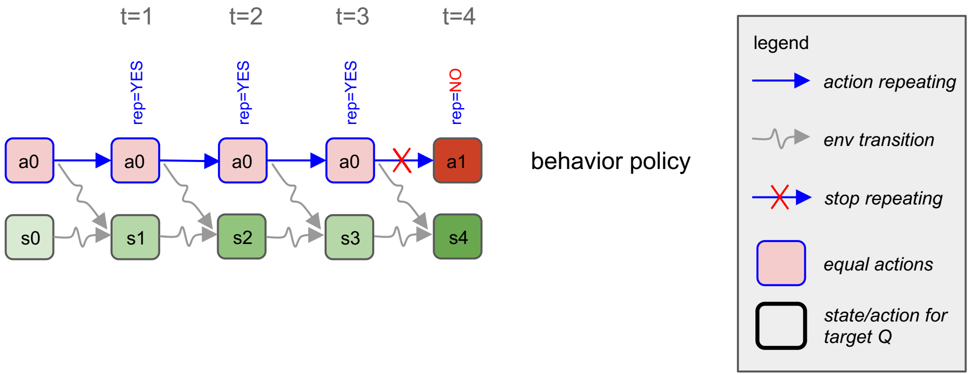


Policy evaluation

A novel compare-through operator for multi-step TD backup

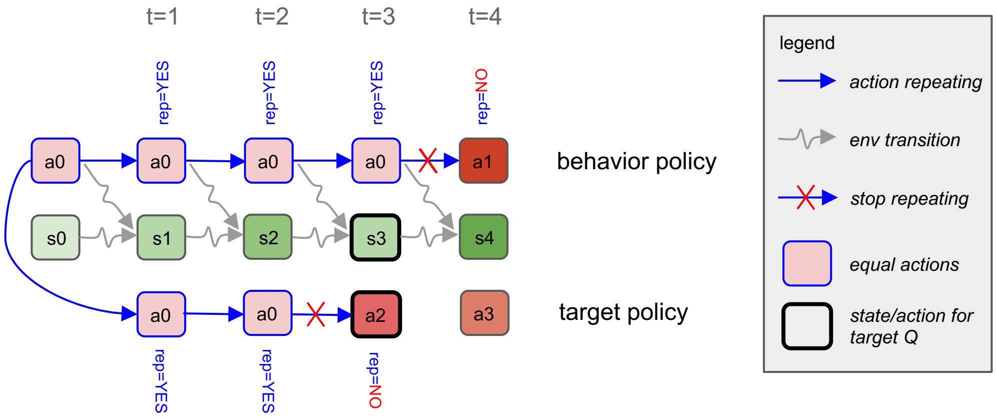
Policy evaluation

A novel compare-through operator for multi-step TD backup



Policy evaluation

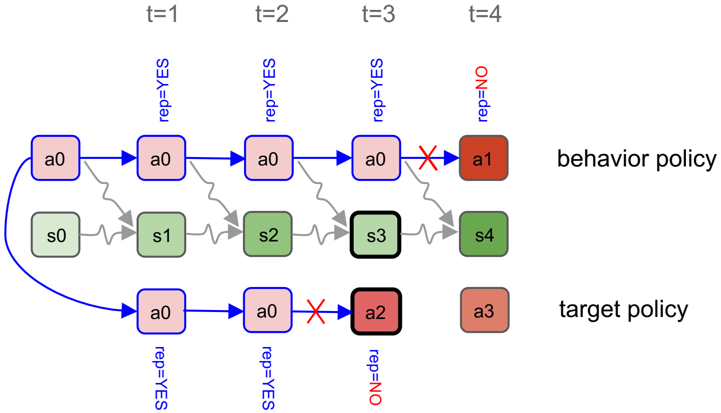
A novel compare-through operator for multi-step TD backup



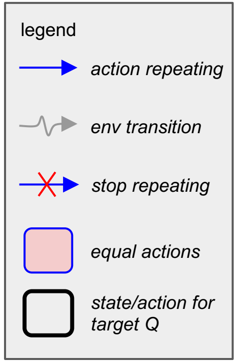
Example 1: We use Q(s3,a2) as the target to bootstrap Q(s0,a0) (3-step TD)

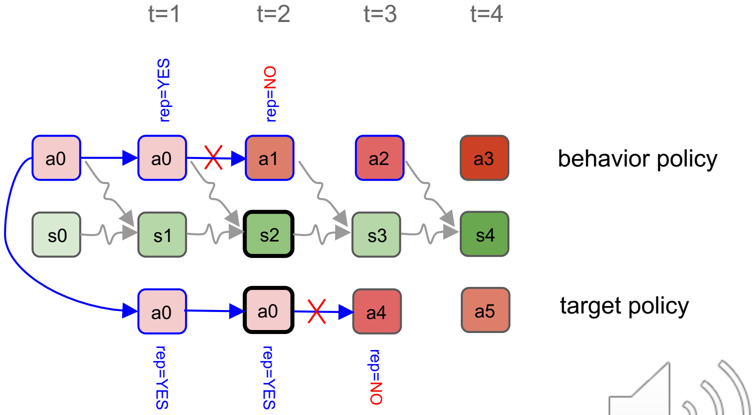
Policy evaluation

A novel compare-through operator for multi-step TD backup



Example 1: We use Q(s3,a2) as the target to bootstrap Q(s0,a0) (3-step TD)





Example 2: We use Q(s2,a0) as the target to bootstrap Q(s0,a0) (2-step TD)

Policy improvement

A closed-form solution for the switching policy to speed up policy learning: sampling two actions by the exponential Q values

Policy improvement

A closed-form solution for the switching policy to speed up policy learning: sampling two actions by the exponential Q values

$$\beta(0)^* = \exp\left(\frac{Q(s, a^-)}{\alpha}\right) / Z(s) \qquad \beta(1)^* = \exp\left(\frac{Q(s, \hat{a})}{\alpha}\right) / Z(s)$$

Policy improvement

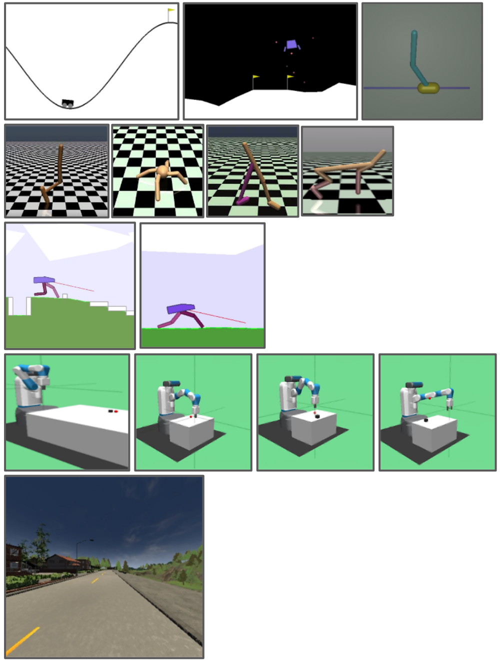
A closed-form solution for the switching policy to speed up policy learning: sampling two actions by the exponential Q values

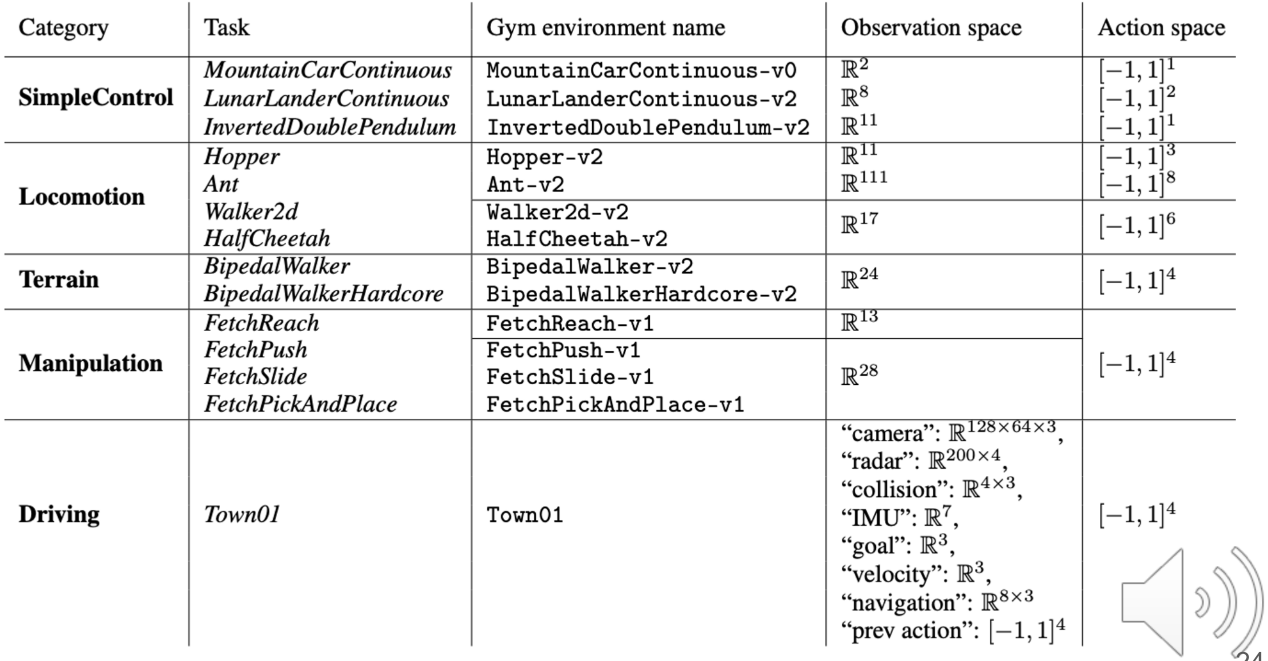
$$\beta(0)^* = \exp\left(\frac{Q(s, a^-)}{\alpha}\right)/Z(s) \qquad \beta(1)^* = \exp\left(\frac{Q(s, \hat{a})}{\alpha}\right)/Z(s)$$

The actor $\pi_\phi(\hat{a}|s, a^-)$is trained similarly as in DDPG [Lillicrap et al., 2016] and SAC [Haarnoja et al., 2018]: $\frac{\partial Q(s, \hat{a})}{\partial \phi}\beta^*(1)$ (this is a good approximation to the full gradient)

Tasks

5 categories of 14 continuous control tasks (13 standard; 1 customized)





Category	Task	Gym environment name	Observation space	Action space
SimpleControl	*MountainCarContinuous*	`MountainCarContinuous-v0`	$\mathbb{R}^2$	$[-1,1]^1$
	LunarLanderContinuous	`LunarLanderContinuous-v2`	$\mathbb{R}^8$	$[-1,1]^2$
	InvertedDoublePendulum	`InvertedDoublePendulum-v2`	$\mathbb{R}^{11}$	$[-1,1]^1$
Locomotion	*Hopper*	`Hopper-v2`	$\mathbb{R}^{11}$	$[-1,1]^3$
	Ant	`Ant-v2`	$\mathbb{R}^{111}$	$[-1,1]^8$
	Walker2d	`Walker2d-v2`	$\mathbb{R}^{17}$	$[-1,1]^6$
	HalfCheetah	`HalfCheetah-v2`		
Terrain	*BipedalWalker*	`BipedalWalker-v2`	$\mathbb{R}^{24}$	$[-1,1]^4$
	BipedalWalkerHardcore	`BipedalWalkerHardcore-v2`		
Manipulation	*FetchReach*	`FetchReach-v1`	$\mathbb{R}^{13}$	$[-1,1]^4$
	FetchPush	`FetchPush-v1`	$\mathbb{R}^{28}$	
	FetchSlide	`FetchSlide-v1`		
	FetchPickAndPlace	`FetchPickAndPlace-v1`		
Driving	*Town01*	`Town01`	“camera”: $\mathbb{R}^{128\times64\times3}$, “radar”: $\mathbb{R}^{200\times4}$, “collision”: $\mathbb{R}^{4\times3}$, “IMU”: $\mathbb{R}^7$, “goal”: $\mathbb{R}^3$, “velocity”: $\mathbb{R}^3$, “navigation”: $\mathbb{R}^{8\times3}$ “prev action”: $[-1,1]^4$	$[-1,1]^4$

Experiment results

Comparison methods

SAC [Haarnoja et al., 2018]: flat RL

SAC-Ntd: SAC + Retrace [Munos et al., 2016] + N-TD

SAC-Nrep: SAC + Fixed action repetition

SAC-Krep: open-loop action repetition [Sharma et al., 2017; Biedenkapp et al., 2021]

SAC-EZ: SAC with EZ-greedy [Dabney et al., 2021]

SAC-Hybrid: closed-loop action repetition formulation from H-MPO [Neunert et al., 2020] with SAC backbone

TAAC-1td: TAAC without the compare-through operator

Experiment results

Comparison methods

SAC [Haarnoja et al., 2018]: flat RL

SAC-Ntd: SAC + Retrace [Munos et al., 2016] + N-TD

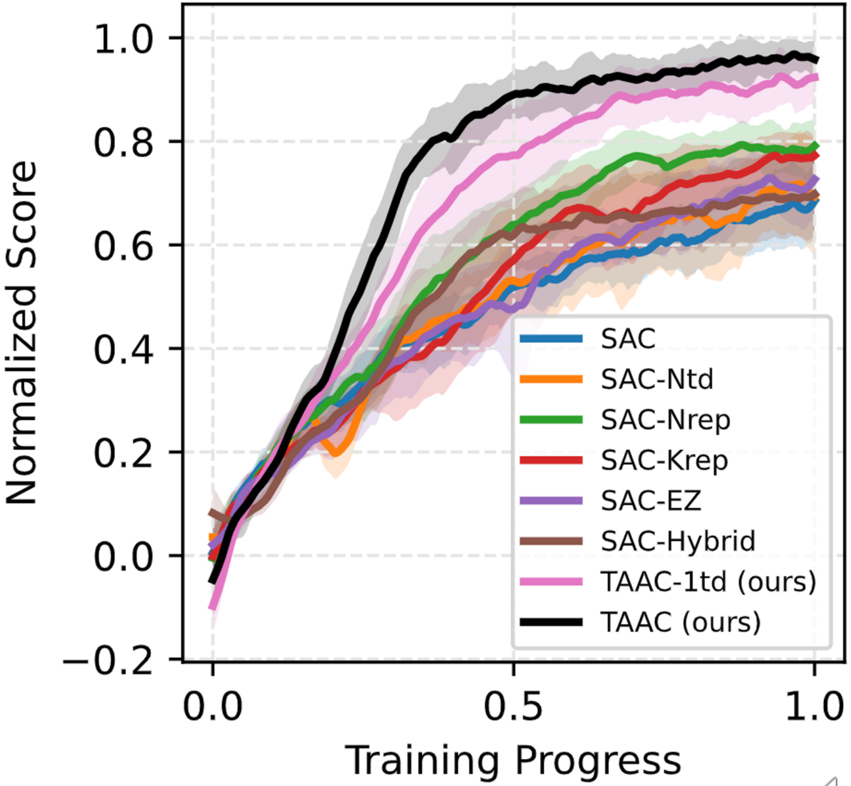
SAC-Nrep: SAC + Fixed action repetition

SAC-Krep: open-loop action repetition [Sharma et al., 2017; Biedenkapp et al., 2021]

SAC-EZ: SAC with EZ-greedy [Dabney et al., 2021]

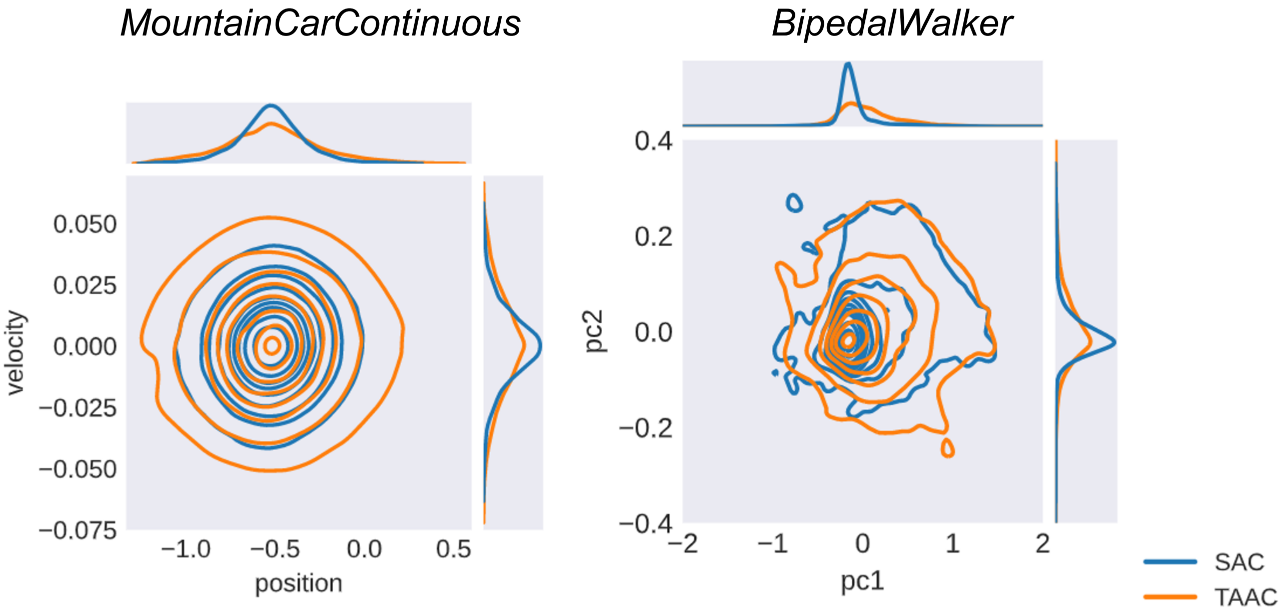
SAC-Hybrid: closed-loop action repetition formulation from H-MPO [Neunert et al., 2020] with SAC backbone

TAAC-1td: TAAC without the compare-through operator



(Normalized and averaged over 14 tasks)

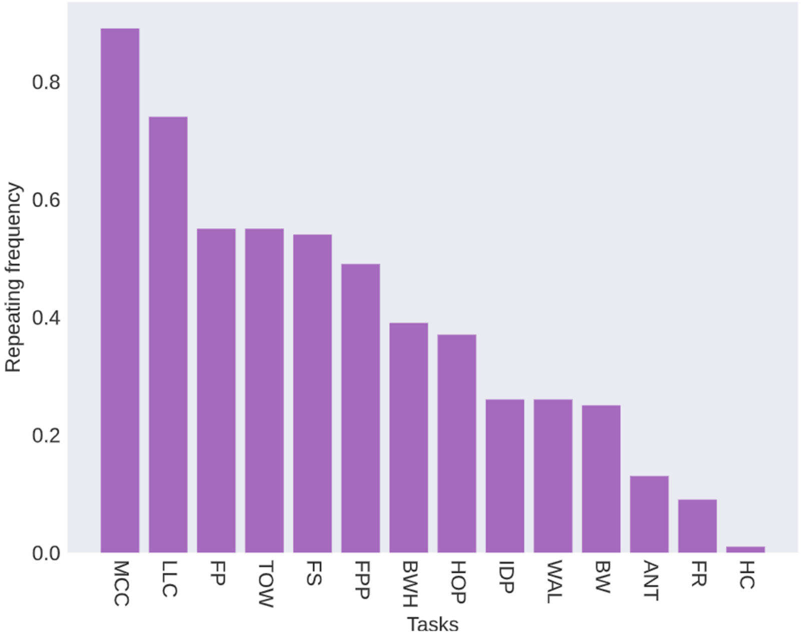
Exploration behavior analysis


MountainCarContinuous

BipedalWalker

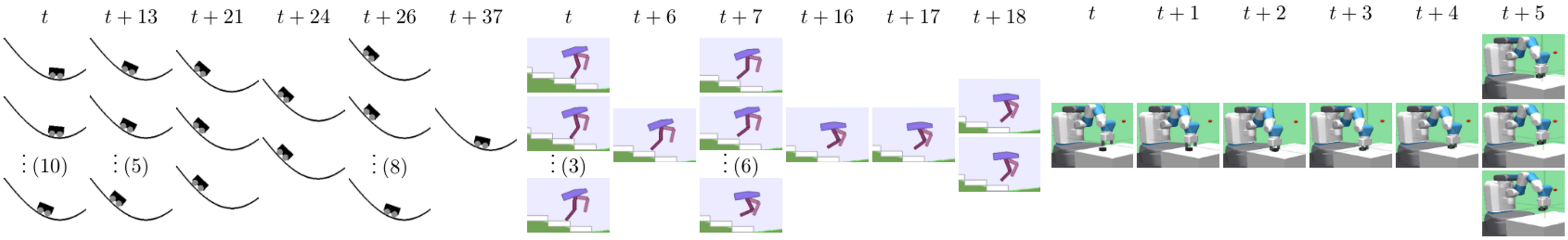
Random policies behaviors

Action repeating frequency



Evaluating a trained TAAC model
for 100 episodes and calculating the repeating frequency

Action repeating patterns

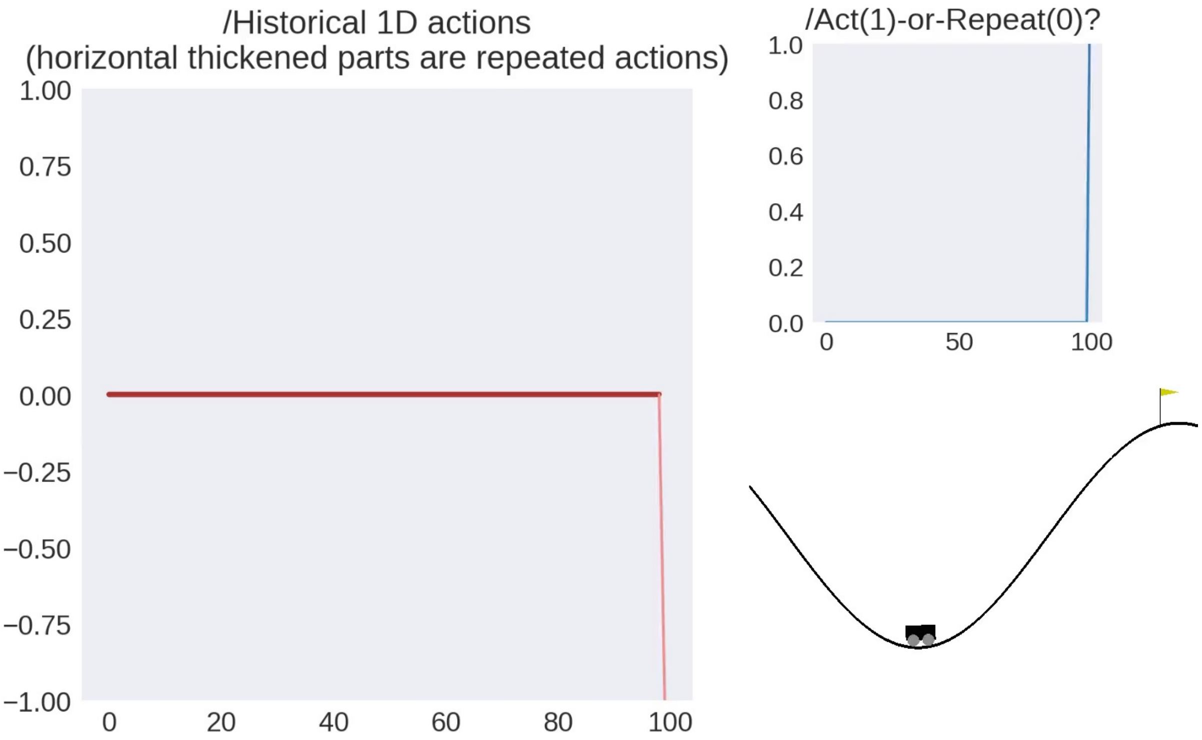


Each column represents the same action

TAAC learns to skip learning to generate new actions at non-critical states, and save the actor network's representational power for critical states!

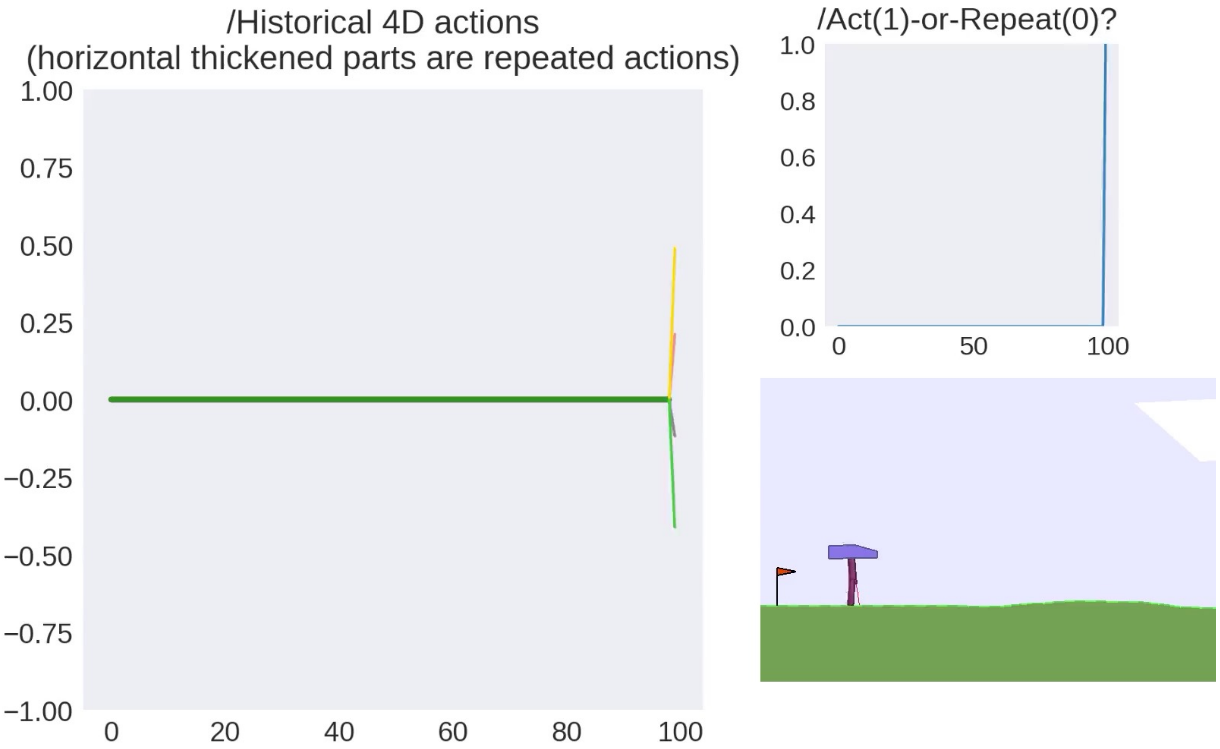
Demos

Demo - *MountainCarContinuous*



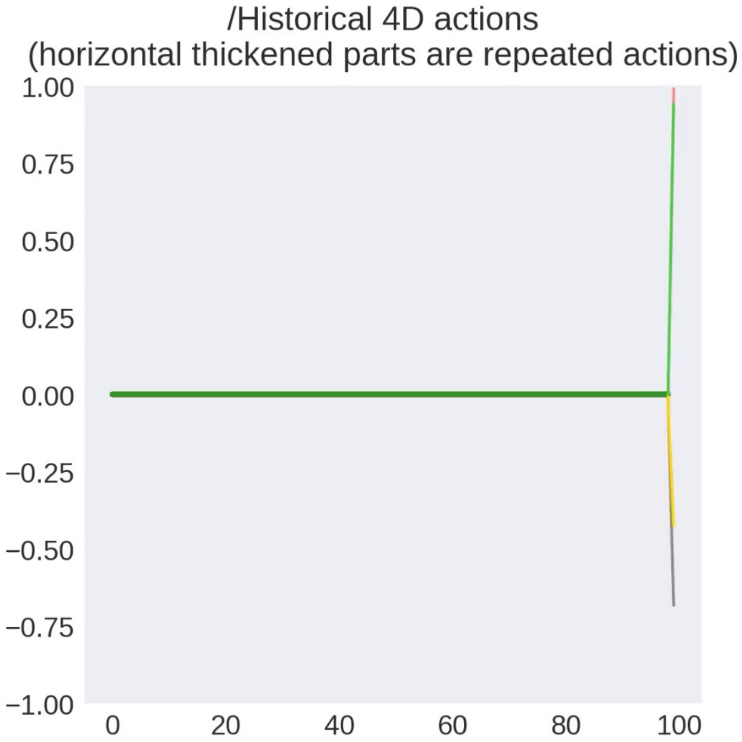
(Played in 0.5x speed for a better view)

Demo - *BipedalWalker*

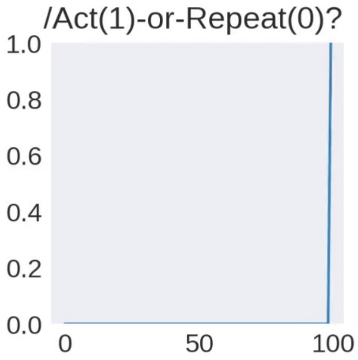


(Played in 0.5x speed for a better view)

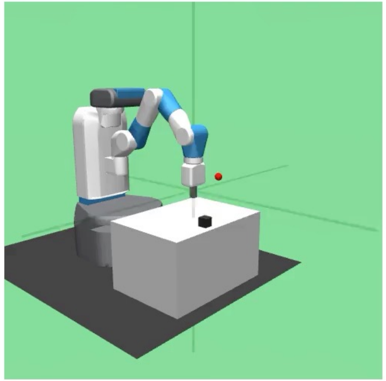
Demo - *FetchPickAndPlace*



/Historical 4D actions
(horizontal thickened parts are repeated actions)



/Act(1)-or-Repeat(0)?



(Played in 0.5x speed for a better view)

TAAC: Temporally Abstract Actor-Critic for Continuous Control

Haonan Yu, *Wei Xu*, and *Haichao Zhang*
Horizon Robotics

Code: https://github.com/hnyu/taac