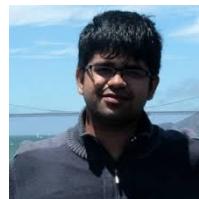
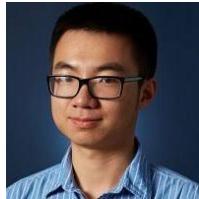


CS294-158 Deep Unsupervised Learning

Lecture 10: Representation Learning in Reinforcement Learning



Pieter Abbeel, Peter Chen, Jonathan Ho, Aravind Srinivas
UC Berkeley

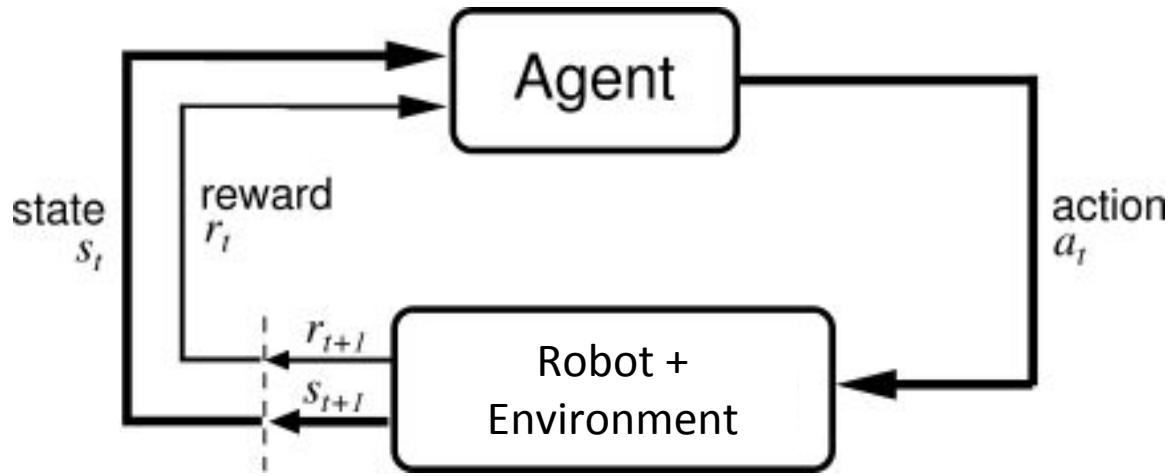
Acknowledgements

Big thank you to

**Abhishek Gupta, Max Jaderberg, David Ha, Martin Riedmiller, Danijar Hafner,
Marvin Zhang, Karol Gregor, Blazej Osinski, Irina Higgins, Pierre Sermanet,
Ashvin Nair, Vitchyr Pong, Rico Jonschkowski, Deepak Pathak, Pulkit Agrawal,
Coline Devin, Chelsea Finn, Amy Zhang, Alessandro Achille, Greg Kahn**

for sharing their insights, illustrations, slides, videos for this
lecture

Reinforcement Learning (RL)

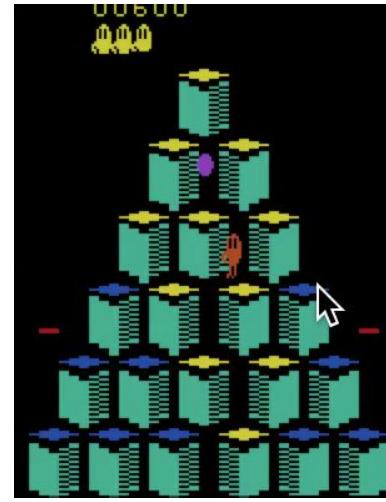
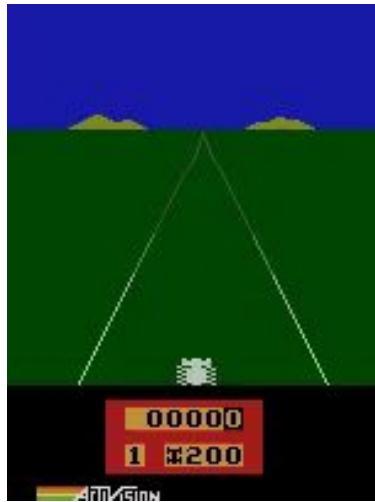


$$\max_{\theta} \mathbb{E}\left[\sum_{t=0}^H R(s_t) | \pi_{\theta}\right]$$

- Compared to supervised learning, additional challenges:
 - Credit assignment
 - Stability
 - Exploration

[Image credit: Sutton and Barto 1998]

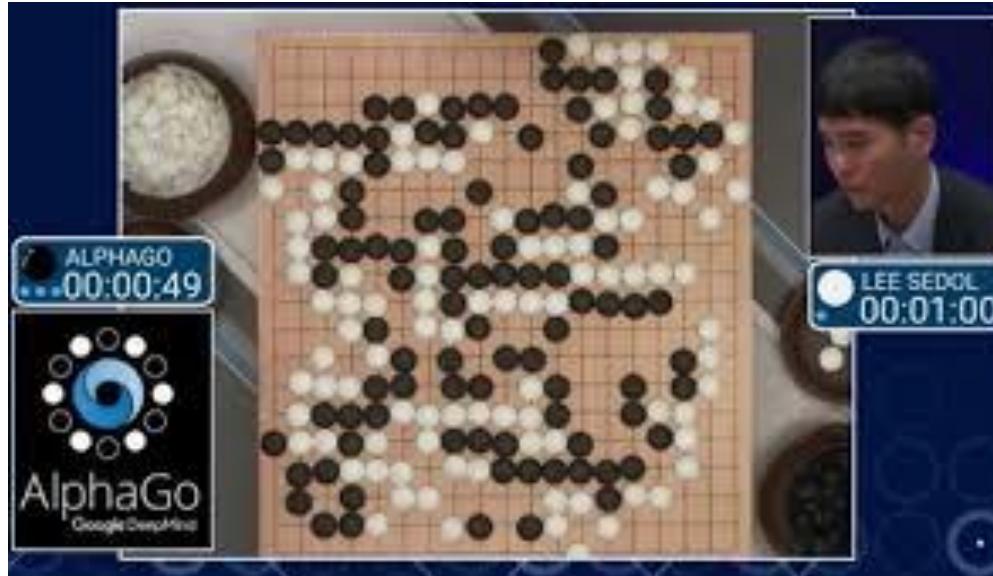
Deep RL Success: Atari



DQN Mnih et al, NIPS 2013 / Nature 2015

MCTS Guo et al, NIPS 2014; TRPO Schulman, Levine, Moritz, Jordan, Abbeel, ICML 2015; A3C Mnih et al, ICML 2016; Dueling DQN Wang et al ICML 2016; Double DQN van Hasselt et al, AAAI 2016; Prioritized Experience Replay Schaul et al, ICLR 2016; Bootstrapped DQN Osband et al, 2016; Q-Ensembles Chen et al, 2017; Rainbow Hessel et al, 2017; Accelerated Stooke and Abbeel, 2018; ...

Deep RL Success: Go



AlphaGo Silver et al, Nature 2015

AlphaGoZero Silver et al, Nature 2017

AlphaZero Silver et al, 2017

Tian et al, 2016; Maddison et al, 2014; Clark et al, 2015

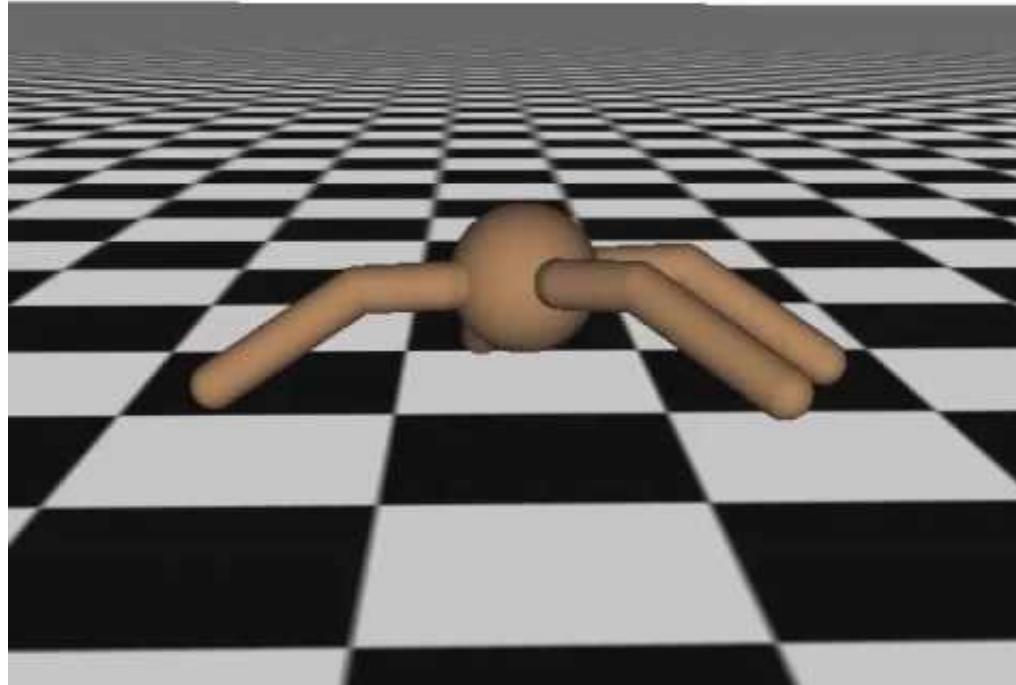
OpenAI's 1v1 Dota [2017] and 5v5 [2018, 2019]

- Super-human agent on a competitive game, enabled by
 - Reinforcement learning
 - Self-play
 - Enough computation
- Cooperation emerges



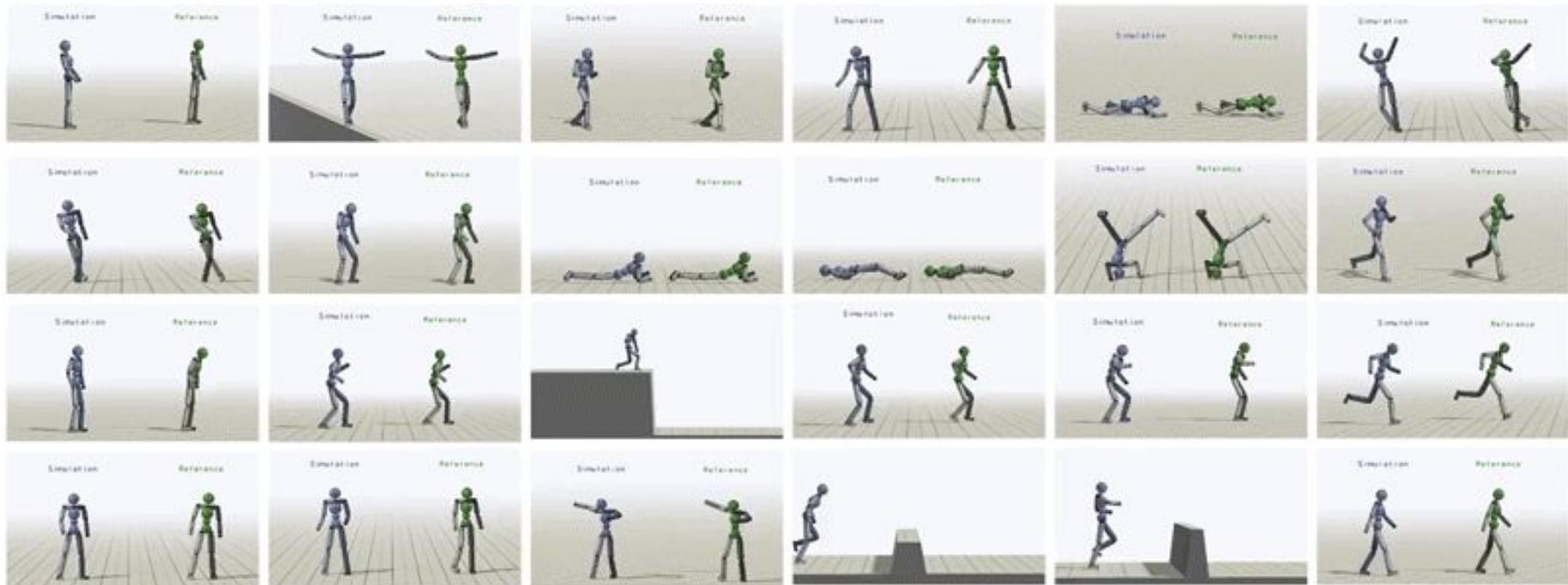
Deep RL Success: Locomotion

Iteration 80



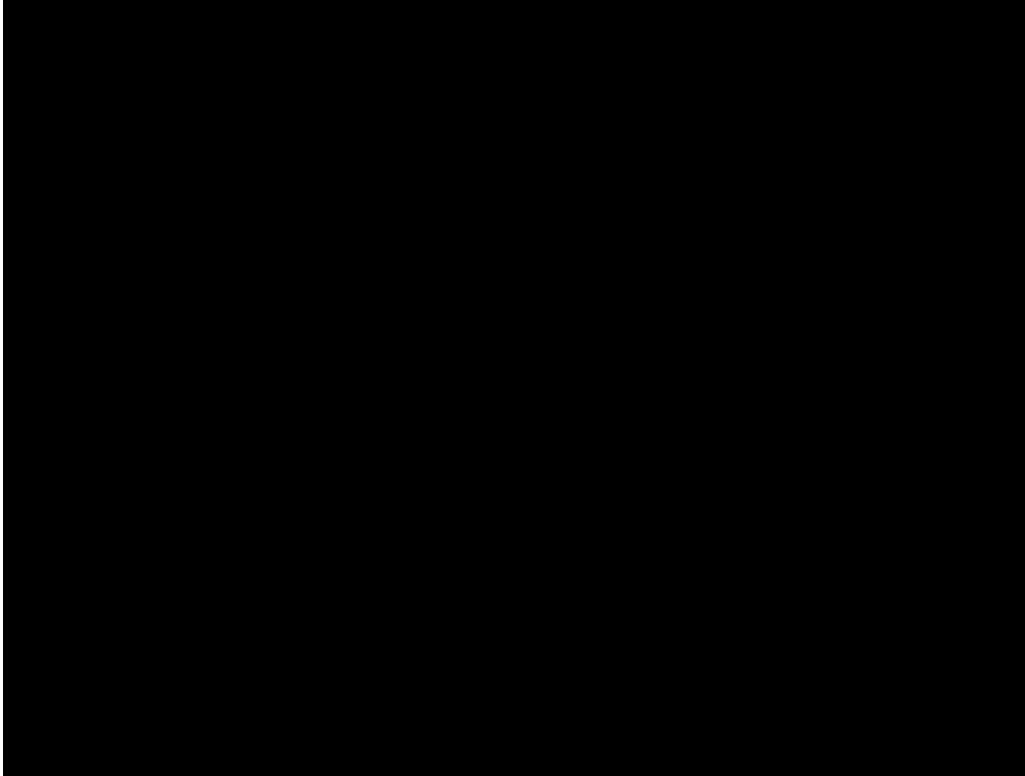
TRPO Schulman, Levine, Moritz, Jordan, Abbeel, 2015 + **GAE** Schulman, Moritz, Levine, Jordan Abbeel, 2016

Deep RL Success: Locomotion



DeepMimic, Peng, Abbeel, Levine, van de Panne 2018

Deep RL Success: Robotic Manipulation



Guided Policy Search, Levine*, Finn*, Darrell, Abbeel, 2016

Catch?

Data inefficiency

Representation Learning in Reinforcement Learning

- Auxiliary losses
- State representation
- Exploration
- Unsupervised skill discovery

Representation Learning in Reinforcement Learning

- ***Auxiliary losses***
- State representation
- Exploration
- Unsupervised skill discovery

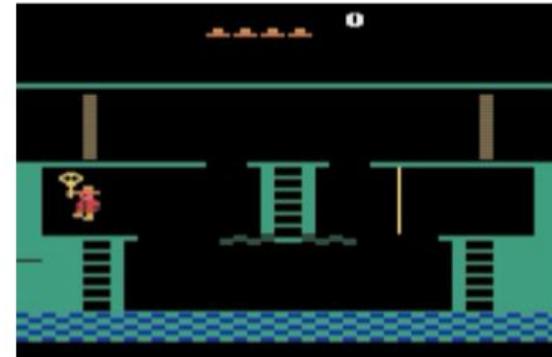
Reinforcement Learning with Unsupervised Auxiliary Tasks

Max Jaderberg*, Volodymyr Mnih*, Wojciech Marian Czarnecki*,
Tom Schaul, Joel Z Leibo, David Silver, Koray Kavukcuoglu

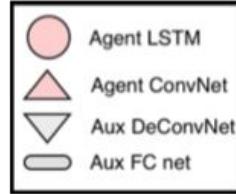


Overview

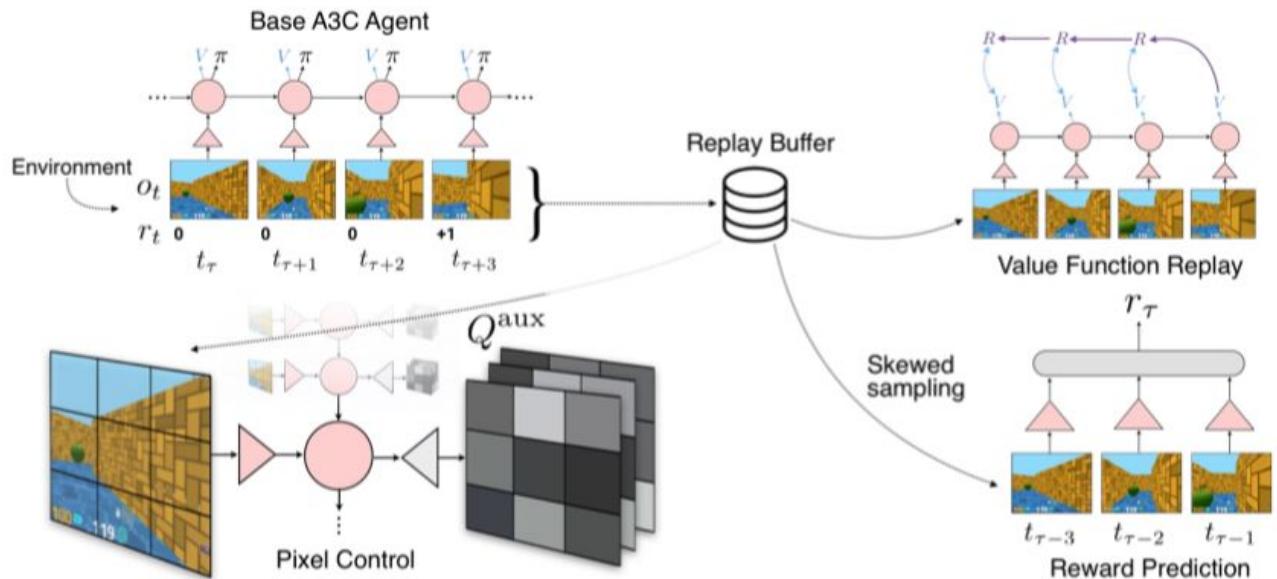
- Problem - Deep reinforcement learning can be very data hungry. Especially with **sparse rewards**.
- This paper - Augment an RL agent with **auxiliary prediction and control tasks**.
- The UNREAL agent - UNsupervised REinforcement and Auxiliary Learning:
 - **10x improvement in data efficiency** over A3C on 3D DeepMind Lab environments.
 - **60% improvement in final scores** over A3C.



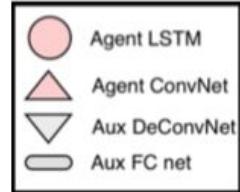
The UNREAL Architecture



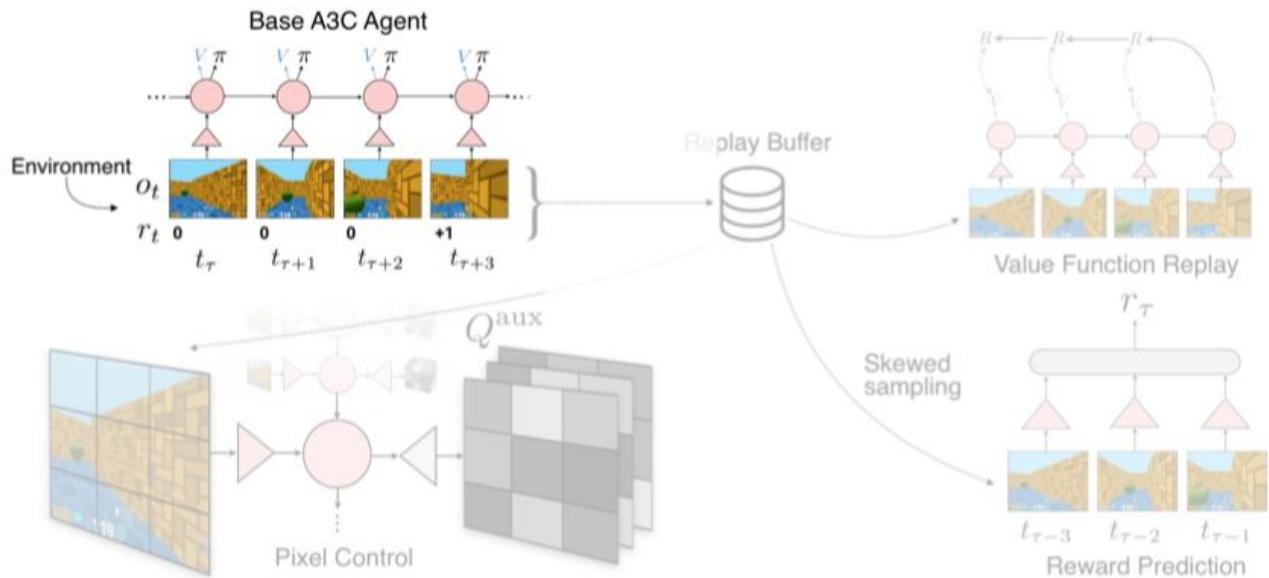
- UNREAL augments an LSTM A3C agent with 3 auxiliary tasks.
- Can be used on top of DQN, DDPG, TRPO or other agents.



The UNREAL Architecture



- Base A3C LSTM agent learns from the environment's scalar reward signal.
- UNREAL acts using the base A3C agent's policy.



Sparse Rewards?

Single scalar reward signal



Sparse Rewards? More Cherries!

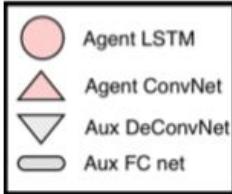
Single scalar reward signal



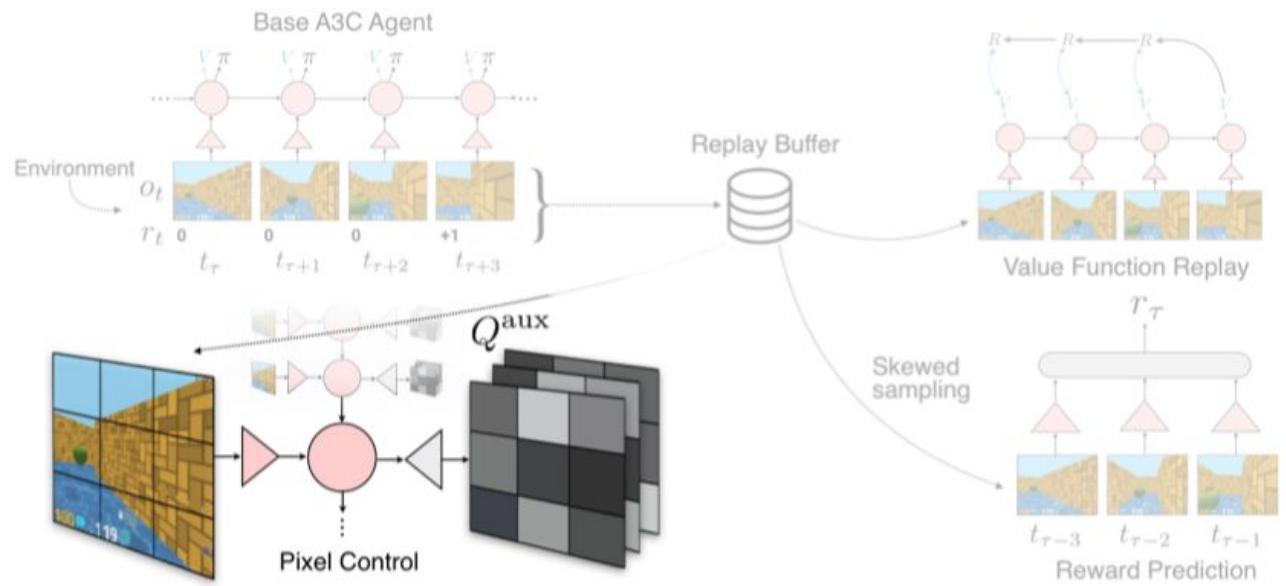
Many reward signals



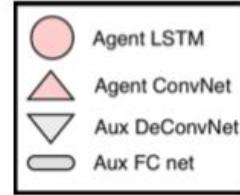
Unsupervised RL



- Augment A3C with many **auxiliary control tasks**.
- Learning to control many aspects of the environment.
- Pixel control - learn to maximally change parts of the screen.
- Feature control (not used by UNREAL) - learn to control the internal representations.

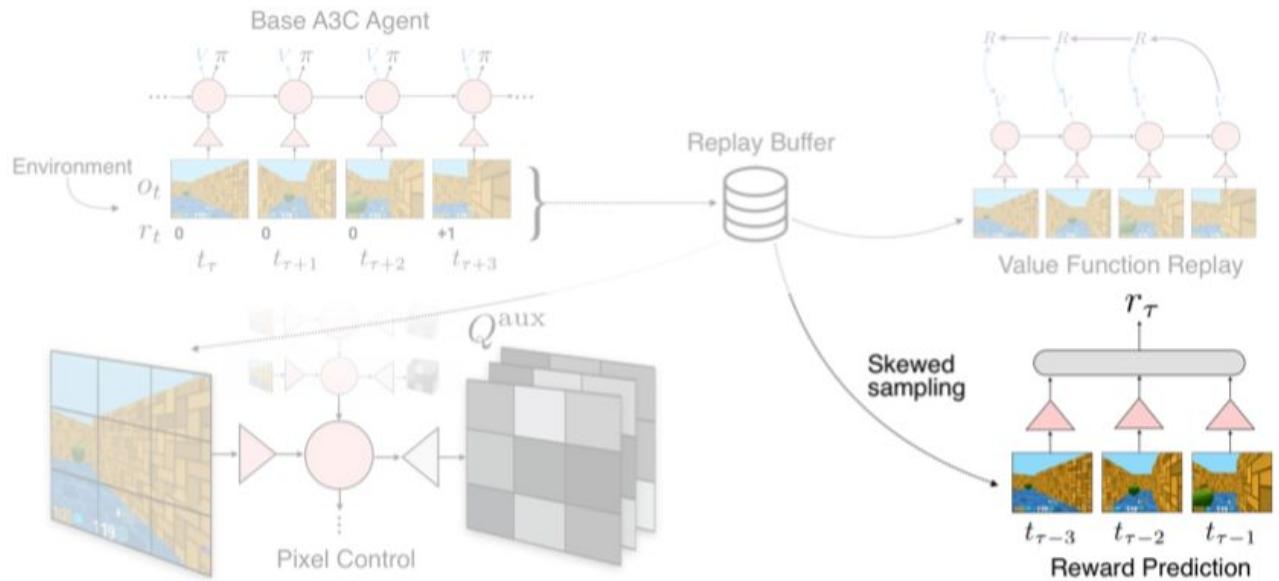


The UNREAL Architecture

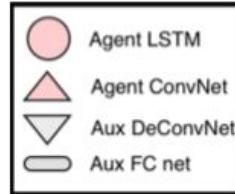


Focusing on rewards:

- Rebalanced reward prediction.
- Shape the agent's CNN by classifying whether a sequence of frames will lead to reward.
- No need to worry about off-policy learning.

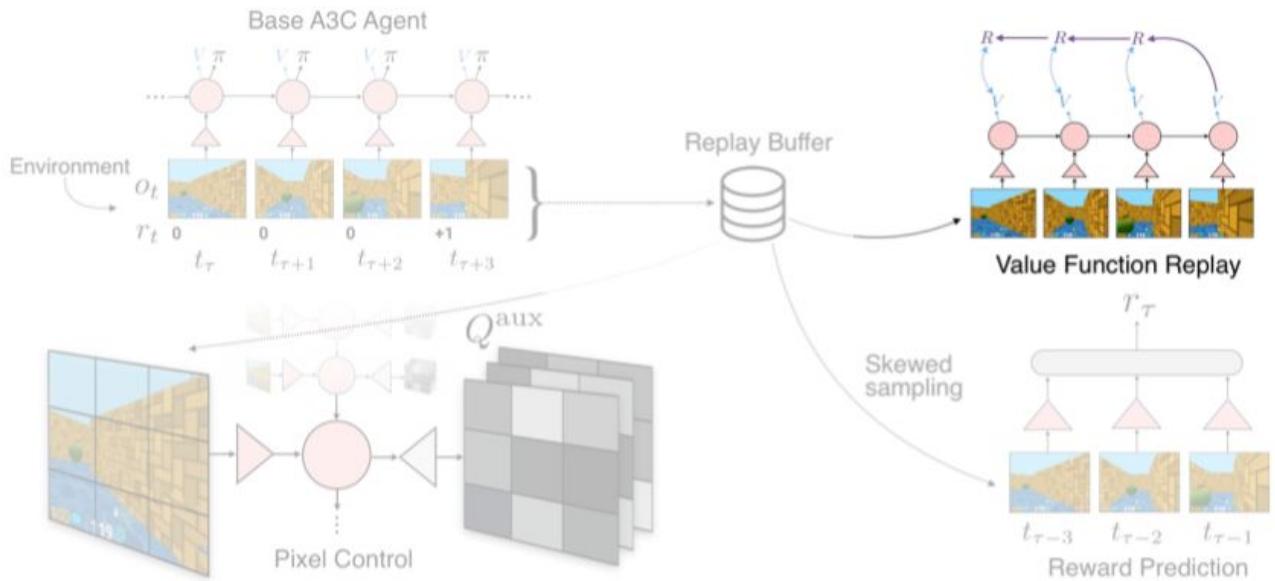


The UNREAL Architecture



Focusing on rewards:

- Value function replay.
- Faster learning of the value function.

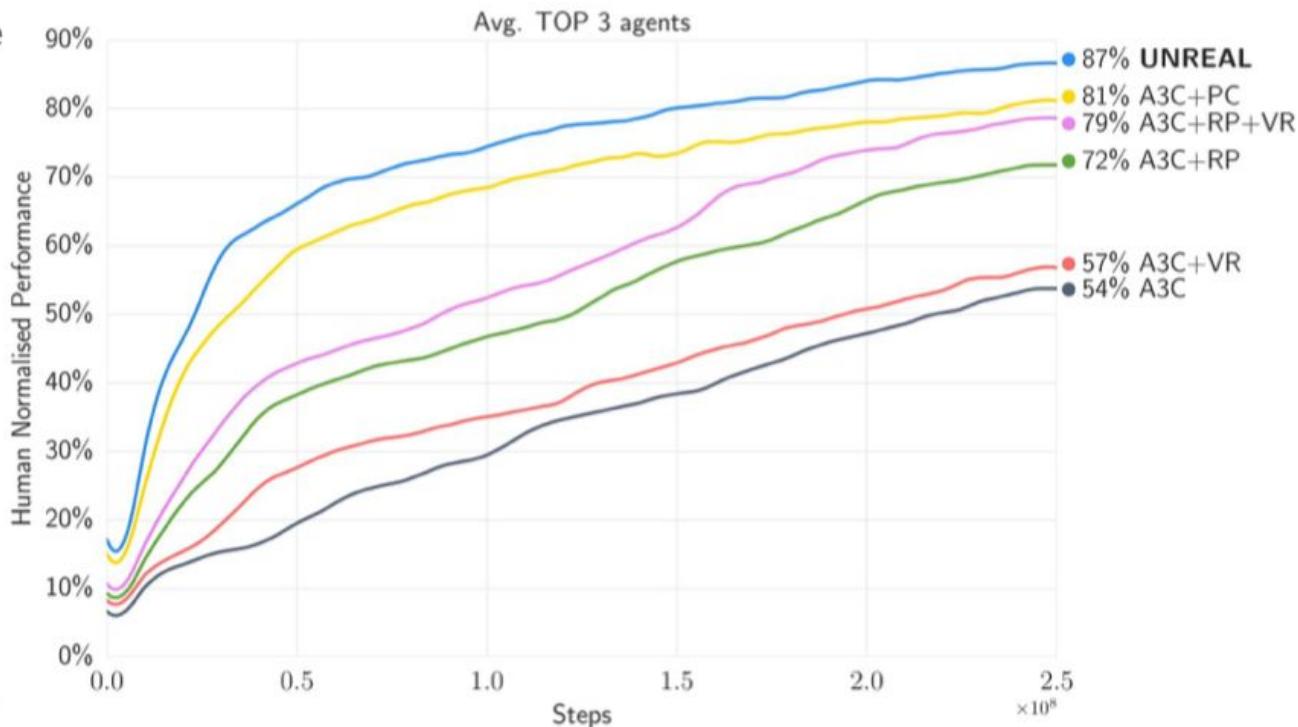




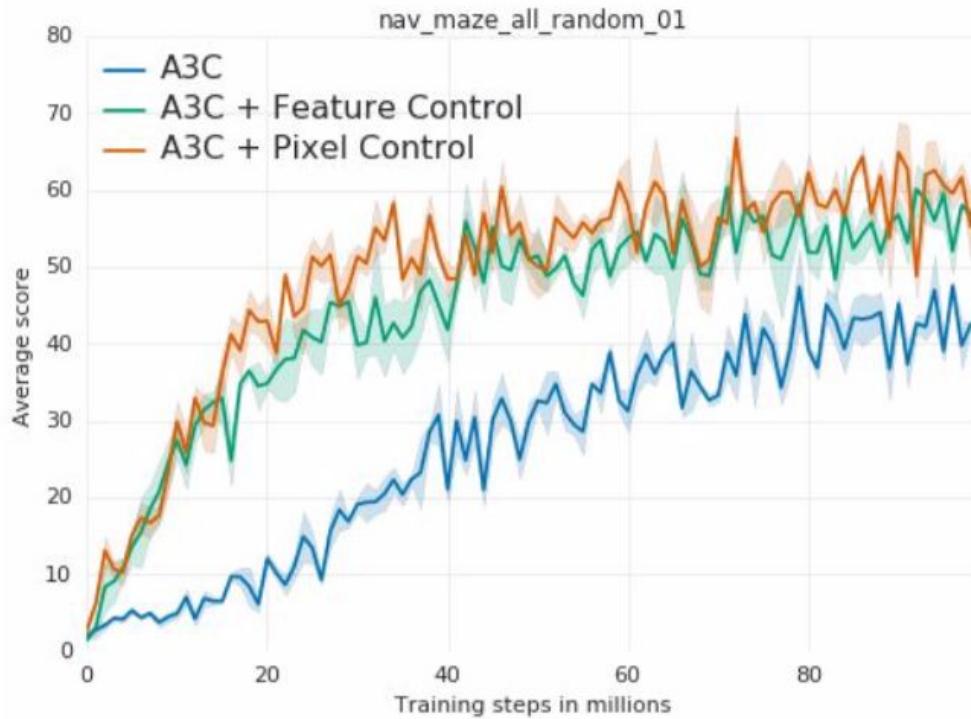
DeepMind Lab

Results

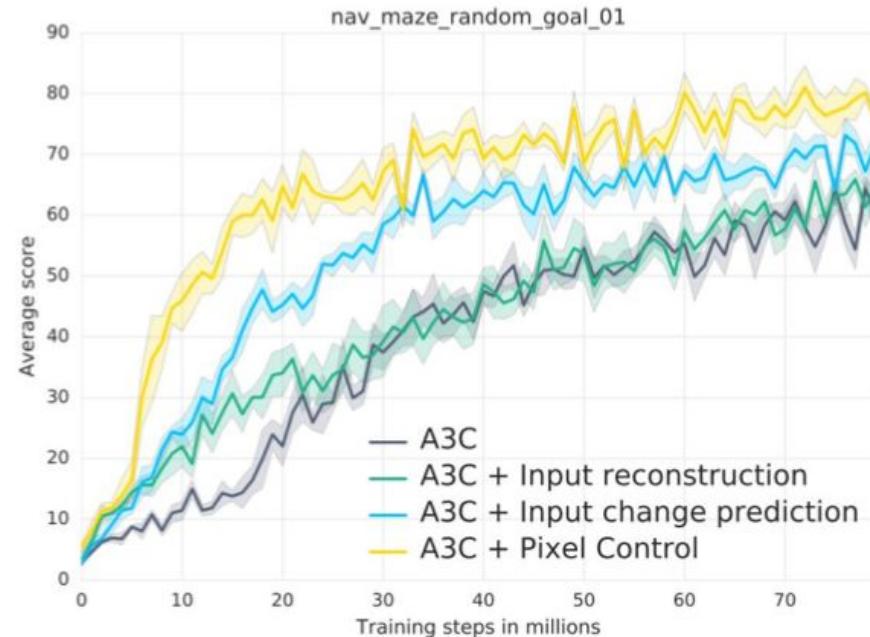
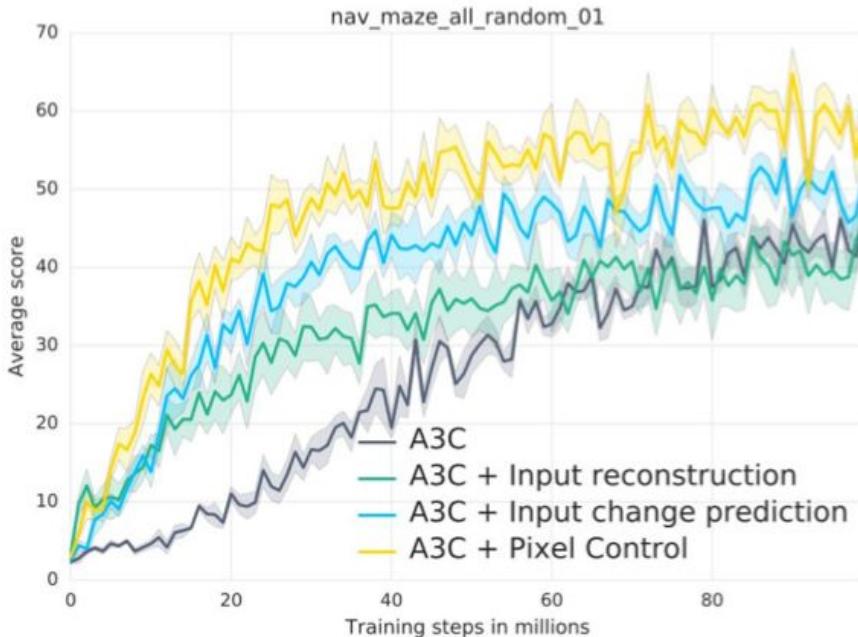
- Average human-normalized performance on 13 3D environments from DeepMind Lab.
- Tasks include random maze navigation and laser tag.
- Roughly a 10x improvement in data efficiency over A3C.
- 60% improvement in final performance.



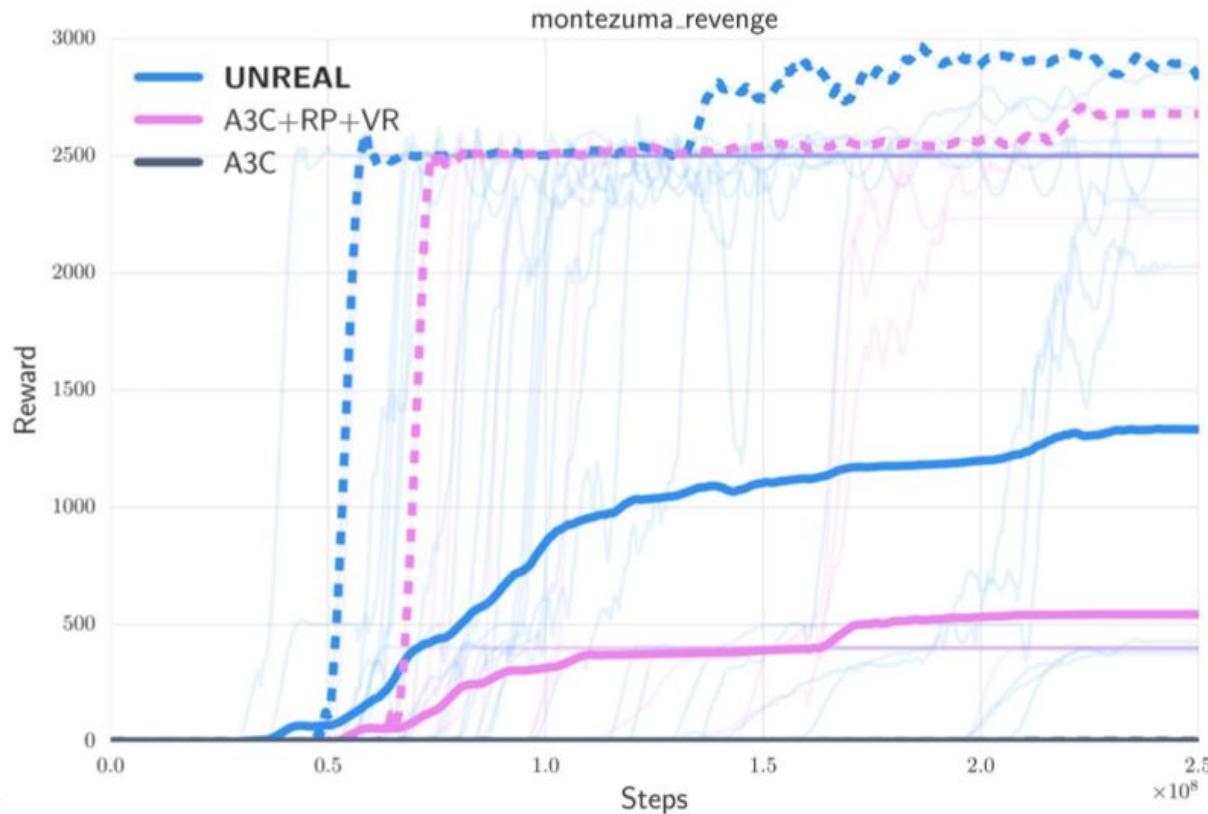
Feature Control



Unsupervised RL Baselines



Montezuma's Revenge



UNREAL playing DeepMind Lab





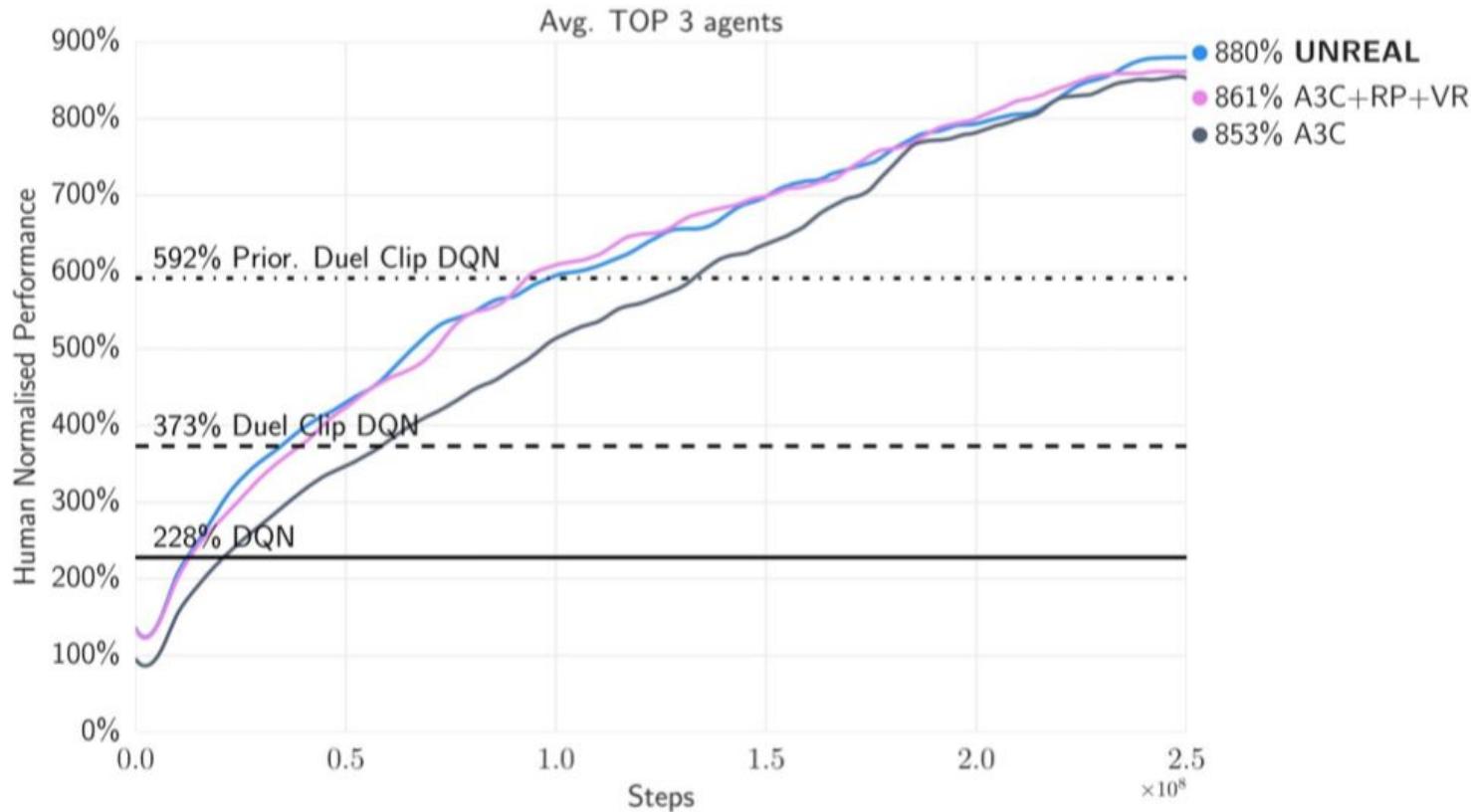
The UNREAL Loss

- The UNREAL loss is defined as:

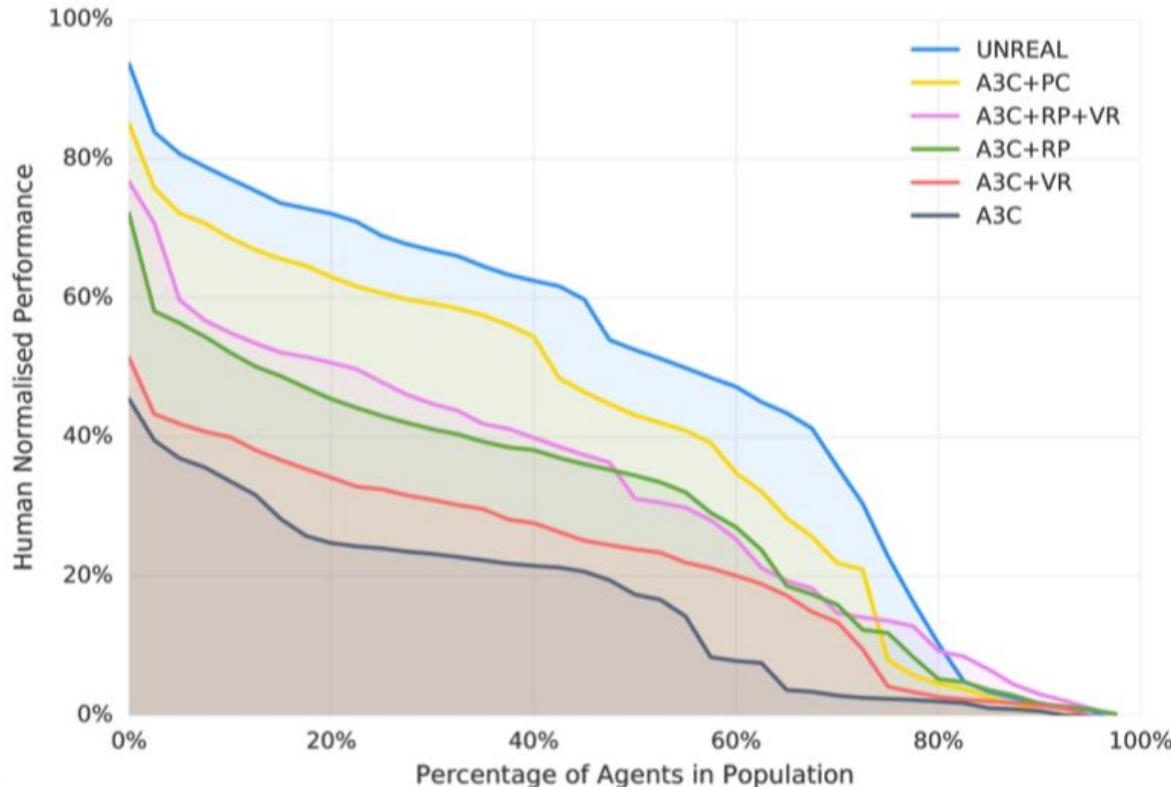
$$\mathcal{L}_{UNREAL}(\theta) = \mathcal{L}_{A3C} + \lambda_{VR}\mathcal{L}_{VR} + \lambda_{PC} \sum_c \mathcal{L}_Q^{(c)} + \lambda_{RP}\mathcal{L}_{RP}$$

- Reward prediction:
 - Equal proportion of rewarding and non-rewarding examples.
- Pixel control:
 - Divide screen into a 20x20 grid of cells.
 - Use n-step Q-learning to learn a policy to maximally change each cell.
 - Use a deconvolutional network to predict the Ax20x20 tensor of Q-values.

Atari Results



Labyrinth Robustness



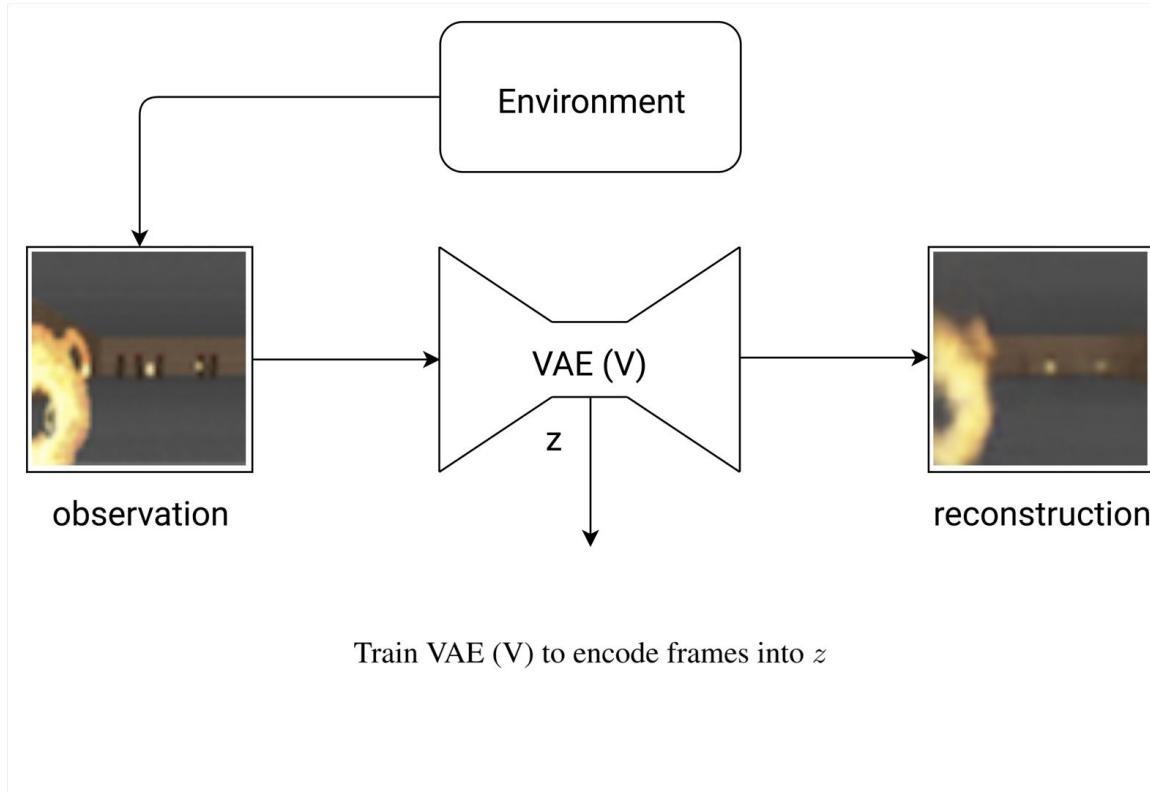
Representation Learning in Reinforcement Learning

- Auxiliary losses
- ***State representation***
- Exploration
- Unsupervised skill discovery

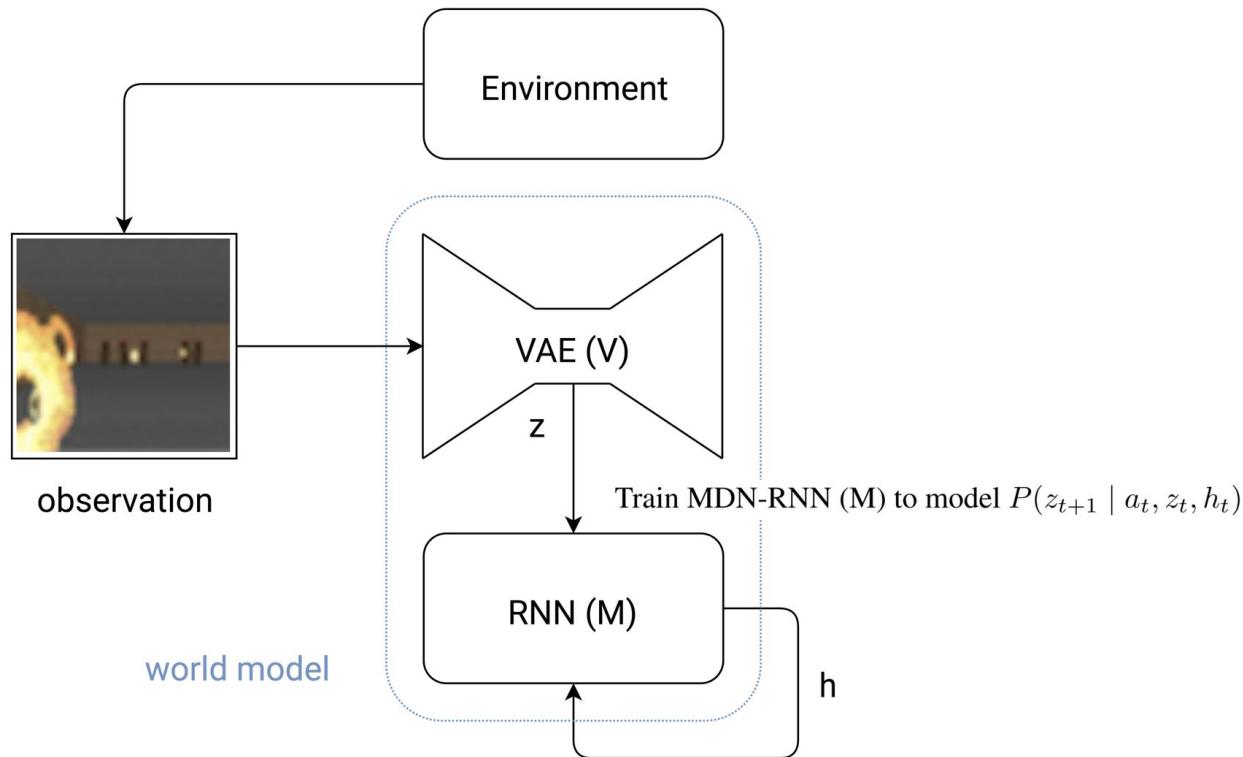
Representation Learning in Reinforcement Learning

- Auxiliary losses
- ***State representation***
 - ***Observation -> State***
- Exploration
- Unsupervised skill discovery

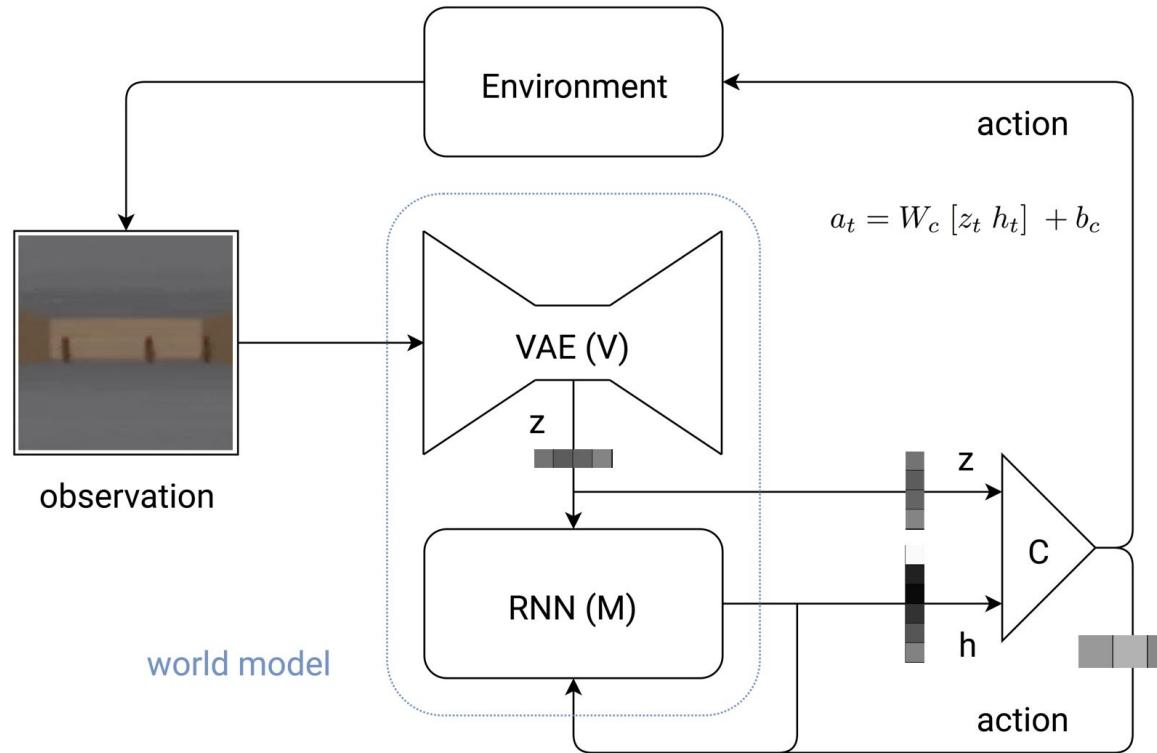
World Models



World Models



World Models



World Models

CarRacing-v0



- ▶ Randomly generated tracks
- ▶ Stay on tiles, travel around track
- ▶ Average Score > 900 (100 trials) to “solve” task

Procedure:

1. Collect 10,000 rollouts from a random policy.
2. Train VAE (V) to encode frames into $z \in \mathcal{R}^{32}$.
3. Train MDN-RNN (M) to model $P(z_{t+1} | a_t, z_t, h_t)$.
4. Evolve linear controller (C) to maximize the expected cumulative reward of a rollout. $a_t = W_c [z_t \ h_t] + b_c$

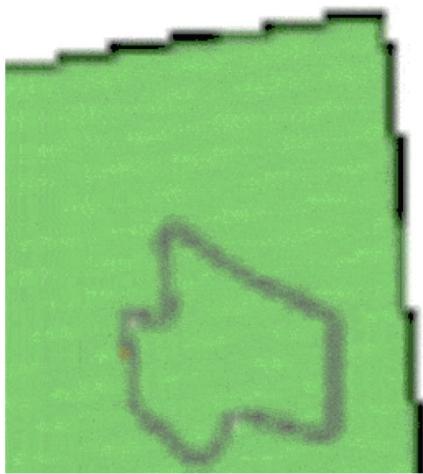
MODEL	PARAMETER COUNT
VAE	4,348,547
MDN-RNN	422,368
CONTROLLER	867



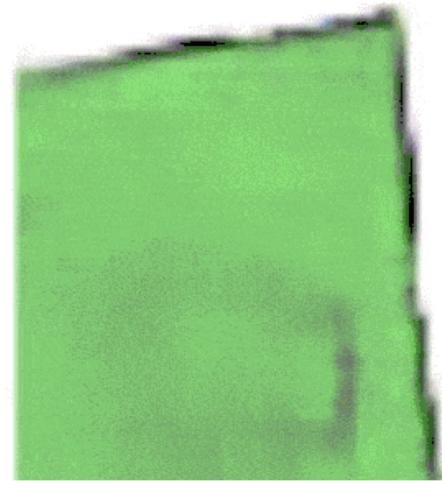
Source: Taste of Home
(www.tasteofhome.com)

World Models

Input Frame (64x64px)



Frame Reconstruction using z

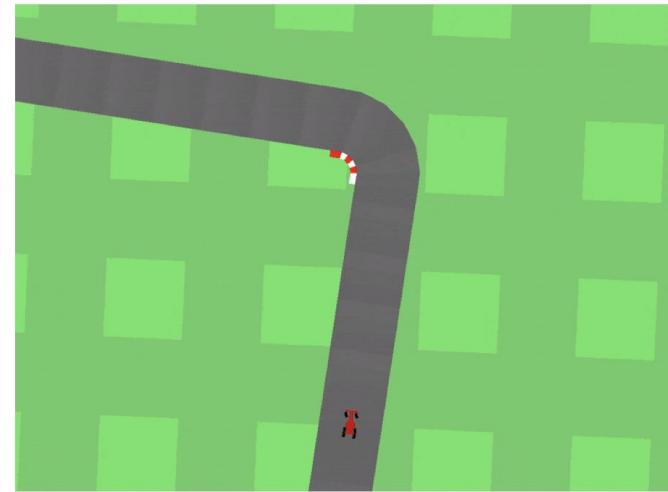


World Models

Car Racing: Spatial Inputs vs Spatial and Temporal Inputs



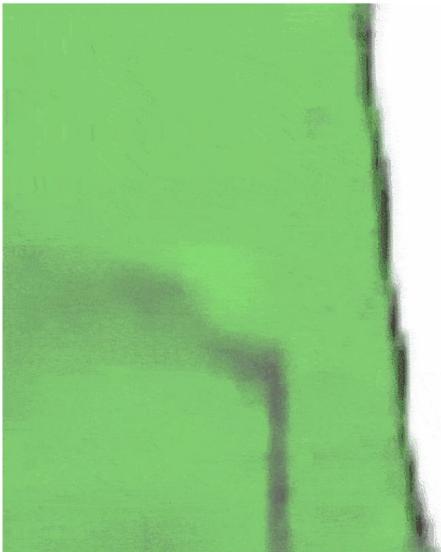
z (spatial) only input



z and h (RNN's hidden state)
as inputs

World Models

CarRacing-v0 Results



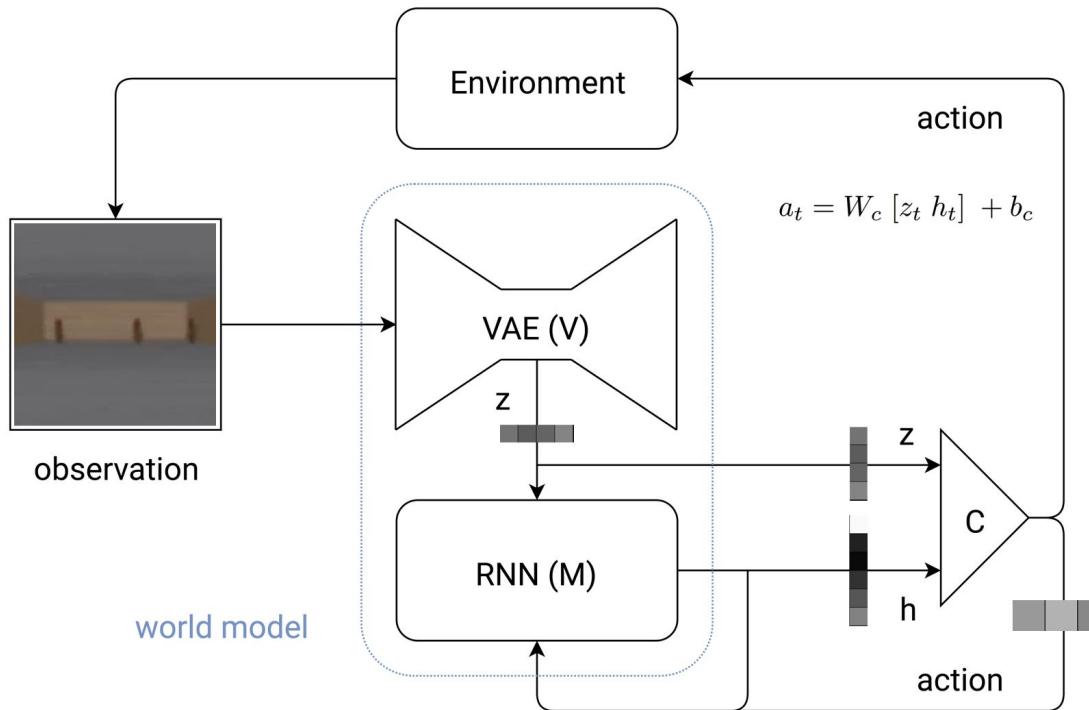
METHOD	AVG. SCORE
DQN (PRIEUR, 2017)	343 ± 18
A3C (CONTINUOUS) (JANG ET AL., 2017)	591 ± 45
A3C (DISCRETE) (KHAN & ELIBOL, 2016)	652 ± 10
CEOBILLIONAIRE (GYM LEADERBOARD)	838 ± 11
V MODEL	632 ± 251
V MODEL WITH HIDDEN LAYER	788 ± 141
FULL WORLD MODEL	906 ± 21

CarRacing-v0 scores achieved using various methods.

METHOD	AVG. SCORE
RANDOM WEIGHTS FOR RNN M (TALLEC ET AL., 2018)	870 ± 120
3-LAYER DQN W/ DROPOUT, CURRICULUM LEARNING (GERBER ET AL., 2018)	$900 \pm ?$

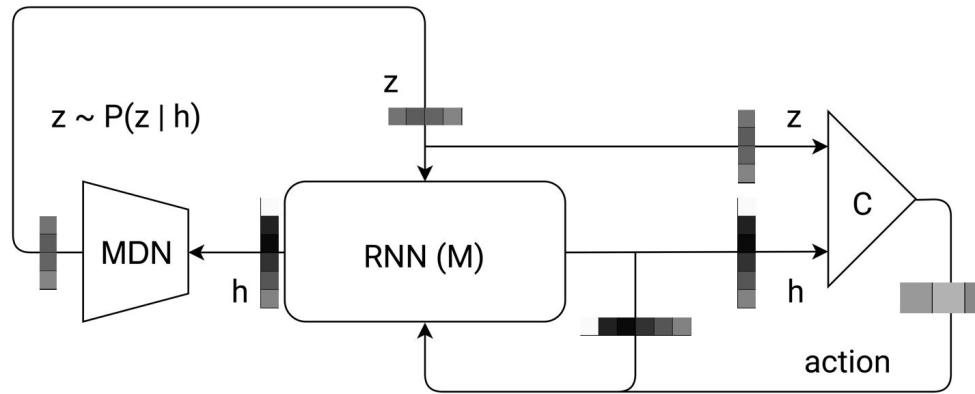
CarRacing-v0 scores achieved using newer methods.

World Models



World Models

Train controller inside latent space environment.



World Models

DoomTakeCover-v0



- ▶ Avoid fireballs from monsters
- ▶ Stay alive for as long as possible
- ▶ Average Score > 750 time steps (100 trials) to “solve” task

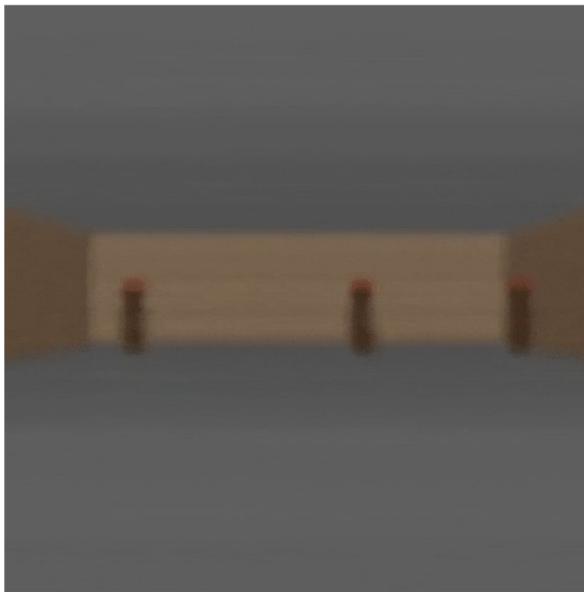
Procedure:

1. Collect 10,000 rollouts from a random policy.
2. Train VAE (V) to encode frames into $z \in \mathcal{R}^{64}$.
3. Train MDN-RNN (M) to model $P(z_{t+1}, d_{t+1} | a_t, z_t, h_t)$.
4. Evolve linear controller (C) to maximize the expected cumulative reward of a rollout.
5. Deploy learned policy from (4) on actual environment.

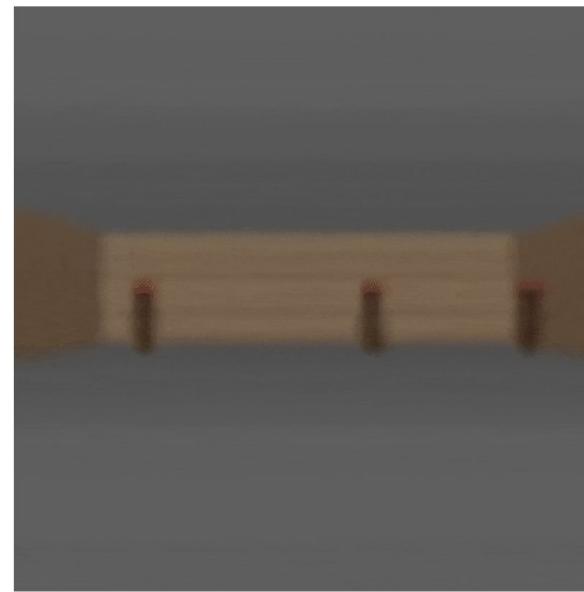
MODEL	PARAMETER COUNT
VAE	4,446,915
MDN-RNN	1,678,785
CONTROLLER	1,088

World Models

Doom TakeCover: Cheating the World Model



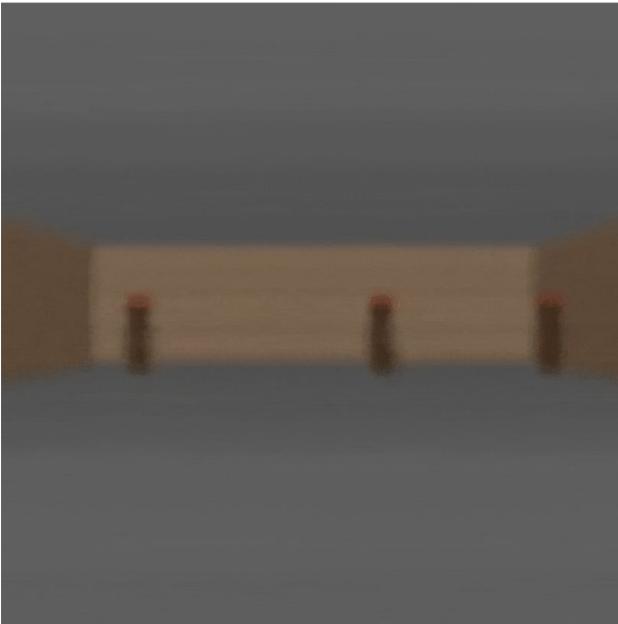
Normal Temperature



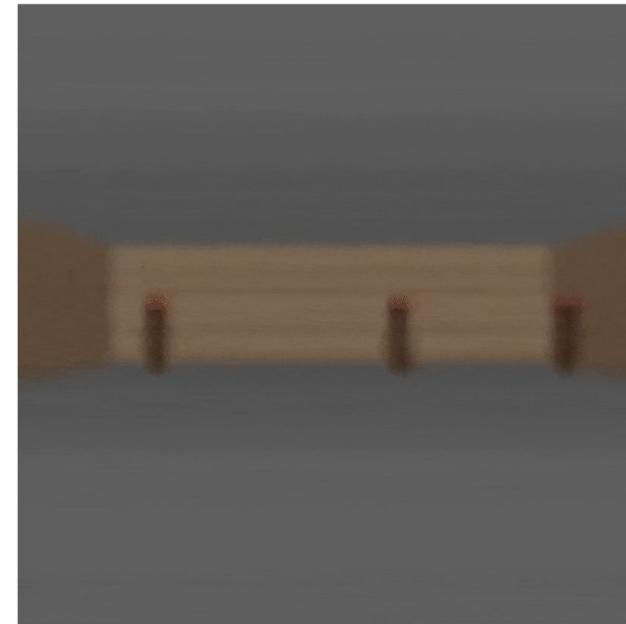
Higher Temperature

World Models

Doom TakeCover: Cheating the World Model



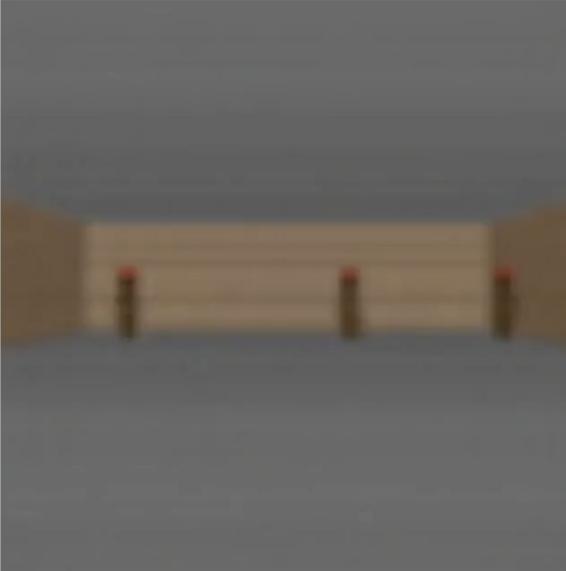
Normal Temperature



Higher Temperature

World Models

DoomTakeCover-v0 Results



TEMPERATURE τ	VIRTUAL SCORE	ACTUAL SCORE
0.10	2086 ± 140	193 ± 58
0.50	2060 ± 277	196 ± 50
1.00	1145 ± 690	868 ± 511
1.15	918 ± 546	1092 ± 556
1.30	732 ± 269	753 ± 139
RANDOM POLICY	N/A	210 ± 108
GYM LEADER	N/A	820 ± 58

DoomTakeCover-v0 scores at various settings of τ .

- ▶ Agent learned actions to take advantage of flaws of virtual environment.
- ▶ Adjust temperature parameter in the sampling to control uncertainty.

World Models

Iterative Training Policy

Procedure:

1. Initialize M, C with random model parameters.
2. Rollout to actual environment N times. Save all actions a_t and observations x_t during rollouts to storage.
3. Train M to model $P(x_{t+1}, r_{t+1}, a_{t+1}, d_{t+1} | x_t, a_t, h_t)$ and train C to optimize expected rewards inside of M.
4. Go back to (2) if task has not been completed.



Iteration #1

World Models

Iterative Training Policy

Procedure:

1. Initialize M, C with random model parameters.
2. Rollout to actual environment N times. Save all actions a_t and observations x_t during rollouts to storage.
3. Train M to model $P(x_{t+1}, r_{t+1}, a_{t+1}, d_{t+1} | x_t, a_t, h_t)$ and train C to optimize expected rewards inside of M.
4. Go back to (2) if task has not been completed.



Iteration #20

World Models

Action-Conditional Video
Prediction using Deep Networks
in Atari Games (2015)

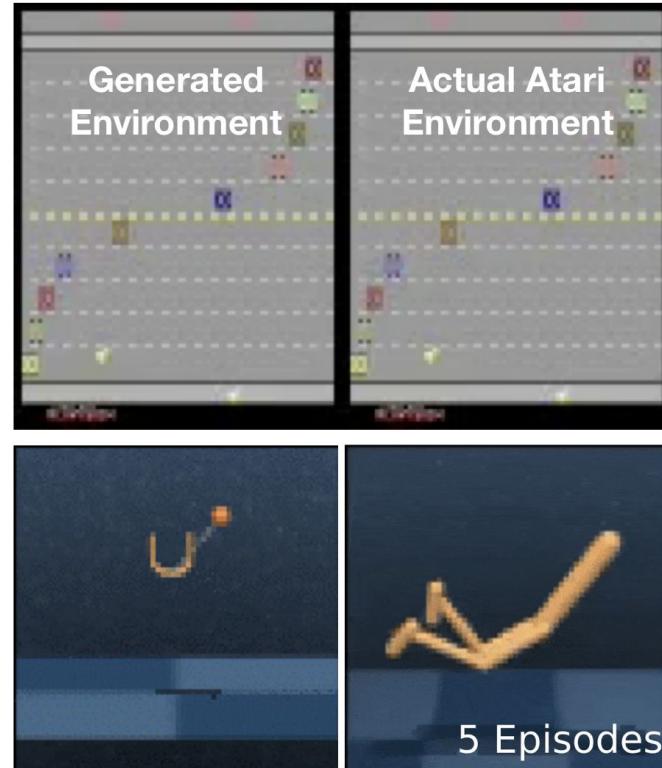
Junhyuk Oh, Xiaoxiao Guo, Honglak Lee,
Richard Lewis and Satinder Singh

Model-Based Reinforcement
Learning for Atari (2018)

Błażej Osiński, Łukasz Kaiser, Mohammad Babaeizadeh,
George Tucker, Dumitru Erhan, Ryan Sepassi, Chelsea Finn,
Sergey Levine, Piotr Kozakowski, Konrad Czechowski,
Piotr Miłos and Henryk Michalewski

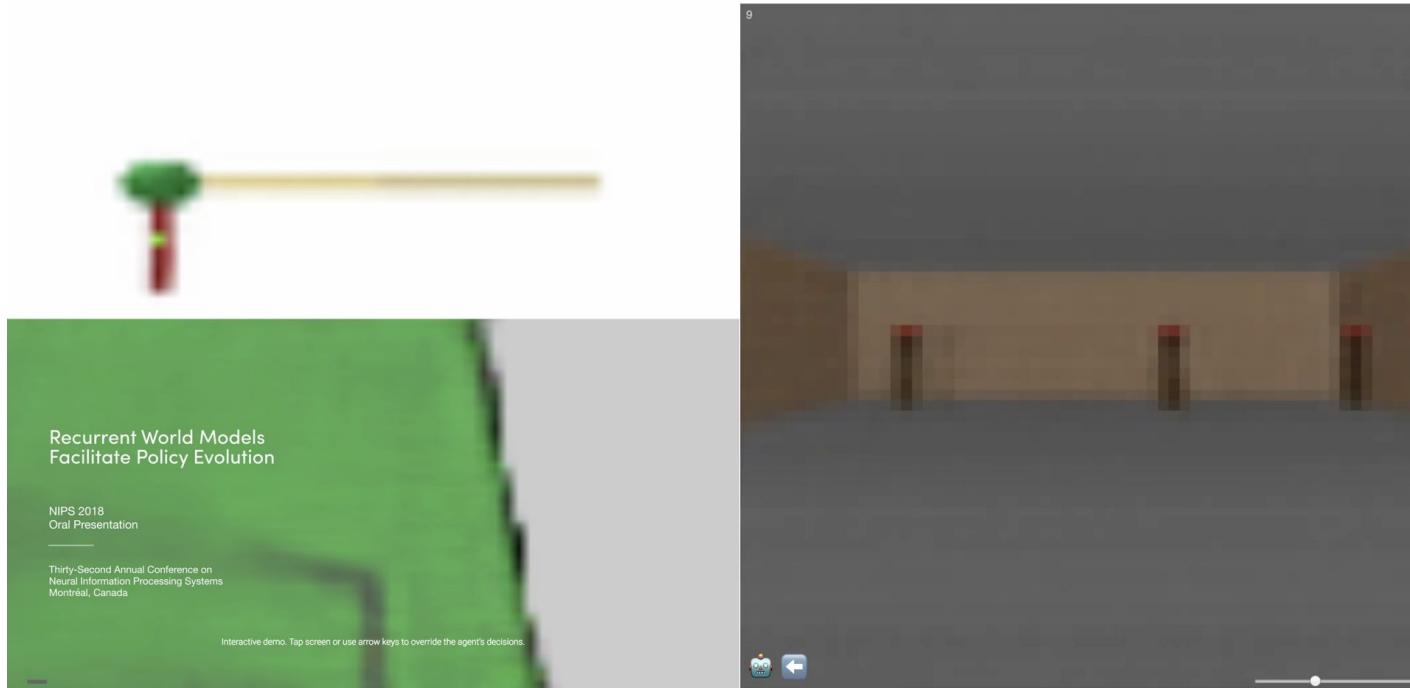
Learning Latent Dynamics for
Planning from Pixels (2018)

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas,
David Ha, Honglak Lee and James Davidson

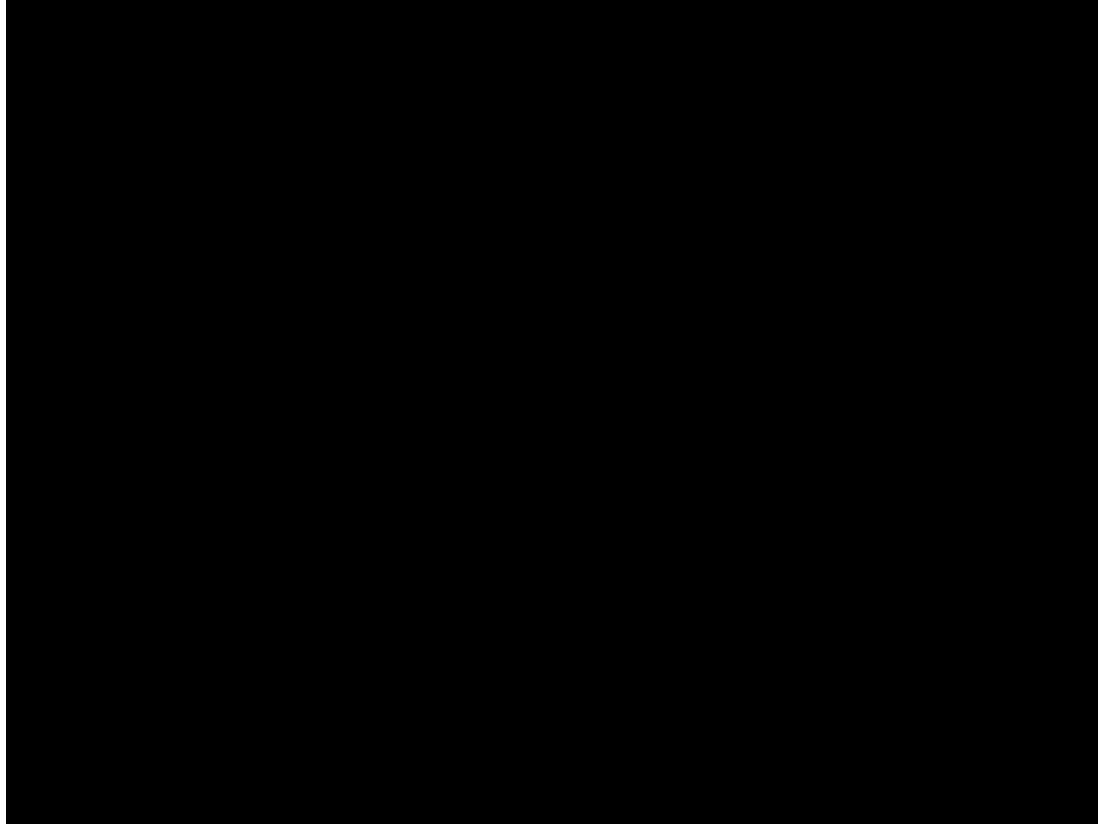


World Models

Code + Demo → <https://worldmodels.github.io>  TensorFlow.js



World Models



Representation Learning in Reinforcement Learning

- Auxiliary losses
- ***State representation***
 - ***Observation -> State***
 - ***Observation -> State + State,Action -> Next State***
- Exploration
- Unsupervised skill discovery

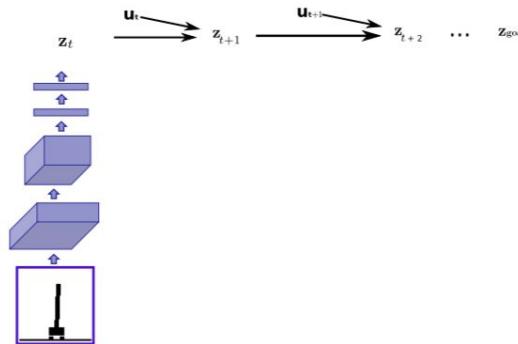
Embed to Control

Embed to Control: Model based RL from raw images



Can we perform model based RL starting from raw images ?

- Standard algorithms would fail in pixel space
- We want to **unsupervisedly** learn latent space z_t for control from images x_t



Related attempts:

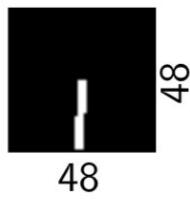
- ▶ [Lange et al.:Deep Learning of Visual Control Policies, 2010]
- ▶ [Wahlstroem et al.: From Pixels to Torques, 2015]

Embed to Control

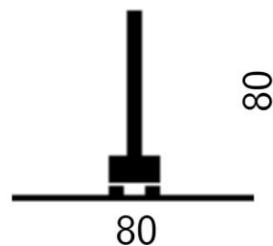
Systems we consider



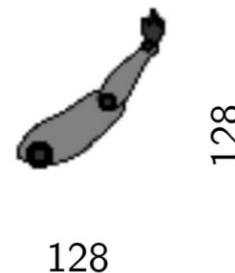
Pendulum
 $z \in \mathbb{R}^3$



Cart-Pole
 $z \in \mathbb{R}^8$



Three-Link-Arm
 $z \in \mathbb{R}^8$



Embed to Control

Ingredients



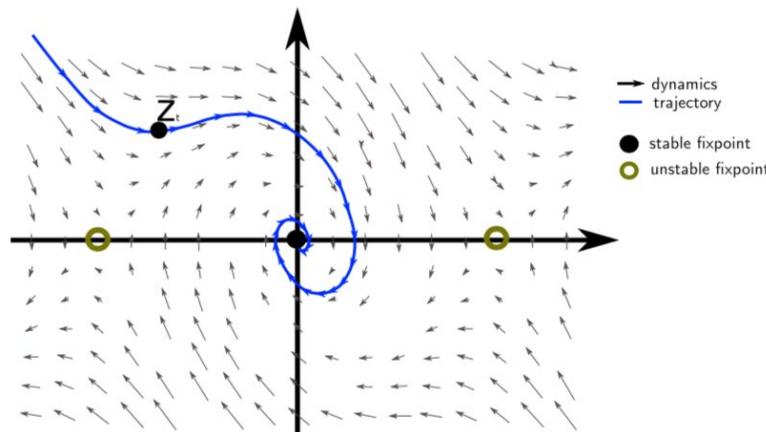
1. **Stochastic optimal control (SOC) in a latent space**
2. E2C latent state space model:
 - ▶ Variational autoencoder
 - ▶ Locally linear latent space model

Embed to Control

Embed to Control - Stochastic Optimal Control



- Local Gaussian z_t , local **accurate** linearization of dynamics:

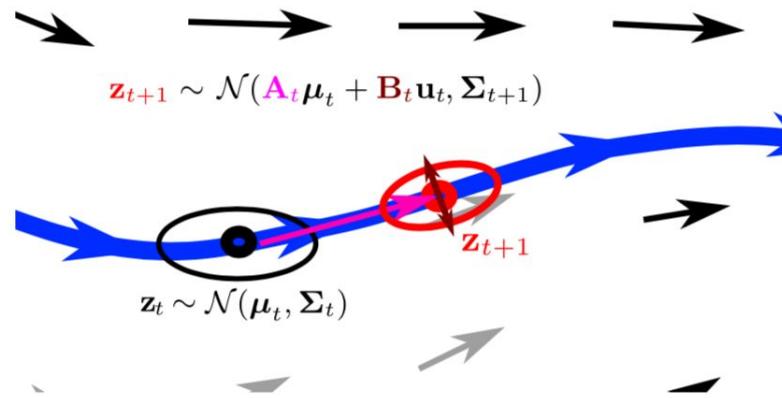


Embed to Control

Embed to Control - Stochastic Optimal Control



- Local Gaussian \mathbf{z}_t , local **accurate** linearization of dynamics:

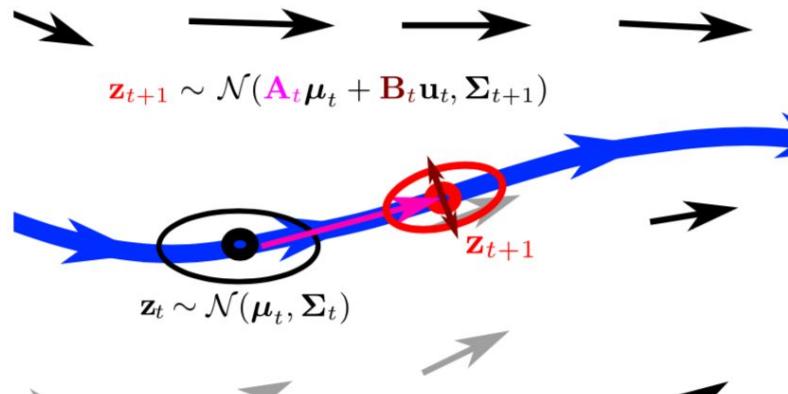


Embed to Control

Embed to Control - Stochastic Optimal Control



- ▶ Local Gaussian \mathbf{z}_t , local **accurate** linearization of dynamics:



- ▶ assume the costs $c(\mathbf{z}_t, \mathbf{u}_t)$ are quadratic in \mathbf{z}_t
- locally optimal controls can be found (iLQR, AICO), if we can learn \mathbf{z}

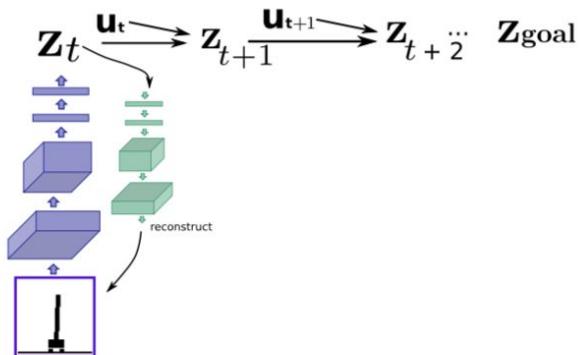
Embed to Control



Key Requirements

Requirements for the inferred latent state space z_t :

- 1 Capture sufficient information about x_t



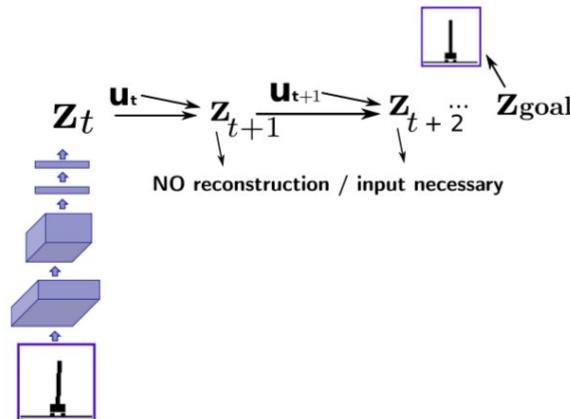
Embed to Control

Key Requirements



Requirements for the inferred latent state space z_t :

- 1 Capture sufficient information about x_t
- 2 Accurate long-term prediction of latent states



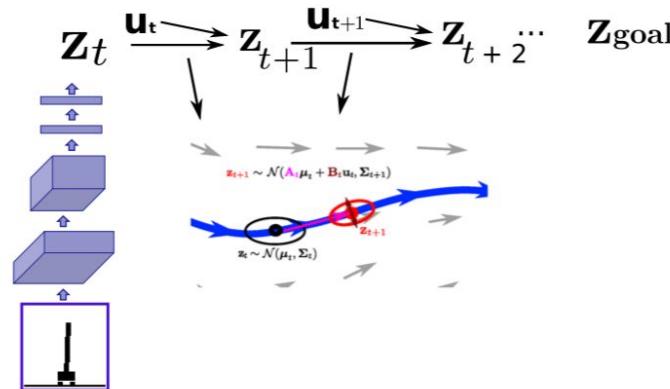
Embed to Control



Key Requirements

Requirements for the inferred latent state space z_t :

- 1 Capture sufficient information about x_t
- 2 Accurate long-term prediction of latent states
- 3 The prediction must be locally linearizable *for all valid control magnitudes*



Embed to Control

Key Requirements



Requirements for the inferred latent state space \mathbf{z}_t :

- 1 Capture sufficient information about \mathbf{x}_t
- 2 Accurate long-term prediction of latent states
- 3 The prediction must be locally linearizable *for all valid control magnitudes*
 - IF we can learn \mathbf{z} , locally optimal controls can be found
 - present model that follows requirements **by construction**

Embed to Control

Ingredients



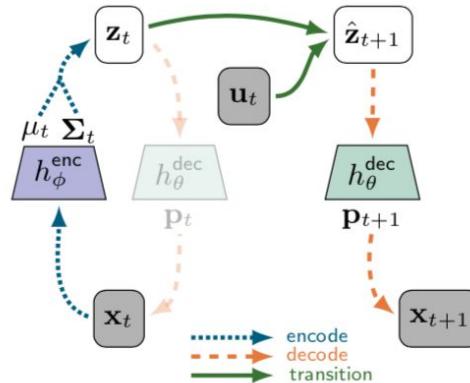
1. Stochastic optimal control (SOC) in a latent space
2. **E2C latent state space model:**
 - ▶ **Variational autoencoder**
 - ▶ Locally linear latent space model

Embed to Control



E2C: Modelling Dynamics

- ▶ From $(\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1})$ to $(\mathbf{z}_t, \mathbf{u}_t, \hat{\mathbf{z}}_{t+1})$
- ▶ Transition Model: 2
 $\mathbf{z}_{t+1} \sim \hat{Q}_\psi(\hat{Z}|Z, \mathbf{u}) = ?$



Embed to Control

Ingredients



1. Stochastic optimal control (SOC) in a latent space
2. **E2C latent state space model:**
 - ▶ Variational autoencoder
 - ▶ Locally linear latent space model

Embed to Control

E2C: Locally Linear Latent Dynamics

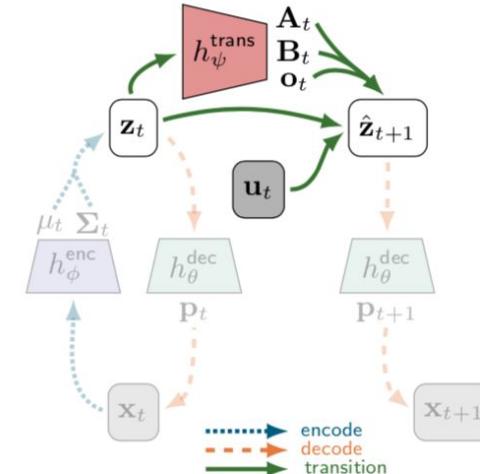


► Transition Model: 3

$$\hat{Q}_\psi(\hat{Z}|Z, \mathbf{u}) = \mathcal{N}(\mathbf{A}_t \mu_t + \mathbf{B}_t \mathbf{u}_t + \mathbf{o}_t, \mathbf{C}_t)$$

$$\mathbf{C}_t = \mathbf{A}_t \Sigma_t \mathbf{A}_t^T + \mathbf{H}_t$$

$$\boldsymbol{\omega}_t \sim \mathcal{N}(0, \mathbf{H}_t)$$



$$\begin{aligned}\mathcal{L}(\mathcal{D}) &= \mathbb{E}_{\mathbf{z}_t \sim Q_\phi} [-\log P_\theta(\mathbf{x}_t | \mathbf{z}_t)] + \text{KL}(Q_\phi || P(Z)) \\ &\quad + \mathbb{E}_{\hat{\mathbf{z}}_{t+1} \sim \hat{Q}_\psi} [-\log P_\theta(\mathbf{x}_{t+1} | \hat{\mathbf{z}}_{t+1})]\end{aligned}$$

Embed to Control



E2C: Locally Linear Latent Dynamics

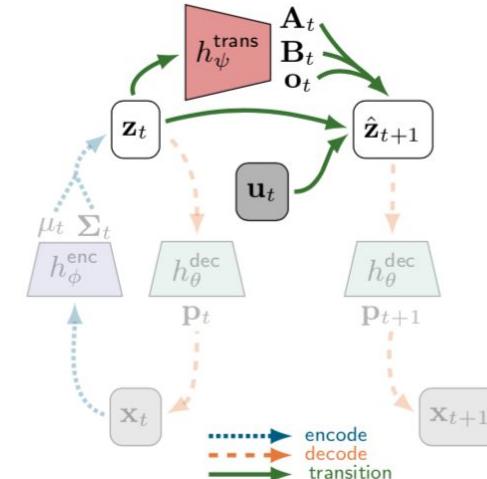
- Transition Model: 3

$$\hat{Q}_\psi(\hat{Z}|Z, \mathbf{u}) = \mathcal{N}(\mathbf{A}_t \boldsymbol{\mu}_t + \mathbf{B}_t \mathbf{u}_t + \mathbf{o}_t, \mathbf{C}_t)$$

$$\mathbf{C}_t = \mathbf{A}_t \Sigma_t \mathbf{A}_t^T + \mathbf{H}_t$$

$$\boldsymbol{\omega}_t \sim \mathcal{N}(0, \mathbf{H}_t)$$

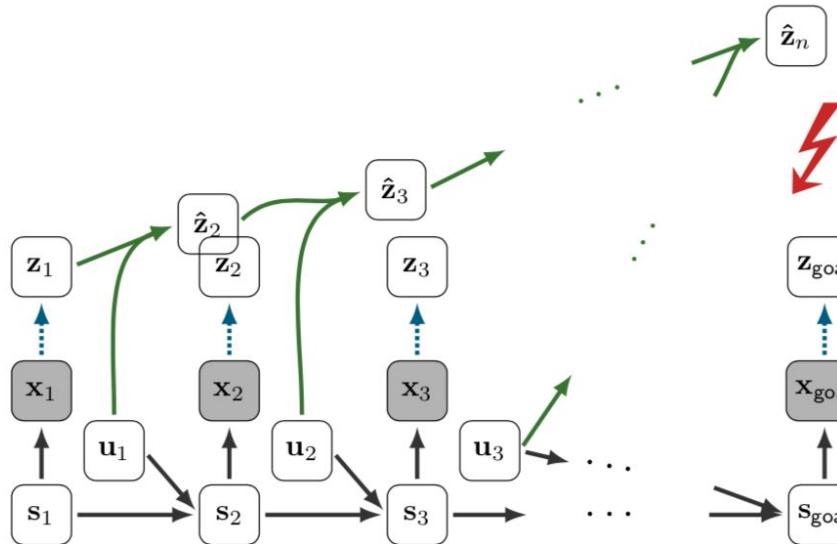
- ⚡ Disagreement of inference & transition model



$$\begin{aligned}\mathcal{L}(\mathcal{D}) = & \mathbb{E}_{\mathbf{z}_t \sim Q_\phi} [-\log P_\theta(\mathbf{x}_t | \mathbf{z}_t)] + \text{KL}(Q_\phi || P(Z)) \\ & + \mathbb{E}_{\hat{\mathbf{z}}_{t+1} \sim \hat{Q}_\psi} [-\log P_\theta(\mathbf{x}_{t+1} | \hat{\mathbf{z}}_{t+1})]\end{aligned}$$

Embed to Control

Problem: Latent State Divergence

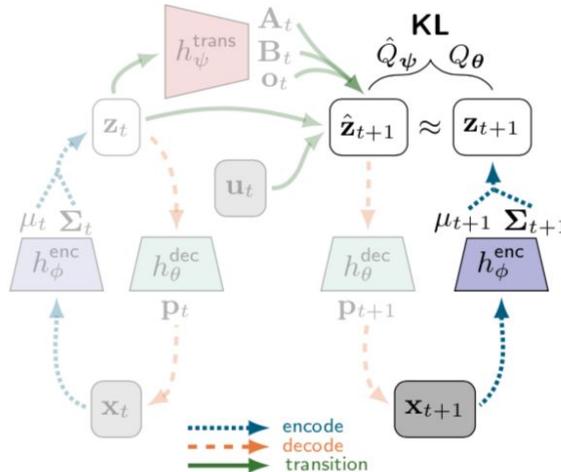


Embed to Control



E2C: Long-term Consistency

→ Force encoder and transition model to work in the same latent space



$$\begin{aligned}\mathcal{L}(\mathcal{D}) = & \mathbb{E}_{\mathbf{z}_t \sim Q_\phi} [-\log P_\theta(\mathbf{x}_t | \mathbf{z}_t)] + \text{KL}(Q_\phi || P(Z)) \\ & + \mathbb{E}_{\hat{\mathbf{z}}_{t+1} \sim \hat{Q}_\psi} [-\log P_\theta(\mathbf{x}_{t+1} | \hat{\mathbf{z}}_{t+1})] + \lambda \text{KL}(\hat{Q}_\psi(\hat{\mathbf{z}} | Z, \mathbf{u}) || Q_\phi(Z | X))\end{aligned}$$

Embed to Control

Requirements



Requirements for the inferred latent state space z_t :

- 1 Capture sufficient information about x_t
→ Autoencoder
- 2 Accurate long-term prediction of latent states
→ Minimize KL of \hat{Q}_ψ and Q_ϕ
- 3 The prediction must be locally linearizable for all valid control magnitudes
→ Predict transformation matrices

Embed to Control



Embed to Control: Full Model

Inference Model:

$$\mathbf{z}_t \sim Q_\phi(Z|X) = \mathcal{N}(Z|\mu, \Sigma_t)$$

Generative Model:

$$\mathbf{x}_t \sim P_\theta(X|Z) = \text{Bernoulli}(\mathbf{p}_t)$$

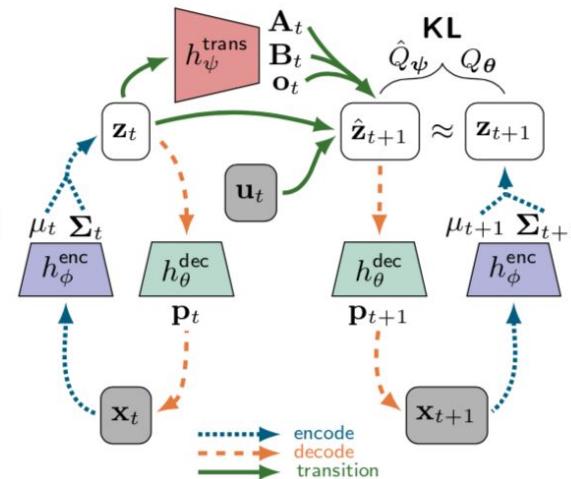
$$\mathbf{x}_{t+1} \sim P_\theta(X|\hat{Z}) = \text{Bernoulli}(\mathbf{p}_{t+1})$$

Transition Model:

$$\mathbf{z}_{t+1} \sim \hat{Q}_\psi(\hat{Z}|Z, \mathbf{u}) = \mathcal{N}(\mathbf{A}_t \mu_t + \mathbf{B}_t \mathbf{u}_t + \mathbf{o}_t, \mathbf{C}_t)$$

$$\mathbf{C}_t = \mathbf{A}_t \Sigma_t \mathbf{A}_t^T + \mathbf{H}_t$$

$$\boldsymbol{\omega}_t \sim \mathcal{N}(0, \mathbf{H}_t)$$



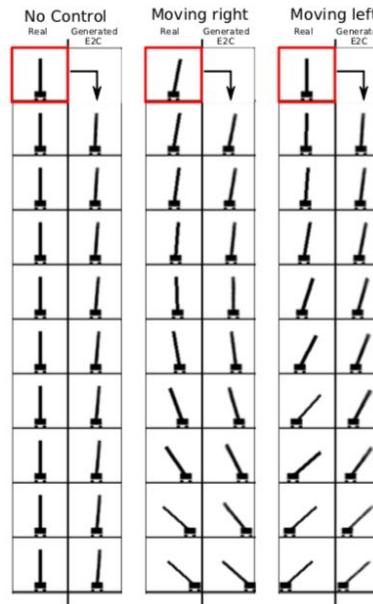
$$\begin{aligned}\mathcal{L}(\mathcal{D}) = & \mathbb{E}_{\mathbf{z}_t \sim Q_\phi} [-\log P_\theta(\mathbf{x}_t|\mathbf{z}_t)] + \text{KL}(Q_\phi||P(Z)) \\ & + \mathbb{E}_{\hat{\mathbf{z}}_{t+1} \sim \hat{Q}_\psi} [-\log P_\theta(\mathbf{x}_{t+1}|\hat{\mathbf{z}}_{t+1})] + \lambda \text{KL}(\hat{Q}_\psi(\hat{Z}|Z, \mathbf{u})||Q_\phi(Z|X))\end{aligned}$$

Embed to Control



Embed to Control: Full Model

Long term predictions



Embed to Control

Results: comparison



Algorithm	Mean State Loss $p(x_t \hat{x}_t)$	Mean Next State Loss $p(x_{t+1} \hat{x}_t, u_t)$	Trajectory Cost		Success percentage
	Planar System			Latent	Real
True Model for s_t	-	-	-	20.24 \pm 4.15	100 %
AE [†]	11.5 \pm 97.8	3538.9 \pm 1395.2	1325.6 \pm 81.2	273.3 \pm 16.4	0 %
VAE [†]	3.6 \pm 18.9	652.1 \pm 930.6	43.1 \pm 20.8	91.3 \pm 16.4	0 %
VAE with slowness [†]	10.5 \pm 22.8	104.3 \pm 235.8	47.1 \pm 20.5	89.1 \pm 16.4	0 %
Non-linear E2C	8.3 \pm 5.5	11.3 \pm 10.1	19.8 \pm 9.8	42.3 \pm 16.4	96.6 %
Global E2C	6.9 \pm 3.2	9.3 \pm 4.6	12.5 \pm 3.9	27.3 \pm 9.7	100 %
E2C	7.7 \pm 2.0	9.7 \pm 3.2	10.3 \pm 2.8	25.1 \pm 5.3	100 %
Inverted Pendulum Swing-Up					
True Model for s_t	-	-	-	9.8 \pm 2.4	100 %
AE [†]	8.9 \pm 100.3	13433.8 \pm 6238.8	1285.9 \pm 355.8	194.7 \pm 44.8	0 %
VAE [†]	7.5 \pm 47.7	8791.2 \pm 17356.9	497.8 \pm 129.4	237.2 \pm 41.2	0 %
VAE with slowness [†]	26.5 \pm 18.0	779.7 \pm 633.3	419.5 \pm 85.8	188.2 \pm 43.6	0 %
E2C no latent KL	64.4 \pm 32.8	87.7 \pm 64.2	489.1 \pm 87.5	213.2 \pm 84.3	0 %
Non-linear E2C	59.6 \pm 25.2	72.6 \pm 34.5	313.3 \pm 65.7	37.4 \pm 12.4	46.6 %
Global E2C	115.5 \pm 56.9	125.3 \pm 62.6	628.1 \pm 45.9	125.1 \pm 10.7	0 %
E2C	84.0 \pm 50.8	89.3 \pm 42.9	275.0 \pm 16.6	15.4 \pm 3.4	90 %

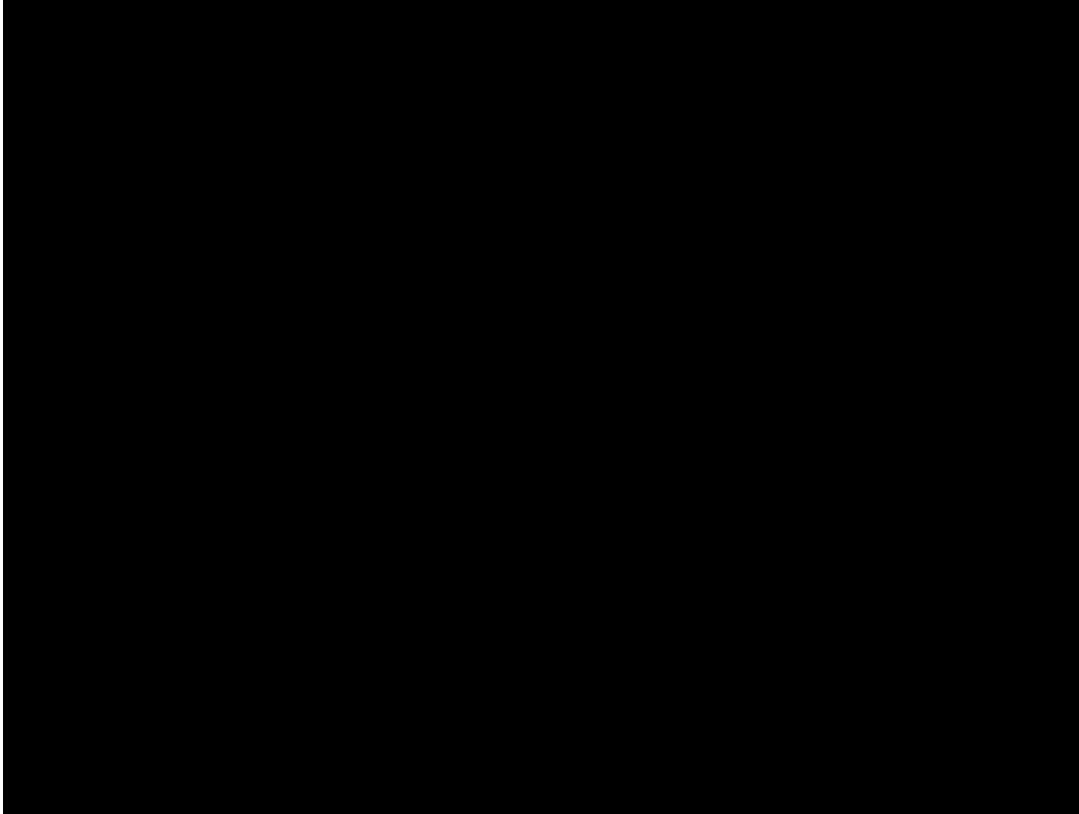
Embed to Control

Results: comparison



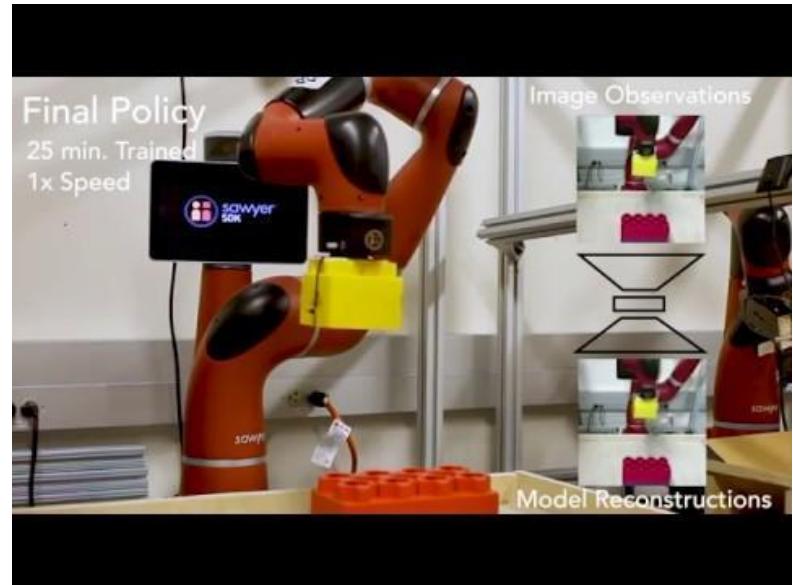
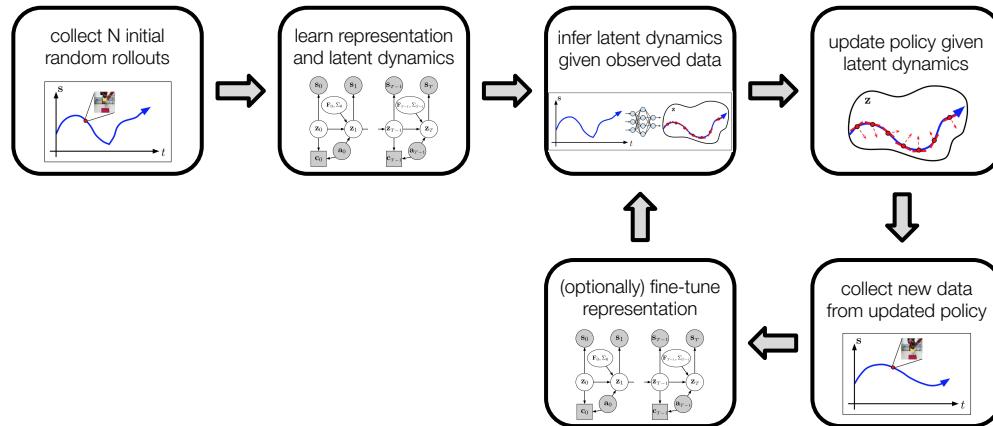
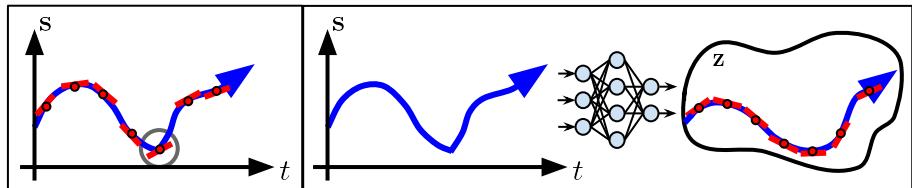
Algorithm	True model	VAE + slownes	E2C no latent KL	Non-linear E2C	E2C
Cart-Pole balance					
Trajectory Cost	15.33 ± 7.70	49.12 ± 16.94	48.90 ± 17.88	31.96 ± 13.26	22.23 ± 14.89
Success %	100 %	0 %	0 %	63 %	93 %
Three-link arm					
Trajectory Cost	59.46	1275.53 ± 864.66	1246.69 ± 262.65	460.40 ± 82.18	90.23 ± 47.38
Success %	100 %	0 %	0 %	40 %	90 %

Embed to Control



SOLAR: Deep Structured Representations for Model-Based Reinforcement Learning

Marvin Zhang*, Sharad Vikram*, Laura Smith, Pieter Abbeel, Matthew Johnson, Sergey Levine



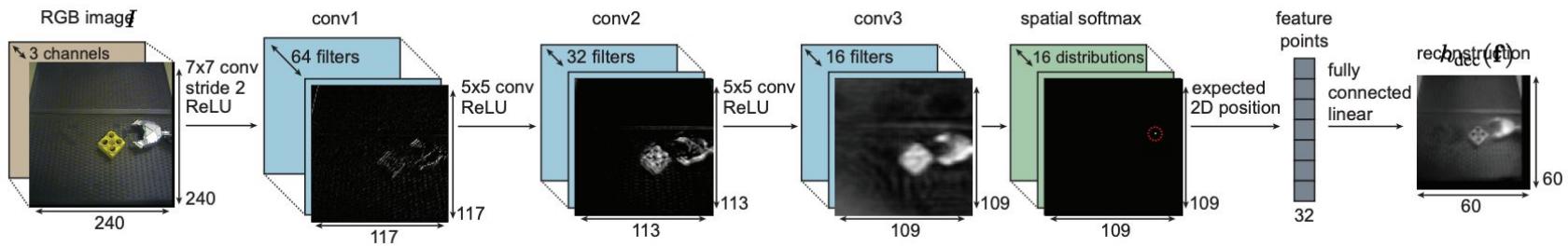
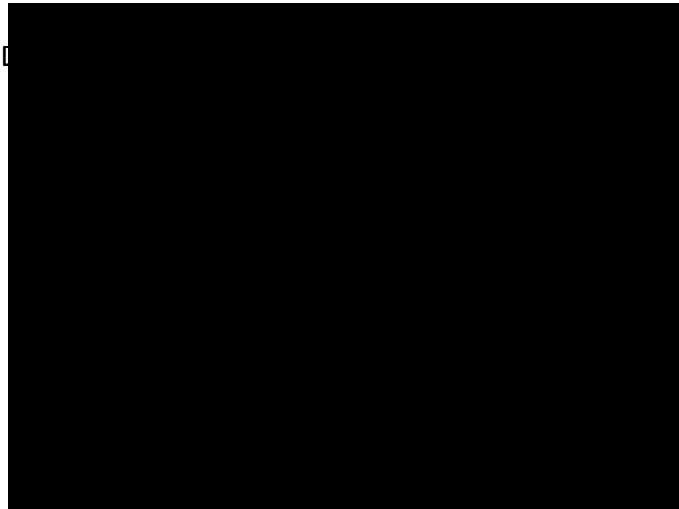
<https://goo.gl/AJKoCL>

Representation Learning in Reinforcement Learning

- Auxiliary losses
- ***State representation***
 - ***Observation -> State***
 - ***Observation -> State + State,Action -> Next State***
- Exploration
- Unsupervised skill discovery

Deep Spatial Autoencoders

- Deep Spatial Autoencoders for Visuomotor Learning, Finn, Tan, Duan, Levine (https://arxiv.org/abs/1509.06113)
 - Train deep spatial autoencoder
 - Model-based RL through iLQR in the latent space



Robotic Priors / PVEs

- PVEs: Position-Velocity Encoders for Unsupervised Learning of Structured State Representations

Rico Jonschkowski, Roland Hafner, Jonathan Scholz, and Martin Riedmiller (<https://arxiv.org/pdf/1705.09805.pdf>)

- Learn an embedding without reconstruct

$$\mathbf{s}_t^{(p)} = \phi(\mathbf{o}_t)$$

$$\mathbf{s}_t^{(v)} = \alpha(\mathbf{s}_t^{(p)} - \mathbf{s}_{t-1}^{(p)})$$

$$L_{\text{conservation}} = \mathbf{E}\left[\left(\|\mathbf{s}_t^{(v)}\| - \|\mathbf{s}_{t-1}^{(v)}\|\right)^2\right]$$

$$L_{\text{variation}} = \mathbf{E}\left[e^{-\|\mathbf{s}_a^{(p)} - \mathbf{s}_b^{(p)}\|}\right]$$

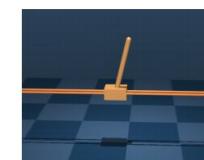
$$\begin{aligned} L_{\text{controlability (i)}} &= e^{-\text{Cov}(\mathbf{a}_{t,i}, \mathbf{s}_{t+1,i}^{(a)})} \\ &= e^{-\mathbf{E}\left[\left(a_{t,i} - \mathbf{E}[a_{t,i}]\right)\left(s_{t+1,i}^{(a)} - \mathbf{E}[s_{t+1,i}^{(a)}]\right)\right]} \end{aligned}$$

$$L_{\text{slowness}} = \mathbf{E}\left[\|\mathbf{s}_t^{(p)} - \mathbf{s}_{t-1}^{(p)}\|^2\right]$$

$$L_{\text{inertia}} = \mathbf{E}\left[\|\mathbf{s}_t^{(v)} - \mathbf{s}_{t-1}^{(v)}\|^2\right] = \mathbf{E}\left[\|\mathbf{s}_t^{(a)}\|^2\right]$$



(a) Inverted pendulum



(b) Cart-pole

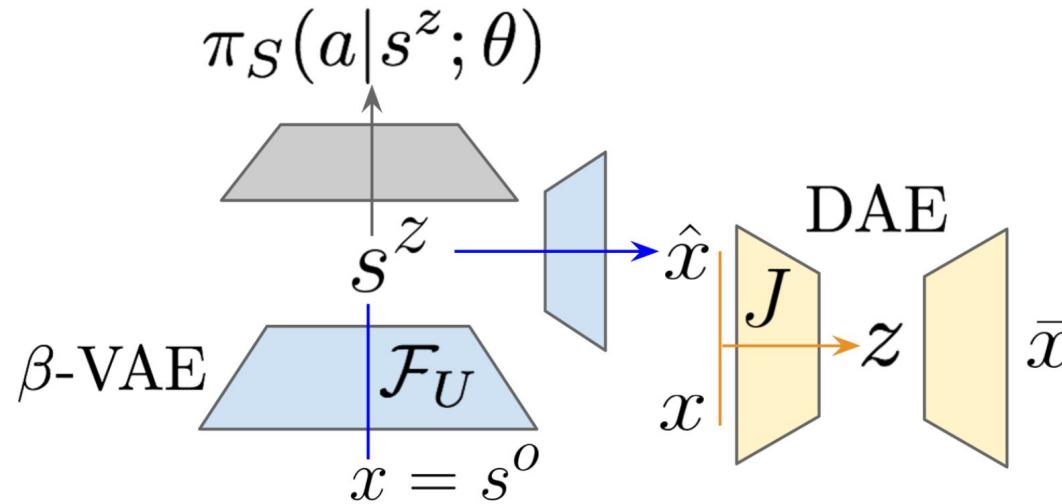


(c) Ball in cup

Disentangled Representation Learning Agent (Darla)

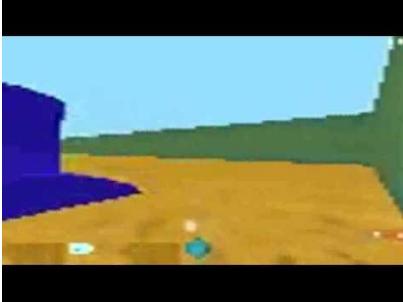
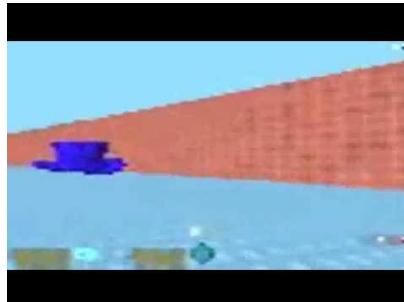
DARLA: Improving Zero-Shot Transfer in Reinforcement Learning

Irina Higgins, Arka Pal, Andrei A. Rusu, Loic Matthey, Christopher P Burgess, Alexander Pritzel, Matthew Botvinick, Charles Blundell, Alexander Lerchner (<https://arxiv.org/abs/1707.08475>)



DeepMind Lab Transfer

DARLA vs DQN baseline

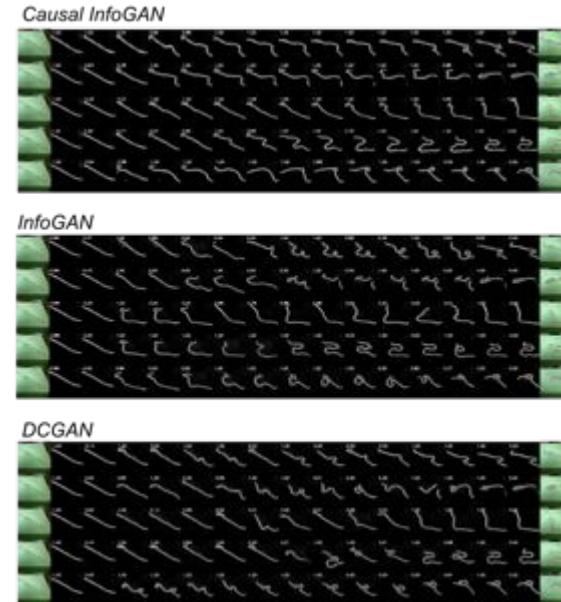
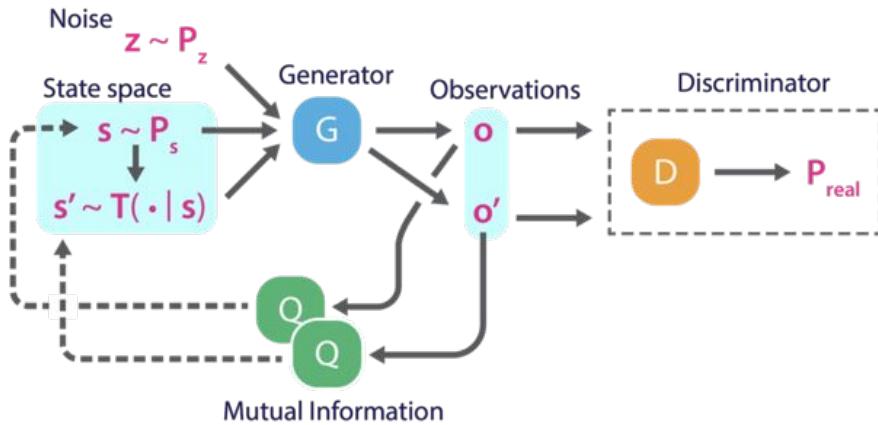
	Train	Transfer
DQN		
DARLA		



Causal InfoGAN

Learning Plannable Representations with Causal InfoGAN

Thanard Kurutach, Aviv Tamar, Ge Yang, Stuart Russell, Pieter Abbeel (<https://arxiv.org/pdf/1807.09341.pdf>)



PlaNet

Learning latent dynamics for planning from pixels

Danijar Hafner, T. Lillicrap, I Fischer, R Villegas, D Ha, H Lee

- Learn latent space dynamics model
- Multi-step prediction
- Planning in latent space



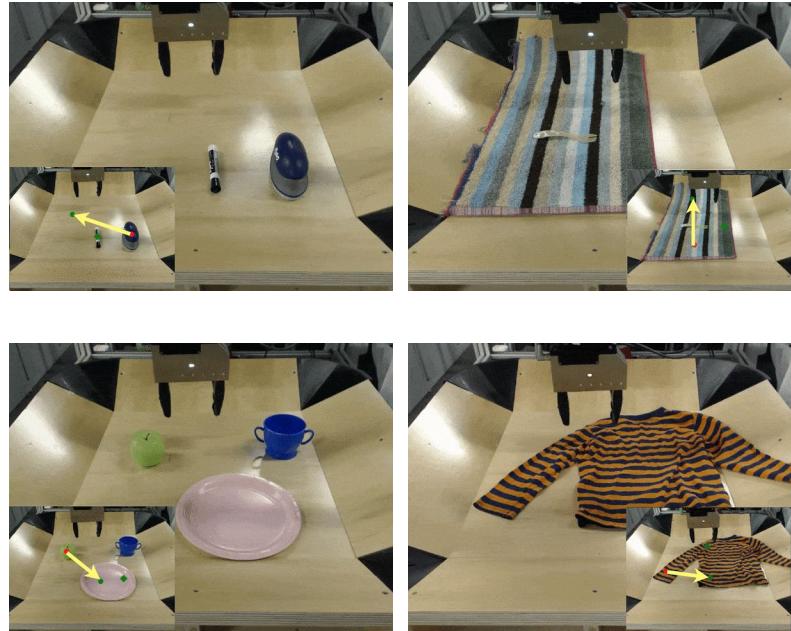
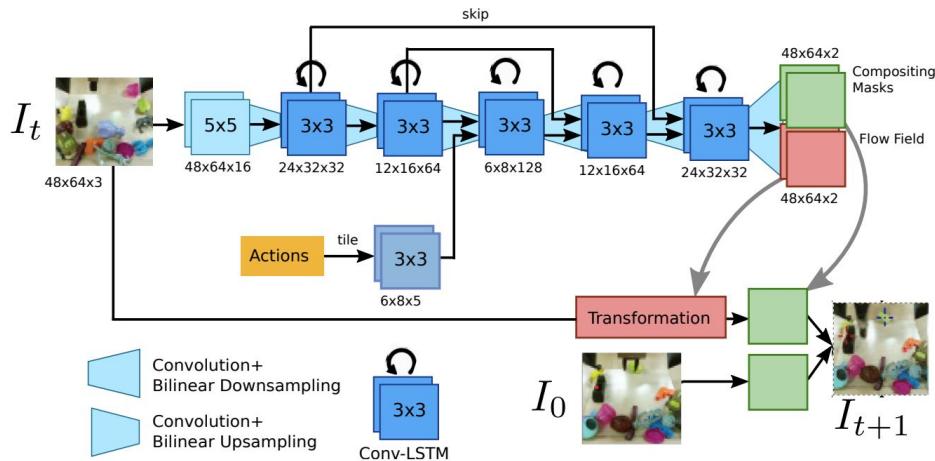
Visual Foresight

Deep Visual Foresight for Planning Robot Motion, Finn and Levine, ICRA 2017 <http://arxiv.org/abs/1610.00696>

Visual Foresight: Model-Based Deep Reinforcement Learning for Vision-Based Robotic Control, Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, Sergey Levine, <https://arxiv.org/abs/1812.00568>,

<https://bair.berkeley.edu/blog/2018/11/30/visual-rl/>

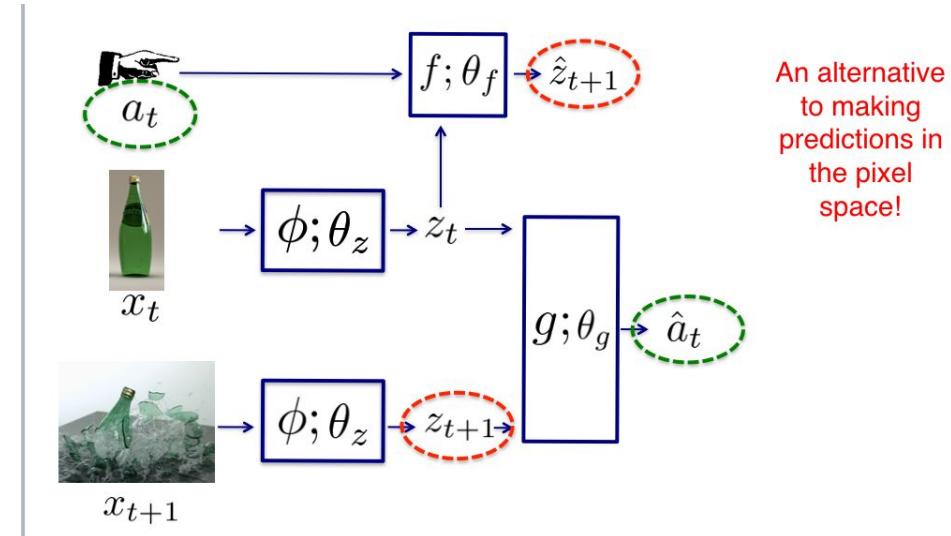
- Video prediction + Cross Entropy Maximization for MPC



Forward + Inverse Dynamics Models

Learning to Poke by Poking: Experiential Learning of Intuitive Physics, Pulkit Agrawal, Ashvin Nair, Pieter Abbeel, Jitendra Malik, Sergey Levine, <https://arxiv.org/abs/1606.07419>

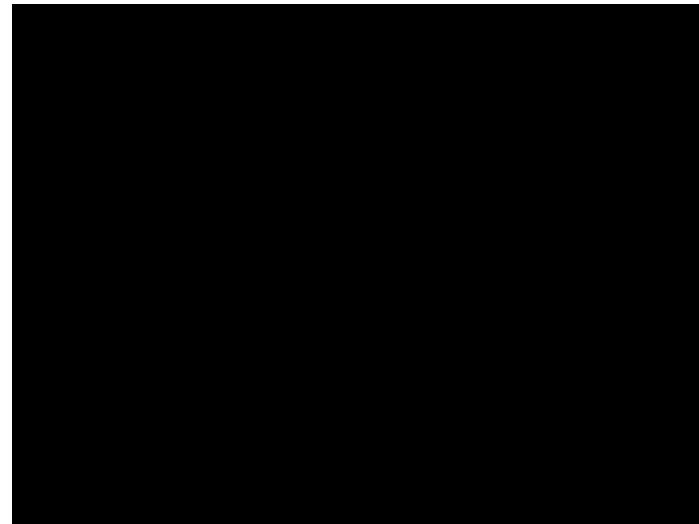
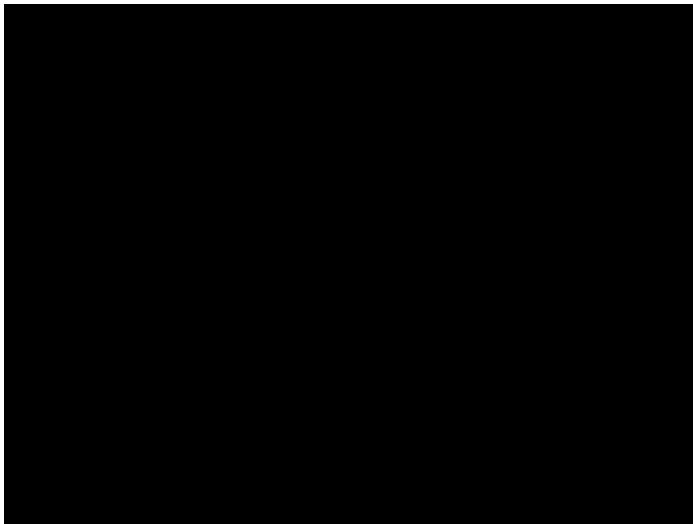
- Learning a forward model in latent space
- BUT: couldn't the latent features always be zero?
- SOLUTION: require the features from t and t+1 to be sufficient to predict a_t



Forward + Inverse Dynamics Models

Learning to Poke by Poking: Experiential Learning of Intuitive Physics, Pulkit Agrawal, Ashvin Nair, Pieter Abbeel, Jitendra Malik, Sergey Levine, <https://arxiv.org/abs/1606.07419>

- Learning a forward model in latent space
- BUT: couldn't the latent features always be zero?
- SOLUTION: require the features from t and t+1 to be sufficient to predict a_t



Representation Learning in Reinforcement Learning

- Auxiliary losses
- ***State representation***
 - ***Observation -> State***
 - ***Observation -> State + State,Action -> Next State***
 - ***Observation -> State + State,Action -> Next State, Future Reward***
- Exploration
- Unsupervised skill discovery

Predictron

The Predictron: End-To-End Learning and Planning

David Silver, Hado van Hasselt, Matteo Hessel, Tom Schaul, Arthur Guez, Tim Harley, Gabriel Dulac-Arnold, David Reichert, Neil Rabinowitz, Andre Barreto, Thomas Degrif (<https://arxiv.org/pdf/1612.08810.pdf>)

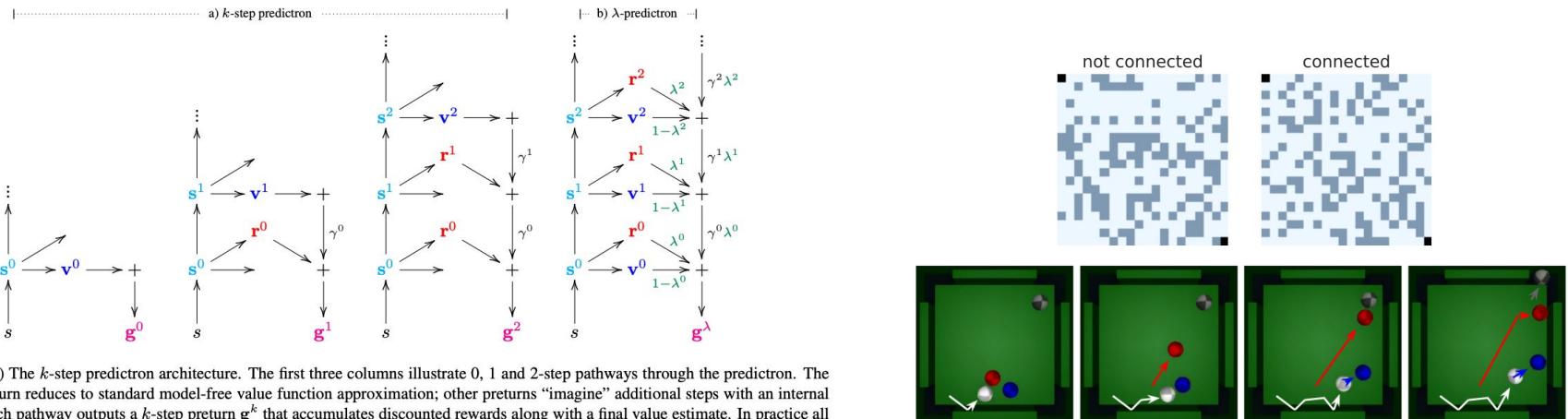


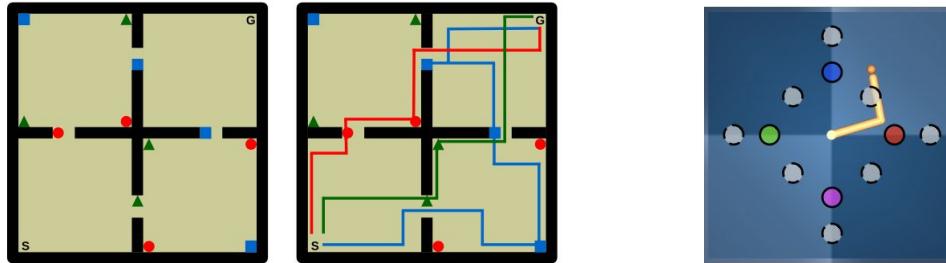
Figure 1. a) The k -step predictron architecture. The first three columns illustrate 0, 1 and 2-step pathways through the predictron. The 0-step preturn reduces to standard model-free value function approximation; other preturns “imagine” additional steps with an internal model. Each pathway outputs a k -step preturn g^k that accumulates discounted rewards along with a final value estimate. In practice all k -step preturns are computed in a single forward pass. b) The λ -predictron architecture. The λ -parameters gate between the different preturns. The output is a λ -preturn g^λ that is a mixture over the k -step preturns. For example, if $\lambda^0 = 1, \lambda^1 = 1, \lambda^2 = 0$ then we recover the 2-step preturn, $g^\lambda = g^2$. Discount factors γ^k and λ -parameters λ^k are dependent on state s^k ; this dependence is not shown in the figure.

Successor Features

Successor Features for Transfer in Reinforcement Learning

André Barreto, Will Dabney, Rémi Munos, Jonathan J. Hunt, Tom Schaul, Hado van Hasselt, David Silver (<https://arxiv.org/abs/1606.05312>)

$$\begin{aligned} Q^\pi(s, a) &= \mathbb{E}^\pi [r_{t+1} + \gamma r_{t+2} + \dots | S_t = s, A_t = a] \\ &= \mathbb{E}^\pi [\phi_{t+1}^\top \mathbf{w} + \gamma \phi_{t+2}^\top \mathbf{w} + \dots | S_t = s, A_t = a] \\ &= \mathbb{E}^\pi \left[\sum_{i=t}^{\infty} \gamma^{i-t} \phi_{i+1}^\top | S_t = s, A_t = a \right]^\top \mathbf{w} = \psi^\pi(s, a)^\top \mathbf{w} \end{aligned}$$



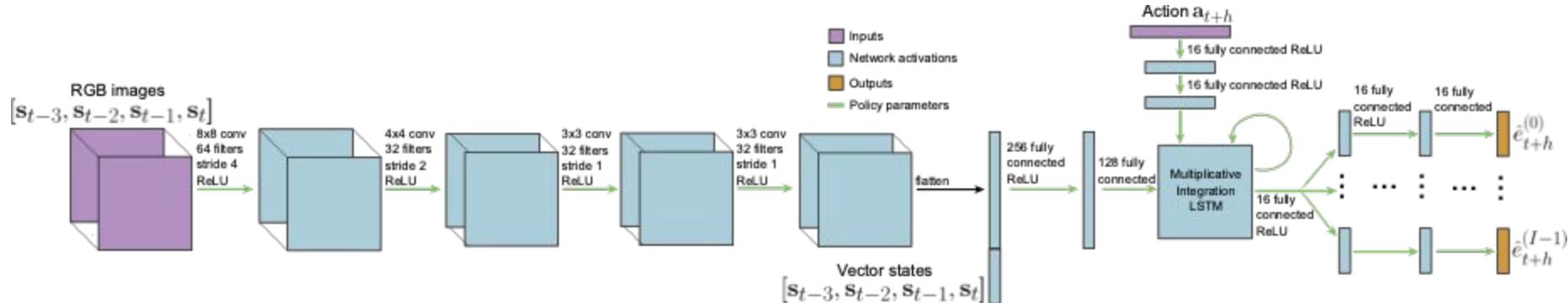
Kahn et al.

Composable Action-Conditioned Predictors: Flexible Off-Policy Learning for Robot Navigation

Gregory Kahn*, Adam Villaflor*, Pieter Abbeel, Sergey Levine, CoRL 2018 (<https://arxiv.org/pdf/1810.07167.pdf>)

Self-supervised Deep Reinforcement Learning with Generalized Computation Graphs for Robot Navigation

Gregory Kahn, Adam Villaflor, Bosen Ding, Pieter Abbeel, Sergey Levine, ICRA 2018 (<https://arxiv.org/pdf/1709.10489.pdf>)



Kahn et al.

Composable Action-Conditioned Predictors: Flexible Off-

Gregory Kahn*, Adam Villaflor*, Pieter Abbeel, Sergey Le

Self-supervised Deep Reinforcement Learning with Gene

Gregory Kahn, Adam Villaflor, Bosen Ding, Pieter Abbeel,



Representation Learning in Reinforcement Learning

- Auxiliary losses
- ***State representation***
 - ***Observation -> State***
 - ***Observation -> State + State,Action -> Next State***
 - ***Observation -> State + State,Action -> Next State, Future Reward***
 - ***Optimal Representations?***
- Exploration
- Unsupervised skill discovery

Some Theory References on State Representations

- From skills to symbols: Learning symbolic representations for abstract high-level planning:
<https://jair.org/index.php/jair/article/view/11175>
- Homomorphism: <https://www.cse.iitm.ac.in/~ravi/papers/KBCS04.pdf>
- Towards a unified theory of state abstraction for mdps:
<https://pdfs.semanticscholar.org/ca9a/2d326b9de48c095a6cb5912e1990d2c5ab46.pdf>
- Model reduction techniques for computing approximately optimal solutions for markov decision processes.<https://arxiv.org/abs/1302.1533>
- Adaptive aggregation methods for infinite horizon dynamic programming
- Transfer via soft homomorphisms. http://www.ifaamas.org/Proceedings/aamas09/pdf/01_Full%20Papers/12_67_FP_0798.pdf
- Near optimal behavior via approximate state abstraction <https://arxiv.org/abs/1701.04113>
- Using PCA to Efficiently Represent State Spaces: <http://irll.eecs.wsu.edu/wp-content/papercite-data/pdf/2015icml-curran.pdf>

A Separation Principle for Control in the Age of Deep Learning

A Separation Principle for Control in the Age of Deep Learning

Alessandro Achille, Stefano Soatto (<https://arxiv.org/abs/1711.03321>)

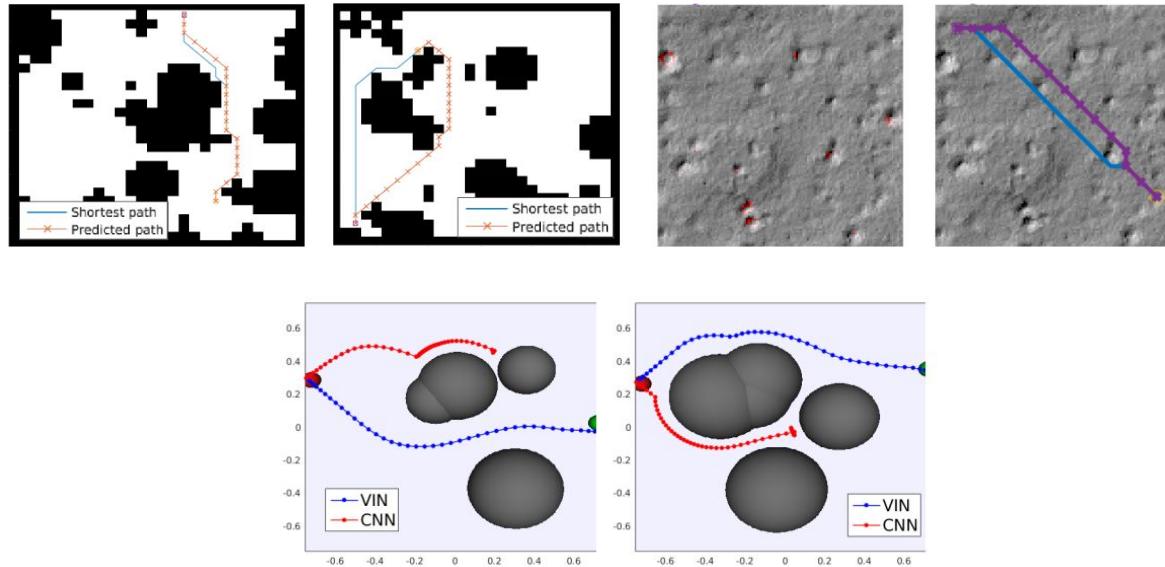
We review the problem of defining and inferring a “state” for a control system based on complex, high-dimensional, highly uncertain measurement streams such as videos. Such a state, or representation, should contain all and only the information needed for control, and discount nuisance variability in the data.

Representation Learning in Reinforcement Learning

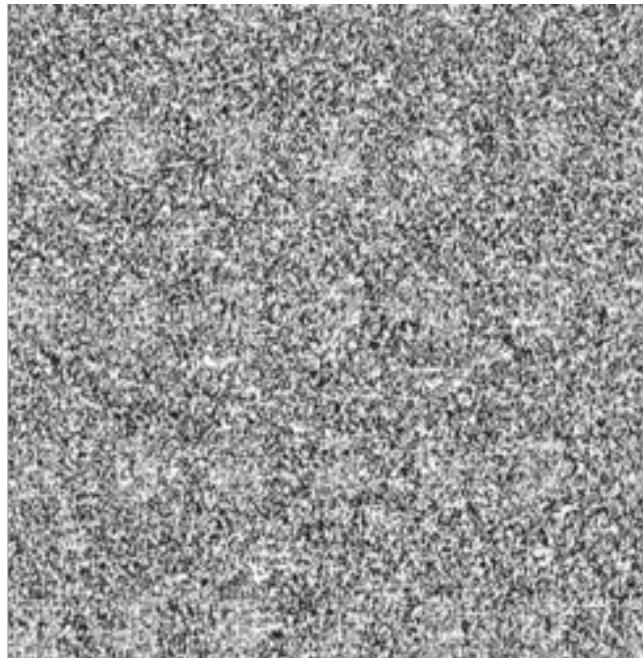
- Auxiliary losses
- ***State representation***
 - ***Observation -> State***
 - ***Observation -> State + State,Action -> Next State***
 - ***Observation -> State + State,Action -> Next State, Future Reward***
 - ***Optimal Representations?***
 - ***“End-to-end”: Learning a representation that’s good for planning***
- Exploration
- Unsupervised skill discovery

Learning Representations for Planning

- Value Iteration Networks, Aviv Tamar, Yi Wu, Garrett Thomas, Sergey Levine, Pieter Abbeel, NeurIPS2016,
<https://arxiv.org/abs/1602.02867>



Pixel-level Video Prediction?



Pixel-level Video Prediction?

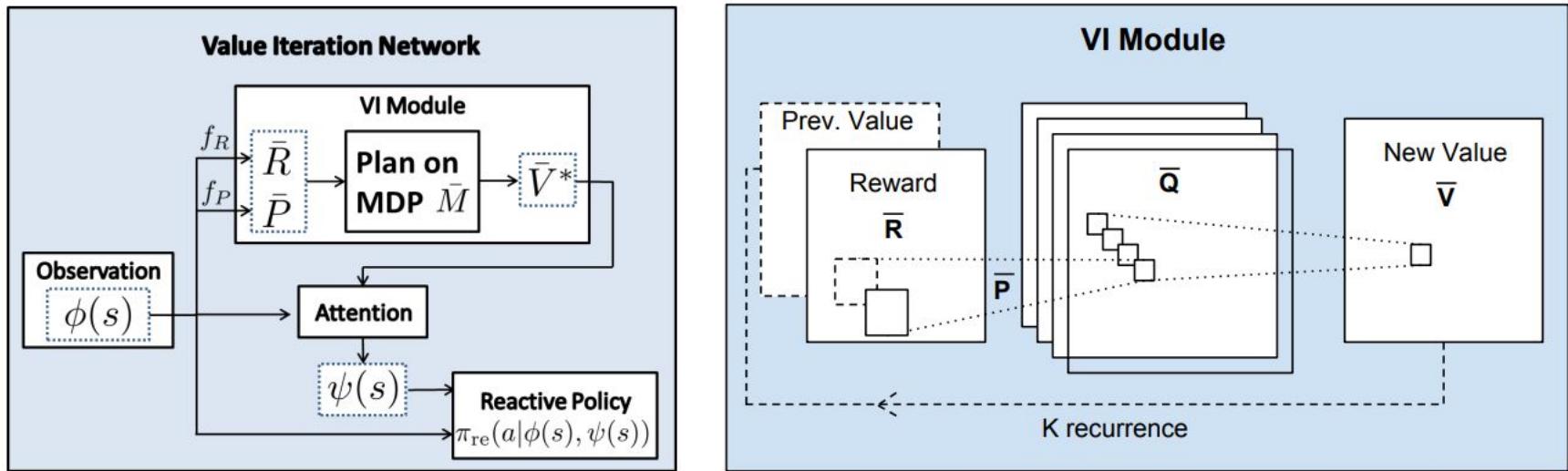


So:

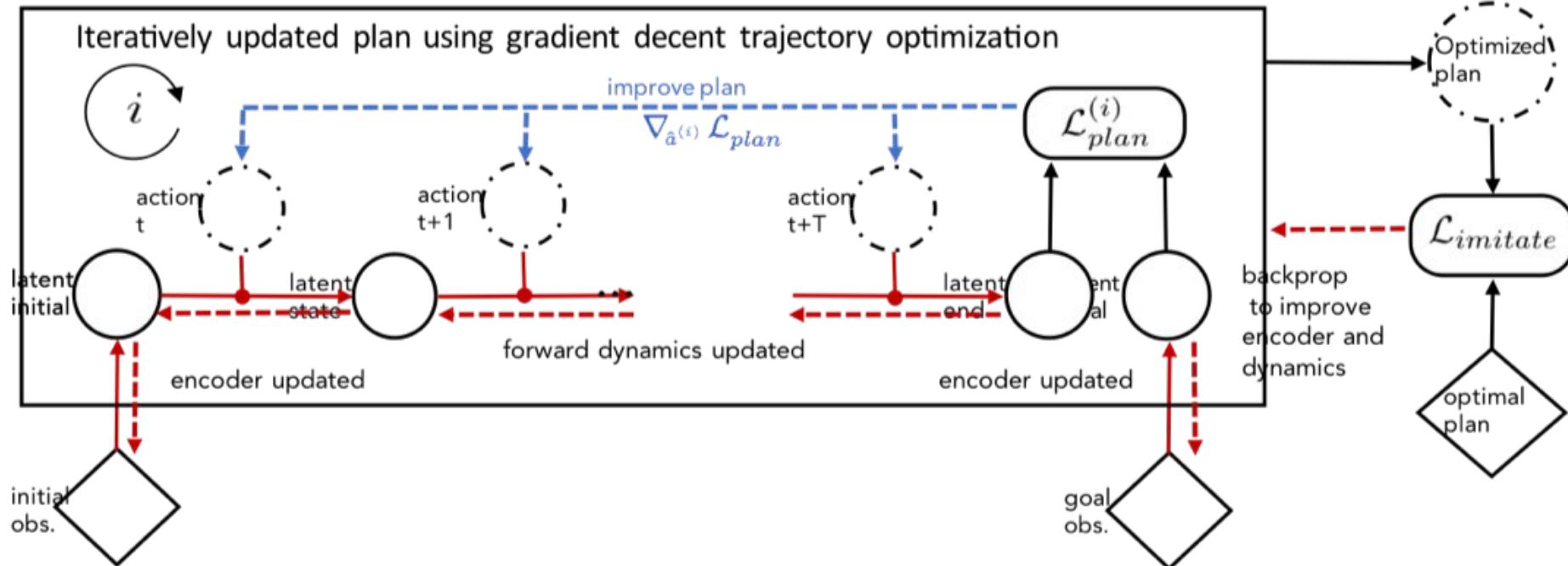
Plannability as the criterion for representation learning

Learning Representations for Planning

- Value Iteration Networks, Aviv Tamar, Yi Wu, Garrett Thomas, Sergey Levine, Pieter Abbeel, NeurIPS2016,
<https://arxiv.org/abs/1602.02867>



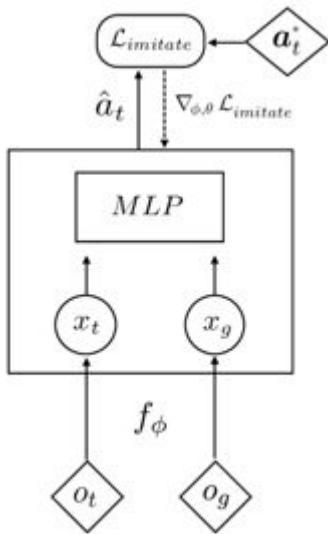
Universal Planning Networks



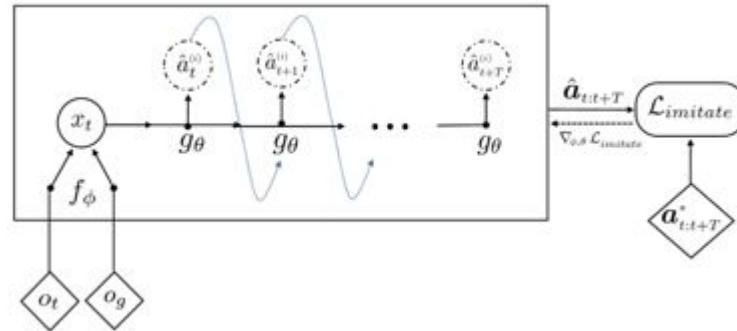
Universal Planning Networks

- Universal (goal-conditioned) policy parameterization with gradient-descent trajectory optimization embedded
- Benefits: better generalization (inductive bias)
- More benefits: learned metric in an abstract space

Approaches Compared With:

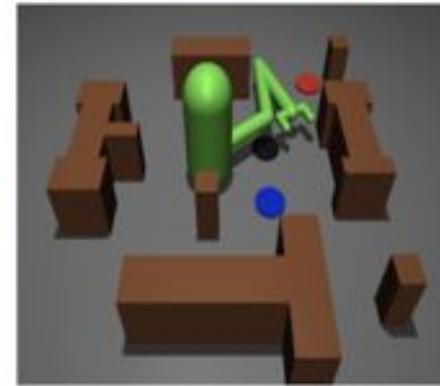
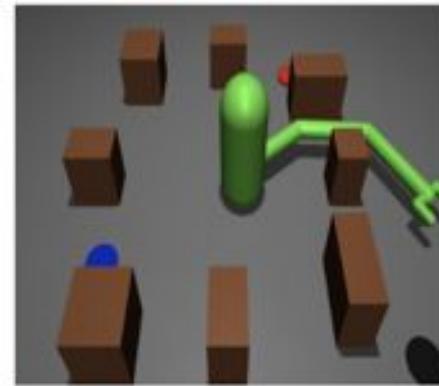
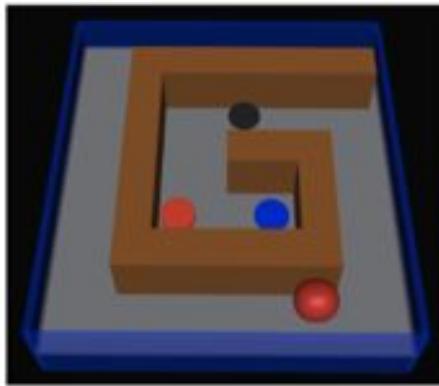


Reactive Imitation Learner



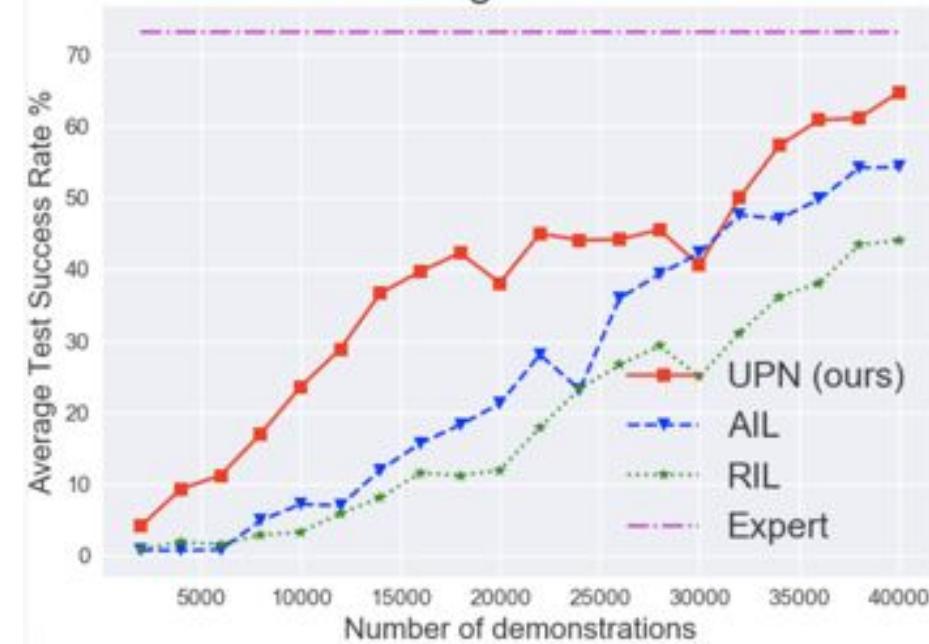
Autoregressive Imitation Learner

Tasks

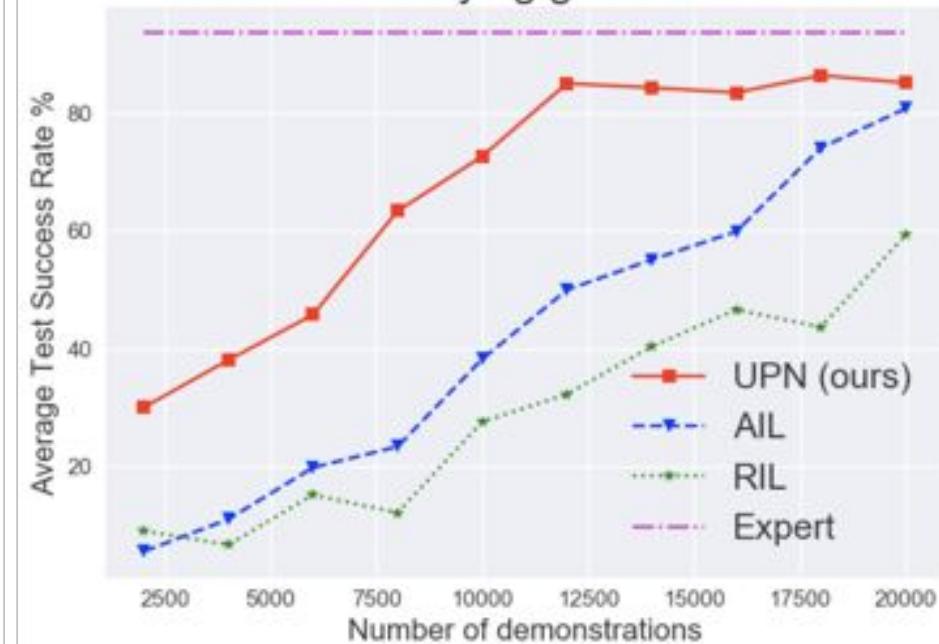


Evaluation

3-Link Reacher with varying obstacles & goals

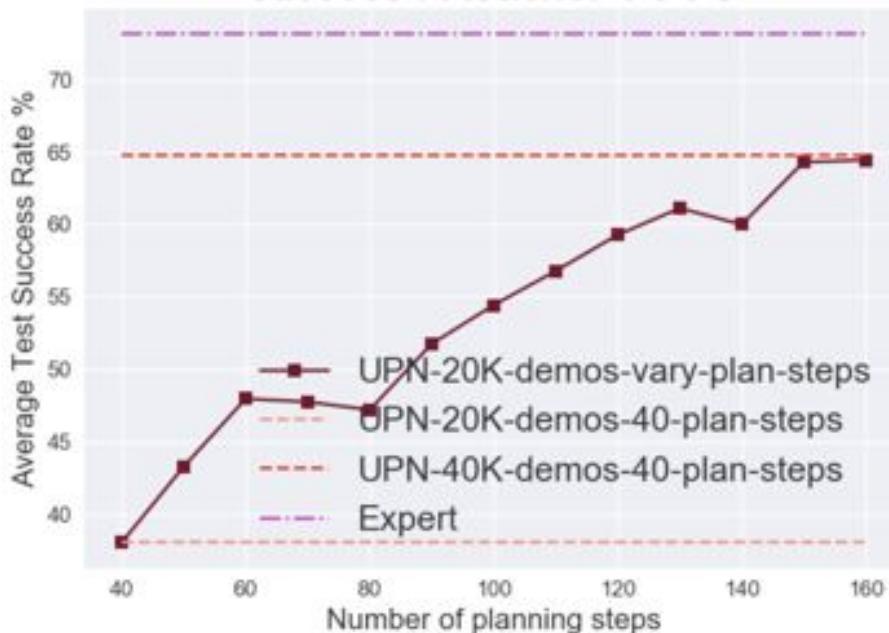


3-Link Reacher with fixed obstacles and varying goals

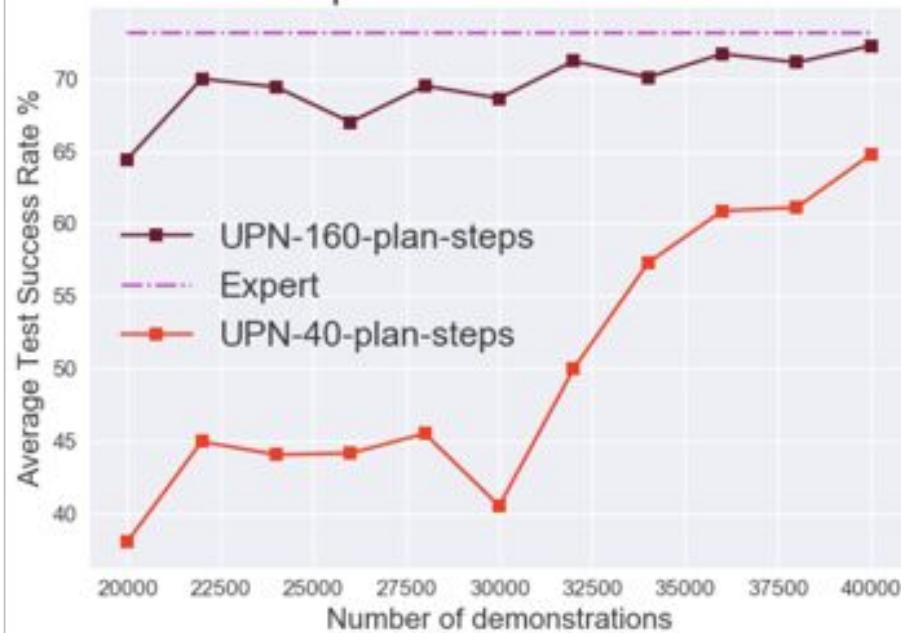


Meta-learning Aspects

Effect of more planning steps on task success : Reacher VOVG

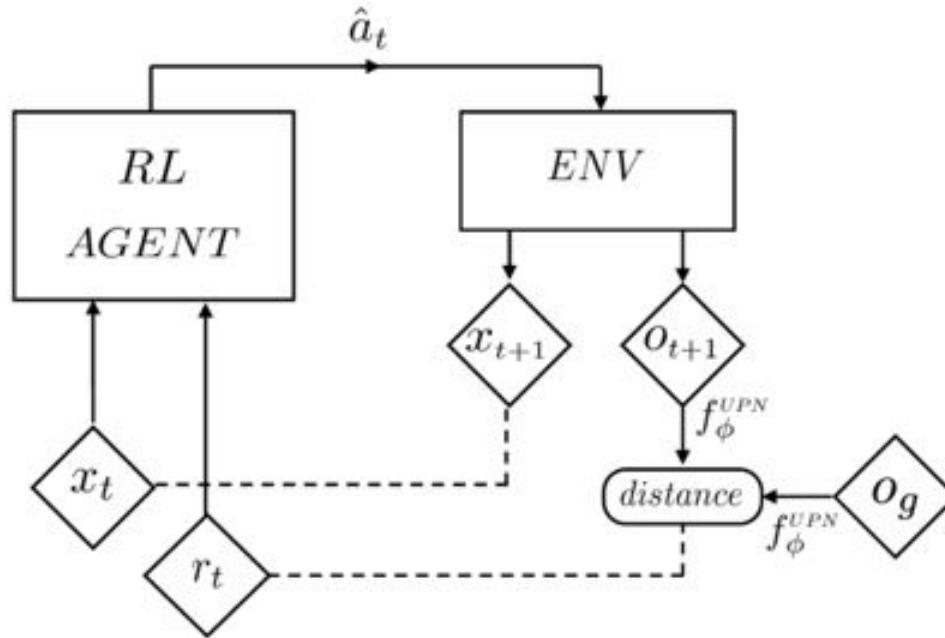


UPN analysis with 40 and 160 planning steps on reacher VOVG



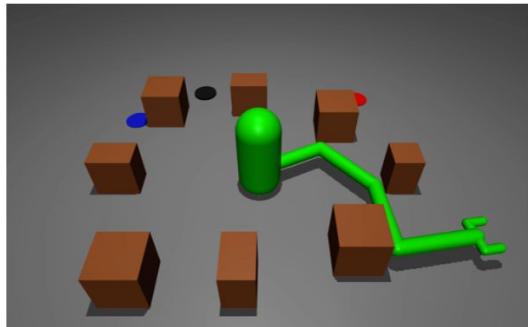
Beyond Imitation...

RL using UPN representations for (shaped) rewards



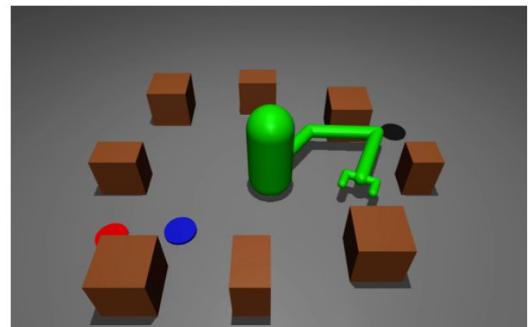
RL using UPN representations

3-link



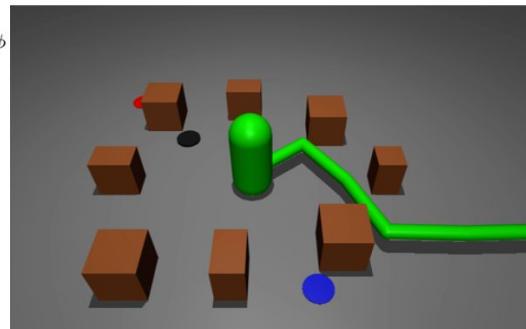
UPN(ϕ, θ) trained with
shared f_ϕ , different g_θ

4-link

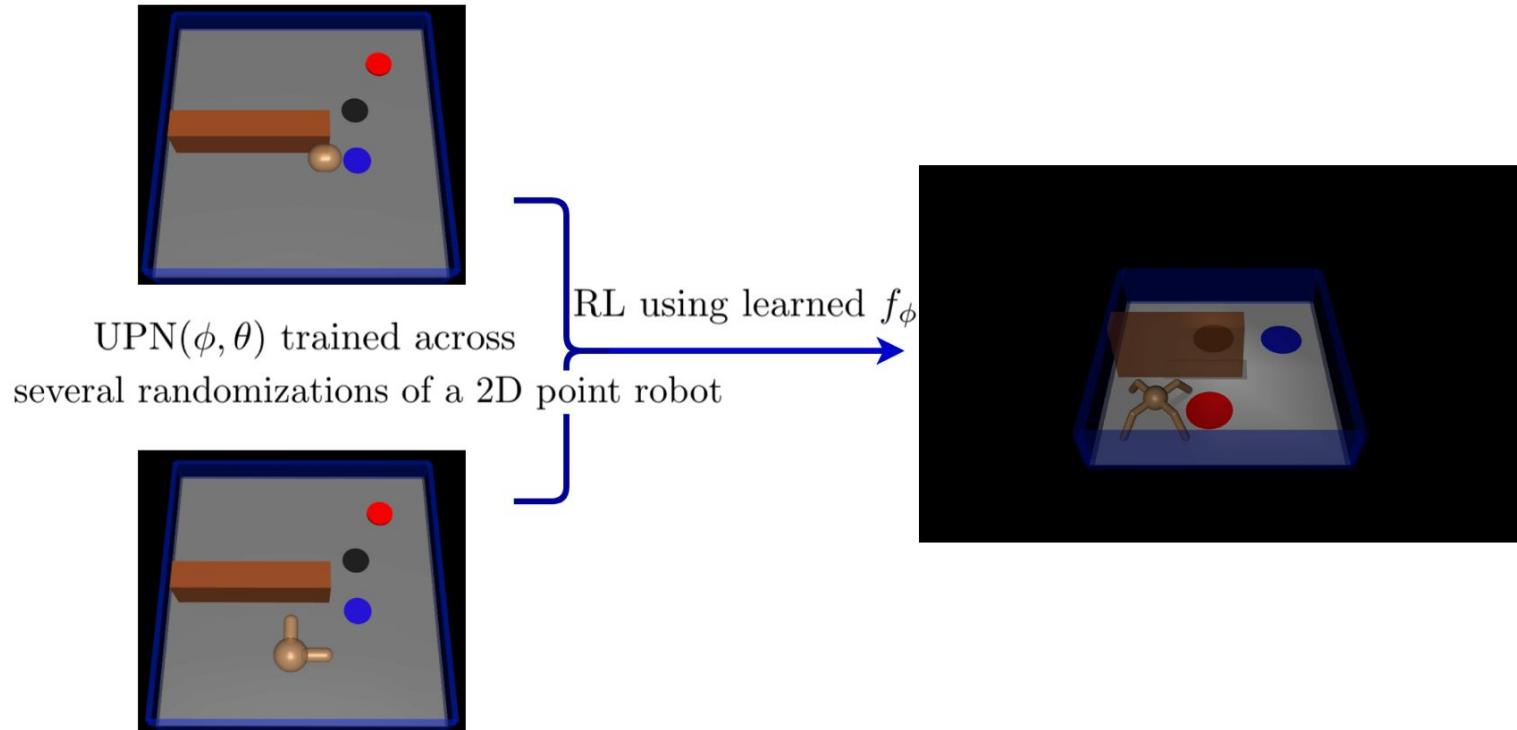


RL using learned f_ϕ

5-link



RL using UPN representations



Representation Learning in Reinforcement Learning

- Auxiliary losses
- ***State representation***
 - ***Observation -> State***
 - ***Observation -> State + State,Action -> Next State***
 - ***Observation -> State + State,Action -> Next State, Future Reward***
 - ***Optimal Representations?***
 - ***“End-to-end”: Learning a representation that’s good for planning***
- Exploration
- Unsupervised skill discovery

Representation Learning in Reinforcement Learning

- Auxiliary losses
- State representation
- ***Exploration***
- Unsupervised skill discovery

Representation Learning in Reinforcement Learning

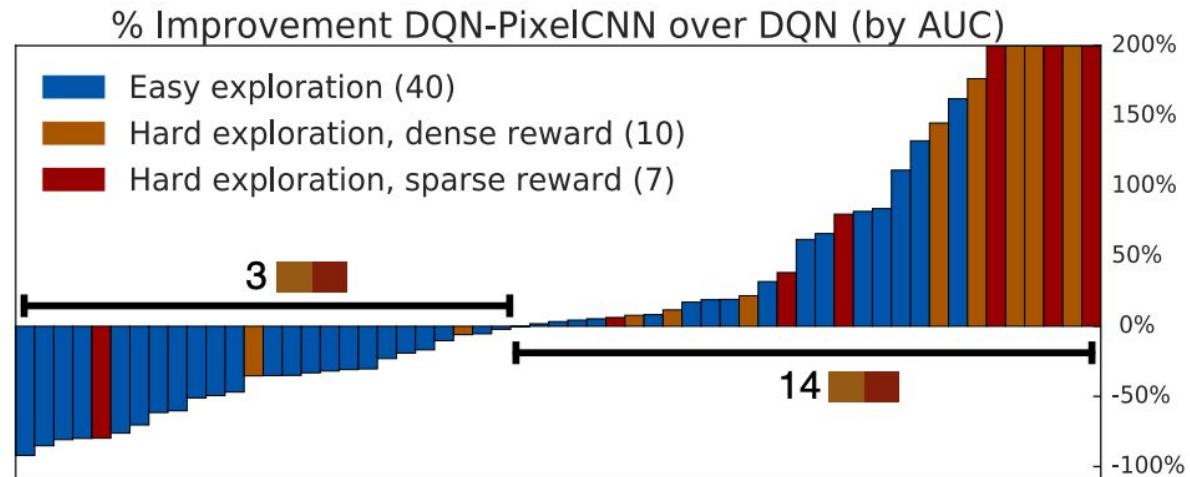
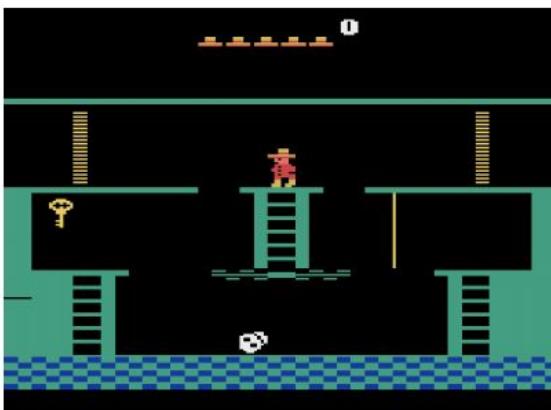
- Auxiliary losses
- State representation
- ***Exploration***
 - ***Exploration bonuses***
- Unsupervised skill discovery

Exploration Bonuses

- Tabular RL exploration bonuses:
 - Give exploration bonus when visiting state that hasn't been visited very often before
- But: impractical for large / continuous state spaces
 - Similarity of states matters, not just individual states

Exploration Bonuses (1)

- Count-Based Exploration with Neural Density Models, Georg Ostrovski, Marc G. Bellemare, Aaron van den Oord, Remi Munos, <https://arxiv.org/abs/1703.01310>
 - PixelCNN for density estimation



Exploration Bonuses (2)

- #Exploration: A Study of Count-Based Exploration for Deep Reinforcement Learning,
Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, Xi Chen, Yan Duan, John Schulman, Filip De Turck, Pieter Abbeel (<https://arxiv.org/abs/1611.04717>)
 - VAE embedding / hash → classical counts in the discrete hash space

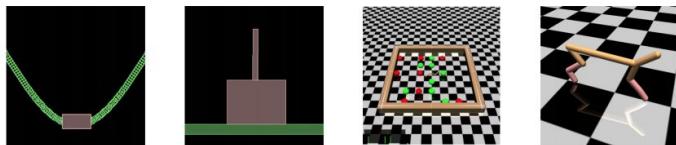


Figure 2: Illustrations of the rllab tasks used in the continuous control experiments, namely MountainCar, CartPoleSwingup, SimmerGather, and HalfCheetah; taken from [8].

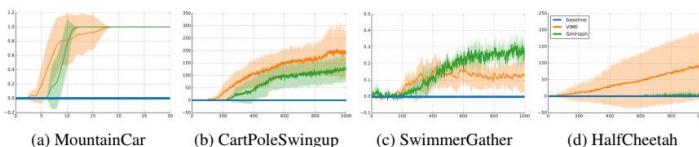
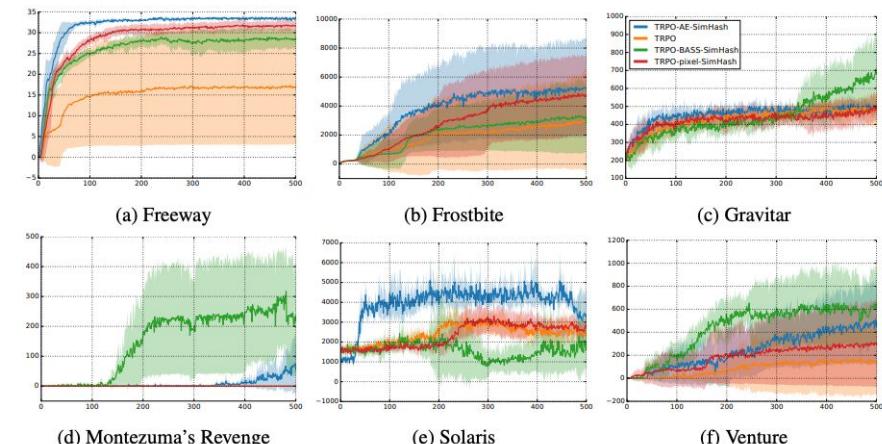


Figure 3: Mean average return of different algorithms on rllab tasks with sparse rewards. The solid line represents the mean average return, while the shaded area represents one standard deviation, over 5 seeds for the baseline and SimHash (the baseline curves happen to overlap with the axis).



Exploration Bonuses (3)

- VIME: Variational Information Maximizing Exploration, Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, Pieter Abbeel (<https://arxiv.org/abs/1605.09674>)
 - Bayesian neural net dynamics model
 - Exploration bonus based on KL divergence between pre and post update model distribution

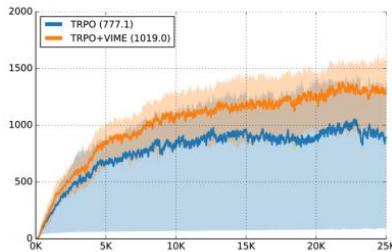


Figure 3: Performance of TRPO with and without VIME on the high-dimensional Walker2D locomotion task.

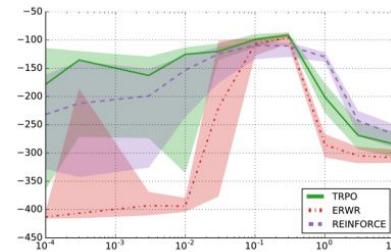


Figure 4: VIME: performance over the first few iterations for TRPO, REINFORCE, and ERWR i.f.o. η on MountainCar.

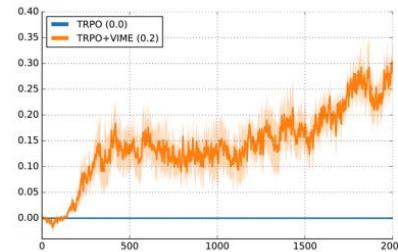


Figure 5: Performance of TRPO with and without VIME on the challenging hierarchical task SwimmerGather.

Exploration Bonuses (4)

- Curiosity:
 - Curiosity-driven Exploration by Self-supervised Prediction
Efros, Trevor Darrell (<https://arxiv.org/abs/1312.0037>)
 - Large-Scale Study of Curiosity-Driven Exploration
Storkey, Trevor Darrell, Alexei A. Efros, and Michael J. Black (CVPR 2014)
 -

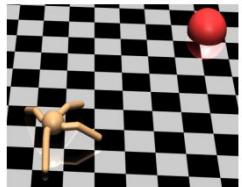


Representation Learning in Reinforcement Learning

- Auxiliary losses
- State representation
- ***Exploration***
 - ***Exploration bonuses***
 - ***Exploration through goal generation***
- Unsupervised skill discovery

GoalGAN

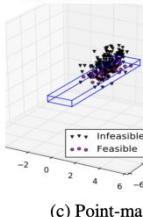
- Automatic Goal Generation for Reinforcement Learning Agents, Carlos Florensa, David Held, Xinyang Geng, Pieter Abbeel (<https://arxiv.org/abs/1705.06366>)
 - Train goal-conditioned policy
 - Achieve curriculum by setting goals that become gradually more difficult
 - How? GAN is continually retrained to generate goals of right level of difficulty based on recent performance on previous goals



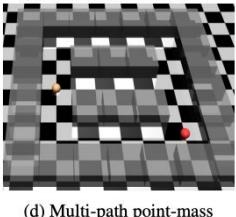
(a) Free Ant Locomotion



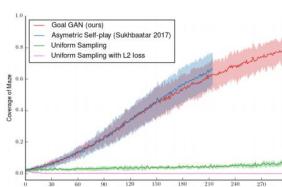
(b) Maze Ant Locomotion



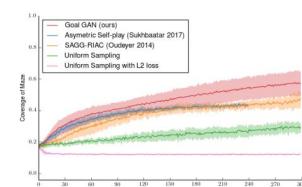
(c) Point-mass 3D



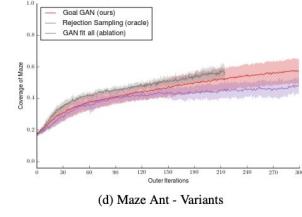
(d) Multi-path point-mass



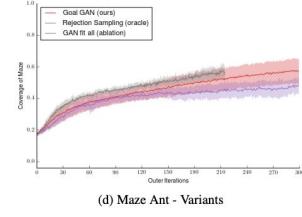
(a) Free Ant - Baselines



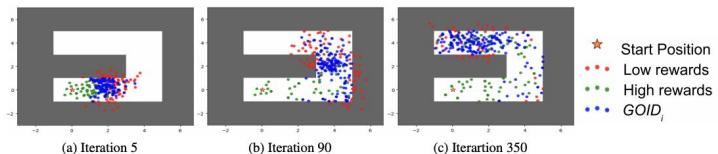
(b) Maze Ant - Baselines



(c) Free Ant - Variants



(d) Maze Ant - Variants



(a) Iteration 5

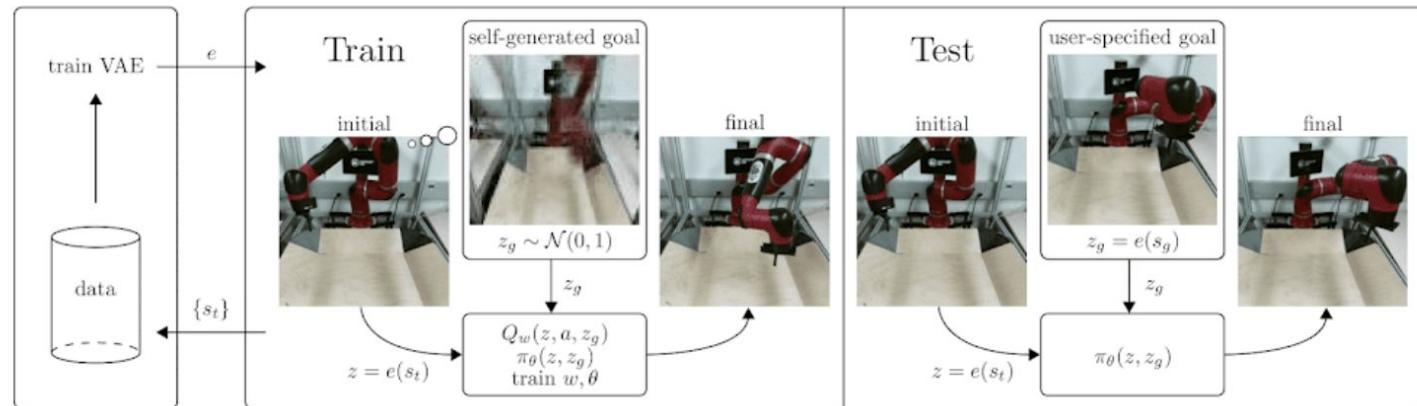
(b) Iteration 90

(c) Iteration 350

★ Start Position
● Low rewards
● High rewards
● GOID_i

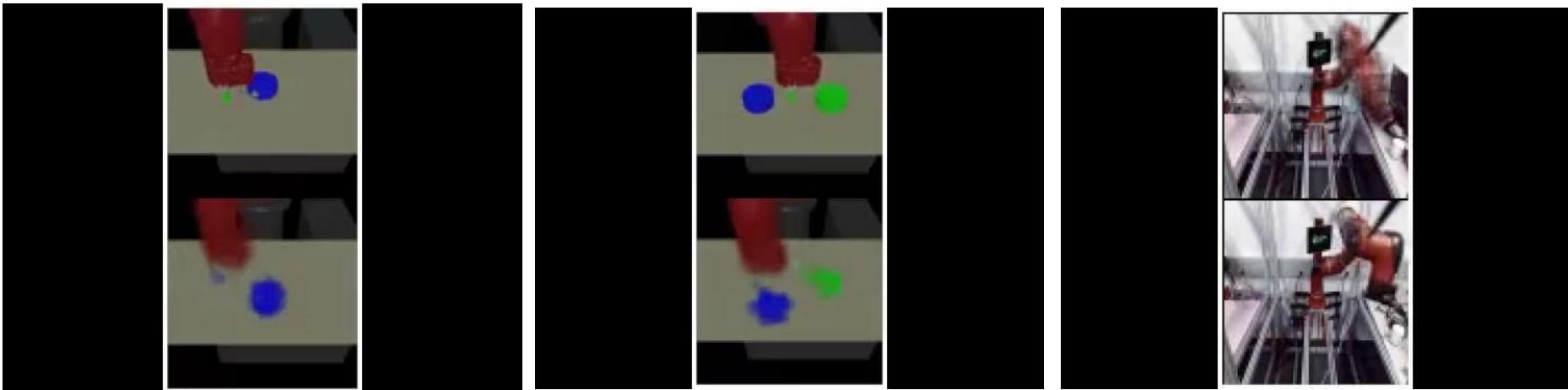
RIG

- Visual Reinforcement Learning with Imagined Goals, Ashvin Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, Sergey Levine (<https://arxiv.org/abs/1807.04742>)
- Learn probabilistic latent variable model (Variational Auto-Encoder)
- Use latent representation for state and goal representation.
- Sample goals for hindsight experience relabeling
- Sample goals for exploration
- Use latent distance for reward



RIG

- Visual Reinforcement Learning with Imagined Goals (RIG)
Ashvin Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, Sergey Levine
(<https://arxiv.org/abs/1807.04742>)

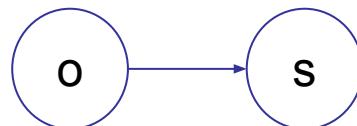


Representation Learning in Reinforcement Learning

- Auxiliary losses
- State representation
- ***Exploration***
 - ***Exploration bonuses***
 - ***Exploration through goal generation***
 - ***Exploration through skill transfer***
- Unsupervised skill discovery

Traditional view on Representation Learning in RL

Typical focus on learning good representations of observations allowing for better learning, modeling, etc



- Embed2Control [Watter et al]
- Deep Predictive Policy Training [Ghadirzadeh et al]
- DARLA [Higgins et al]
- Robotic Priors [Jonschkowski and Brock]
- UPN [Aravind, Allan, Chelsea, etc]
- Causal InfoGAN [Kurutach, Tamar et al]

Especially useful when trying to learn from pixel inputs and complex observations

Alternative View on Representation Learning

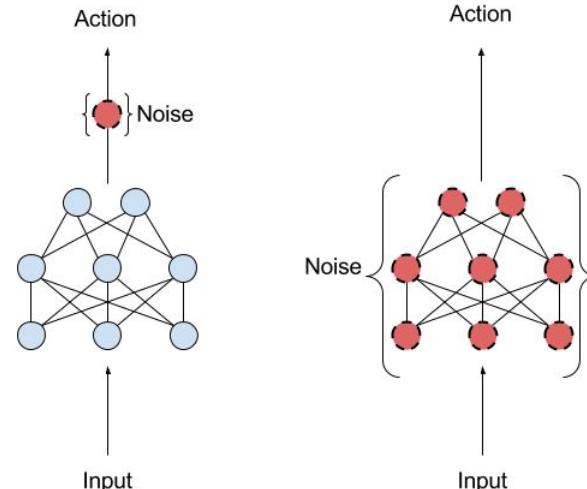
Can we learn representations of things other than the mapping from observations to low level hidden state

- Trajectories
- Behaviors/tasks?
- ...

Question: How do we supervise representation learning for these things?

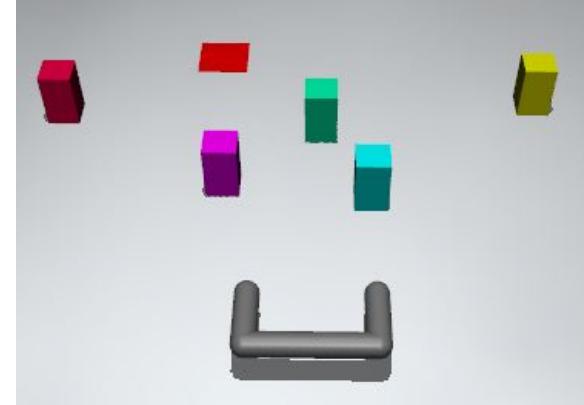
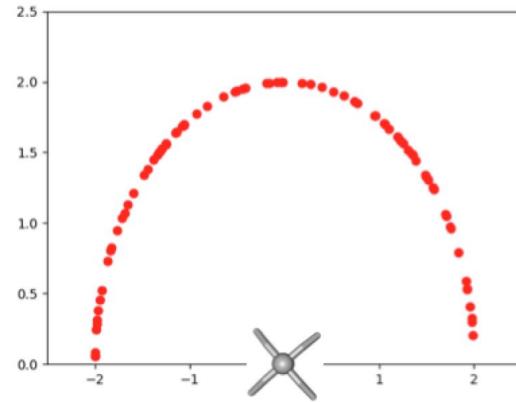
Representation learning for Exploration

Exploration typically with stochasticity in the space of actions or in parameter space



Can we better inform
or experience?

Why is this important?

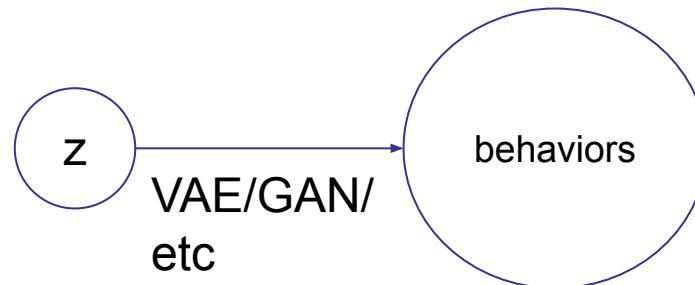


More efficient exploration for test tasks based on prior knowledge

Latent Exploration Spaces

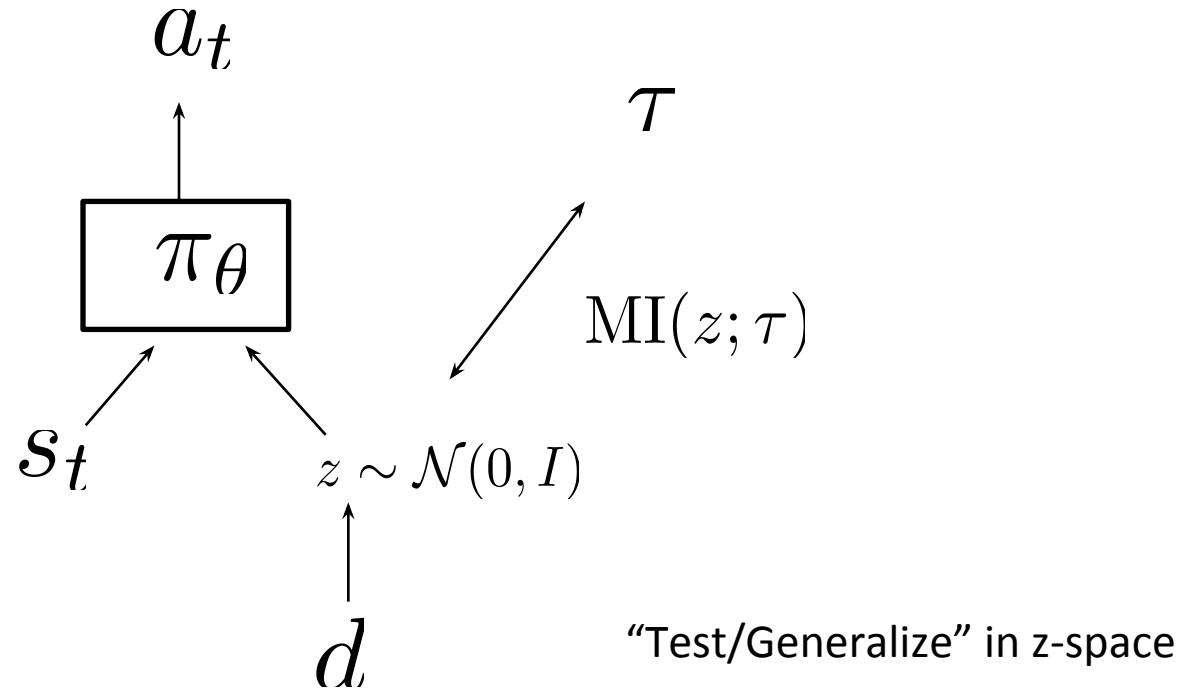
Can we learn better spaces to inject noise instead of simply into action/parameter space?

- What if we learned a latent space of behaviors which spans the input task distribution? i.e can we learn a generative model of behaviors and use it to generate novel exploratory behavior



Approach 1: Hausman et al 2018

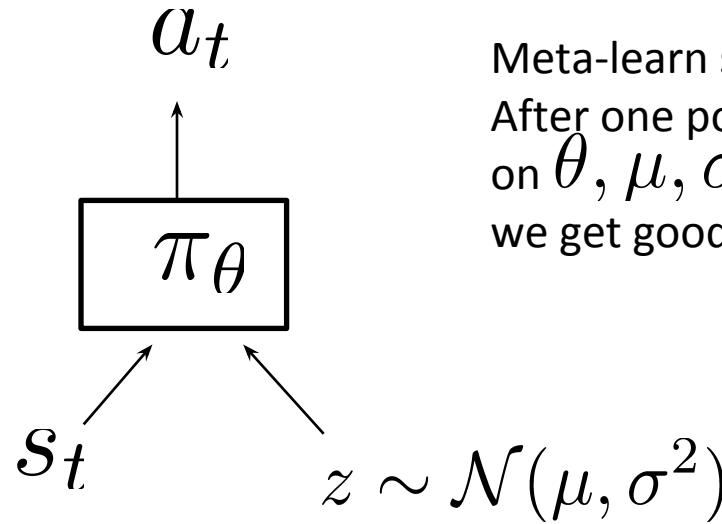
Train against many tasks
(each task indexed by d):



Learning an Embedding Space for Transferrable Skill, Hausman, Springenberg, Wang, Heess, Riedmiller, ICLR 2018

Approach 2: Gupta et al 2018

Train against many tasks



Meta-learn such that:
After one policy gradient update
on θ, μ, σ
we get good behavior

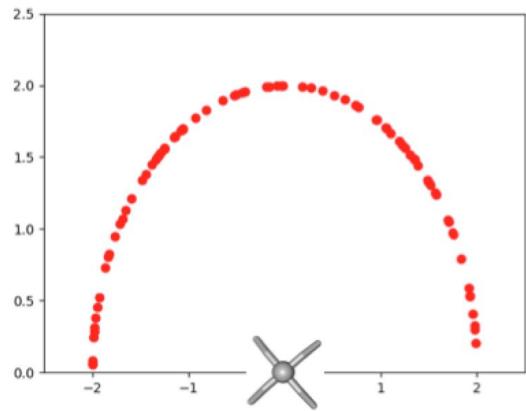
“Test/Generalize” in z-space

Model-Agnostic Exploration with Structured Noise (MAESN), Gupta, Mendonca, Liu, Abbeel, Levine, 2018

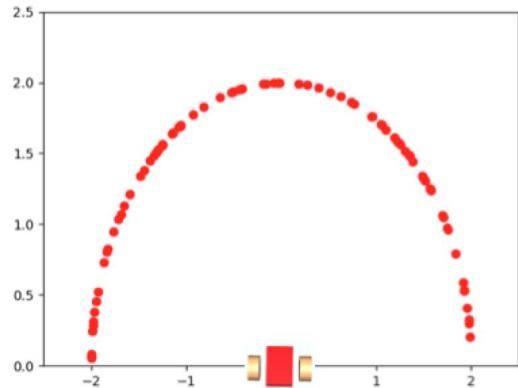
Experiments

3 task families:

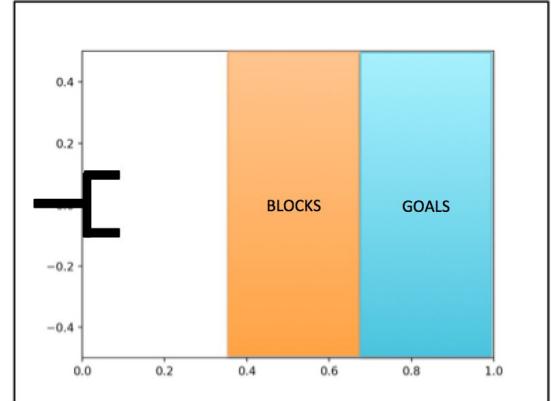
Ant



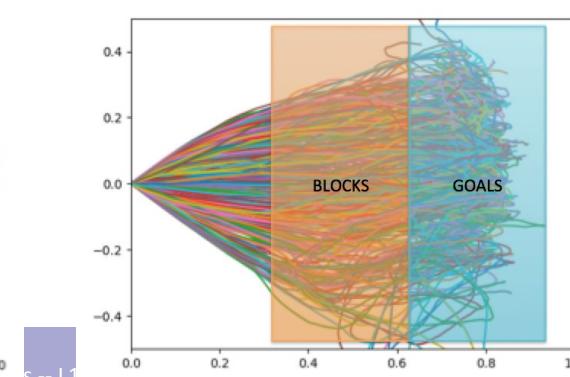
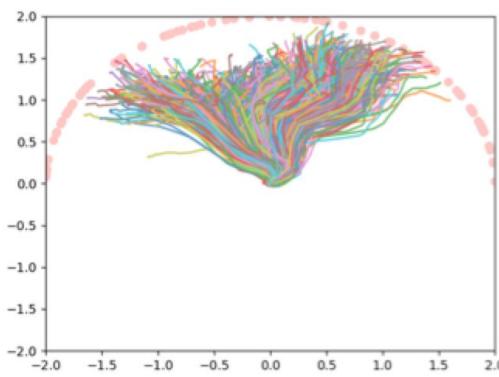
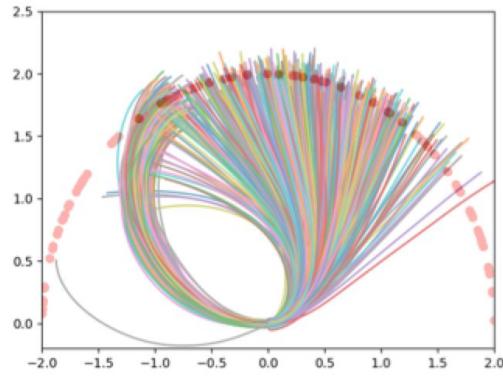
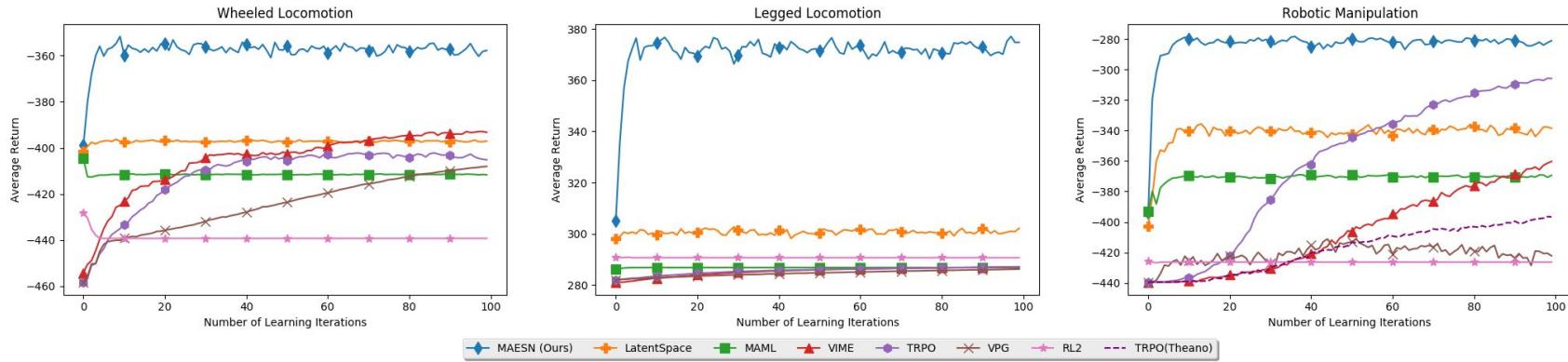
Wheeled Locomotion



Block Manipulation



Experiments



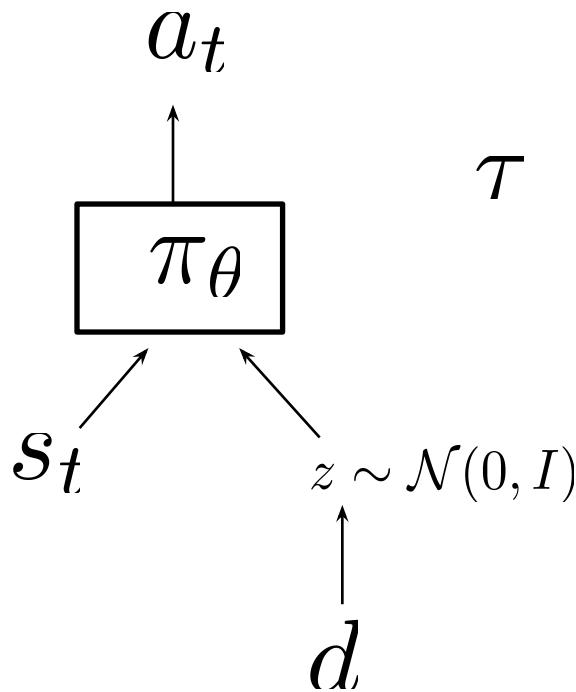
Representation Learning in Reinforcement Learning

- Auxiliary losses
- State representation
- ***Exploration***
 - ***Exploration bonuses***
 - ***Exploration through goal generation***
 - ***Exploration through skill transfer***
- Unsupervised skill discovery

Representation Learning in Reinforcement Learning

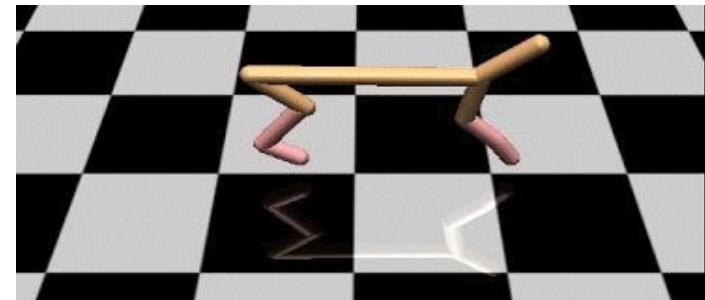
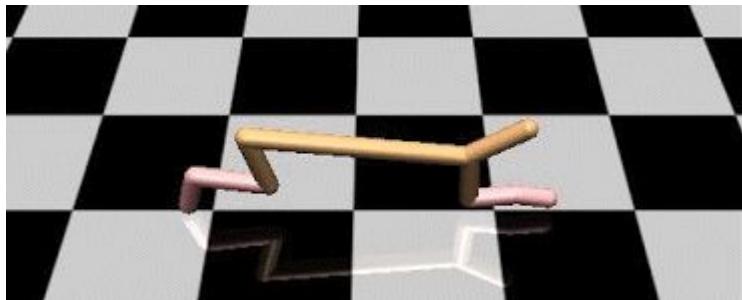
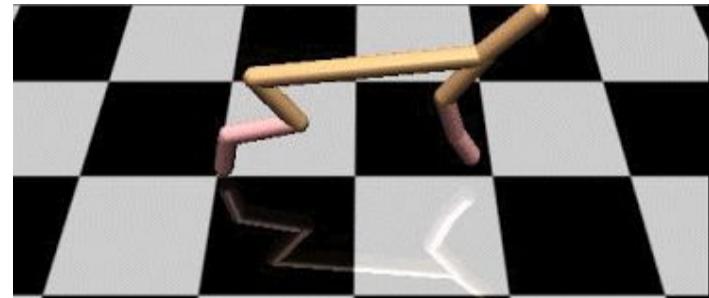
- Auxiliary losses
- State representation
- Exploration
- ***Unsupervised skill discovery***

Unsupervised Skill Discovery

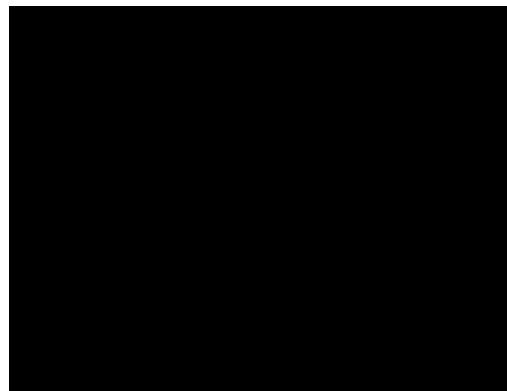
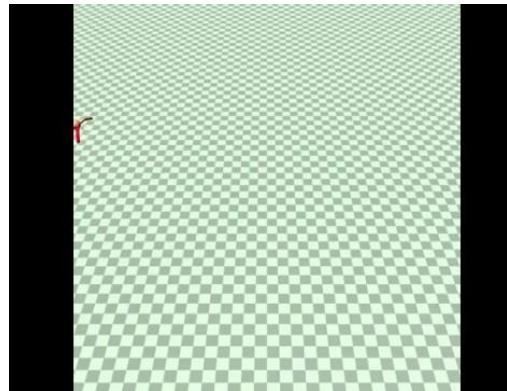
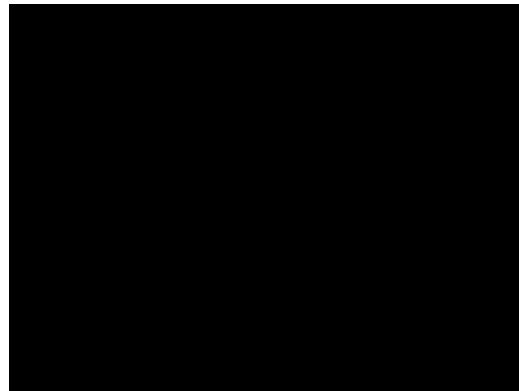


- SSN4HRL (Florensa, Duan, Abbeel, 2016)
 $\text{MI}(d; \tau)$
- Variational Intrinsic Control (Gregor et al 2016)
 $\text{MI}(z; s_H)$
- Diversity is all you need (Eysenbach et al 2018)
 $\sum_t \text{MI}(z; s_t)$
- VALOR (Achiam et al 2018)
 $\text{MI}(z; \tau)$

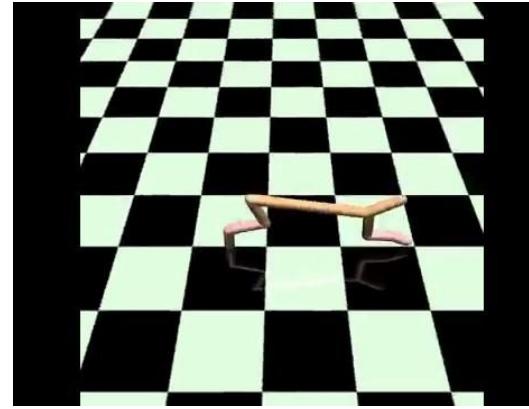
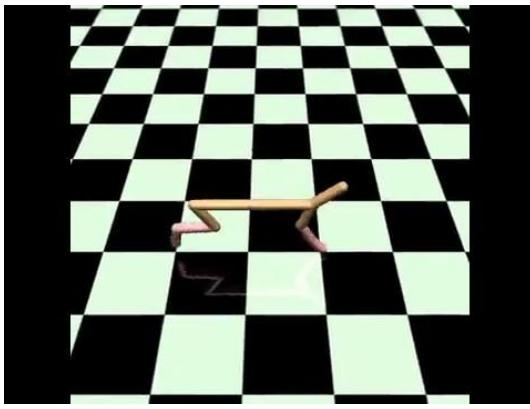
Here are cool videos from Eysenbach et al



Here are cool videos from Achiam et al



Here are cool videos from Achiam et al



Humanoid



High-dimensional problems like Humanoid:

- All learned behaviors were on the ground, none of them get humanoid to get up and run
- Maybe something more is needed beyond the MI objective...

Summary: Representation Learning in Reinforcement Learning

- Auxiliary losses
- State representation
- Exploration
- Unsupervised skill discovery

Some recommended readings we didn't cover

- **Temporal Difference Variational Auto-Encoder**

Karol Gregor, George Papamakarios, Frederic Besse, Lars Buesing, Theophane Weber

<https://arxiv.org/abs/1806.03107>

- **Model-Based Reinforcement Learning for Atari**

Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Ryan Sepassi, George Tucker, Henryk Michalewski

<https://arxiv.org/abs/1903.00374>

- **Time-Contrastive Networks: Self-Supervised Learning from Video**

Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine

<https://arxiv.org/abs/1704.06888>

- **Learning Actionable Representations with Goal-Conditioned Policies**

Dibya Ghosh, Abhishek Gupta, Sergey Levine

<https://arxiv.org/abs/1811.07819>

- **Deep Object-Centric Representations for Generalizable Robot Learning**

Coline Devin, Pieter Abbeel, Trevor Darrell, Sergey Levine

<https://arxiv.org/abs/1708.04225>

- **Decoupling Dynamics and Reward for Transfer Learning**

Amy Zhang, Harsh Satija, Joelle Pineau

<https://arxiv.org/abs/1804.10689>