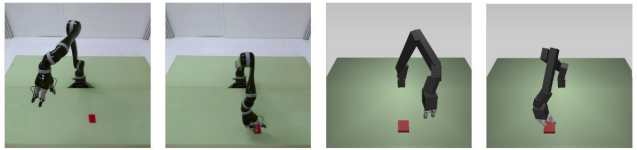
Recent Advances and Frontiers in Deep RL

Volodymyr Mnih

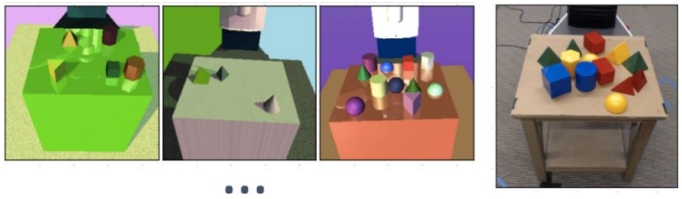
Deep RL Bootcamp, Berkeley

Deep RL Frontiers

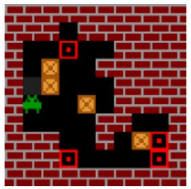
- What are the exciting areas in Deep RL research?
  - HRL?
  - Deep RL for robotics?
  - Model-based RL?
- Hard to keep up in the age of arXiv.



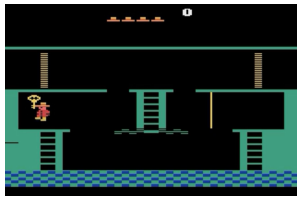
"Sim to Real Robot Learning from Pixels with Progressive Nets" Rusu et al. (2016)



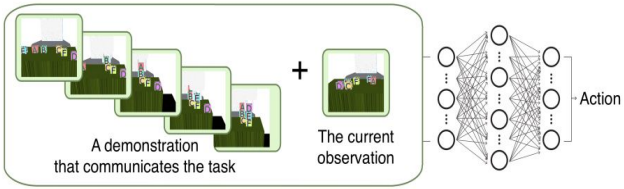
"Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World", Tobin et al. (2017)



"Imagination-Augmented Agents for Deep RL", Weber, Racanière, Reichert et al. (2017)



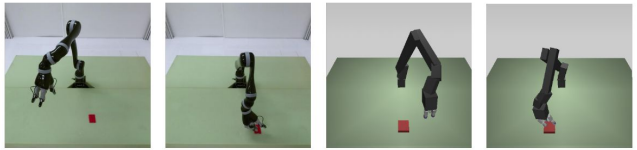
Feudal Networks, Vezhnevets et al. (2017)



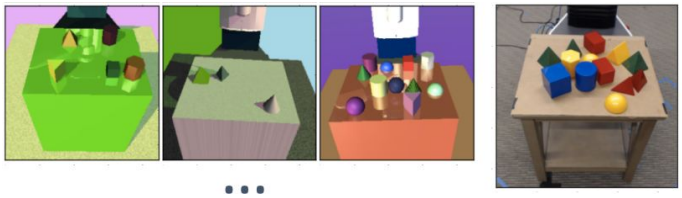
"One-Shot Imitation Learning", Duan et al. (2017)

Deep RL Frontiers

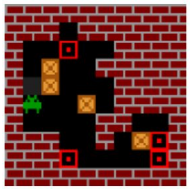
- What are the exciting areas in Deep RL research?
  - HRL?
  - Deep RL for robotics?
  - Model-based RL?
- **Everything in deep RL is still exciting.**



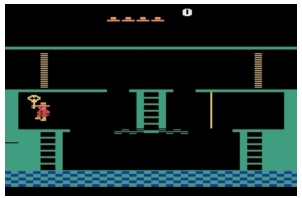
"Sim to Real Robot Learning from Pixels with Progressive Nets" Rusu et al. (2016)



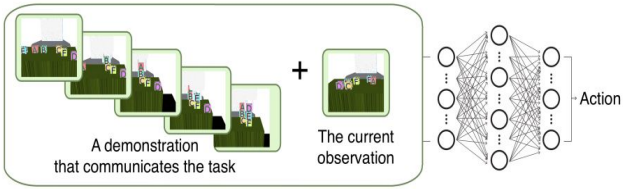
"Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World", Tobin et al. (2017)



"Imagination-Augmented Agents for Deep RL", Weber, Racanière, Reichert et al. (2017)



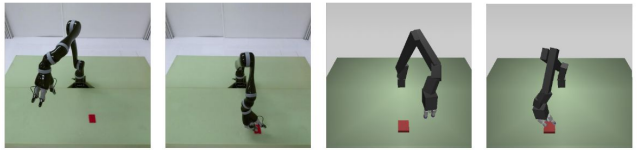
Feudal Networks, Vezhnevets et al. (2017)



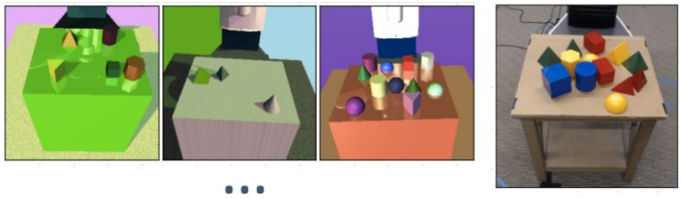
"One-Shot Imitation Learning", Duan et al. (2017)

Deep RL Frontiers

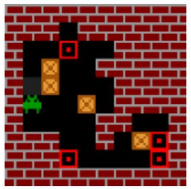
- What are the exciting areas in Deep RL research?
  - HRL?
  - Deep RL for robotics?
  - Model-based RL?
- **Everything in deep RL is still exciting.**
- **Nothing is solved!**
  - How to learn a value function. How to learn a policy. How to train a deep network. Which nonlinearity to use ...



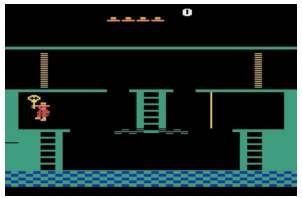
"Sim to Real Robot Learning from Pixels with Progressive Nets" Rusu et al. (2016)



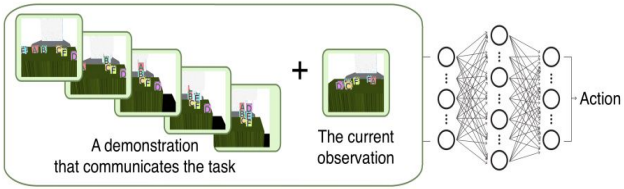
"Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World", Tobin et al. (2017)



"Imagination-Augmented Agents for Deep RL", Weber, Racanière, Reichert et al. (2017)



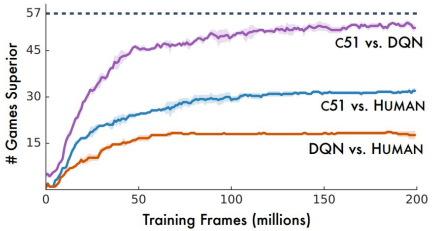
Feudal Networks, Vezhnevets et al. (2017)

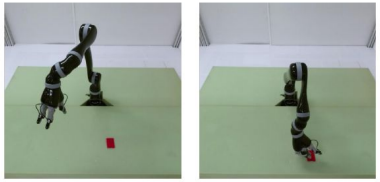


"One-Shot Imitation Learning", Duan et al. (2017)

Two Problems

Data/Sample Efficiency





Exploration

Distributional RL

- The value function gives the **expected** future discounted return.
- This ignores variance and multi-modality.
- Bellman equation:

$$Q(s,a) = \mathbb{E}R(s,a,s') + \gamma \mathbb{E}Q(s',a')$$

- Bellemare and Dabney argue for modelling the full distribution of the return.
- Distributional Bellman equation:

$$Z(s,a) \overset{D}{=} R(s,a,S') + \gamma Z(S',A')$$

"A Distributional Perspective on Reinforcement Learning", Bellemare*, Dabney*, Munos (2017)

Distributional RL - The C51 Algorithm

- C51 outputs a 51-way softmax for each Q-value.
- Bin the return into 51 equally-sized bins between -10 and 10.
- The Q-value update becomes categorical (instead of the usual regression).

Algorithm 1 Categorical Algorithm

input A transition $x_t, a_t, r_t, x_{t+1}, \gamma_t \in [0,1]$

$Q(x_{t+1}, a) := \sum_i z_i p_i(x_{t+1}, a)$

$a^* \leftarrow \arg\max_a Q(x_{t+1}, a)$

$m_i = 0, \quad i \in 0, \ldots, N-1$

for $j \in 0, \ldots, N-1$ **do**

Compute the projection of $\hat{\mathcal{T}} z_j$ onto the support $\{z_i\}$

$\hat{\mathcal{T}} z_j \leftarrow [r_t + \gamma_t z_j]_{V_{\text{MIN}}}^{V_{\text{MAX}}}$

$b_j \leftarrow (\hat{\mathcal{T}} z_j - V_{\text{MIN}})/\Delta z \quad \# b_j \in [0, N-1]$

$l \leftarrow \lfloor b_j \rfloor, u \leftarrow \lceil b_j \rceil$

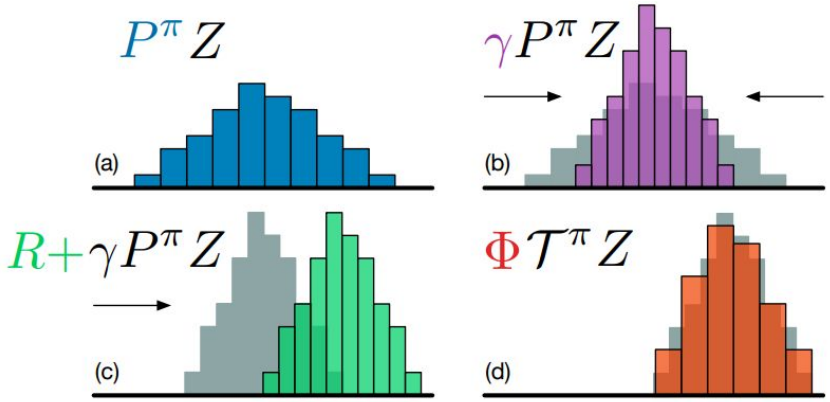
Distribute probability of $\hat{\mathcal{T}} z_j$

$m_l \leftarrow m_l + p_j(x_{t+1}, a^*)(u - b_j)$

$m_u \leftarrow m_u + p_j(x_{t+1}, a^*)(b_j - l)$

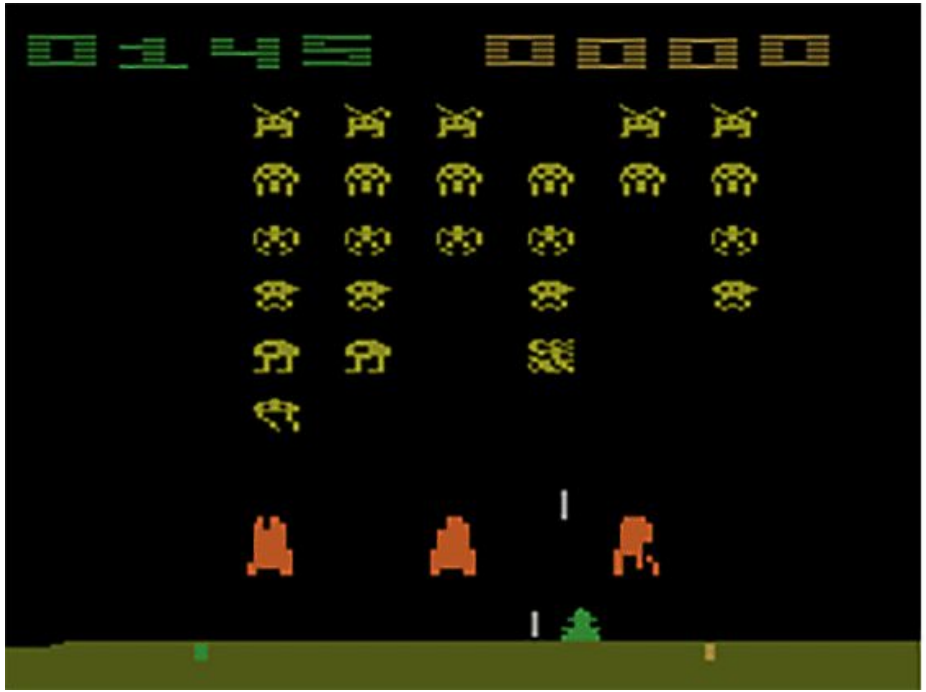
end for

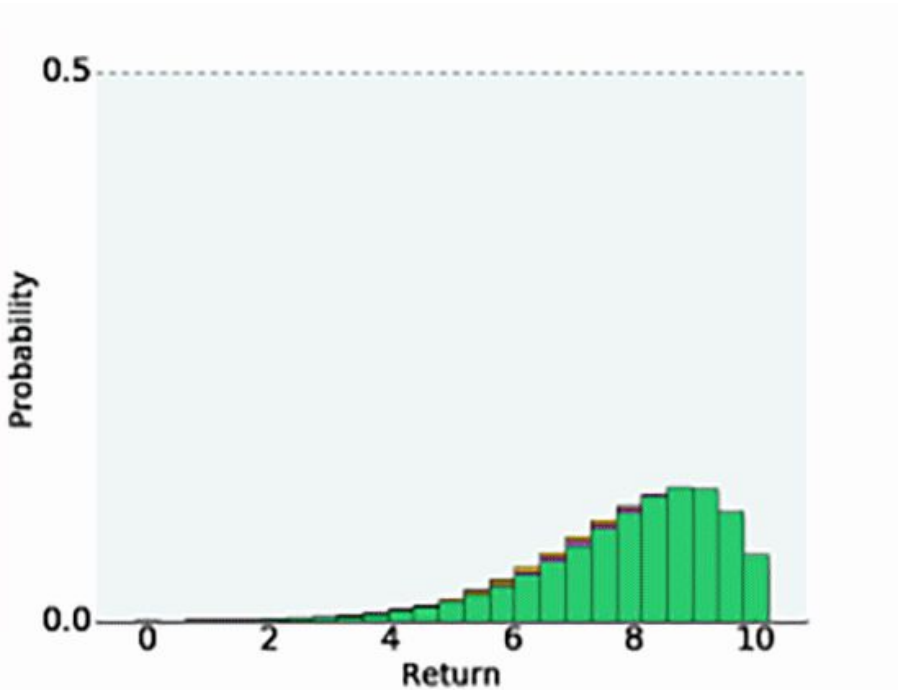
output $-\sum_i m_i \log p_i(x_t, a_t)$ # Cross-entropy loss



"A Distributional Perspective on Reinforcement Learning", Bellemare*, Dabney*, Munos (2017)

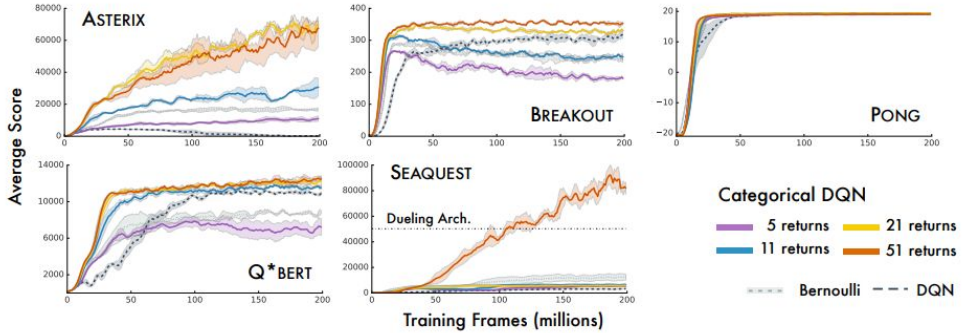
C51 Visualization



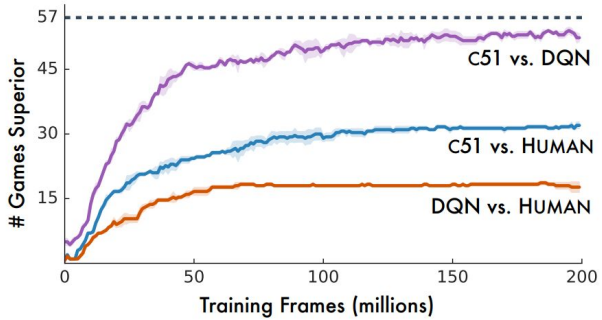


"A Distributional Perspective on Reinforcement Learning", Bellemare*, Dabney*, Munos (2017)

C51 Atari Results



	Mean	Median	> H.B.	> DQN
DQN	228%	79%	24	0
DDQN	307%	118%	33	43
DUEL.	373%	151%	37	50
PRIOR.	434%	124%	39	48
PR. DUEL.	592%	172%	39	44
C51	**701%**	**178%**	**40**	**50**
UNREAL†	880%	250%	-	-

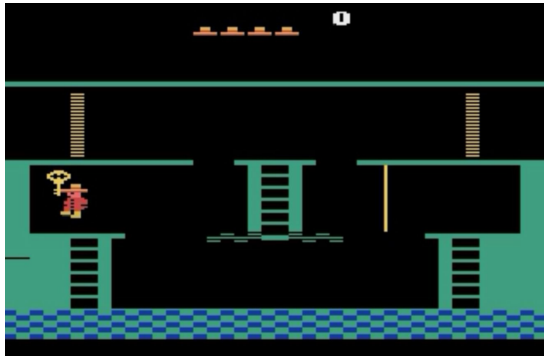


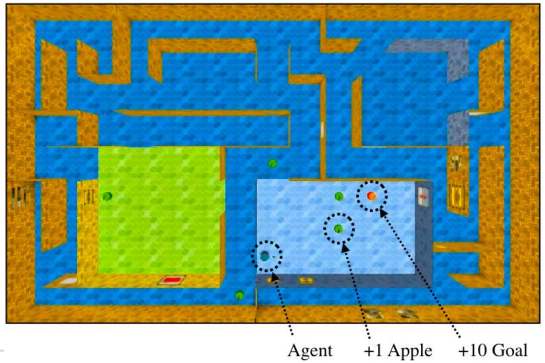
Why does C51 work so well?

"A Distributional Perspective on Reinforcement Learning", Bellemare*, Dabney*, Munos (2017)

Unsupervised Reinforcement Learning

- The best deep RL methods are still very data hungry. Especially with **sparse rewards**.
- Obvious solution - Learn about the environment.
- We can augment an RL agent with **auxiliary prediction and control tasks** to improve data efficiency.
- The UNREAL agent - UNsupervised REinforcement and Auxiliary Learning.
  - “Reinforcement Learning with Unsupervised Auxiliary Tasks”, Jaderberg et al. (2017)

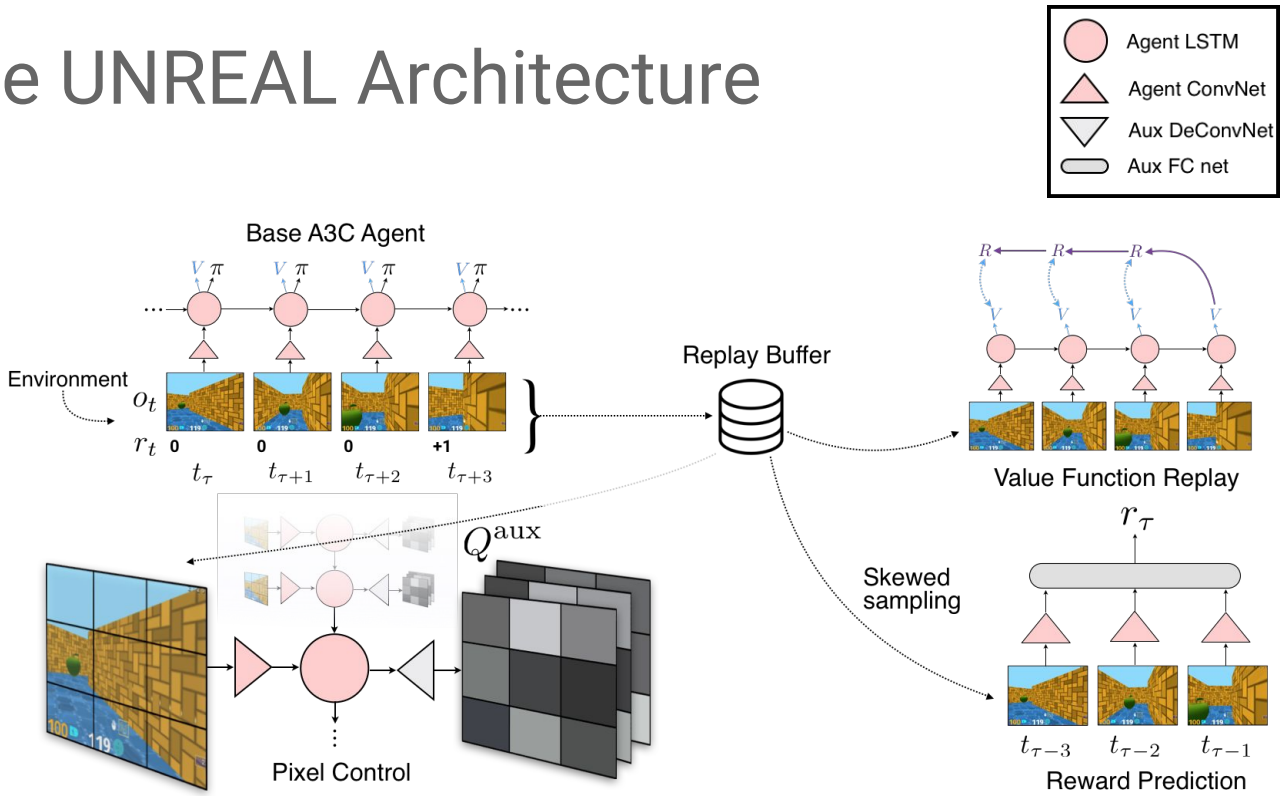




“Reinforcement Learning with Unsupervised Auxiliary Tasks”, Jaderberg, Mnih, Czarnecki et al. (2017)

The UNREAL Architecture

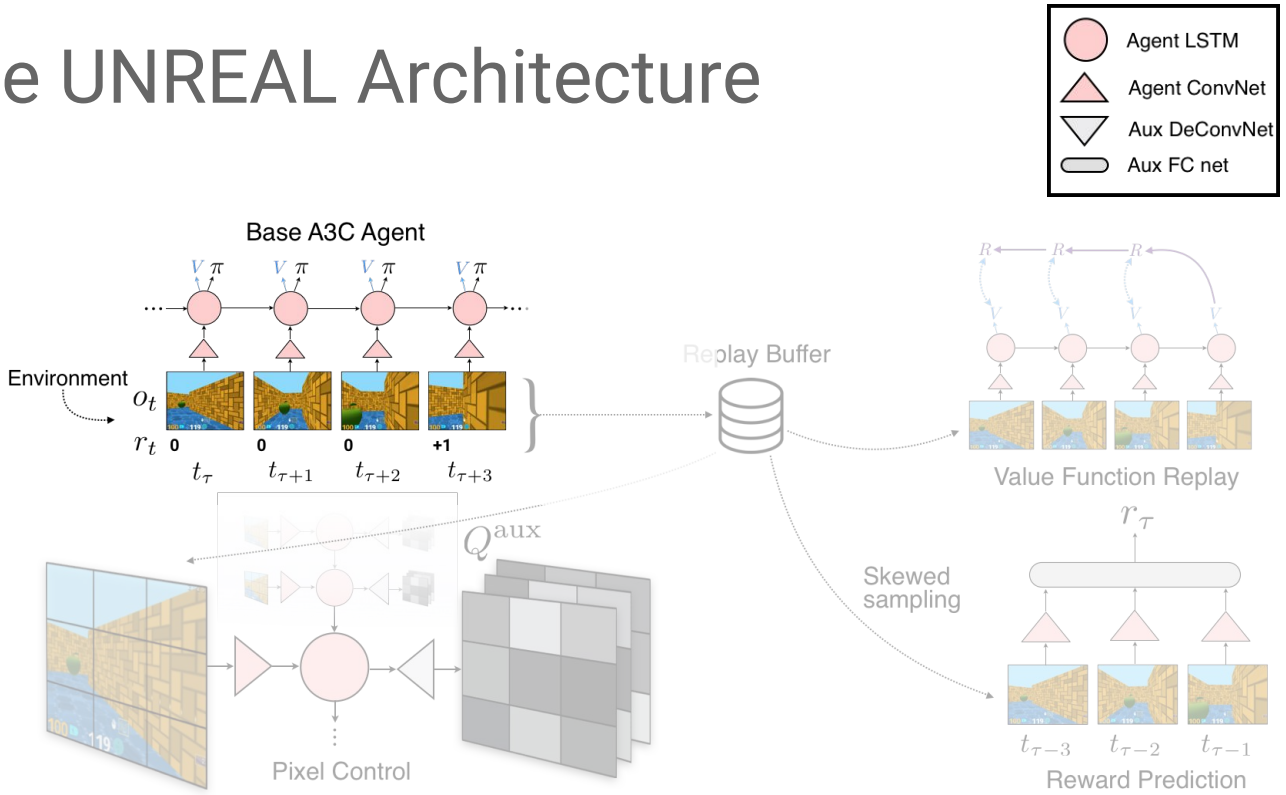
- UNREAL augments an LSTM A3C agent with 3 auxiliary tasks.
- Can be used on top of DQN, DDPG, TRPO or other agents.



"Reinforcement Learning with Unsupervised Auxiliary Tasks", Jaderberg, Mnih, Czarnecki et al. (2017)

The UNREAL Architecture

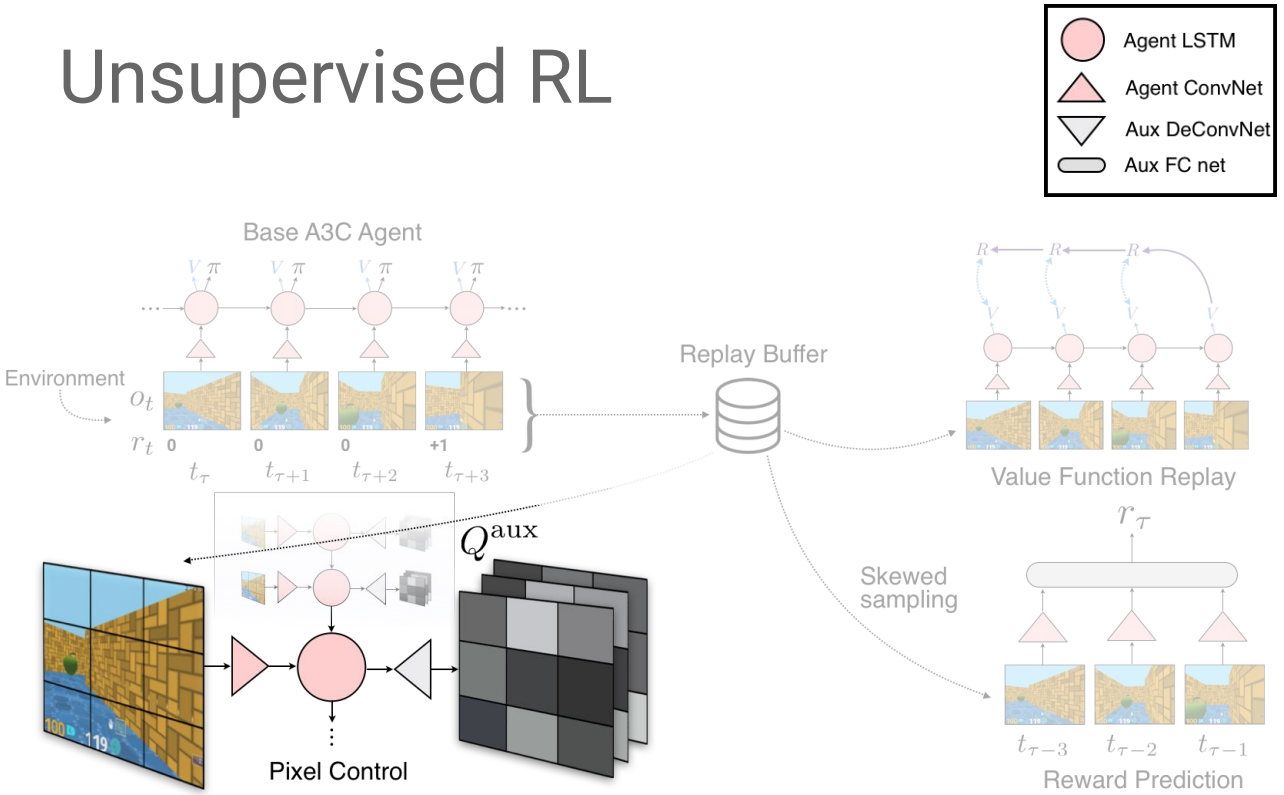
- Base A3C LSTM agent learns from the environment's scalar reward signal.
- UNREAL acts using the base A3C agent's policy.



"Reinforcement Learning with Unsupervised Auxiliary Tasks", Jaderberg, Mnih, Czarnecki et al. (2017)

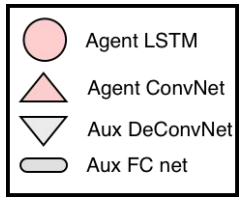
Unsupervised RL

- 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.



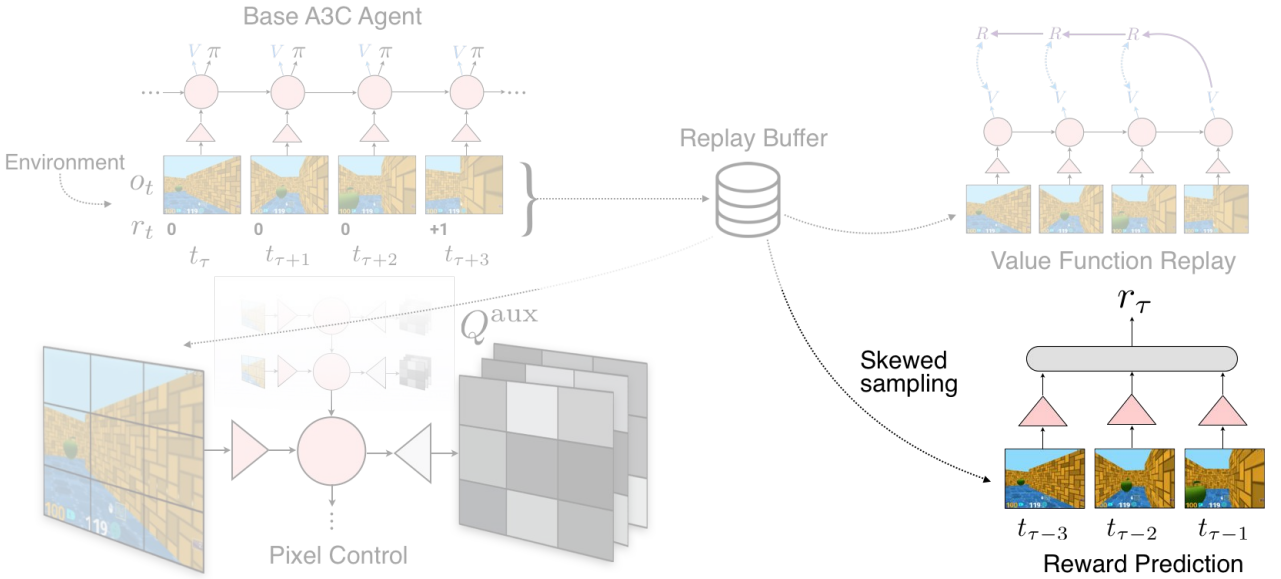
"Reinforcement Learning with Unsupervised Auxiliary Tasks", Jaderberg, Mnih, Czarnecki et al. (2017)

The UNREAL Architecture



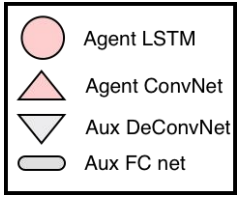
Focusing on rewards:

- 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.



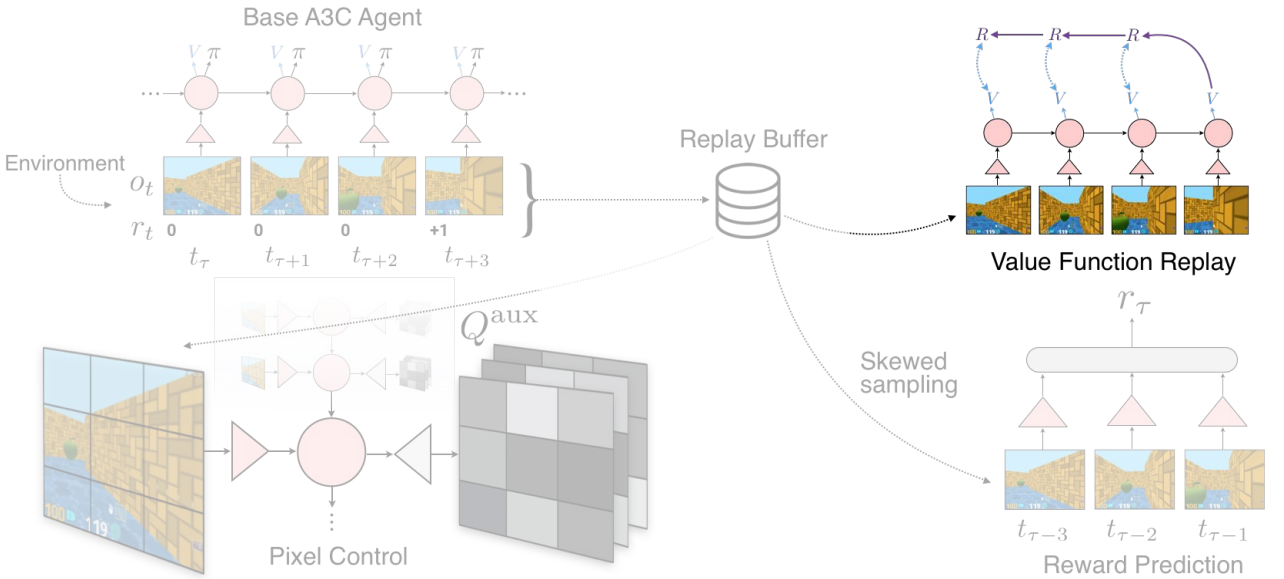
"Reinforcement Learning with Unsupervised Auxiliary Tasks", Jaderberg, Mnih, Czarnecki et al. (2017)

The UNREAL Architecture



Focusing on rewards:

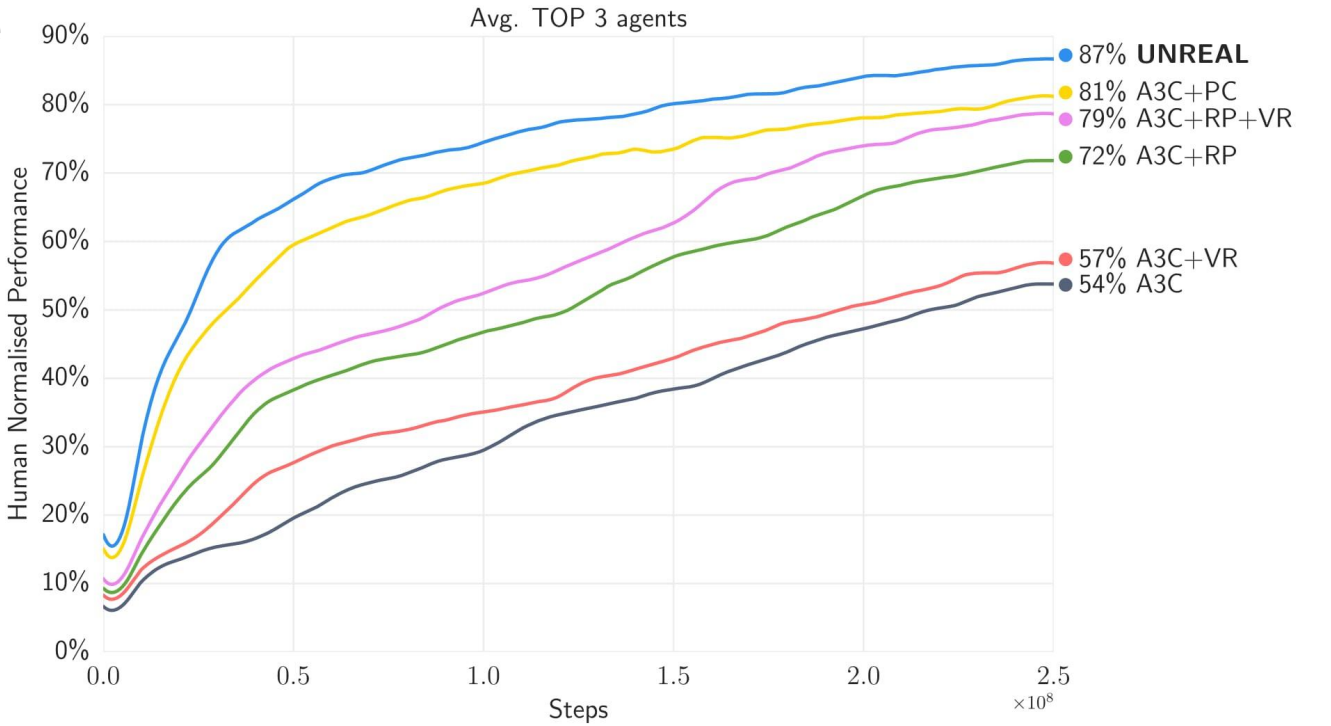
- Value function replay.
- Faster learning of the value function.



"Reinforcement Learning with Unsupervised Auxiliary Tasks", Jaderberg, Mnih, Czarnecki et al. (2017)

DeepMind Lab Results

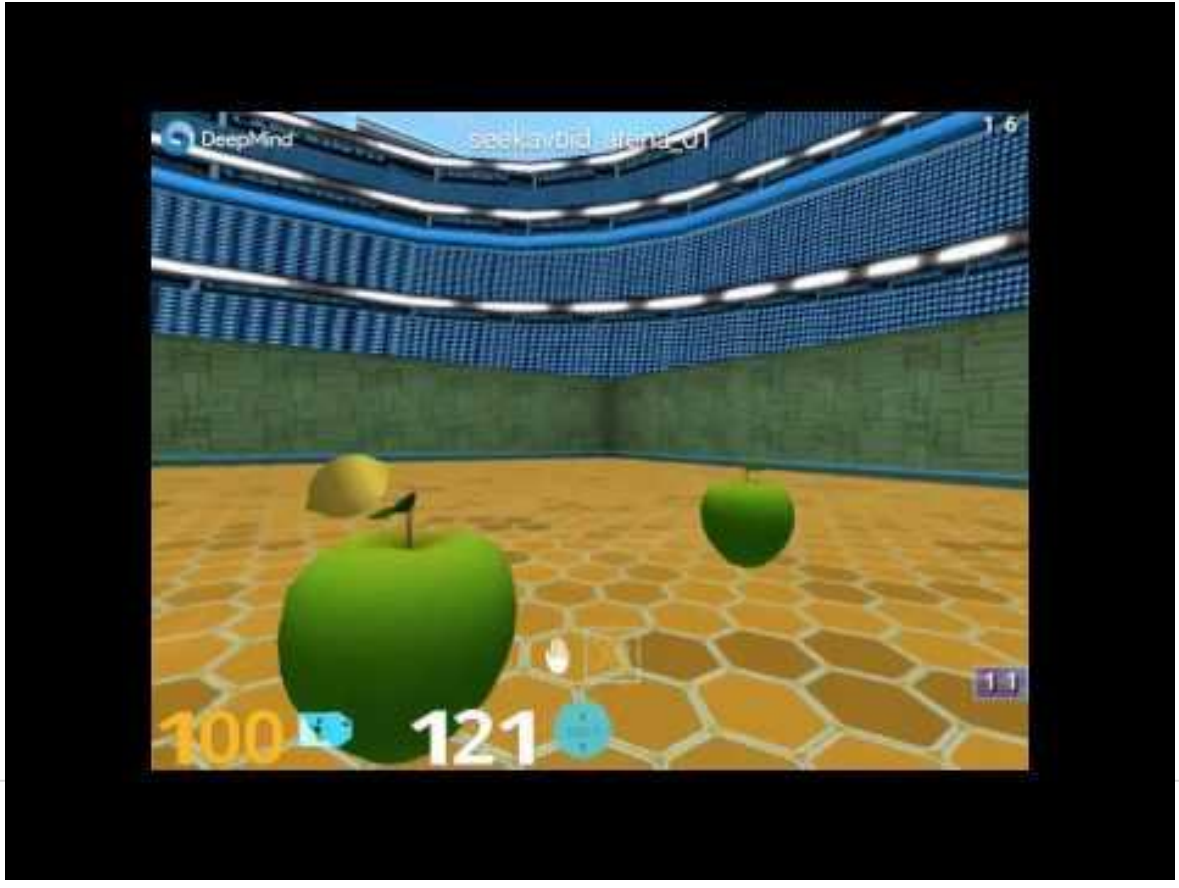
- Average human-normalized performance on 13 3D environments from DeepMind Lab.
- Tasks include random maze navigation and laser tag.
- Roughly a 10x improvement in data efficiency over A3C.
- 60% improvement in final performance.



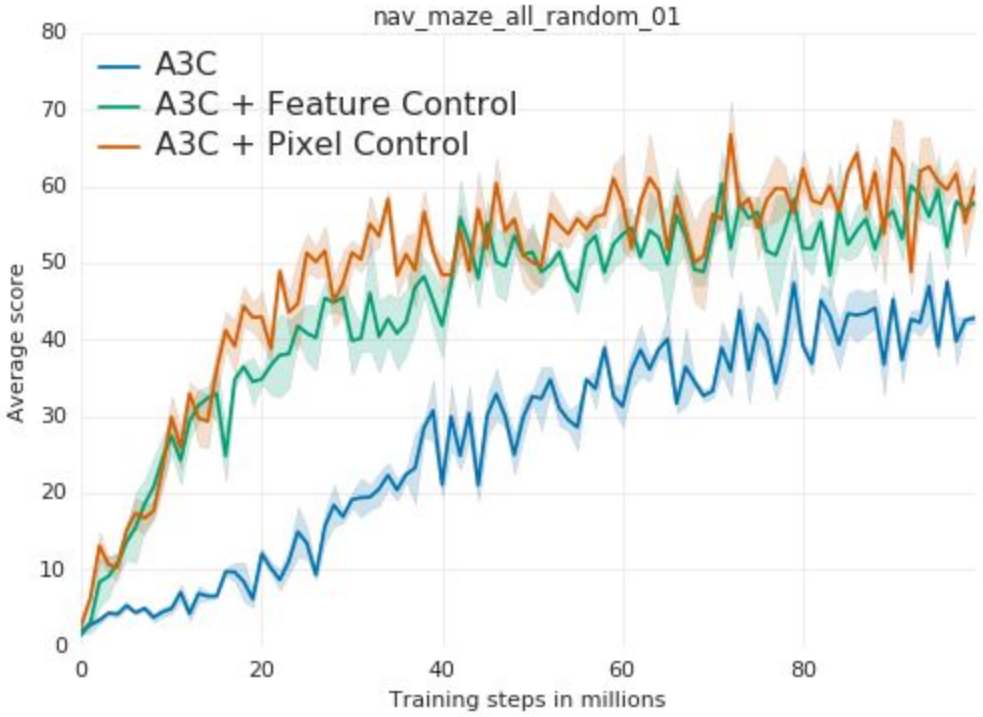
DeepMind

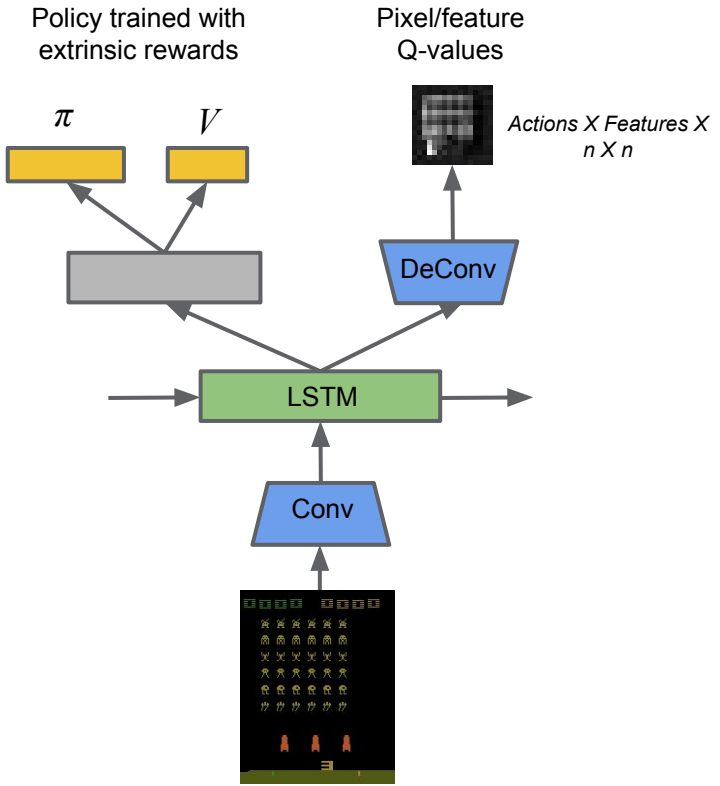
"Reinforcement Learning with Unsupervised Auxiliary Tasks", Jaderberg, Mnih, Czarnecki et al. (2017)

UNREAL playing DeepMind Lab



Feature Control





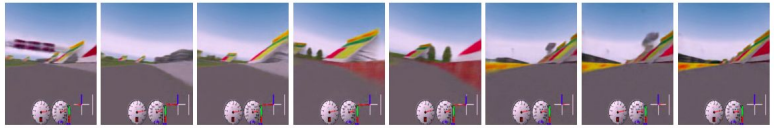
DeepMind

"Reinforcement Learning with Unsupervised Auxiliary Tasks", Jaderberg, Mnih, Czarnecki et al. (2017)

Model-Based RL

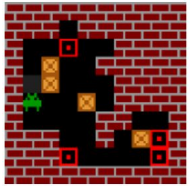
- There is huge promise in model-based RL.
- A perfect model allows evaluating any policy or learning an optimal policy.
- But learning a model is hard - model-based RL is just beginning to work in rich environments.

Explicit Environment Models



"Action-Conditional Video Prediction using Deep Networks in Atari Games", Oh et al., (2016)

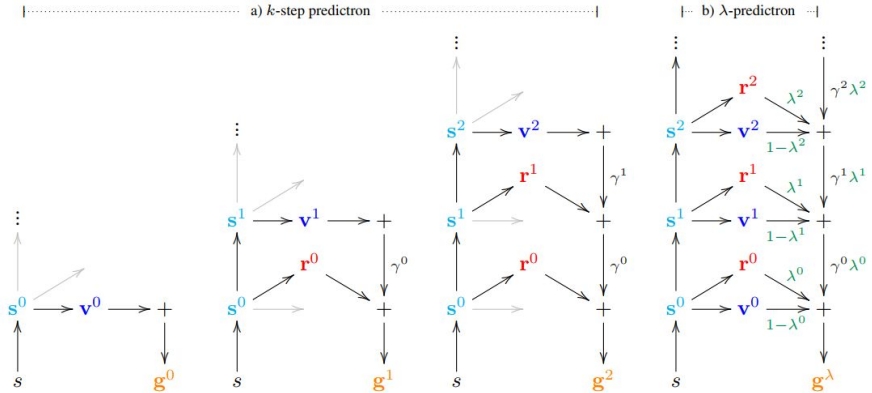
"Recurrent Environment Simulators", Chiappa et al. (2017)



Interpreting Imperfect Explicit Models

"Imagination-Augmented Agents for Deep RL", Weber, Racanière, Reichert et al. (2017)

Implicit Environment Models

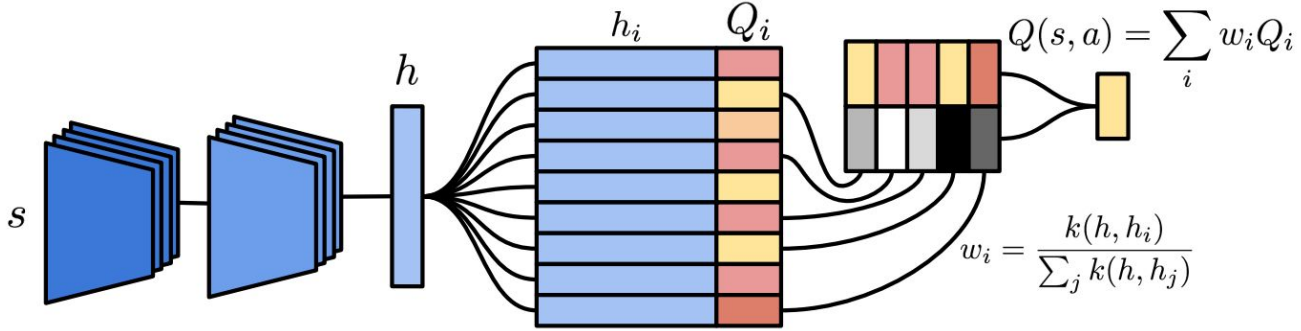


"Value Iteration Networks", Tamar et al. (2016)

"The Predictron: End-to-End Learning and Planning", Silver et al. (2017)

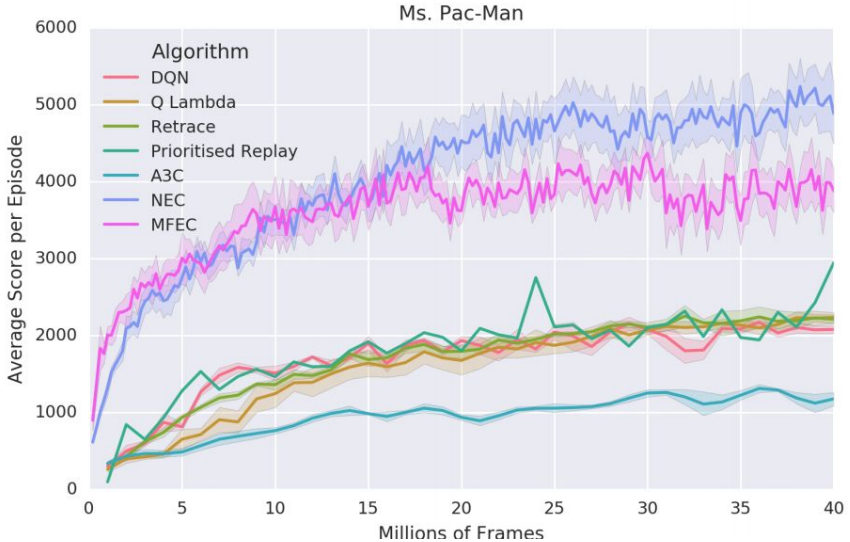
Neural Episodic Control

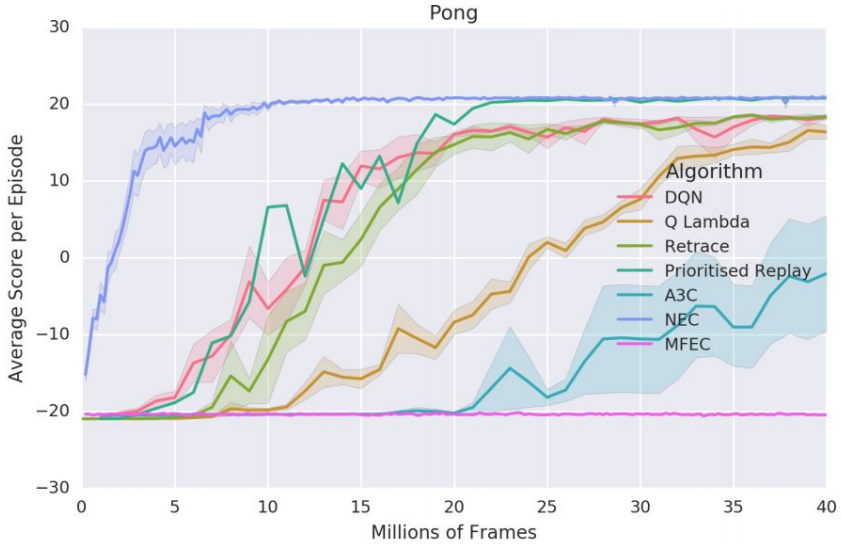
- Purely parametric approaches to deep RL are very data inefficient.
- A hybrid parametric/non-parametric method can be much more efficient.
- NEC - Represent the action value function as a table:
  - Slowly changing learned keys.
  - Rapidly changing Q-value estimates.



"Neural Episodic Control", Pritzel et al. (2017)

Neural Episodic Control



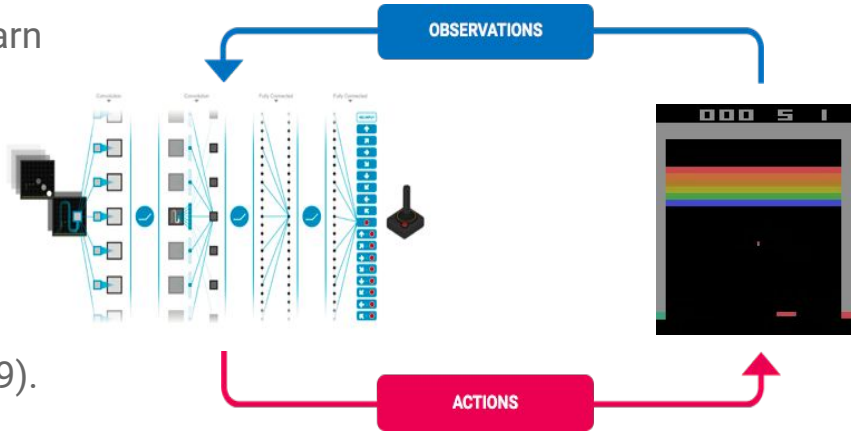


DeepMind

"Neural Episodic Control", Pritzel et al. (2017)

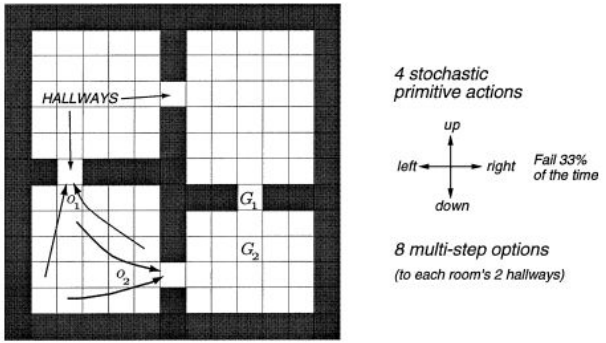
Hierarchical Reinforcement Learning

- Deep RL architectures like DQN use ConvNets to learn hierarchical structure in the visual inputs.



- Structure is also present in the space of actions/policies.
  - Motor primitives or *options* (Sutton et al., 1999).

- Capturing and exploiting this structure is one of the goals of hierarchical reinforcement learning.
  - Better exploration.
  - Faster learning through skill reuse.



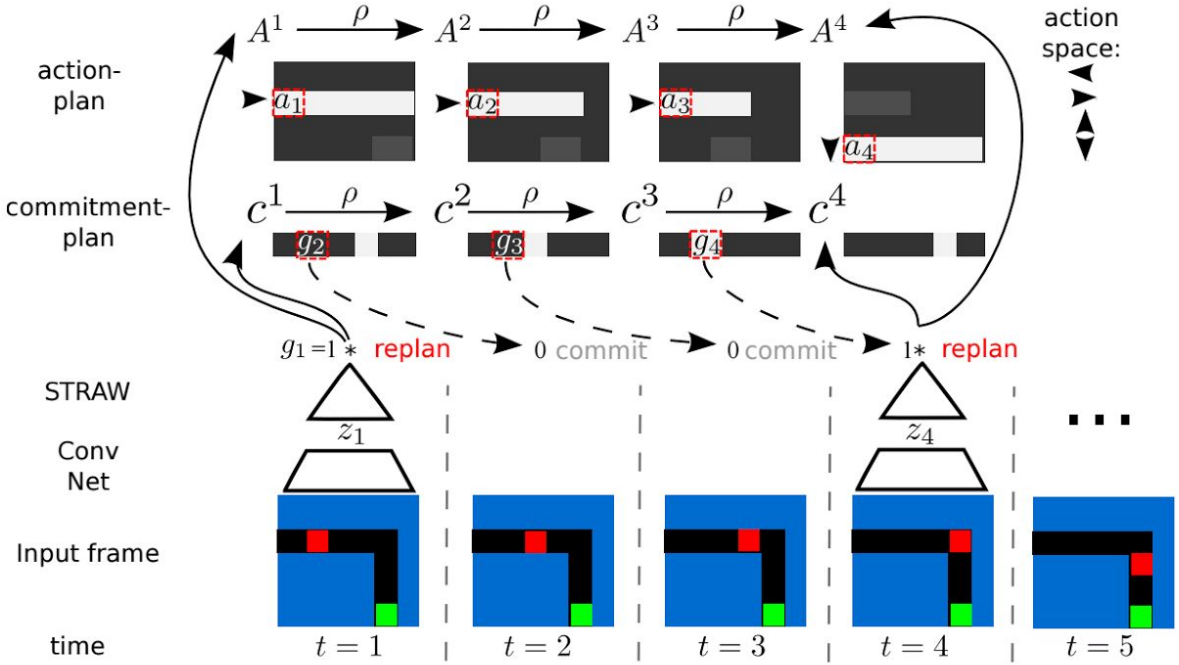
STRAW - Strategic Attentive Writer

- Strategic Attentive Writer for Learning Macro-Actions (Vezhnevets et al., 2016).
- Learning macro-actions (sequences of primitive actions). Simpler form of options.
- Learning from reward only. No hardcoded sub-goals or pseudo-rewards.

- What's the benefit?
  - Structured exploration.
  - Economic computation - the model sleeps while executing a macro-action.

- How does it work?
  - Two key components of the STRAW architecture:
    - An action plan for the future.
    - A commitment plan - how long the action plan can be followed without replanning.
  - Macro-actions arise implicitly from the architecture.

The STRAW Architecture



"Strategic Attentive Writer for Learning Macro-Actions", Vezhnevets et al. (2017)

STRAW Results

- STRAW outperforms A3C agents on games requiring longer term planning:

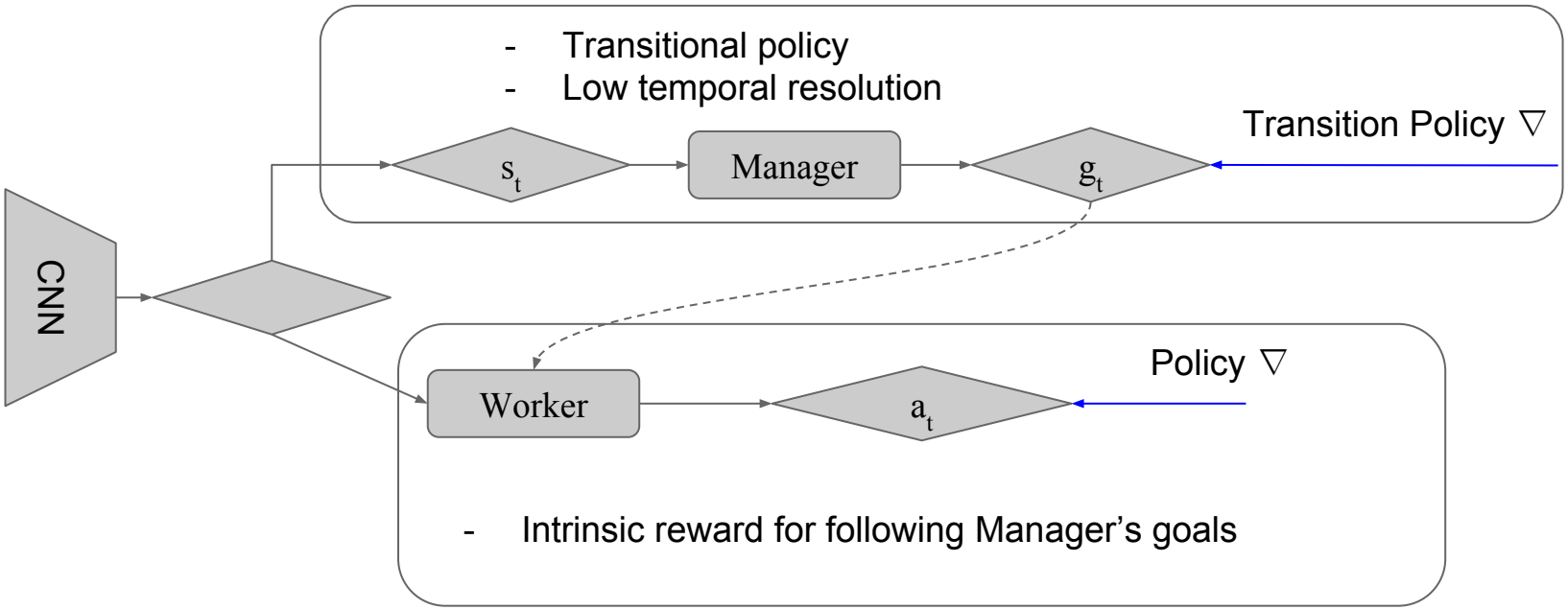
	Frostbite	Ms. Pacman	Q-bert	Hero	Crazy cl.	Alien	Amidar	Breakout
STRAWe	**8074**	**6673**	**23430**	36948	142686	**3191**	1833	363
STRAW	4138	6557	21350	35692	**144004**	2632	**2120**	423
LSTM	1409	4135	21591	35642	128351	2917	1382	**632**
FF	1543	2302	22281	**39610**	128146	2302	1235	95

Feudal Reinforcement Learning

- Agent with a two level hierarchy: **manager** and **worker**.
- Manager:
  - Does not act in the environment directly.
  - Sets goals for the worker.
  - Gets **rewarded for setting good goals** with the true reward.
- Worker:
  - Acts in the environment.
  - Gets **rewarded for achieving goals** set by the manager.
  - This is potentially a much richer learning signal.
- Key problems: how to represent goals and determine when they've been achieved.

"Feudal Reinforcement Learning", Dayan and Hinton (1993)

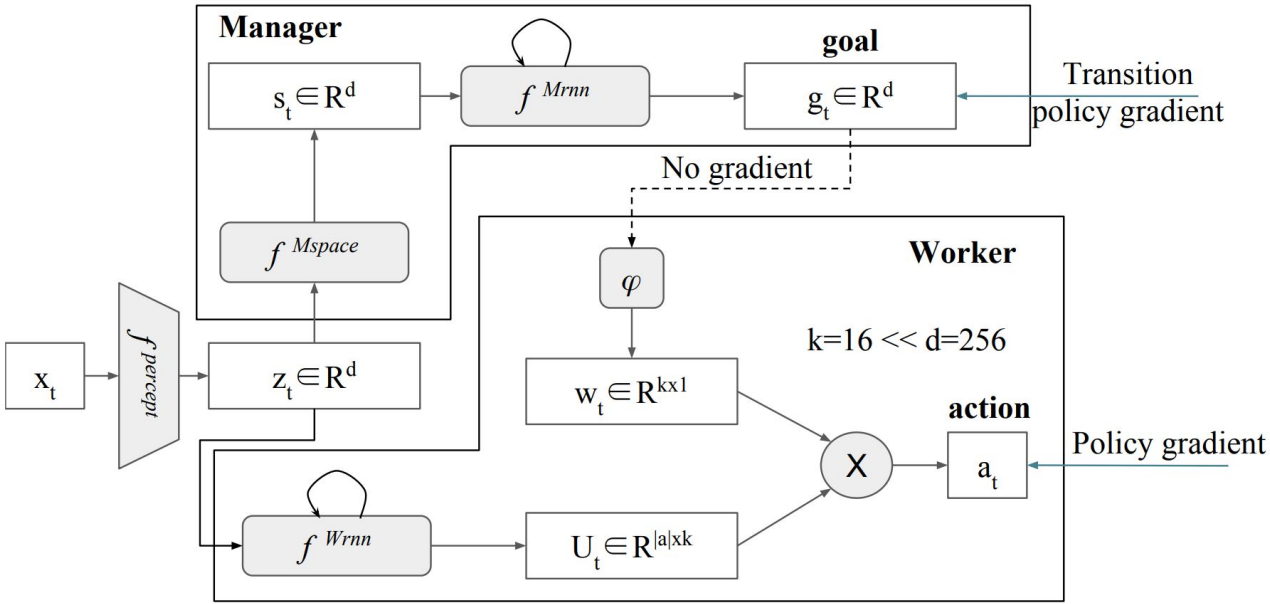
Feudal Networks Architecture



"Feudal Networks for Hierarchical Reinforcement Learning", Vezhnevets et al. (2017)

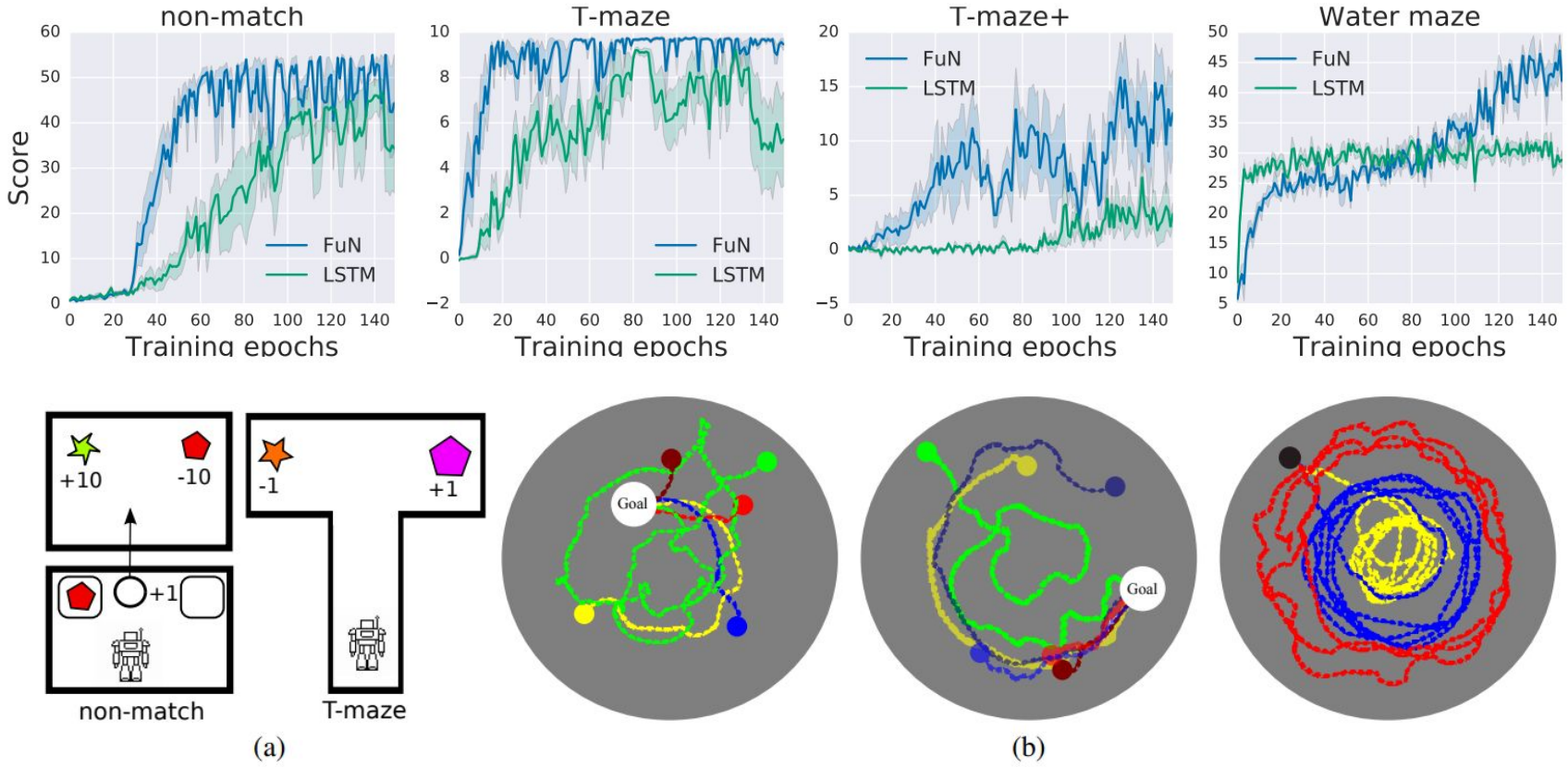
Feudal Networks Architecture

- Practical instantiation by Vezhnevets et al.
- Key idea - **represent goals in a shared feature space**.



"Feudal Networks for Hierarchical Reinforcement Learning", Vezhnevets et al. (2017)

FuN Results

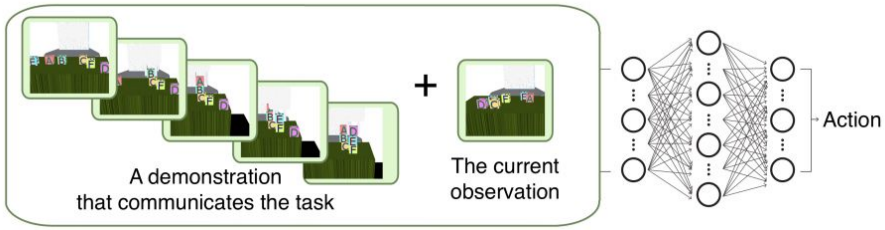


"Feudal Networks for Hierarchical Reinforcement Learning", Vezhnevets et al. (2017)

Deep RL for Real Robots

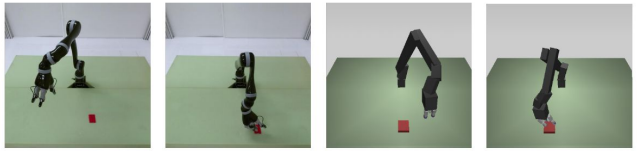
- Already a lot of exciting work from Levine, Abbeel, Finn, and many others.
- Feels like a breakthrough is coming.
- But how can slow, data inefficient deep RL help with robots?

Imitation Learning

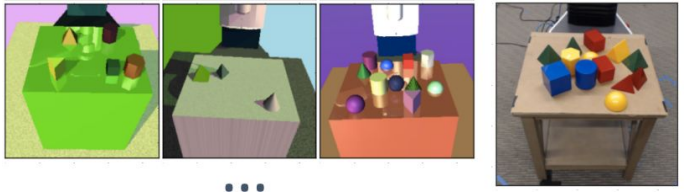


"One-Shot Imitation Learning", Duan et al. (2017)

Sim-to-Real Learning



"Sim to Real Robot Learning from Pixels with Progressive Nets"
Rusu et al. (2016)



"Domain Randomization for Transferring Deep Neural Networks
from Simulation to the Real World", Tobin et al. (2017)

Questions?