Advanced Topics in Deep Reinforcement Learning

Learning Environments

Theresa Eimer

Previously in ADRL

Previously in ADRL

- Model learning in model-based RL

Previously in ADRL

- Model learning in model-based RL
- Transition function learning in next state prediction, e.g. in exploration

Previously in ADRL

- Model learning in model-based RL
- Transition function learning in next state prediction, e.g. in exploration
- Learning rewards via meta-learning or LLMs [Klissarov et al. 2023]

Previously in ADRL

- Model learning in model-based RL
- Transition function learning in next state prediction, e.g. in exploration
- Learning rewards via meta-learning or LLMs [Klissarov et al. 2023]
- Composite actions/options [Klissarov & Precup 2021, Dockhorn & Kruse 2023]

Previously in ADRL

- Model learning in model-based RL
- Transition function learning in next state prediction, e.g. in exploration
- Learning rewards via meta-learning or LLMs [Klissarov et al. 2023]
- Composite actions/options [Klissarov & Precup 2021, Dockhorn & Kruse 2023]
- State and context abstractions

Why Then Learn Full Environments?

Why Then Learn Full Environments?

- Learning simulators online can lead to errors

Why Then Learn Full Environments?

- Learning simulators online can lead to errors
- One time expense which can be re-used

Why Then Learn Full Environments?

- Learning simulators online can lead to errors
- One time expense which can be re-used
- Environment components depend on one another, so end-to-end learning might be easier than learning everything on its own

Why Then Learn Full Environments?

- Learning simulators online can lead to errors
- One time expense which can be re-used
- Environment components depend on one another, so end-to-end learning might be easier than learning everything on its own
- Synthetic environments can cut out non-informative paths in the environment to accelerate learning

Why Then Learn Full Environments?

- Learning simulators online can lead to errors
- One time expense which can be re-used
- Environment components depend on one another, so end-to-end learning might be easier than learning everything on its own
- Synthetic environments can cut out non-informative paths in the environment to accelerate learning
- We can try to learn new environments instead of imitating existing ones

Representing Environments

What we want:

Representing Environments

What we want:

- Incorporating temporal dependencies
- As easy to learn as possible
- Able to generate complex transitions
- Fast inference for RL training

Representing Environments

- NN predicting (s_{t+1}, r_{t+1})

Representing Environments

- NN predicting (s_{t+1}, r_{t+1}) -> Temporal dependencies?

Representing Environments

- NN predicting (s_{t+1}, r_{t+1}) -> Temporal dependencies?
- RNN predicting (s_{t+1}, r_{t+1})

Representing Environments

- NN predicting (s_{t+1}, r_{t+1}) -> Temporal dependencies?
- RNN predicting (s_{t+1}, r_{t+1}) -> Easy to learn?

Representing Environments

- NN predicting (s_{t+1}, r_{t+1}) -> Temporal dependencies?
- RNN predicting (s_{t+1}, r_{t+1}) -> Easy to learn?
- Separate transition and reward prediction

Representing Environments

- NN predicting (s_{t+1}, r_{t+1}) -> Temporal dependencies?
- RNN predicting (s_{t+1}, r_{t+1}) -> Easy to learn?
- Separate transition and reward prediction -> Complex transition?

Representing Environments

- NN predicting (s_{t+1}, r_{t+1}) -> Temporal dependencies?
- RNN predicting (s_{t+1}, r_{t+1}) -> Easy to learn?
- Separate transition and reward prediction -> Complex transition?

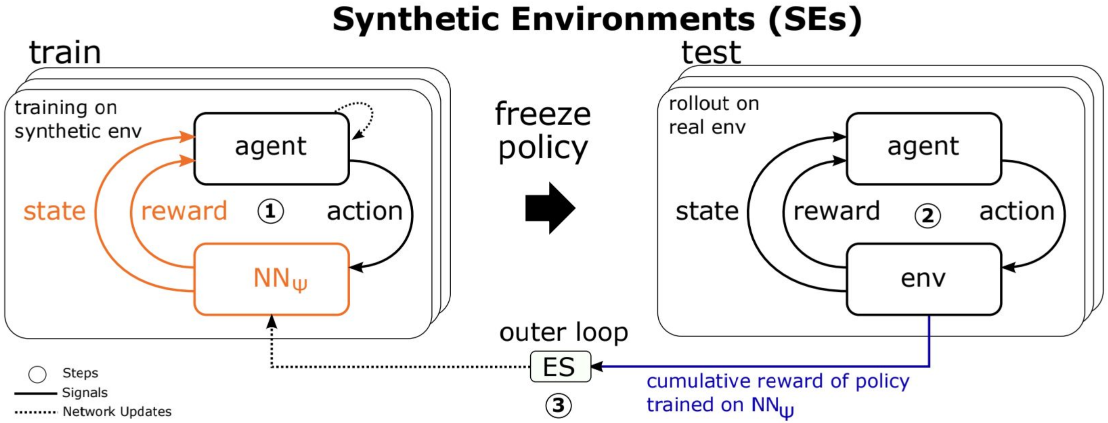
This question is possibly the key to learning environments

Imitating Environments [Ferreira et al. 2022]

- Relatively small NN for transitions and rewards
- Evolved using NES (over quite a few generations)
- Targets Acrobot and CartPole and tests one learning algorithm
- Score for the NES loop: difference between current policy on synthetic environment and score on actual environment

Imitating Environments [Ferreira et al. 2022]

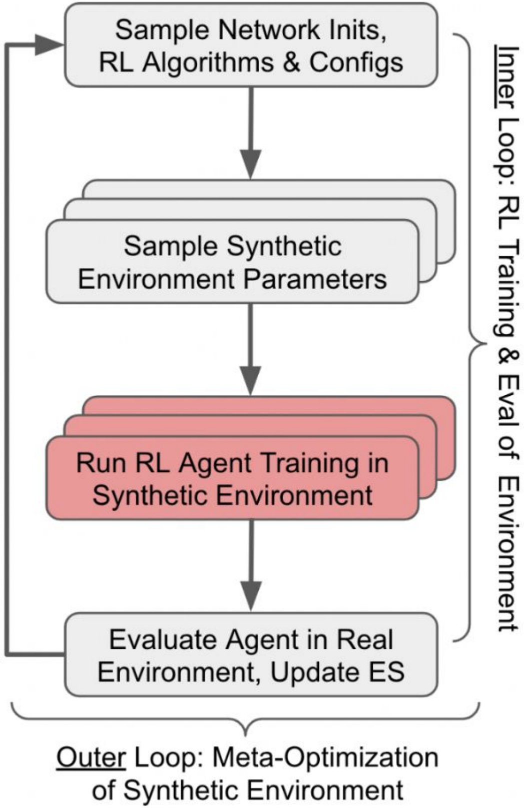
- Relatively small NN for transitions and rewards
- Evolved using NES (over quite a few generations)
- Targets Acrobot and CartPole and tests one learning algorithm
- Score for the NES loop: difference between current policy on synthetic environment and score on actual environment



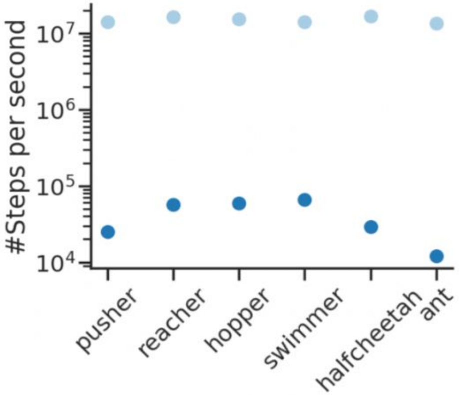
Discovering Environments [Liesen et al. 2024]

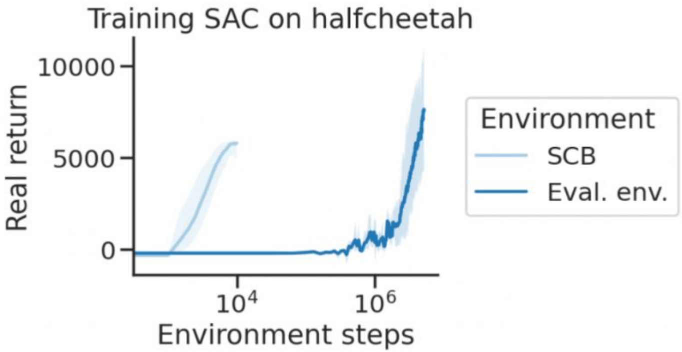
- Discovery: training contextual bandits is enough to learn a synthetic environment
- Contextual bandit: based on state, choose a bandit arm at each step yielding a reward -> approach learns bandit weights
- Difference to MDP: no transition function, the bandit is “stateless”
- Targeting continuous control environments from brax
- Three different RL algorithms with hyperparameter distribution used

Discovering Environments [Liesen et al. 2024]



Algorithm	PPO			SAC			DDPG		
Environment	S	F	R	S	F	R	S	F	R
pusher	-92.3	-57.1	-44.7	-43.0	-35.9	-54.0	-38.5	-36.8	-41.9
reacher	-30.1	-24.7	-5.1	-11.8	-9.4	-5.9	-7.6	-9.6	-4.3
hopper	916.3	1062.1	2521.9	2718.0	3521.6	3119.4	3002.7	3085.4	1465.8
swimmer	351.6	31.9	83.6	360.8	152.5	124.8	365.6	365.3	345.0
halfcheetah	1207.9	4790.3	8696.7	5784.1	4122.3	7735.5	6082.7	146.8	3966.8
ant	-261.8	760.0	3026.5	-3.5	3135.0	6011.1	-11.5	-20.0	3503.3





What About The Real World?

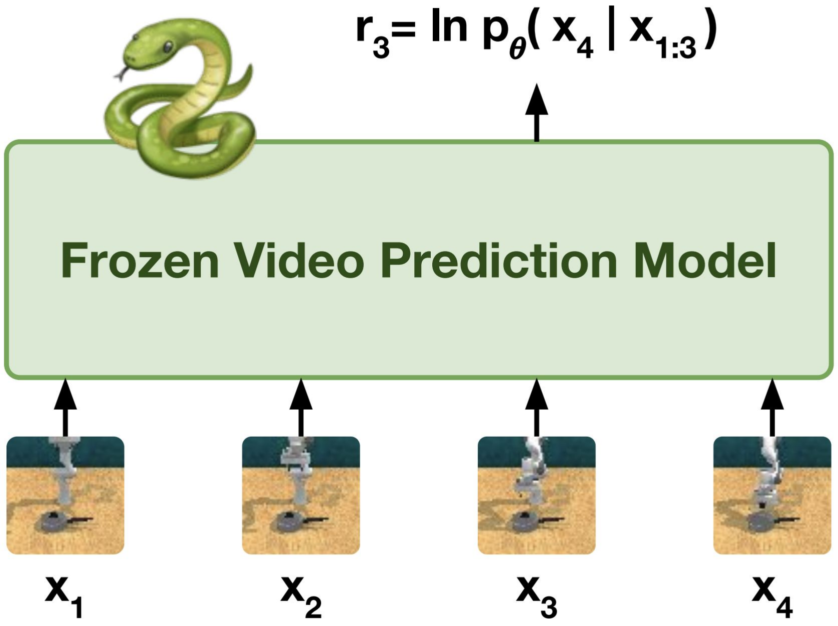
- So far: imitation of simulations
- Problem: most RL simulations are vastly more simple than real-world settings
- The hardest thing about solving a task with RL can be the modelling
- Can we skip the initial hand designed environment and directly learn real-world tasks?

Interactive Rewards [Escontrela et al. 2023]

- Idea: reward is a major bottleneck in modelling, why not try to infer it from videos in an unsupervised manner?
- Actions do not factor in directly
- Predict rewards based on how likely a frame sequence is in the training data
- Method: learn to predict video frames as tokens to extract probabilities

Interactive Rewards [Escontrela et al. 2023]

- Idea: reward is a major bottleneck in modelling, why not try to infer it from videos in an unsupervised manner?
- Actions do not factor in directly
- Predict rewards based on how likely a frame sequence is in the training data
- Method: learn to predict video frames as tokens to extract probabilities

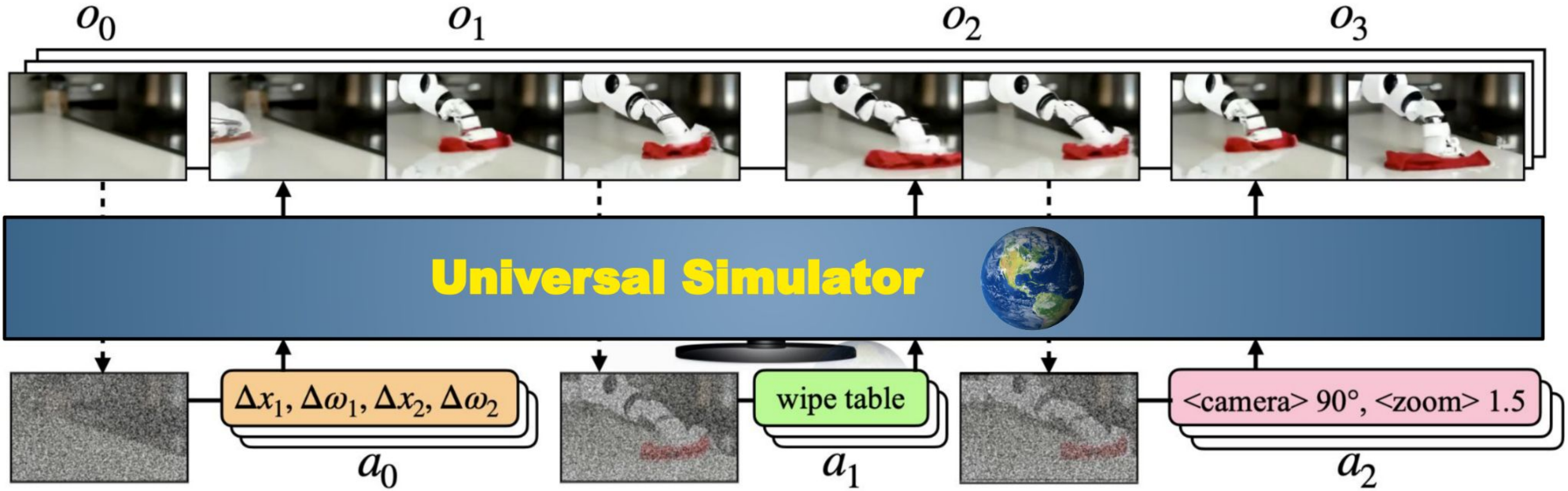


Interactive Simulators [Yang et al. 2024]

- Action-conditioned diffusion model
- Pre-trained on internet data and compatible with different action modalities

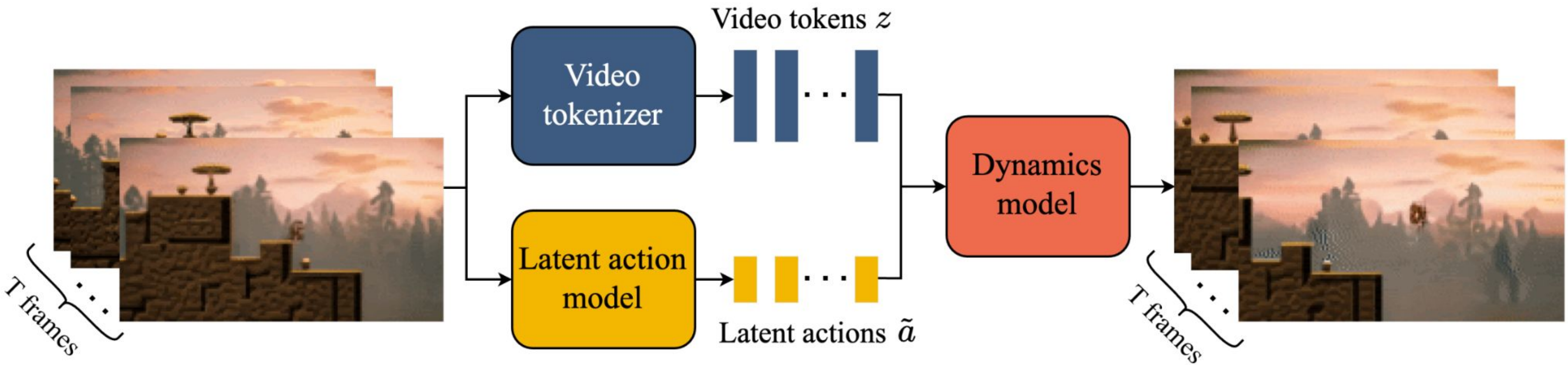
Interactive Simulators [Yang et al. 2024]

- Action-conditioned diffusion model
- Pre-trained on internet data and compatible with different action modalities

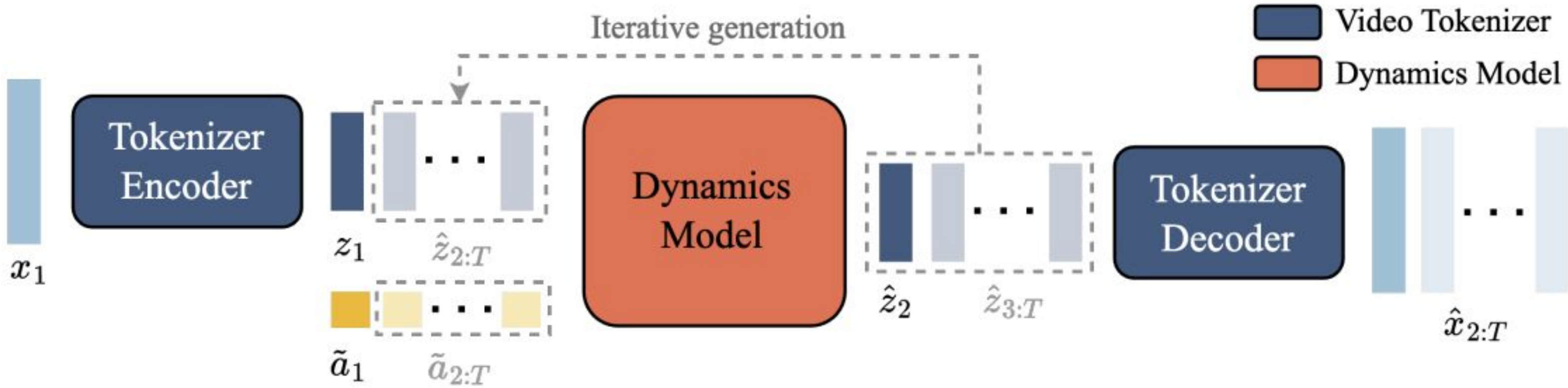


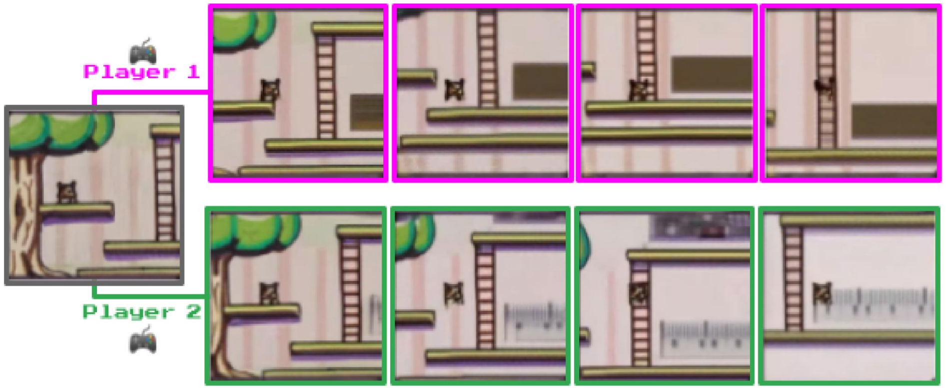
Generative Environments [Bruce et al. 2024]

- Idea: use text prompt to generate video game environment
- Number of possible actions is specified in advance
- Unsupervised matching of actions to frames using internet data in a latent action model



Generative Environments [Bruce et al. 2024]





Practicalities

- Cheap training seems possible, but only from pre-existing simulation
- Learning from the real world or generating from scratch is expensive at RL training time due to inference cost
- Learning the environment itself can be a huge undertaking
- But: it is hard to overstate how hard modelling is, so maybe still the quickest option for truly hard problems

Possible Future Directions

- Faster RL training on complex synthetic environments
  - Chaining of real-world learning and synthetic abstraction?
  - Predicting policy performance outright?

Possible Future Directions

- Faster RL training on complex synthetic environments
  - Chaining of real-world learning and synthetic abstraction?
  - Predicting policy performance outright?
- More prompt-able environments
  - Constrained generation, e.g. for gridworlds?
  - Curricula using generative environments?

Possible Future Directions

- Faster RL training on complex synthetic environments
  - Chaining of real-world learning and synthetic abstraction?
  - Predicting policy performance outright?
- More prompt-able environments
  - Constrained generation, e.g. for gridworlds?
  - Curricula using generative environments?
- Hopefully: more open-source synthetic environments to experiment with

My Understanding of Synthetic Environments

- ❑ I understand why learning environments is useful
- ❑ I can contrast it with model-based RL
- ❑ I know an example of learning an environment
- ❑ I understand the current limitations
- ❑ I know 1-2 approaches in detail
- ❑ I can discuss some future directions