Deep RL for Robotics

Raia Hadsell

The plan

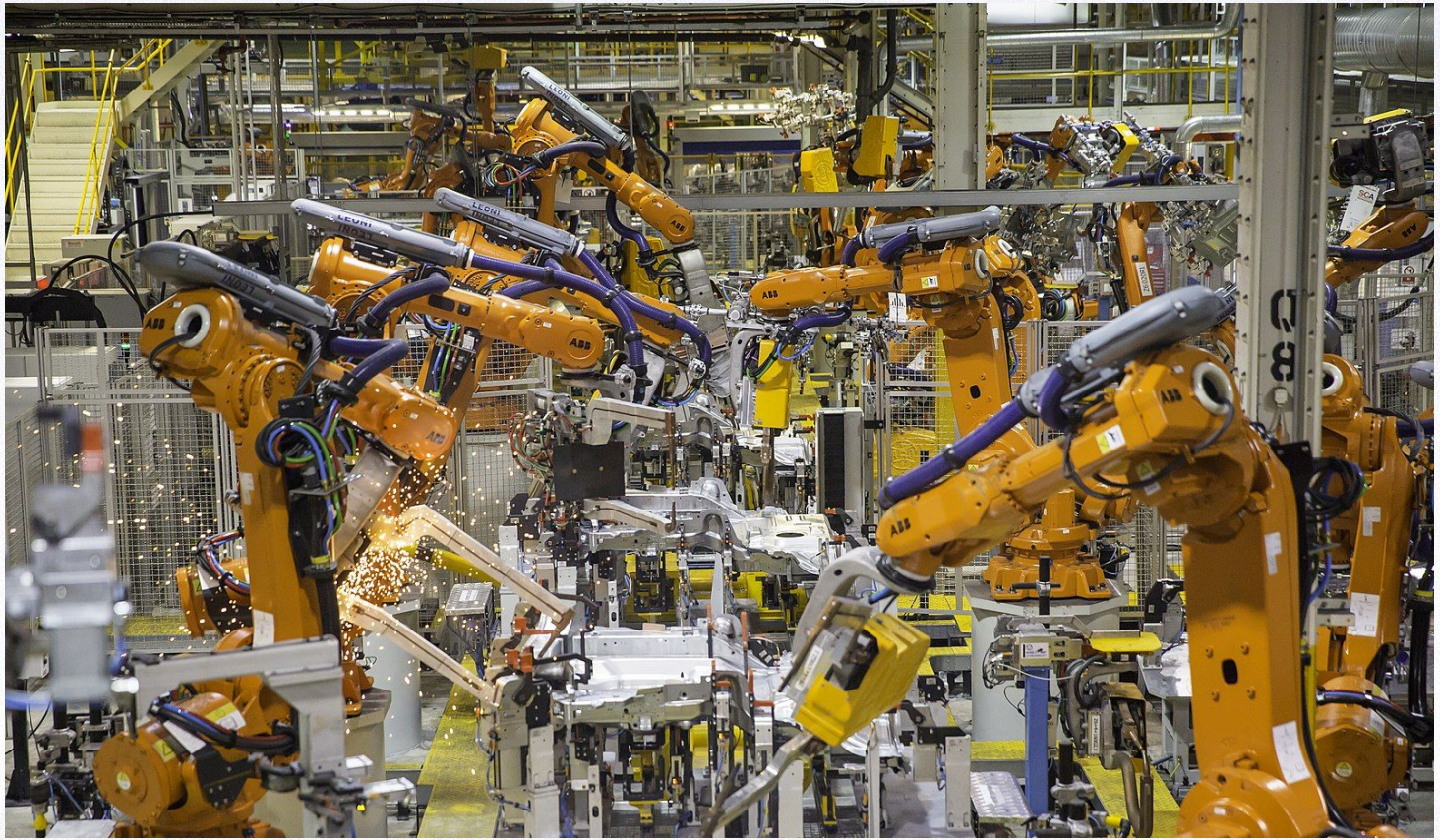
- Why should we care about end-to-end deep RL for robots?
- Deep RL for navigation
- Challenges for learning with real robots
- Possible solutions - which also happen to be important topics for RL generally

What do robots offer RL?

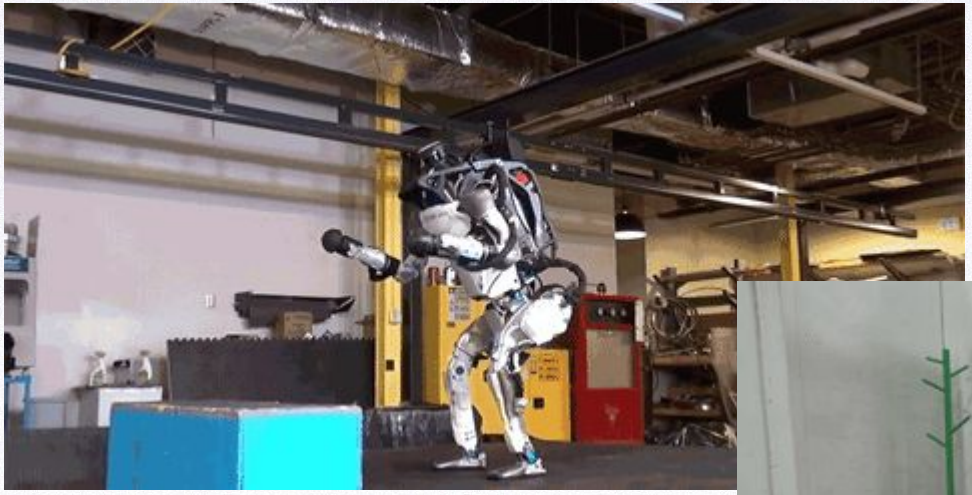
- Robots offers grounding for RL research - a rich domain of meaningful challenges
- A natural way to understand the action space, the environment, and the observation
- Biological intelligence is embodied

→ so we haven’t solved AGI until we’ve solved embodied AGI

What does RL offer robots?

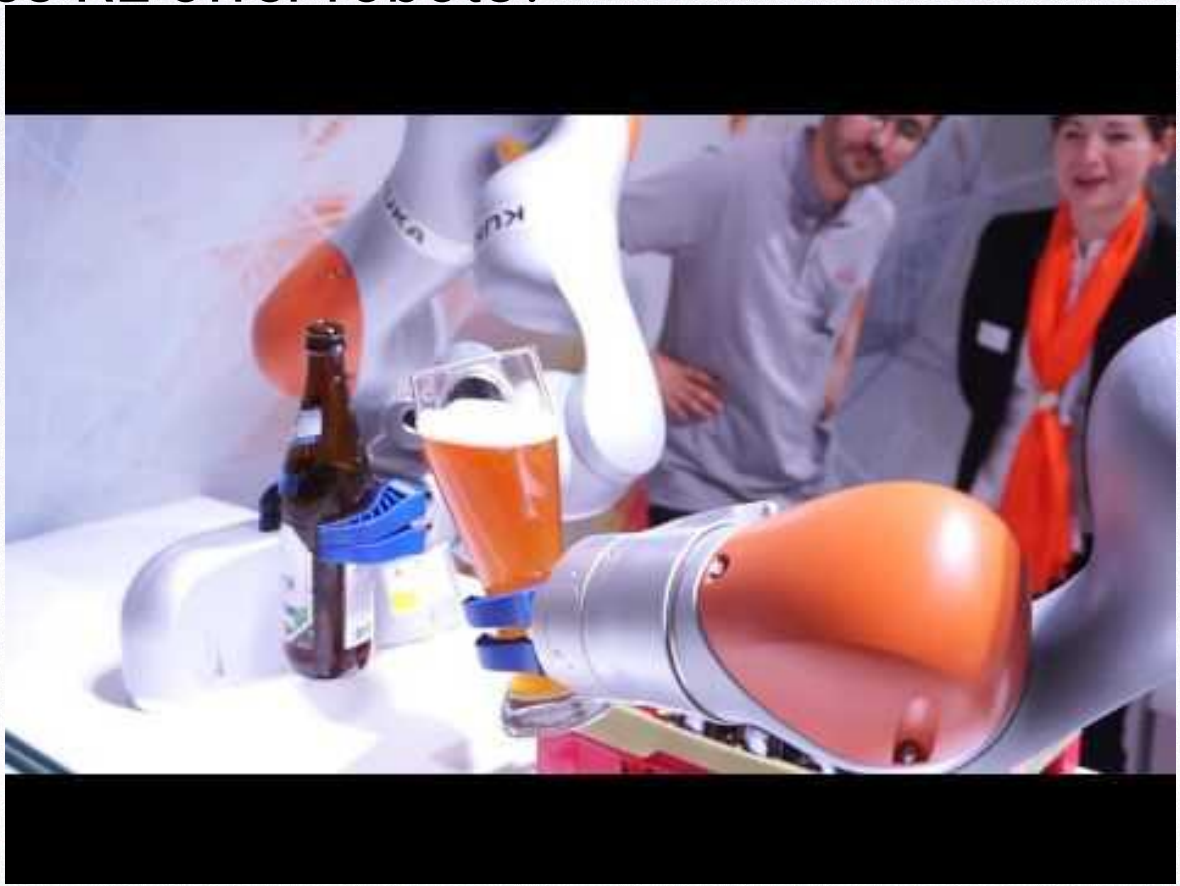


What does RL offer robots?



https://www.youtube.com/user/BostonDynamics

What does RL offer robots?



KUKA Robot pours a beer at Hannover Messe 2017 -- https://www.youtube.com/watch?v=c6mqv4vf2mg

What does RL offer robots?

- **BUT** we still believe that by learning end-to-end there will be a huge advantage
- Why? we can overcome some of the limitations
  - minimal contacts
  - constrained environments and tasks
  - slowness
  - brittleness

End-to-end Deep Learning for robots?

2010: Speech Recognition

Audio → Acoustic Model → Phonetic Model → Language Model → Text

2012: Computer Vision

Pixels → Key Points → SIFT features → Deformable Part Model → Labels

2014: Machine Translation

Text → Reordering → Phrase Table/Dictionary → Language Model → Text

Robotics?

Sensors → Perception → World Model → Planning → Control → Action

End-to-end Deep Learning for robots?

2010: Speech Recognition

2012: Computer Vision

2014: Machine Translation

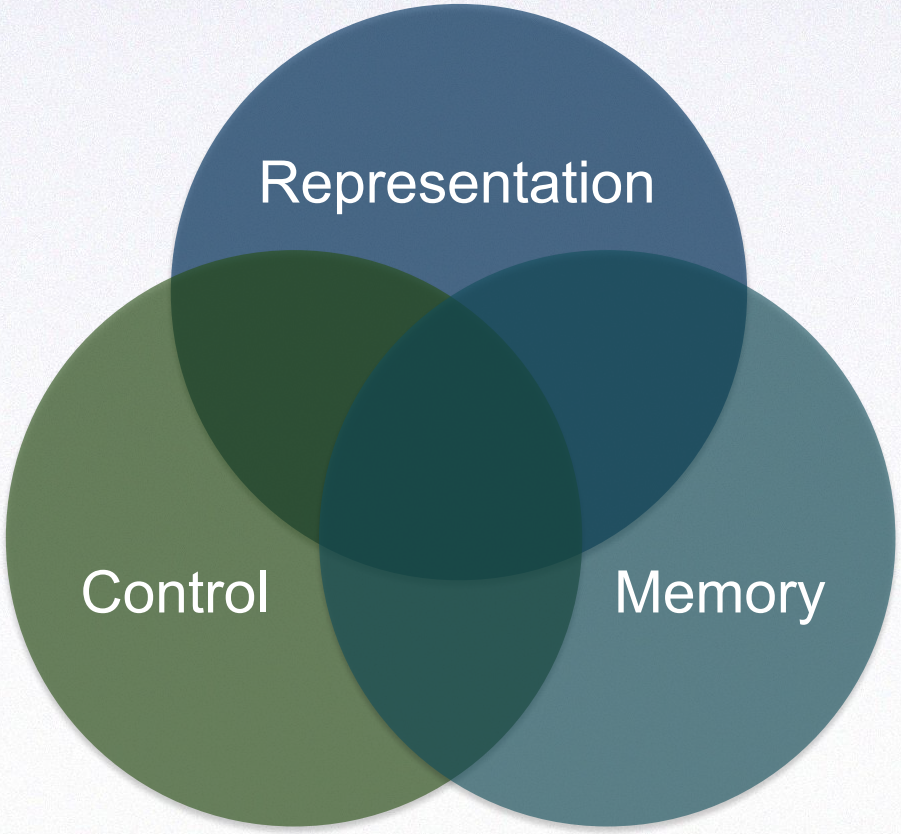
Robotics?

Sensors → Perception → World Model → Planning → Control → Action

DeepMind

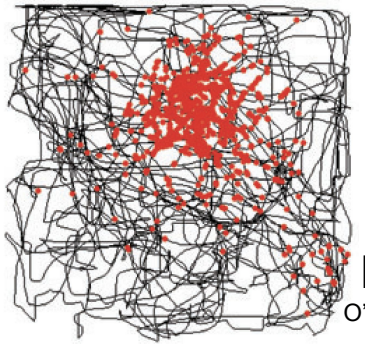
Navigation

- where am I?
- where did I come from?
- where am I going?
- what is the shortest path from A to B?
- have I ever been here before?

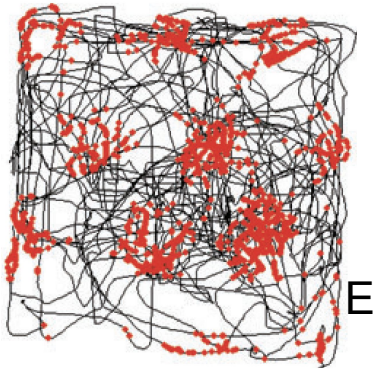

Representation
Control
Memory

Representation

Place and Grid cells

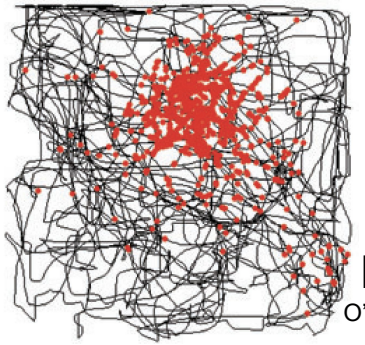


Place cell
Hippocampus
O'Keefe & Dostrovsky (1971)

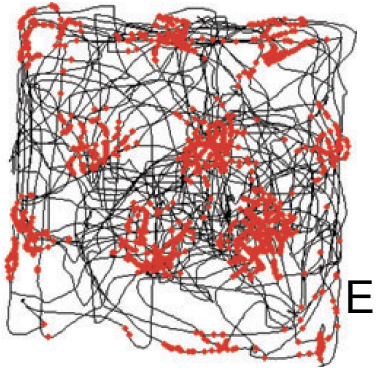


Grid cell
Entorhinal cortex
Fyhn et al. (2004)

Place and Grid cells



Place cell
Hippocampus
O'Keefe & Dostrovsky (1971)



Grid cell
Entorhinal cortex
Fyhn et al. (2004)

Vector-based Navigation using Grid-like Representations in Artificial Agents

Andrea Banino, Caswell Barry, et al.

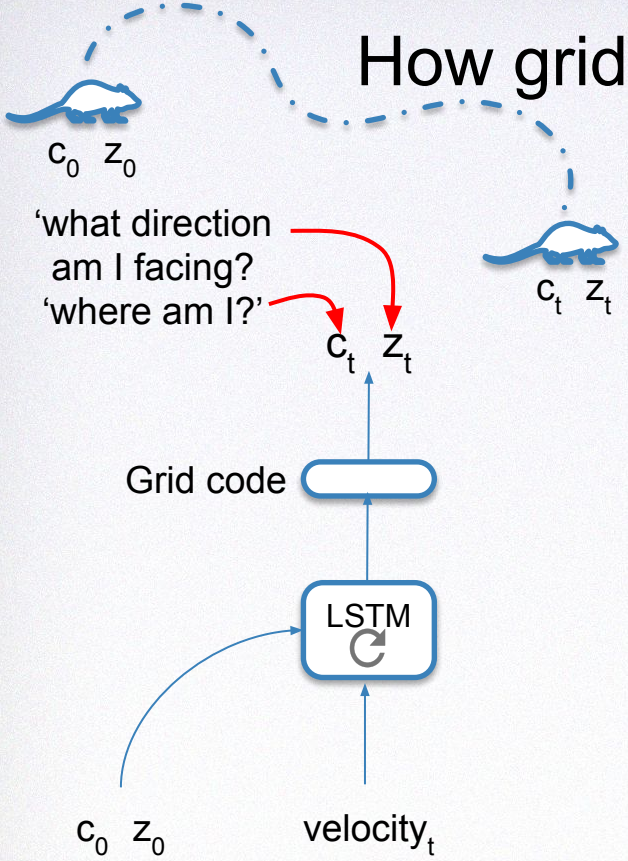
Main claims of paper:

1. **Grid cells can emerge by training an RNN to do path integration**
   a. Online gradient descent training
   b. Place cell activations serve as targets

2. **Grid cells are an effective basis for vector based navigation**
   a. Useful auxiliary representation for Deep RL
   b. Metric representation of space

- *Previous and concurrent work:*
- *Milford et al (2004) "RatSLAM: a hippocampal model for simultaneous localization and mapping”*
- *Franzius et al (2007) “Slowness and sparseness lead to place, head-direction, and spatial-view cells”*
- *Independent confirmation of grid cell emergence from RNN training:*
- *Cueva & Wei (2018) “Emergence of grid-like representations by training RNN for spatial localization”*

How grid cells emerge in a neural net



Inputs:

1. Relative velocity (lateral and angular)
2. LSTM initialization at start of sequence

Targets:

1. Place cells (c): mixture of Gaussians
2. Head direction cells (z): mixture of Von Mises

Training:

1. Dropout on the linear, g, layer
2. Gradient clipping on weights from LSTM to output

$$c_i = \frac{e^{-\frac{||\vec{x}-\vec{\mu}_i^{(c)}||_2^2}{2(\sigma^{(c)})^2}}}{\sum_{j=1}^{N} e^{-\frac{||\vec{x}-\vec{\mu}_j^{(c)}||_2^2}{2(\sigma^{(c)})^2}}} \qquad z_i = \frac{e^{\kappa^{(z)}\cos(\varphi-\mu_i^{(z)})}}{\sum_{j=1}^{M} e^{\kappa^{(z)}\cos(\varphi-\mu_j^{(z)})}}$$

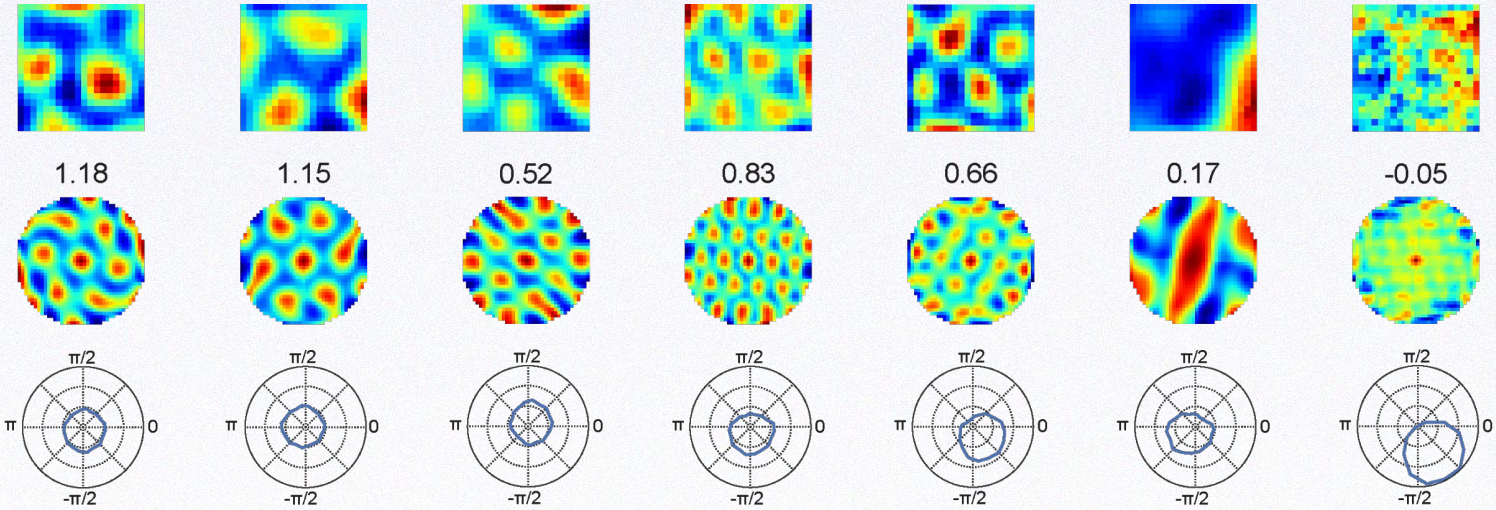
DeepMind

Activation in the linear layer

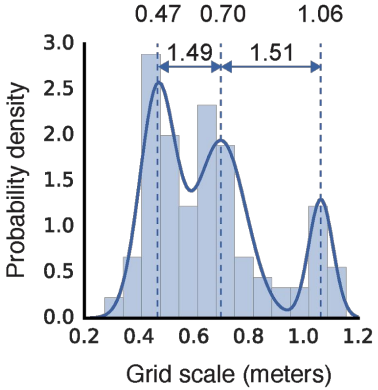
Rate maps in 2D space

Spatial auto-correlograms

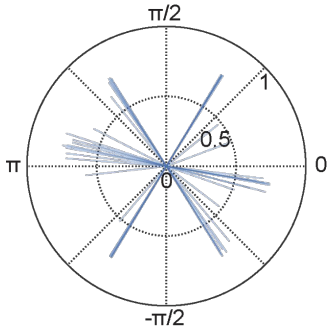
Activity vs. head direction



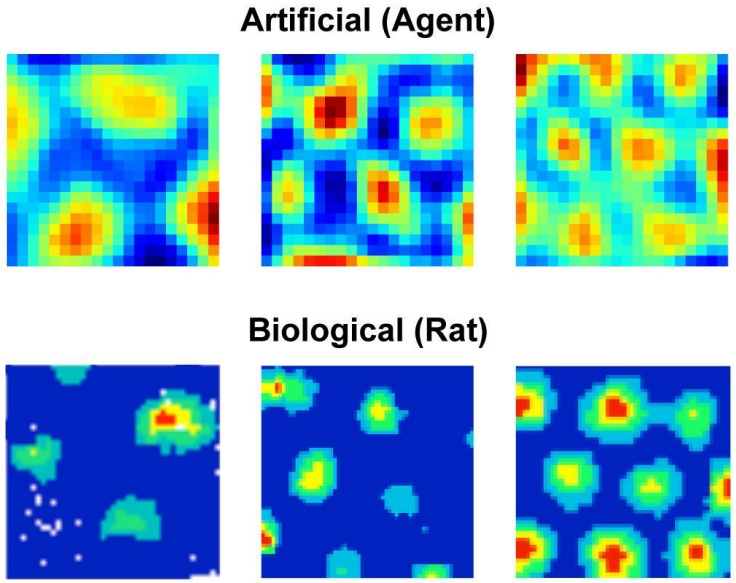
Strikingly similar to biological cells!



Spatial scale of grid-like units

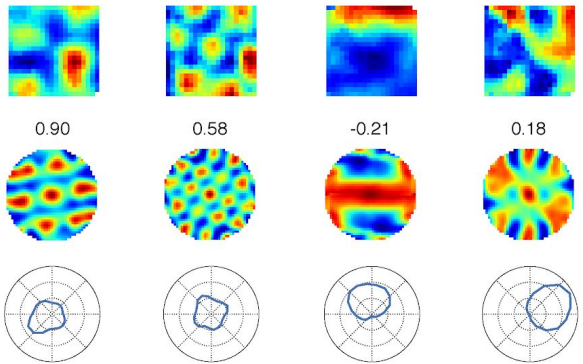


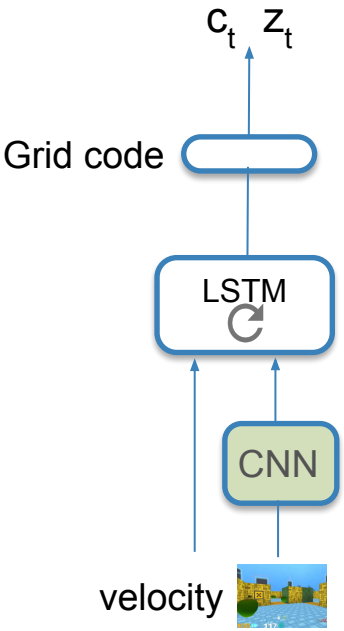
6-fold tuning of head direction units



Integrating visual place estimates

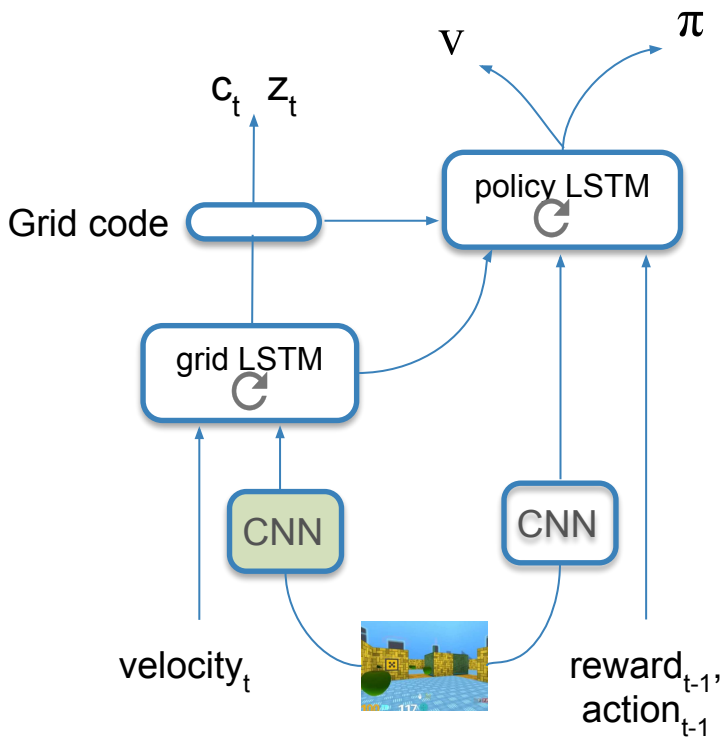
- Remove ground truth initial conditions
- CNN pretrained to estimate place and head direction targets
- Visual estimate of position is integrated with velocities



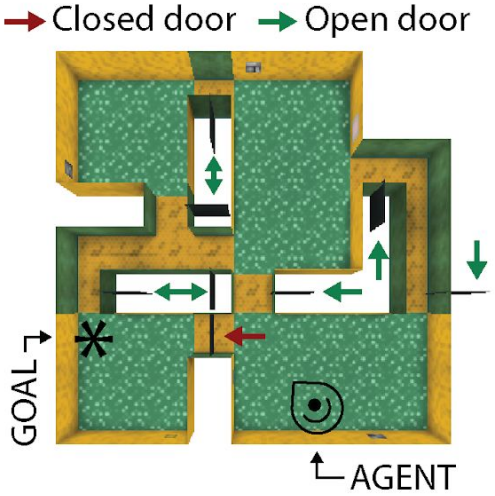


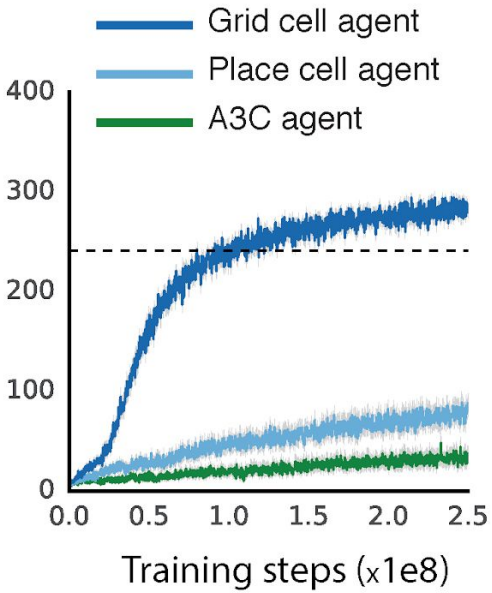
The Grid Cell Agent

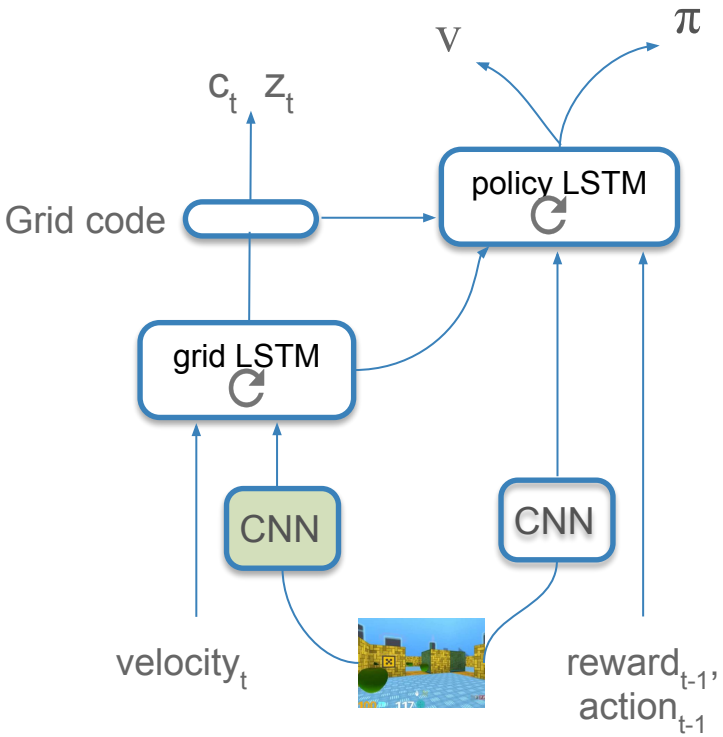
- Grid code is input to an LSTM with actor-critic outputs
- Grid code is memorized at reward
- No gradient from A3C to grid network



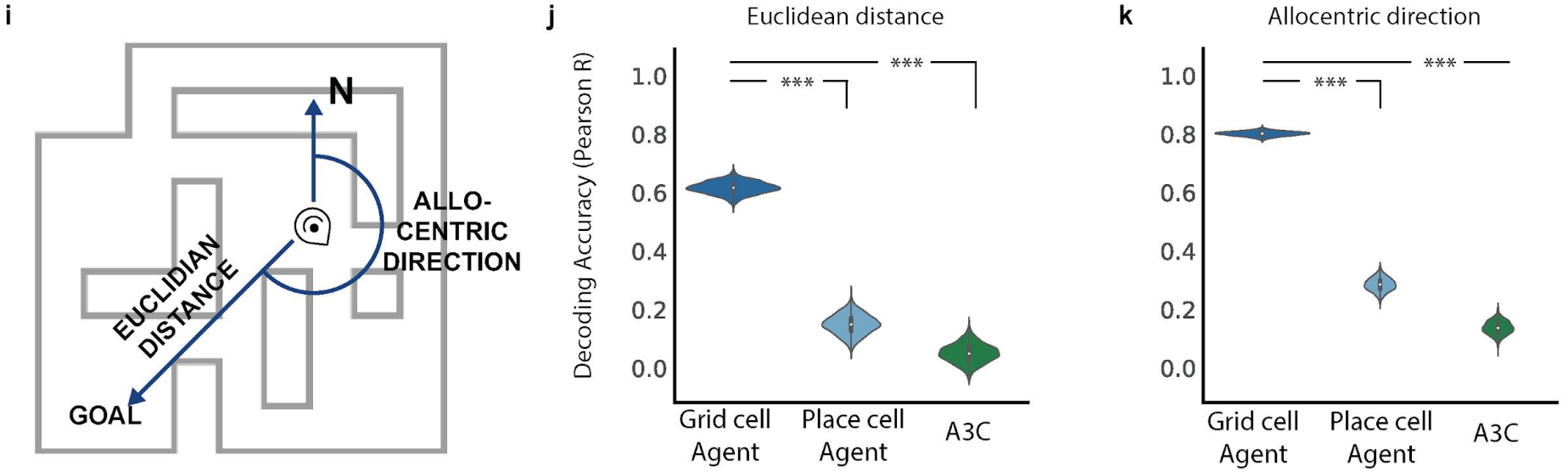
Experiment 1: **goal-finding in mazes with dynamic doors**



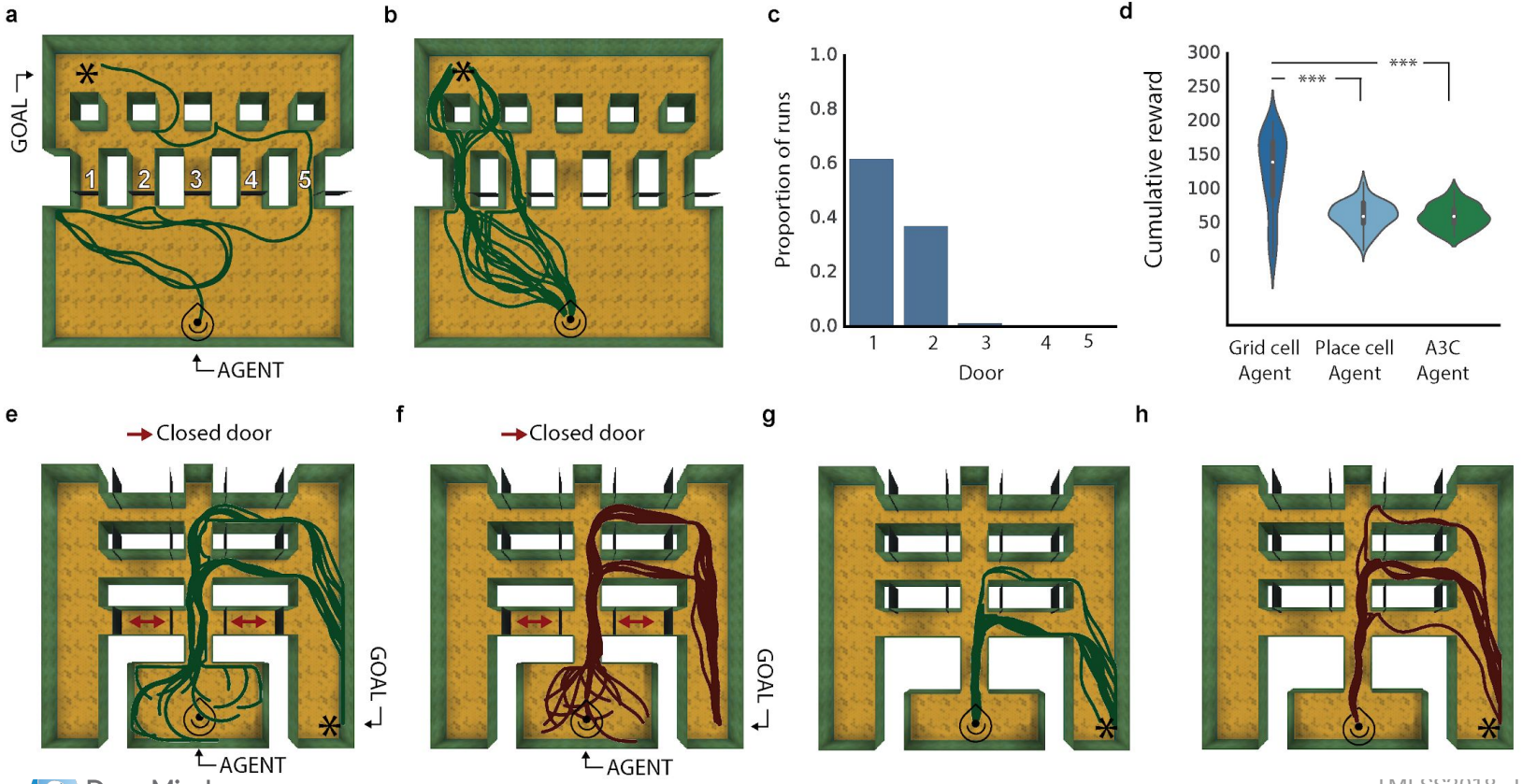




Experiment 2: **Vector based navigation**



Experiment 3: **Shortcut behaviour**



DeepMind

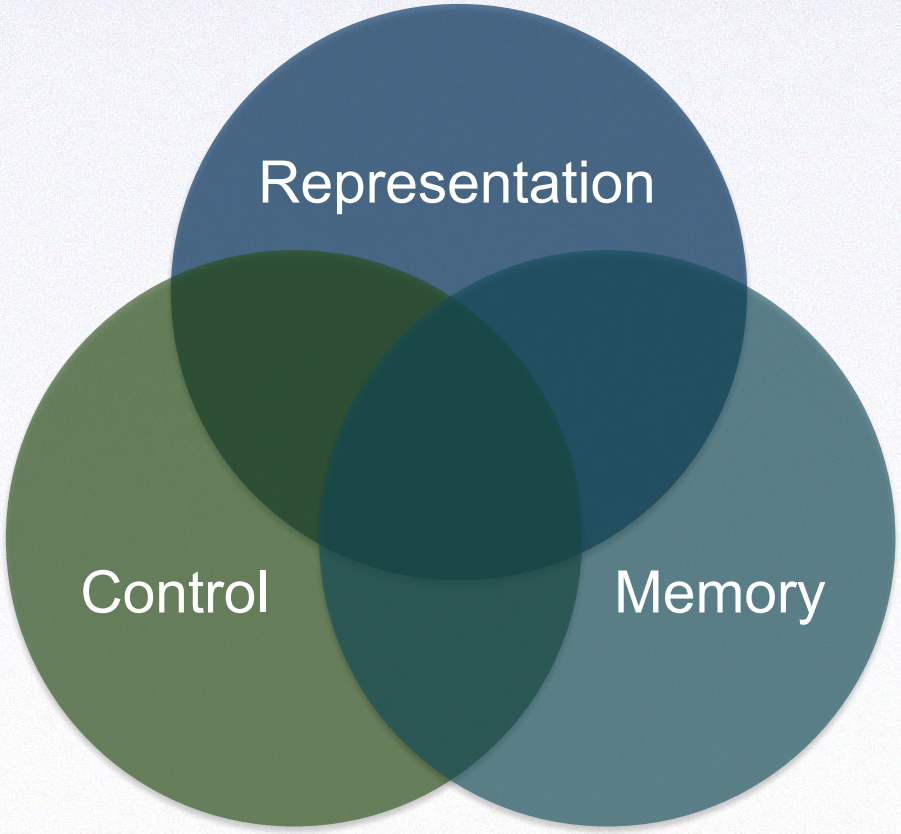
LETTER

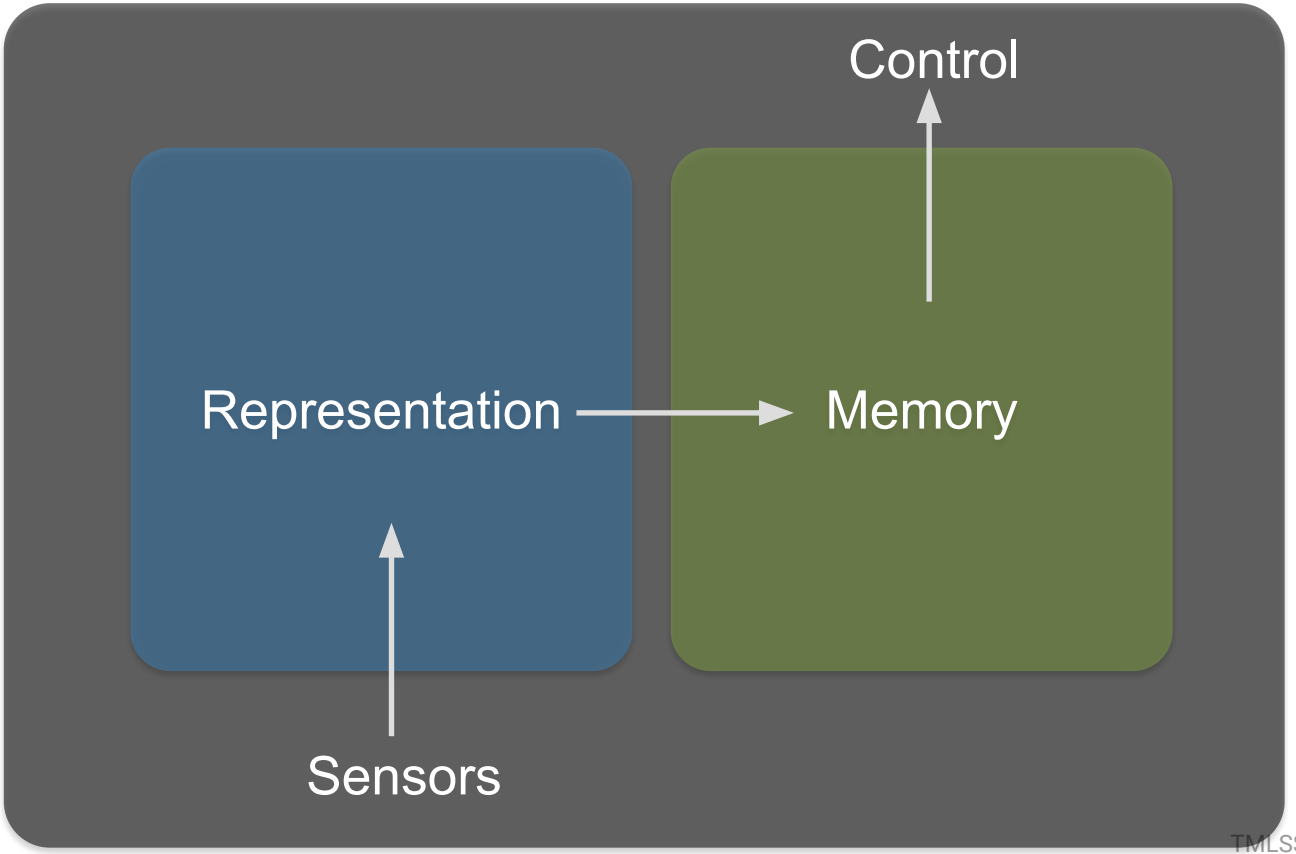
https://doi.org/10.1038/s41586-018-0102-6

Vector-based navigation using grid-like representations in artificial agents

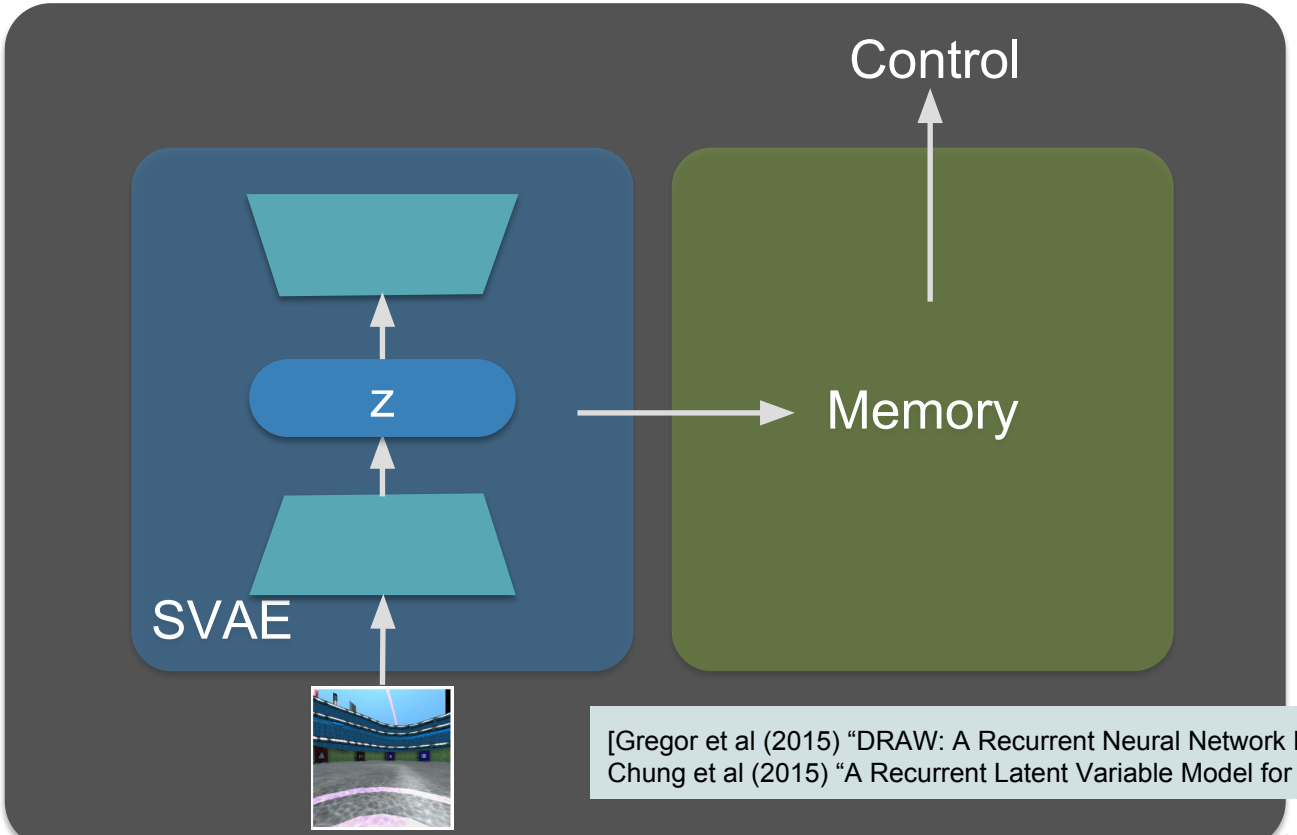
Andrea Banino[1,2,3,5]*, Caswell Barry[2,5]*, Benigno Uria[1], Charles Blundell[1], Timothy Lillicrap[1], Piotr Mirowski[1], Alexander Pritzel[1], Martin J. Chadwick[1], Thomas Degris[1], Joseph Modayil[1], Greg Wayne[1], Hubert Soyer[1], Fabio Viola[1], Brian Zhang[1], Ross Goroshin[1], Neil Rabinowitz[1], Razvan Pascanu[1], Charlie Beattie[1], Stig Petersen[1], Amir Sadik[1], Stephen Gaffney[1], Helen King[1], Koray Kavukcuoglu[1], Demis Hassabis[1,4], Raia Hadsell[1] & Dharshan Kumaran[1,3]*

> “In this incredible study, Banino and colleagues found that an artificial agent with access to only a few sensory-motor cues spontaneously formed grid cells with properties that were remarkably similar to those in brain… This truly special paper highlights the power of studying artificial intelligence for understanding our own brains.”
>
> —Edvard Moser


Representation
Control
Memory

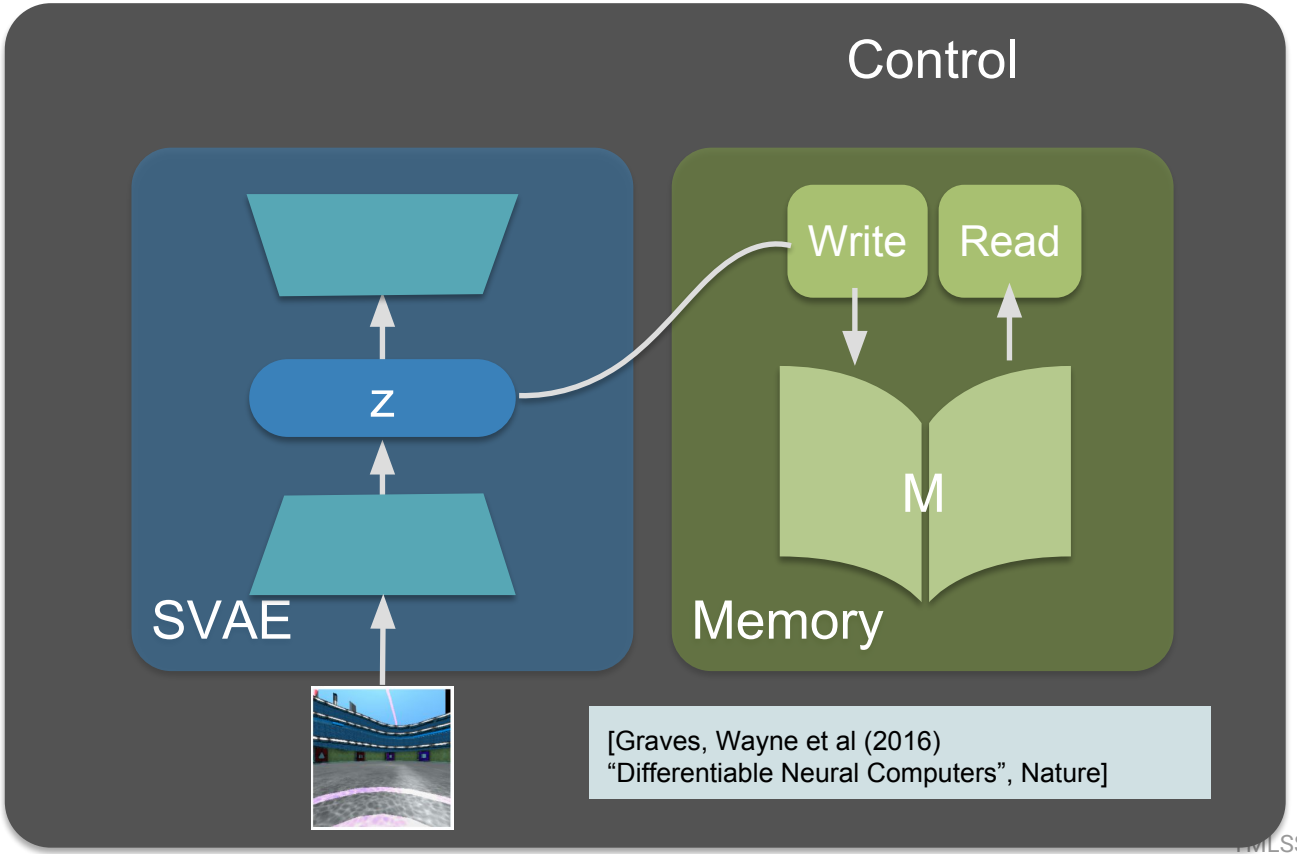

Control
Representation
Memory
Sensors

Representation: Sequential Variational Autoencoder

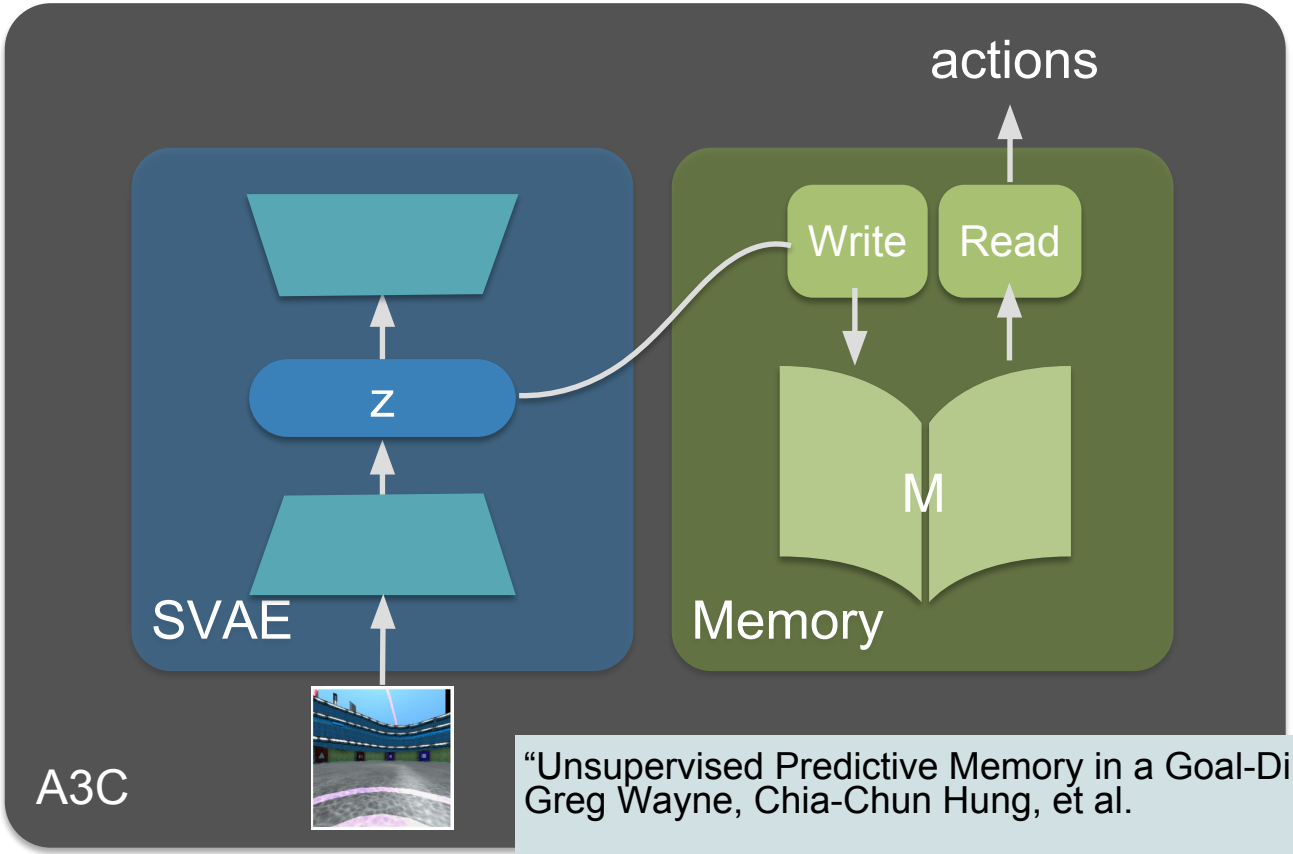


[Gregor et al (2015) “DRAW: A Recurrent Neural Network For Image Generation”; Chung et al (2015) “A Recurrent Latent Variable Model for Sequential Data”]

Memory: Differentiable Neural Computer



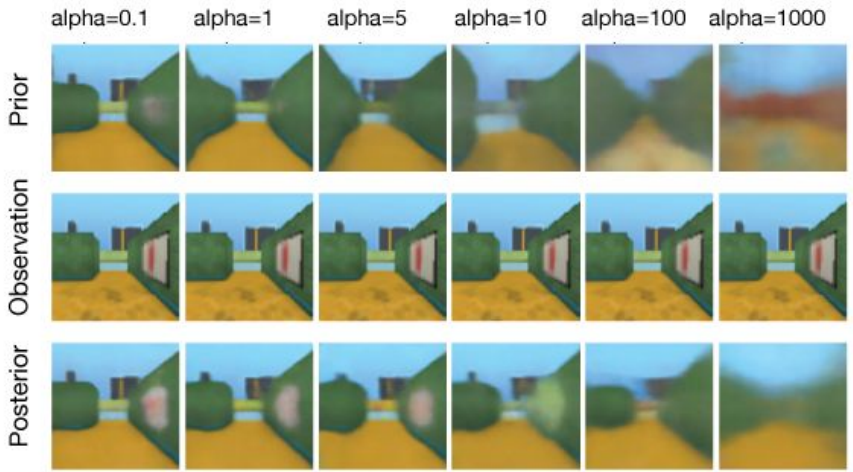
MERLIN: Memory-Enabled Reinforcement Learning

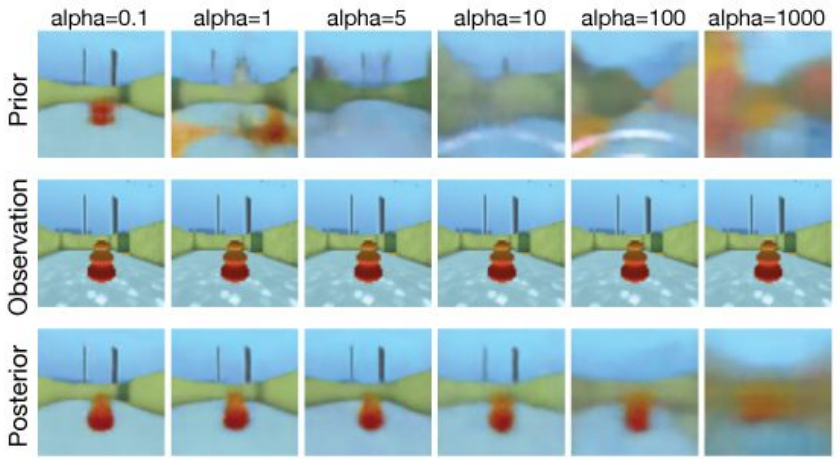


"Unsupervised Predictive Memory in a Goal-Directed Agent"
Greg Wayne, Chia-Chun Hung, et al.

DeepMind

Store Compressed Representations

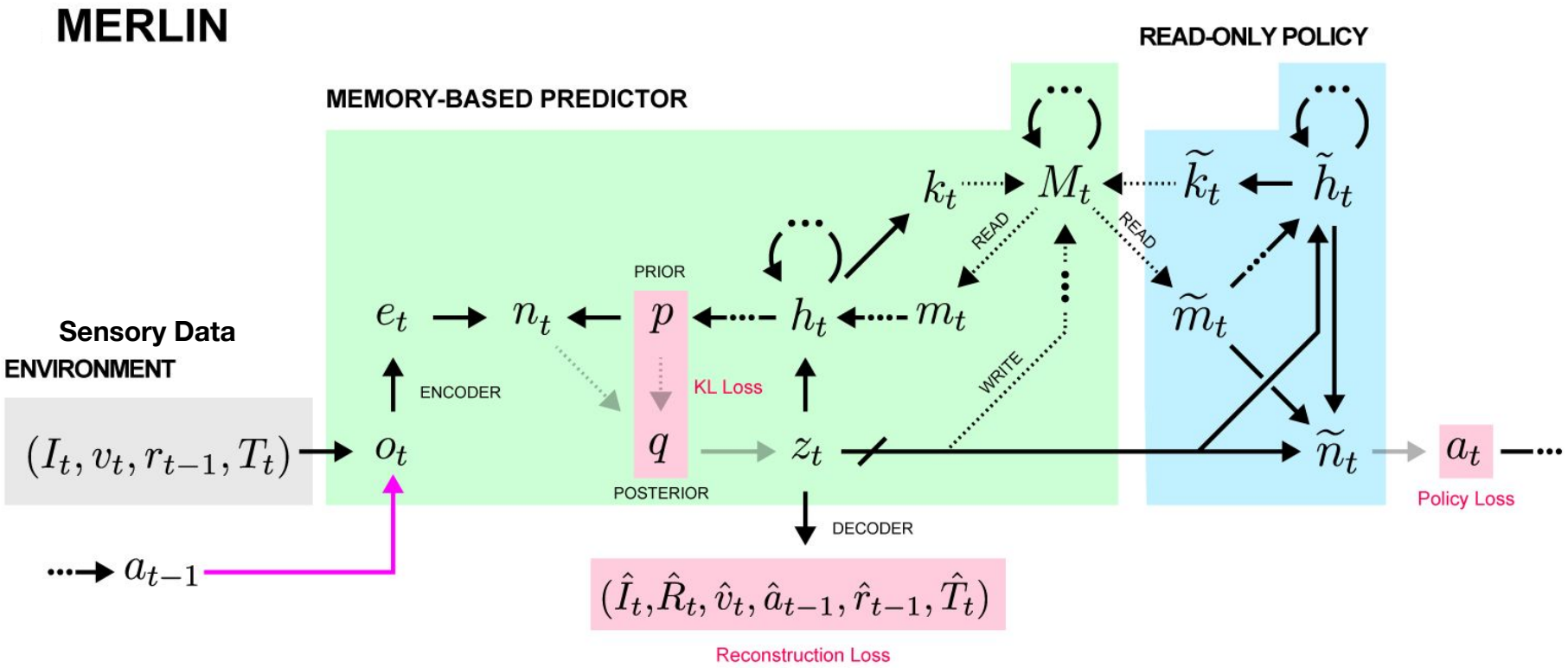




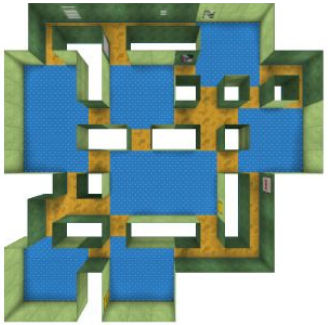
Prediction error vs. return cost coefficients

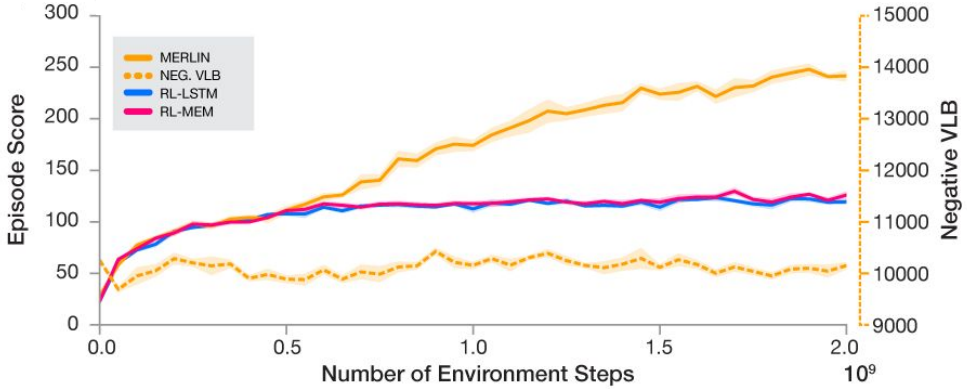
slide from G. Wayne

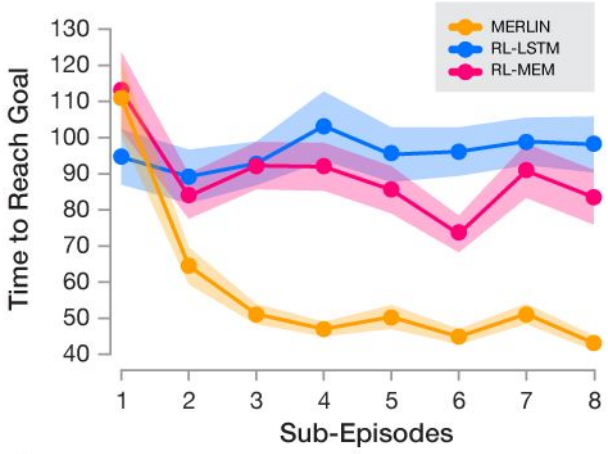
MERLIN, the full architecture



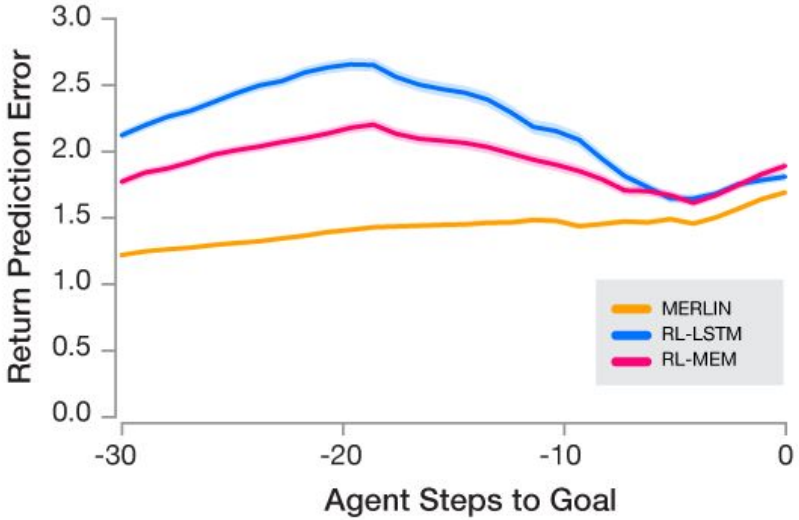
Navigation

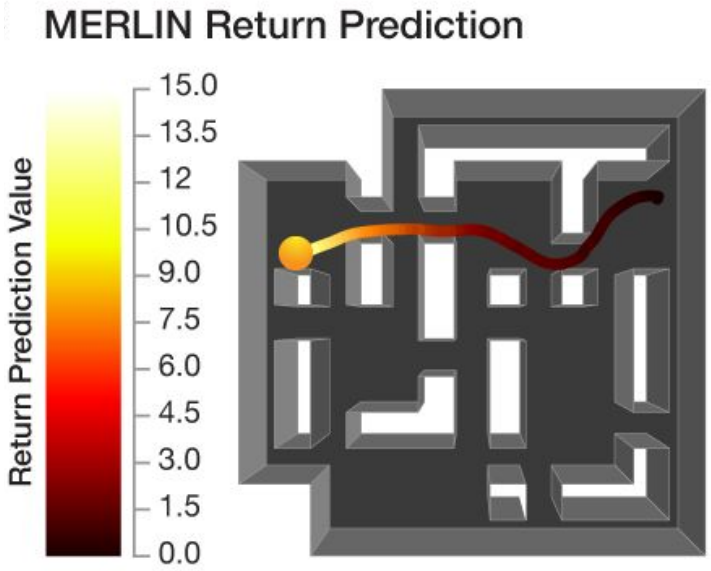



MERLIN
NEG. VLB
RL-LSTM
RL-MEM
Episode Score
Negative VLB
Number of Environment Steps
10^9


MERLIN
RL-LSTM
RL-MEM
Time to Reach Goal
Sub-Episodes

Return Prediction

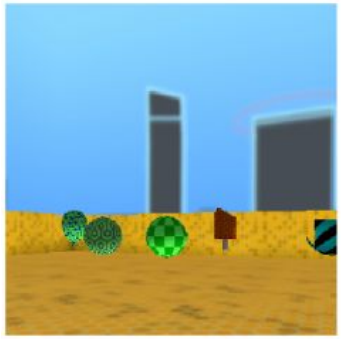


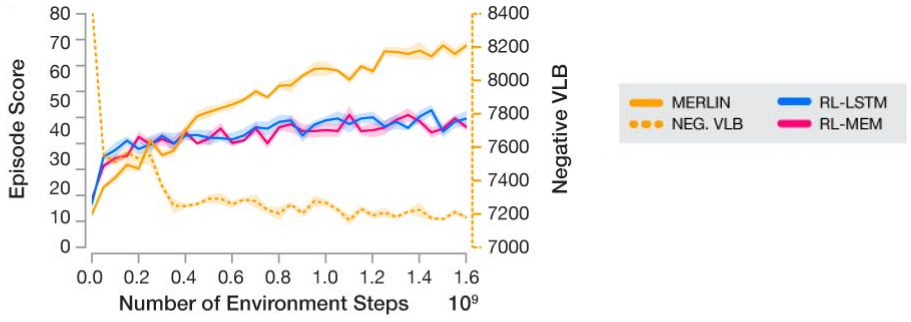


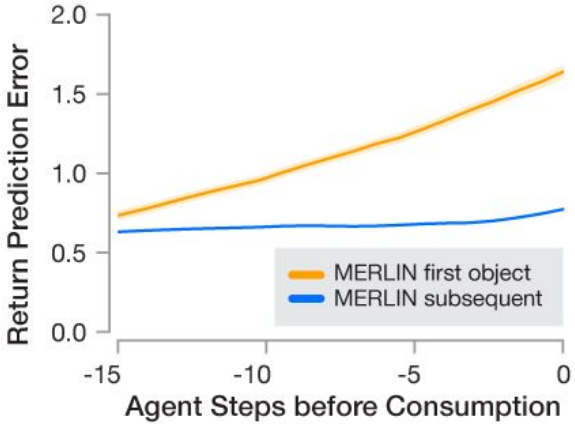

MERLIN
RL-LSTM
acc. rewards: 0
acc. rewards: 0
RL-LSTM learns to explore efficiently but does not remember where the goal is

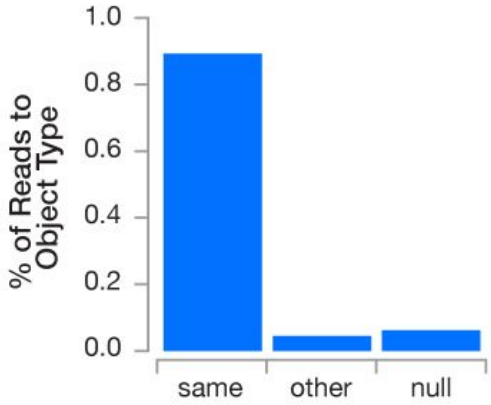

MERLIN
RL-LSTM
acc. rewards: 0
acc. rewards: 0
RL-LSTM learns to explore efficiently but does not remember where the goal is

Rapid Reward Valuation (One-Shot)







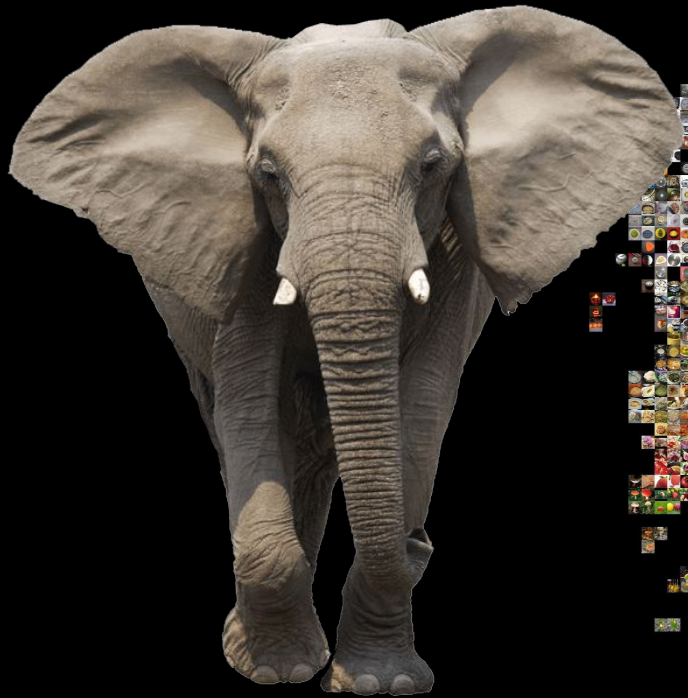


slide from G. Wayne


MERLIN
RL-LSTM
acc. rewards: 5
acc. rewards: 5

- 3D, first person environment
- partially observed
- procedural variations

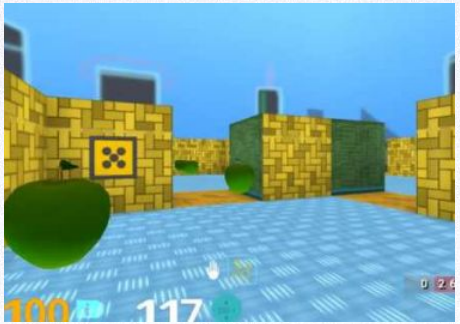
… but it's not real



DeepMind

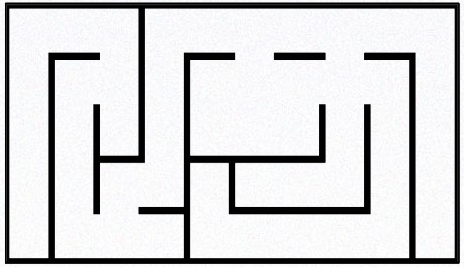
The real challenges: complexity and scale

Can we solve navigation tasks in the real world?

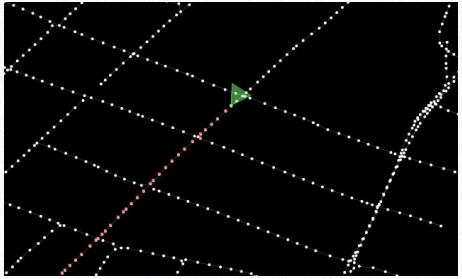


observation

observation



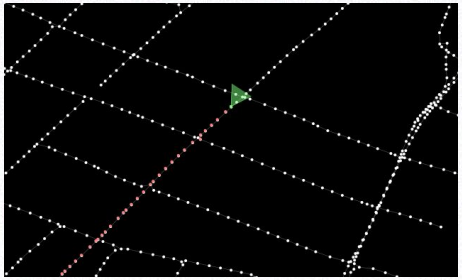
structure



structure

StreetView as an RL environment: **StreetLearn**

streetview images



google maps

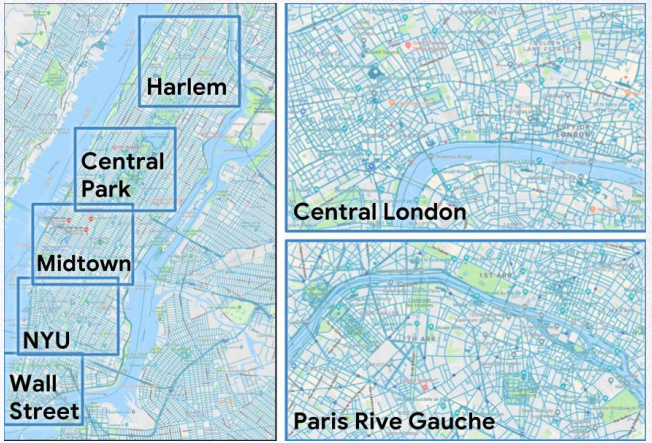
- RGB image cropped from panorama (84x84)

Actions: move to next node, rotate view 20° or 60°

Learning to navigate in cities without a map

Piotr Mirowski, Raia Hadsell, Alex Graves

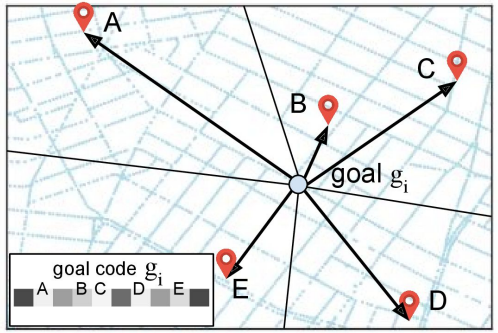
NYC, London, Paris



- 14,000 to 60,000 nodes (panoramas) per city, covering range of 3-5km per city
- Discrete action space allows rotating in place and stepping to next node
- Multi-city dataset and RL environment will be released later this year

The Courier Task

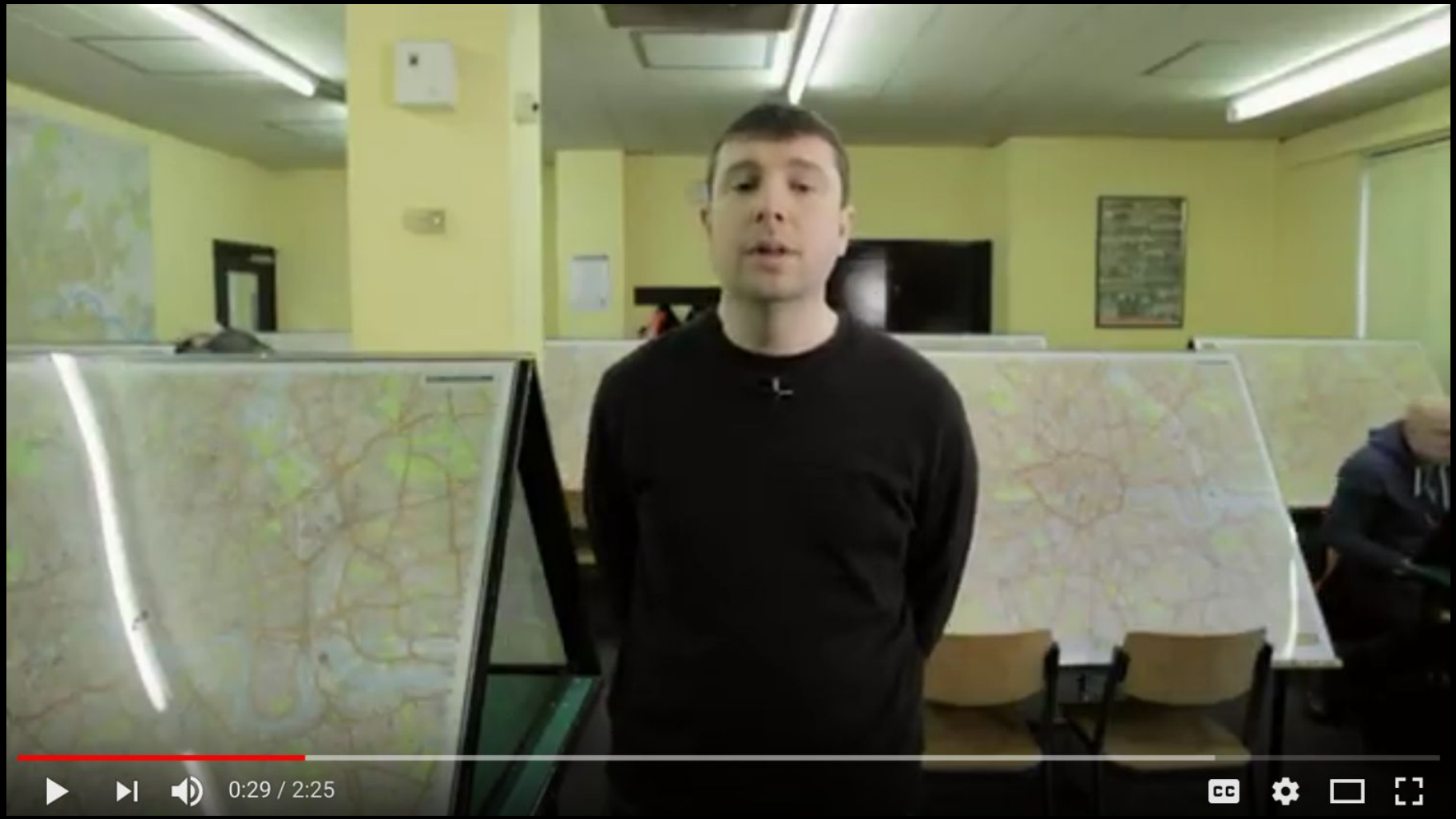
- **Random start/end** navigation without a map
- Reward when close to goal
- Actions: rotate left, right, or step forward
- Inputs for the agent at every time point *t*:
  - 84x84 RGB **image observations**
  - landmark-based **goal description**



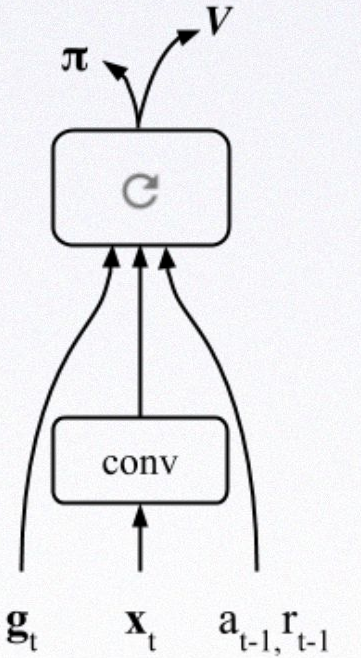
The Knowledge

- Test to get a black cab license in London
- Candidates study for 3-4 years
- Memorize 25,000 roads
- and 20,000 named locations
- By the time they've passed the exam,
  their hippocampi are 'significantly enlarged'.

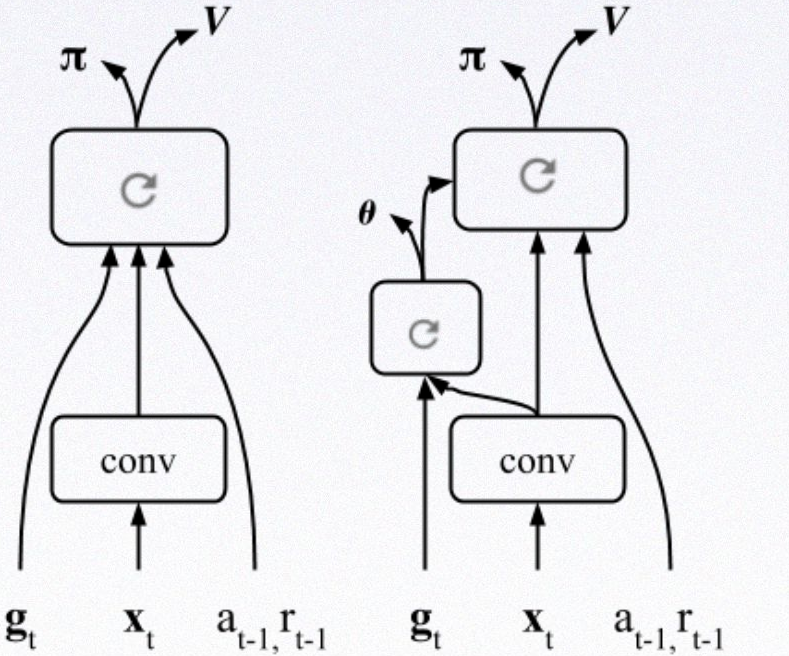
Woollett & Maguire. 2011. Acquiring ''the Knowledge'' of London's Layout Drives Structural Brain Changes. Current Biology


0:29 / 2:25

https://www.nytimes.com/video/t-magazine/100000003223621/applying-the-knowledge.html

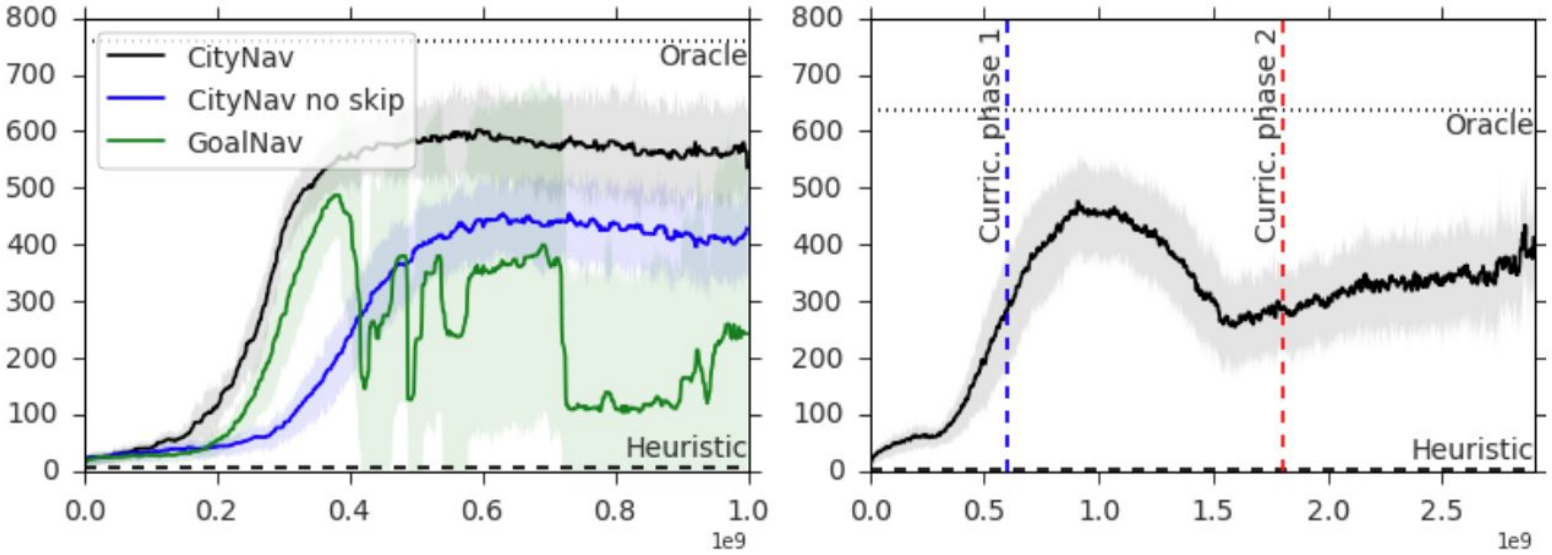


a. *GoalNav* agent



a. *GoalNav* agent b. *CityNav* agent

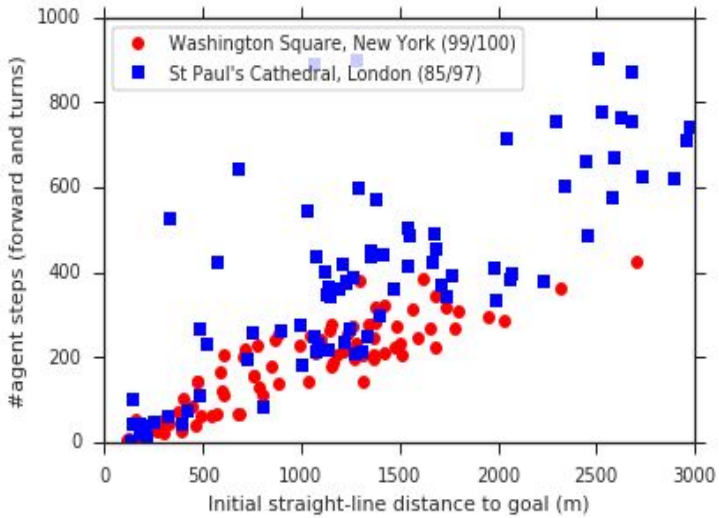
Successful learning of all 3 cities, with curriculum

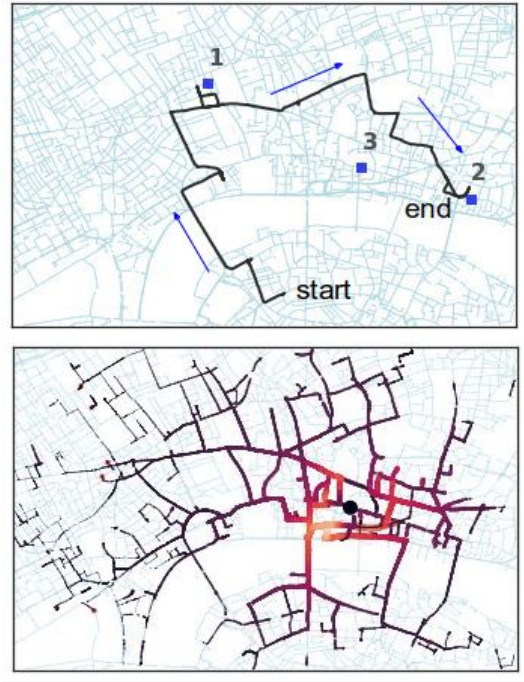


video here:
https://sites.google.com/corp/view/streetlearn

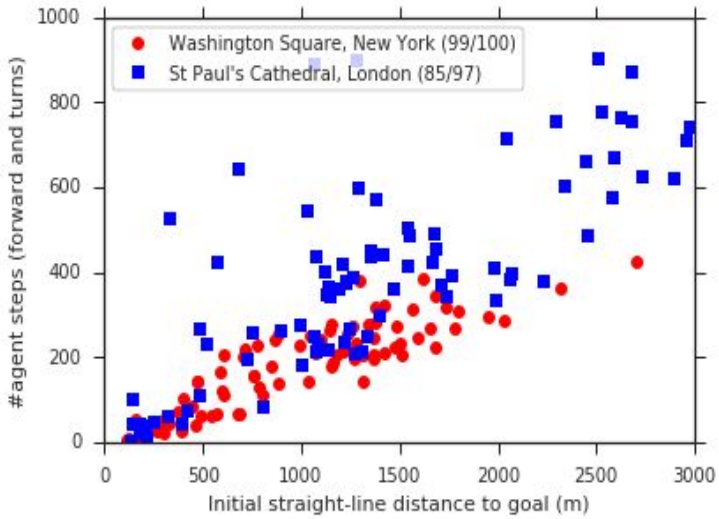
video here:
https://sites.google.com/corp/view/streetlearn

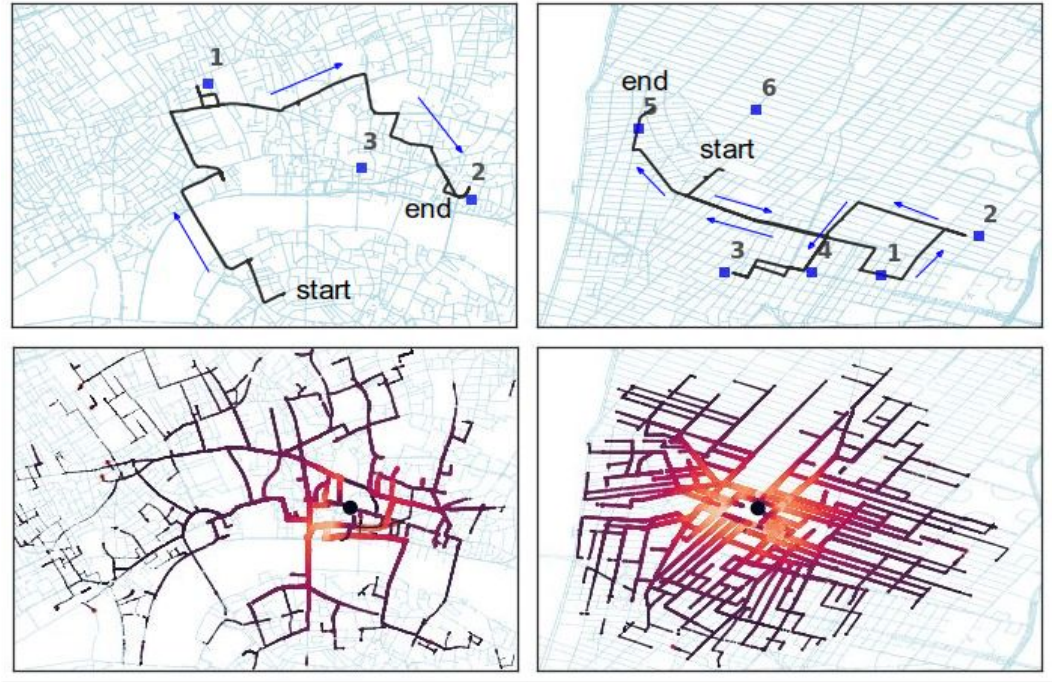
How well does it do?

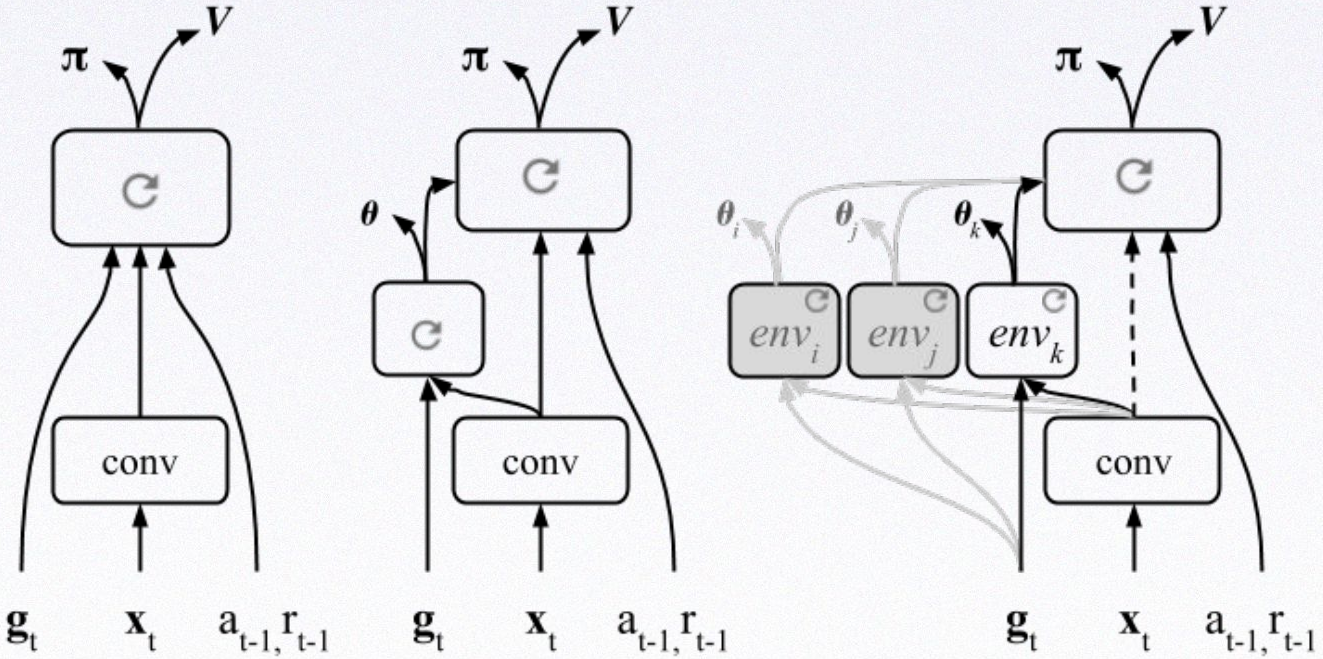




How well does it do?



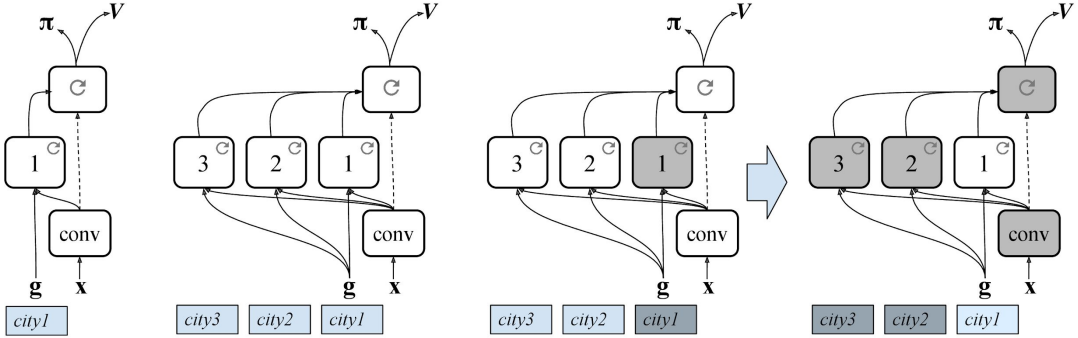




a. *GoalNav* agent b. *CityNav* agent c. *MultiCityNav* agent

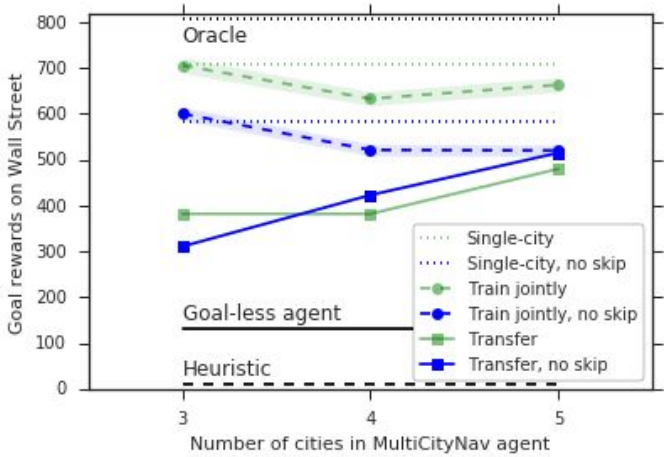
Multi-city modular transfer

Given a sequence of cities (regions of NYC), compare the following



a. b. c.

single **joint** **modular transfer**



- **Successful navigation in target city, even though the convnet and policy LSTM are frozen and only the goal LSTM is trained.**
- Moreover, we note that the transfer success is correlated to number of cities seen during pre-training.

video here:
https://sites.google.com/corp/view/streetlearn

Challenges for deep RL with real robots

Top 6 challenges for deep RL with real robots

1. **Perception**
    - Real robots see the world with real cameras - how sensitive are RL policies to shadows, motion blur, viewpoint change, and clutter? (spoiler: very)
    - Deep learning standard approach is to pre-train on ImageNet - is this applicable?

Top 6 challenges for deep RL with real robots

1. **Perception**
2. **Simulation**
    - Simulation should be an invaluable asset, used to vet algorithms, search hyperparameters, pre-train models, etc.
    - However simulators are ‘doomed to succeed’, making their usage potentially misleading, or even distracting.
    - Simulators model the real world **poorly** in precisely the areas that we need end-to-end robotics to excel in: contacts, complex dynamics, and deformable or articulated objects.
    - Reliance on simulators might actually prevent us from making breakthroughs in robotics.

Top 6 challenges for deep RL with real robots

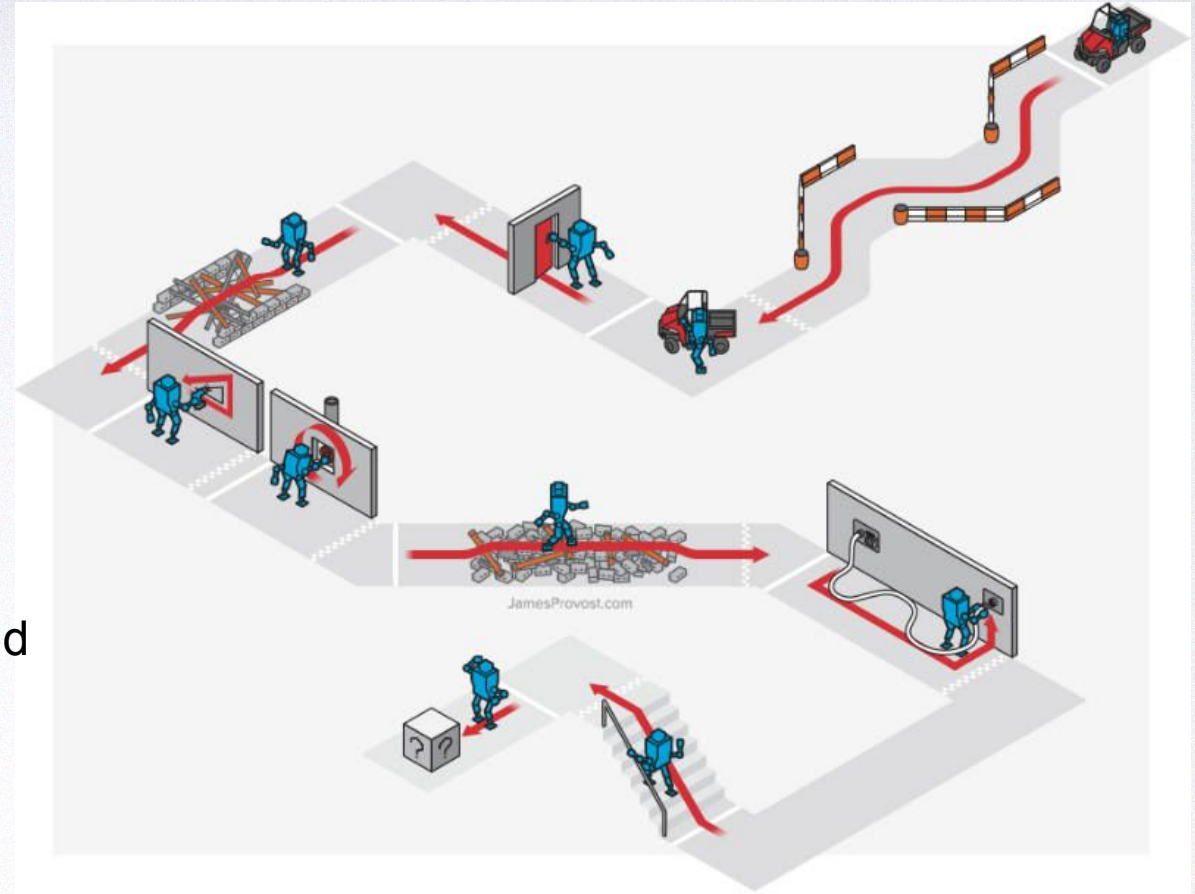
1. **Perception**
2. **Simulation**
3. **Rewards**
    - Hard to engineer, even harder to make foolproof
        - How do I know if the robot reaches the goal? (answer use SLAM)
        - How do I know if the block is lifted? (answer use background subtraction)
        - How do I know if the legos are stacked? (answer train a nn classifier)
    - Hard to learn from; the reward can be arbitrarily sparse!
    - Easy to hack by a smart agent - tipping a brick up a corner or tossing off the table, circling the goal, etc

Top 6 challenges for deep RL with real robots

1. **Perception**
2. **Simulation**
3. **Rewards**
4. **Task scaling and complexity**
    - We don’t want robots that can perform simple short tasks
    - There are very few deep RL methods that attempt to learn long, complex, multiparted tasks (and none on real robots).
    - Building a piece of furniture (even IKEA) or cleaning a room is far beyond current art.

DARPA Robot Challenge Final:

1. Drive a car (!)
2. Open a door
3. Cross a rubble field
4. Use a power tool
5. Turn a crank
6. Cross another rubble field
7. Plug in a cable/hose
8. Climb stairs
9. Mystery challenge



https://www.roboticsbusinessreview.com

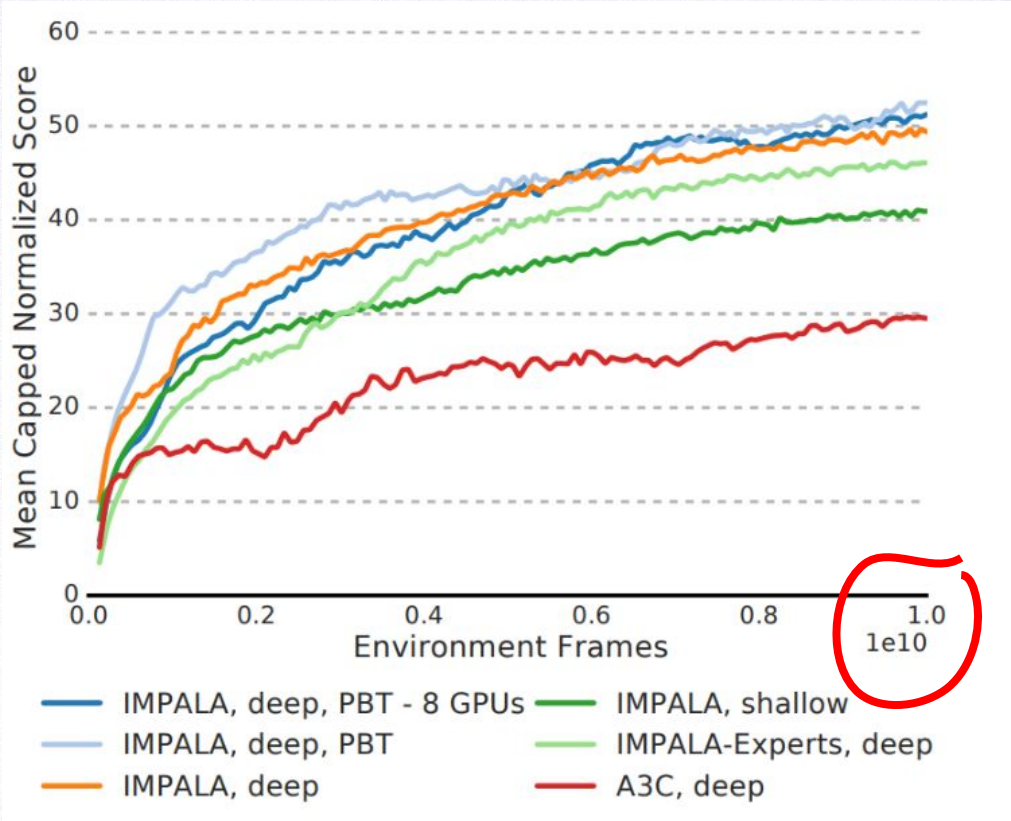
Top 6 challenges for deep RL with real robots

1. **Perception**
2. **Simulation**
3. **Rewards**
4. **Task scaling and complexity**
5. **Data efficiency**

Top 6 challenges for deep RL with real robots

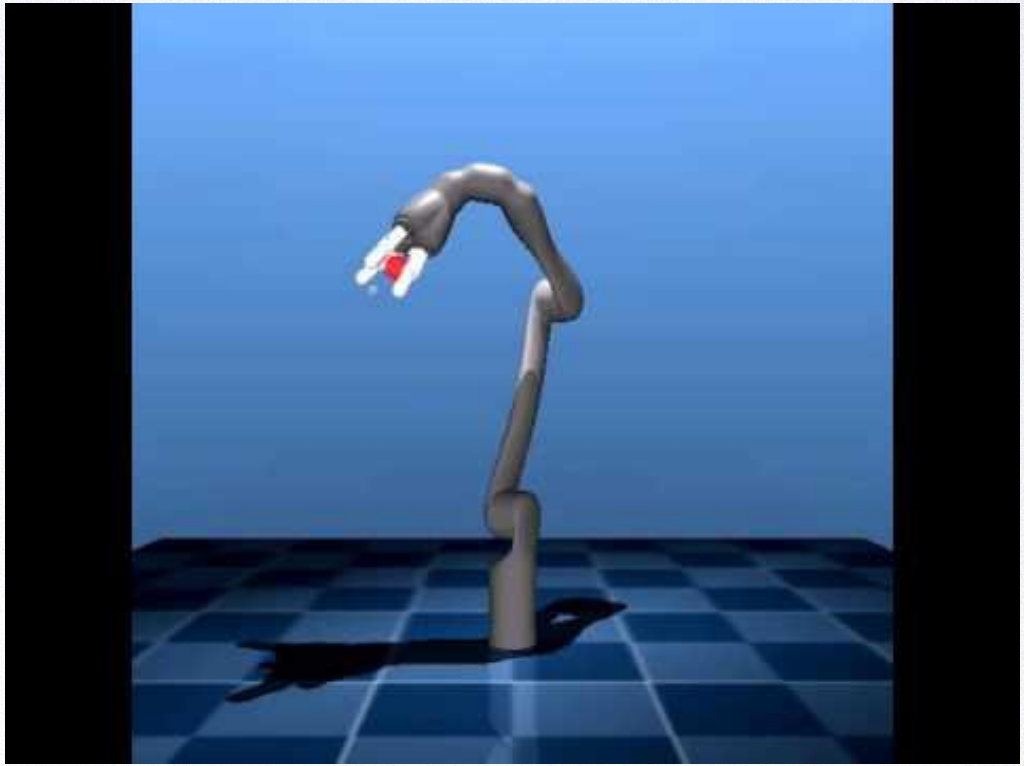
1. **Perception**
2. **Simulation**
3. **Rewards**
4. **Task scaling and complexity**
5. **Data efficiency**
   - Example 1: Learning to play a suite of 30 games

[IMPALA: Importance Weighted Actor-Learner Architectures, Espeholt, Soyer, Munos et al. 2018]



Top 6 challenges for deep RL with real robots

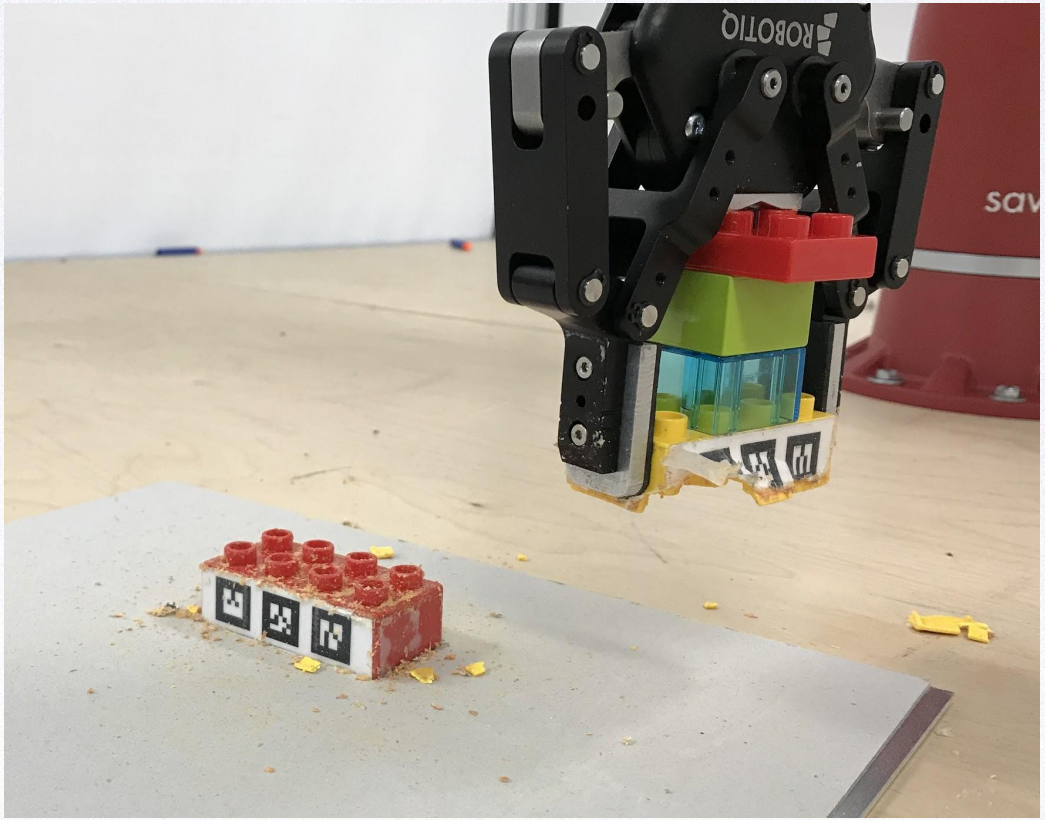
1. **Perception**
2. **Simulation**
3. **Rewards**
4. **Task scaling and complexity**
5. **Data efficiency**
   - Example 2: learning the kinematic chain → **half a day** in simulation, **2 weeks** on the real robot.



https://www.youtube.com/watch?v=u0M3PvTgTcE

Top 6 challenges for deep RL with real robots

1. **Perception**
2. **Simulation**
3. **Rewards**
4. **Task scaling and complexity**
5. **Data efficiency**
   - Example 3: **powdered Legos.**



Top 6 challenges for deep RL with real robots

1. **Perception**
2. **Simulation**
3. **Rewards**
4. **Task scaling and complexity**
5. **Data efficiency**
6. **The robots themselves**
    - Roboticists have focused on building robots that are precise
    - But robots need to be robust, sensorised, and low-level controllable
    - Moreover, the field needs common platforms and benchmarks

Top 6 challenges for deep RL with real robots

1. **Perception**
2. **Simulation**
3. **Rewards**
4. **Task scaling and complexity**
5. **Data efficiency**
6. **The robots themselves**

Quick survey: Which do you think is the biggest challenge?

Possible solutions and research directions

Sim2Real: Transfer policies from sim to real robot

challenges: simulation, data efficiency

Seems straightforward, but natural images and unmodeled dynamics make this challenging. Small variations break transfer.

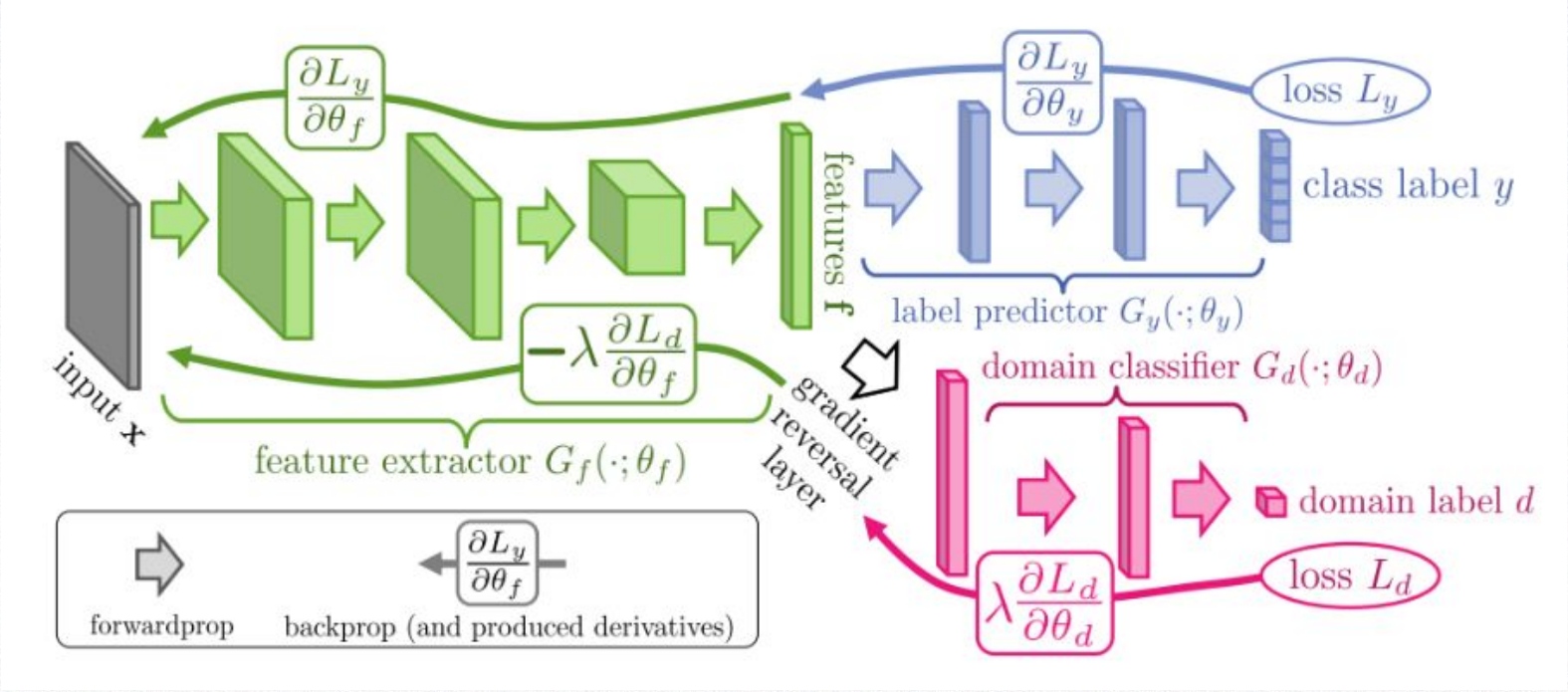
- First approach: Domain variations
  - CAD2RL: Real Single-Image Flight without a Single Real Image (Sadeghi & Levine, 2016)
  - Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World (Tobin et al, 2017)
  - Transferring End-to-End Visuomotor Control from Simulation to Real World for a Multi-Stage Task (James, Davison, Johns, 2017)
  - Sim-to-Real Transfer of Robotic Control with Dynamics Randomization (Peng et al, 2017)

Transfer policies from sim to real robot (sim2real)

["Transferring End-to-End Visuomotor Control from Simulation to Real World", James, Davison, Johns CoRL 2017]

Transfer policies from sim to real robot (sim2real)

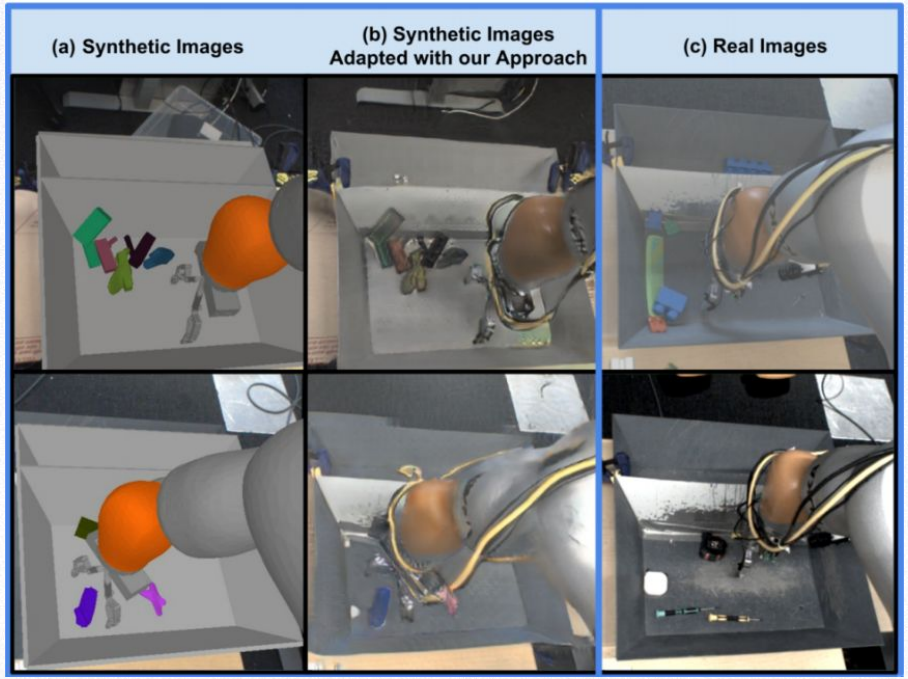
- Second approach: Domain Adversarial Nets



["Domain-Adversarial Training of Neural Networks" Ganin et al, JMLR 2016]

Transfer policies from sim to real robot (sim2real)

- Third approach: Grasp GAN: Use a conditional GAN to adapt the sim images to look like real images.



["Using Simulation and Domain Adaptation to Improve Efficiency of Deep Robotic Grasping" Bousmalis et al, ICRA 2018]

Intrinsic motivation: Exploration, Curiosity, Surprise, etc.

challenges: rewards, data efficiency

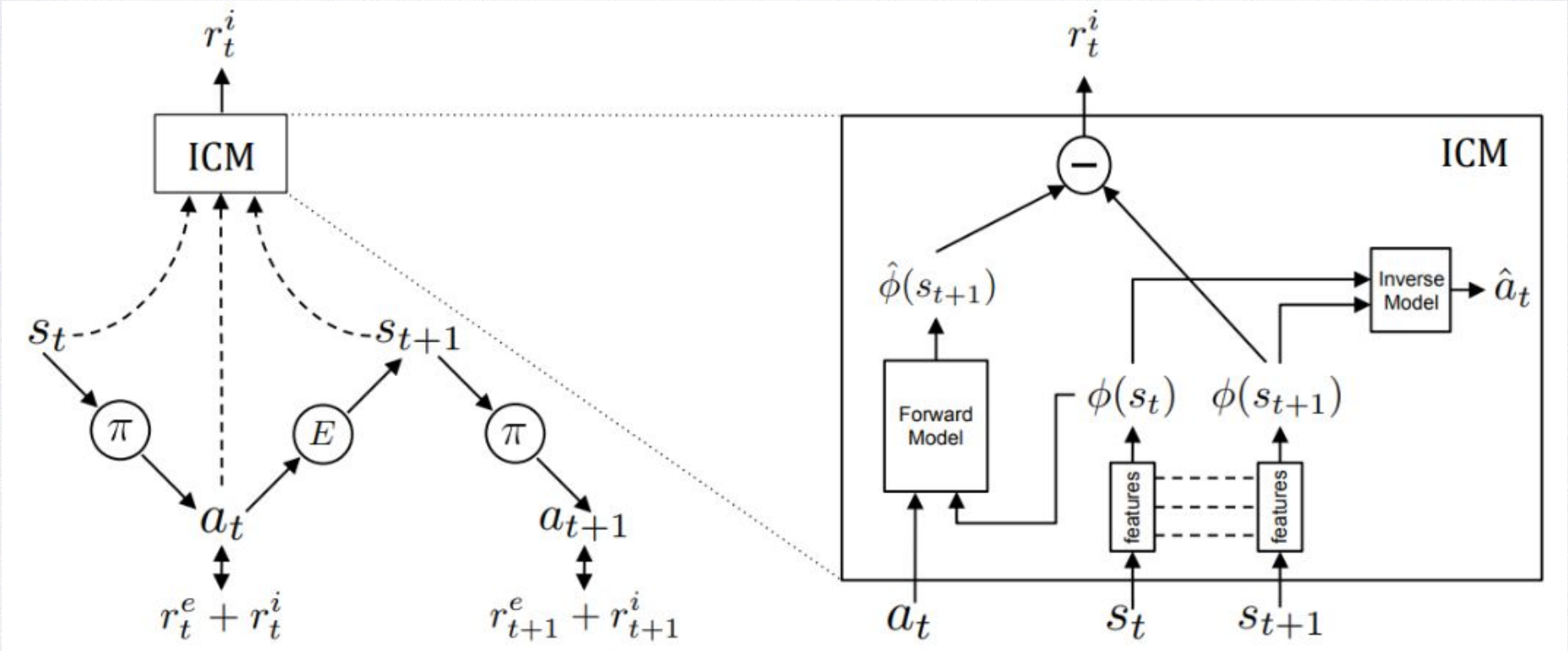
Learning solely from external rewards is unnatural and unnecessary. Recent research has sought to encode intrinsic motivations such as curiosity in a general way.

- Intrinsic motivation systems for autonomous mental development (Oudeyer, Kaplan, Hafner, 2007)
- Unifying count-based exploration and intrinsic motivation (Bellemare et al, 2016)
- Curiosity-driven Exploration by Self-supervised Prediction (Pathak et al, 2017)
- Learning to play with intrinsically motivated self-aware agents (Haber et al, 2018)

→ *Auxiliary tasks and losses also fall in this solution category*

Intrinsic motivation: Exploration, Curiosity, Surprise, etc.

challenges: rewards, data efficiency

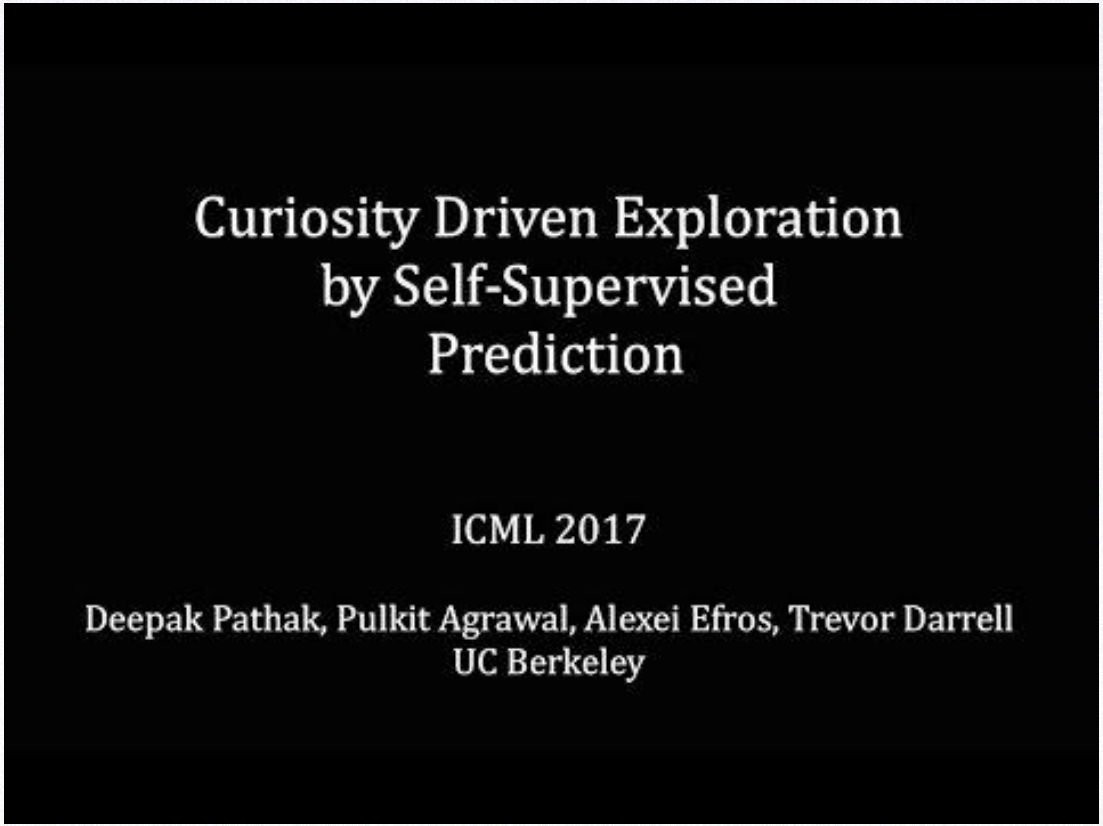


DeepMind

["Curiosity-driven Exploration by Self-supervised Prediction", Pathak, Agrawal, Efros, Darrell, 2017)

Intrinsic motivation: Exploration, Curiosity, Surprise, etc.

challenges: rewards, data efficiency



["Curiosity-driven Exploration by Self-supervised Prediction", Pathak, Agrawal, Efros, Darrell, 2017)

Continual (lifelong) learning and Meta-learning

challenges: simulation, data efficiency, task scaling and complexity

Continual learning implies a learning framework or algorithm that enables **forward transfer** (learning new tasks faster), and **not forgetting** (retaining previous skills). Simulation could be considered just another task.

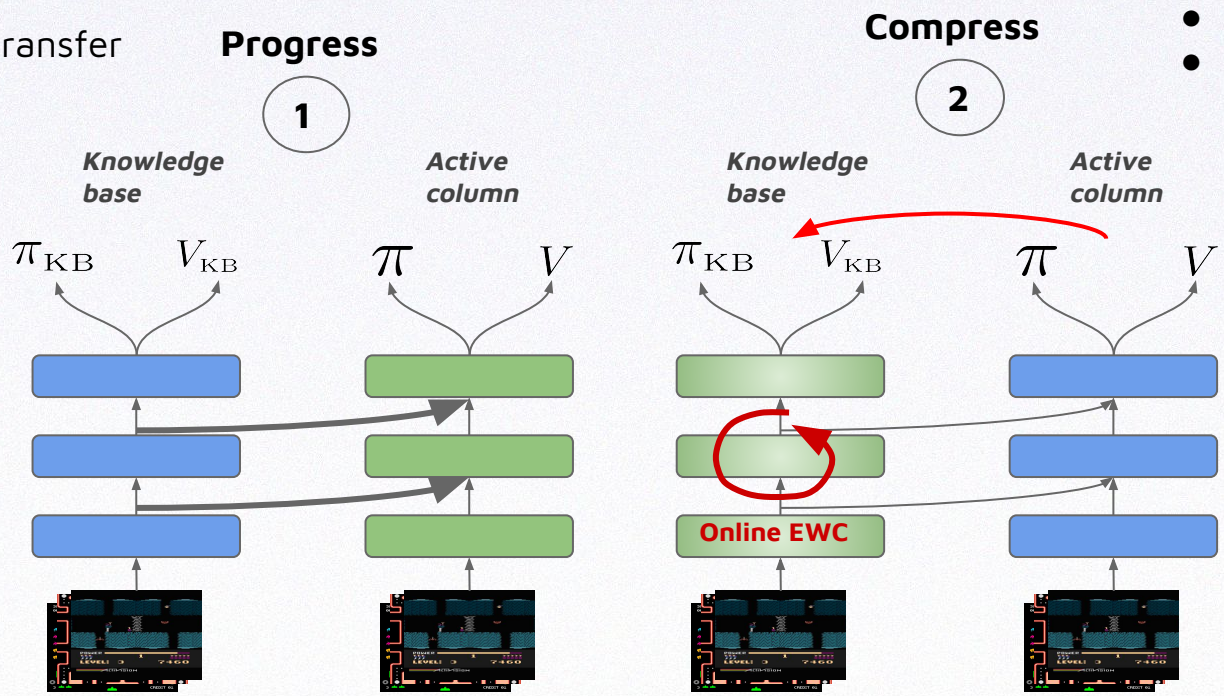
Theoretically, continual learning could allow skill composition, thus task scaling and complexity.

- Lifelong robot learning (Thrun, 1995)
- CHILD: A first step towards continual learning (Ring, 1997)
- Progressive Neural Nets (Rusu et al, 2016) and Progress & Compress (Schwarz et al, 2018)
- Model Agnostic Meta-Learning (Finn, Abbeel, Levine, 2017)
- Continuous Adaptation via Meta-Learning in Nonstationary and Competitive Environments (Al-Shedivat et al, 2017)

Continual (lifelong) learning and Meta-learning

challenges: simulation, data efficiency, task scaling and complexity

- Forward transfer
- No Forgetting
- Scalability

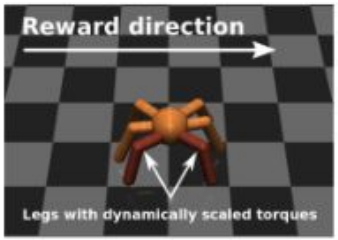


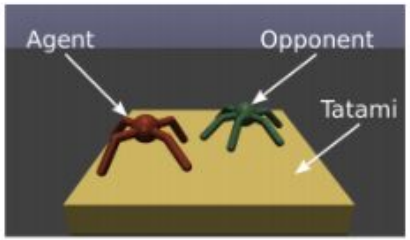
DeepMind

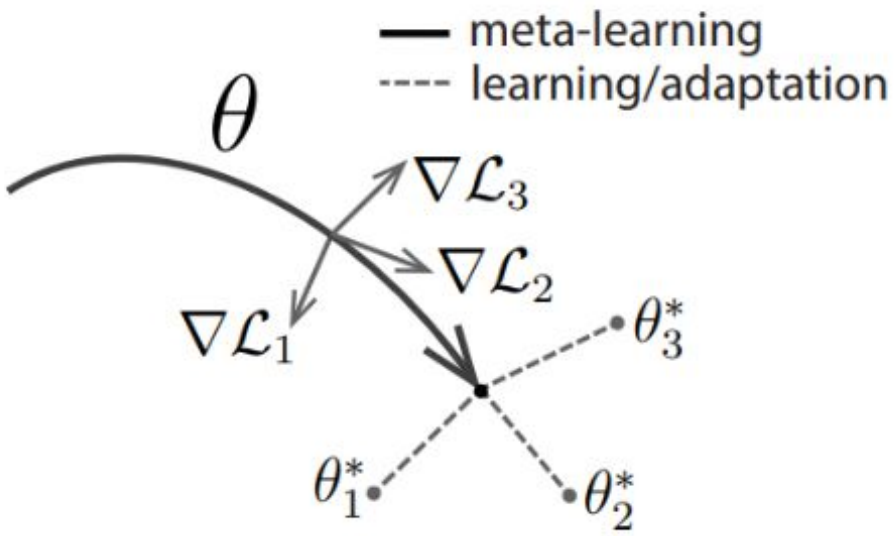
["Progress & Compress: A scalable framework for continual learning", Schwarz et al, 2018)

Continual (lifelong) learning and Meta-learning

challenges: simulation, data efficiency, task scaling and complexity







["Model Agnostic Meta-Learning for Fast Adaptation of Deep Networks" Finn, Abbeel, Levine, 2017]
["Continuous Adaptation via Meta-Learning in Nonstationary and Competitive Envs", Al-Shedivat et al, 2017]

Hierarchical RL

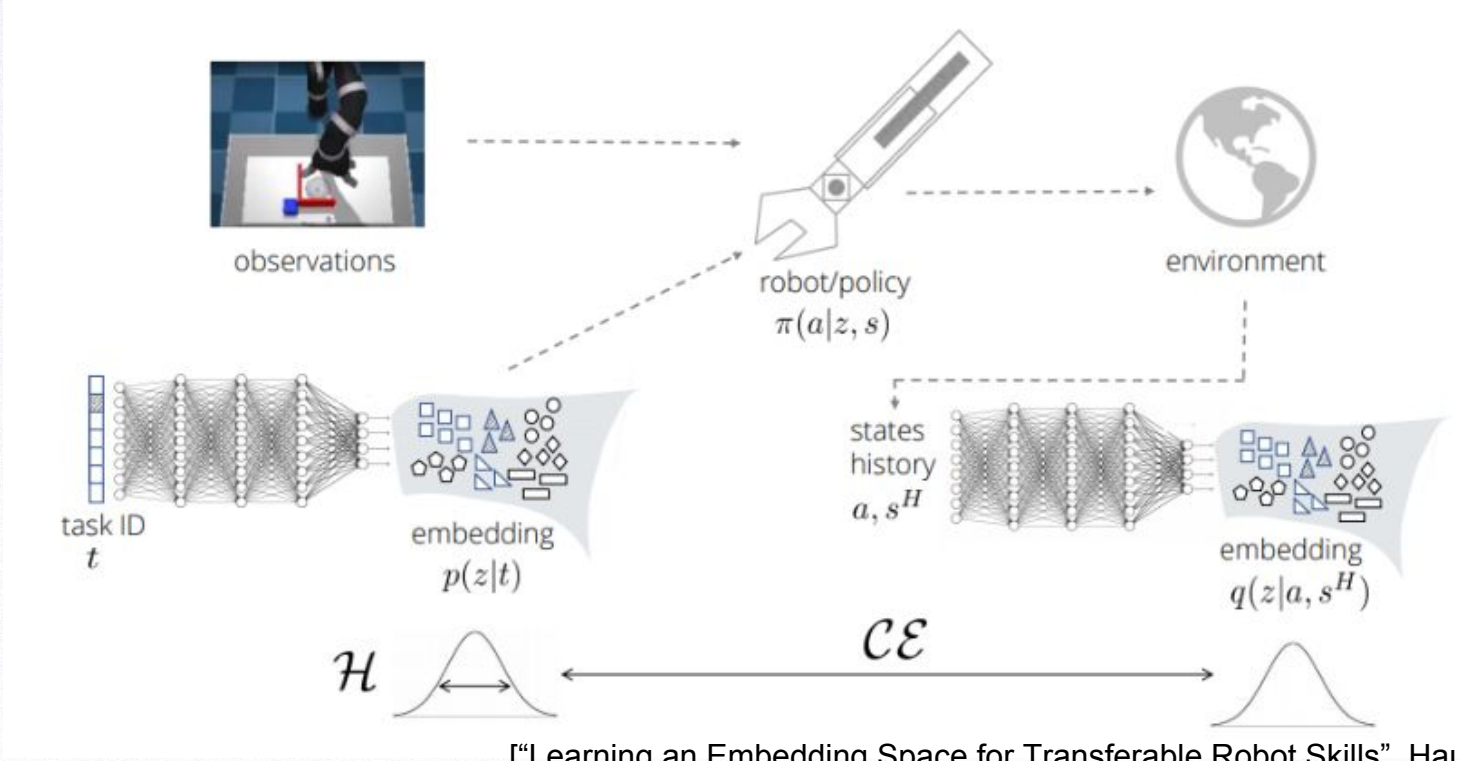
challenges: rewards, task scaling and complexity

Similarly to how convolutional nets leverage spatial pooling and convolutions to build abstraction, the objective of HRL is to build temporal abstraction. This enables long-term credit assignment, high-level planning, skill composition, etc.

- FeUdal Networks for Hierarchical Reinforcement Learning (Vezhnevets et al, 2017)
- Neural Task Programming: Learning to Generalize Across Hierarchical Tasks (Xu et al, 2017)
- Learning an Embedding Space for Transferable Robot Skills (Hausman et al, 2018)

Hierarchical RL

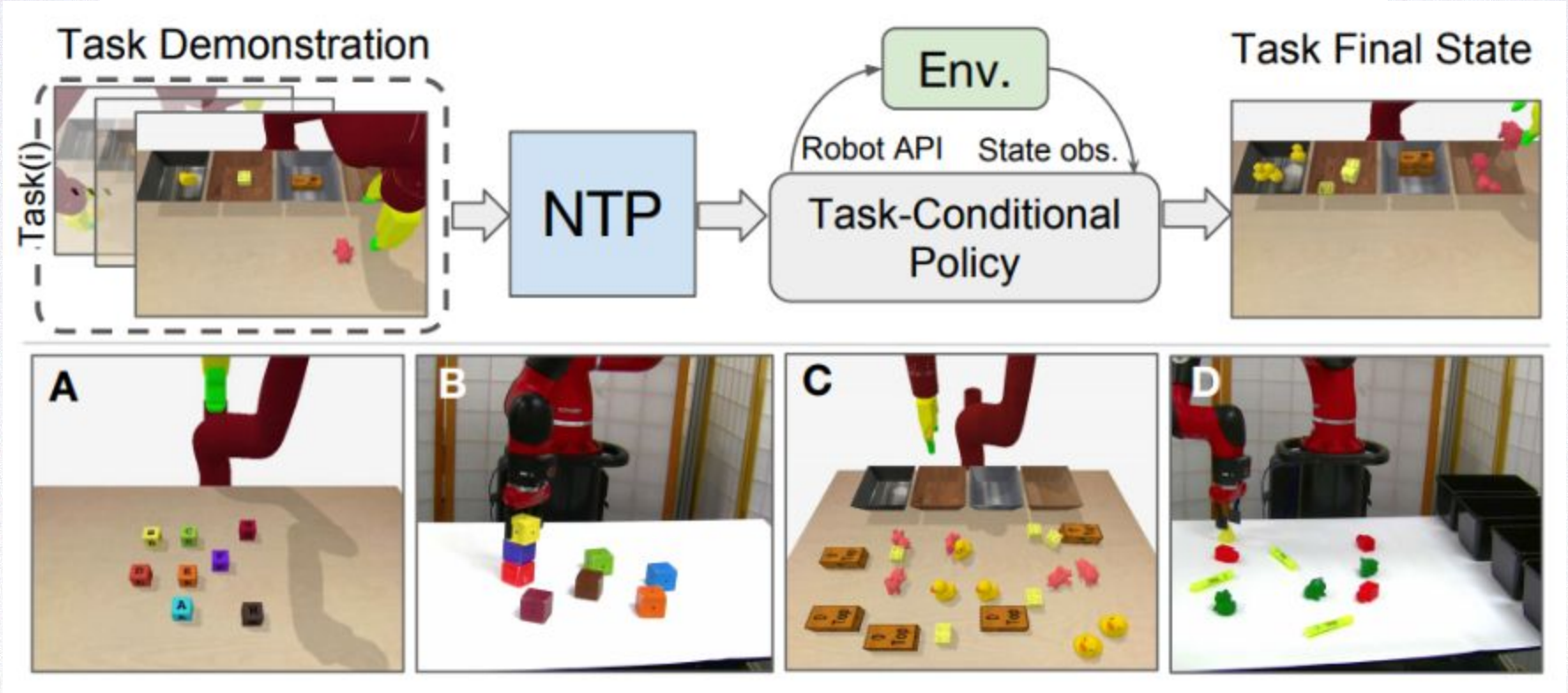
challenges: rewards, task scaling and complexity



["Learning an Embedding Space for Transferable Robot Skills", Hausman, 2018]

Hierarchical RL

challenges: rewards, task scaling and complexity



["Neural Task Programming: Learning to Generalize Across Hierarchical Tasks" (Xu et al, 2017)]

Learning from Demonstrations/Imitation learning

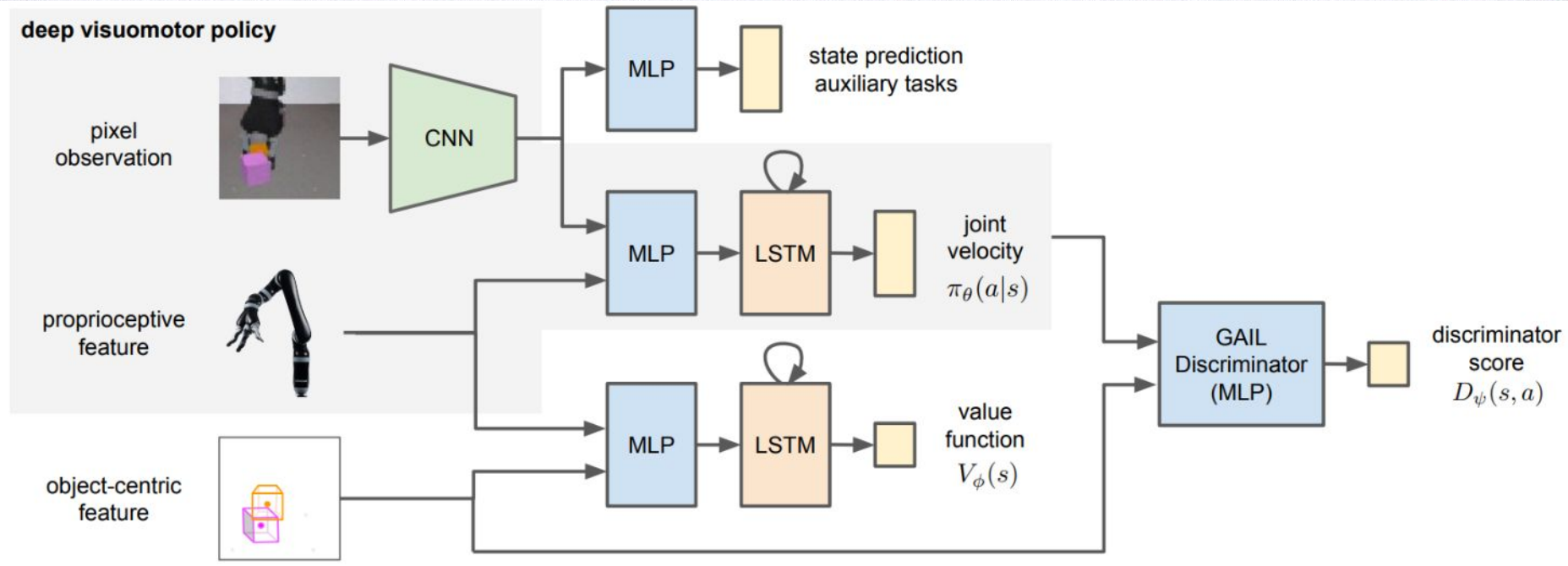
challenges: rewards, data efficiency, task scaling and complexity

Human and animal learning relies heavily on imitation learning, and arguably this is a necessary component of any robot learning approach.

- Neural Task Graphs: Generalizing to Unseen Tasks from a Single Video Demonstration (Huang et al, 2018)
- Reinforcement and Imitation Learning for Diverse Visuomotor Skills, (Zhu et al, 2018)
- Time-Contrastive Networks: Self-Supervised Learning from Video, (Sermanet et al, 2017)

Learning from Demonstrations/Imitation learning

challenges: rewards, data efficiency, task scaling and complexity



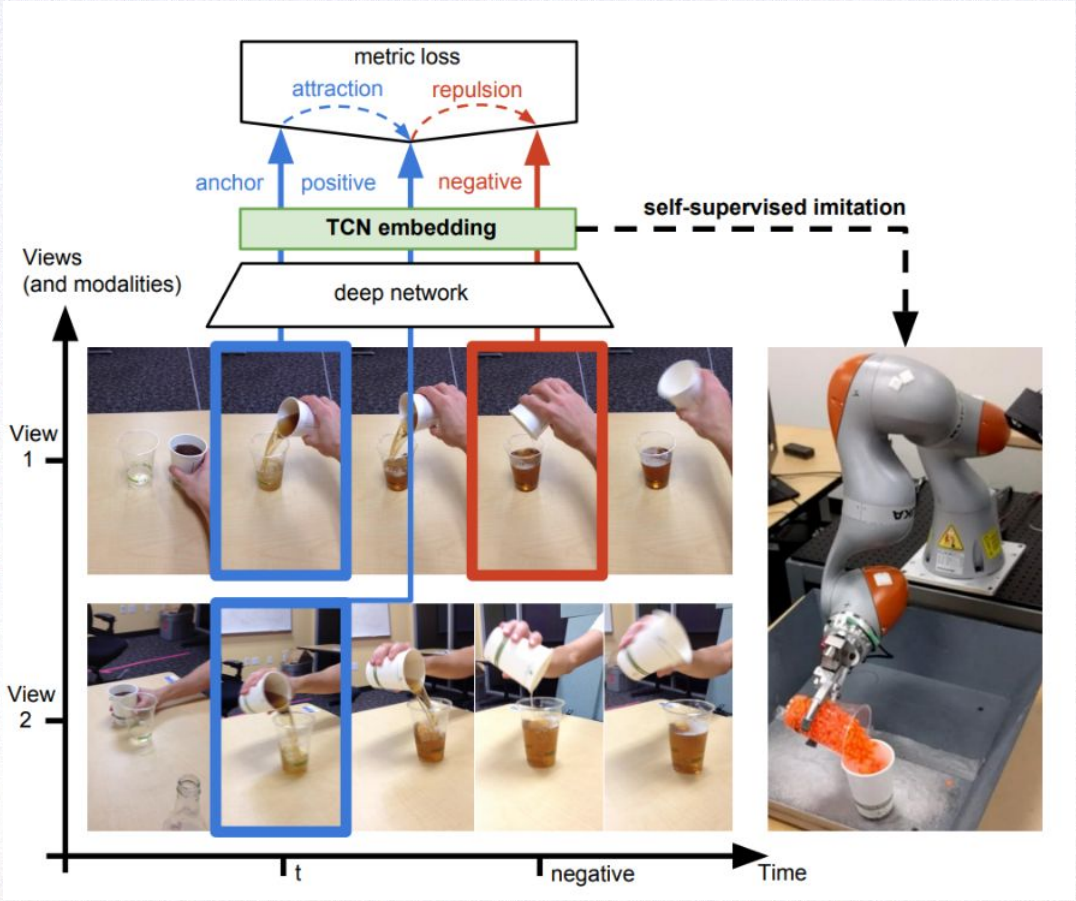
["Reinforcement and Imitation Learning for Diverse Visuomotor Skills", (Zhu et al, 2018)]

Learning from Demonstrations/Imitation learning

https://www.youtube.com/watch?v=IKB_G2k_ECE

["Reinforcement and Imitation Learning for Diverse Visuomotor Skills", (Zhu et al, 2018)]

Learning from Demonstrations/Imitation learning

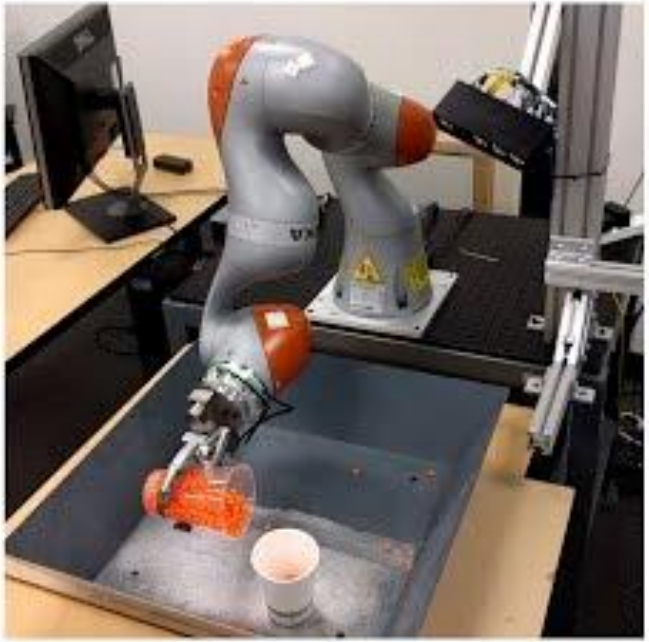


["Time-Contrastive Networks: Self-Supervised Learning from Video", (Sermanet et al, 2017)]

Learning to imitate, from video, without supervision

3rd-person observation

Imitating



Learned policy

Possible solutions and research directions

Sim2Real: Transfer policies from sim to real robot

- challenges: simulation, data efficiency

Intrinsic motivation: Exploration, Curiosity, Surprise, etc.

- challenges: rewards, data efficiency

Continual (lifelong) learning and Meta-learning

- challenges: simulation, data efficiency, task scaling and complexity

Hierarchical RL

- challenges: rewards, task scaling and complexity

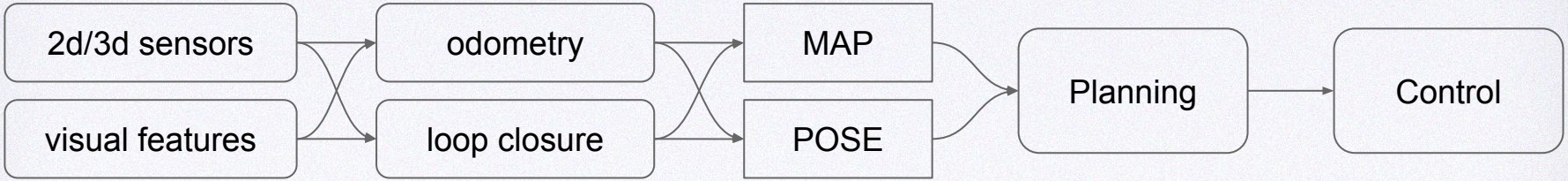
Learning from Demonstrations/Imitation learning

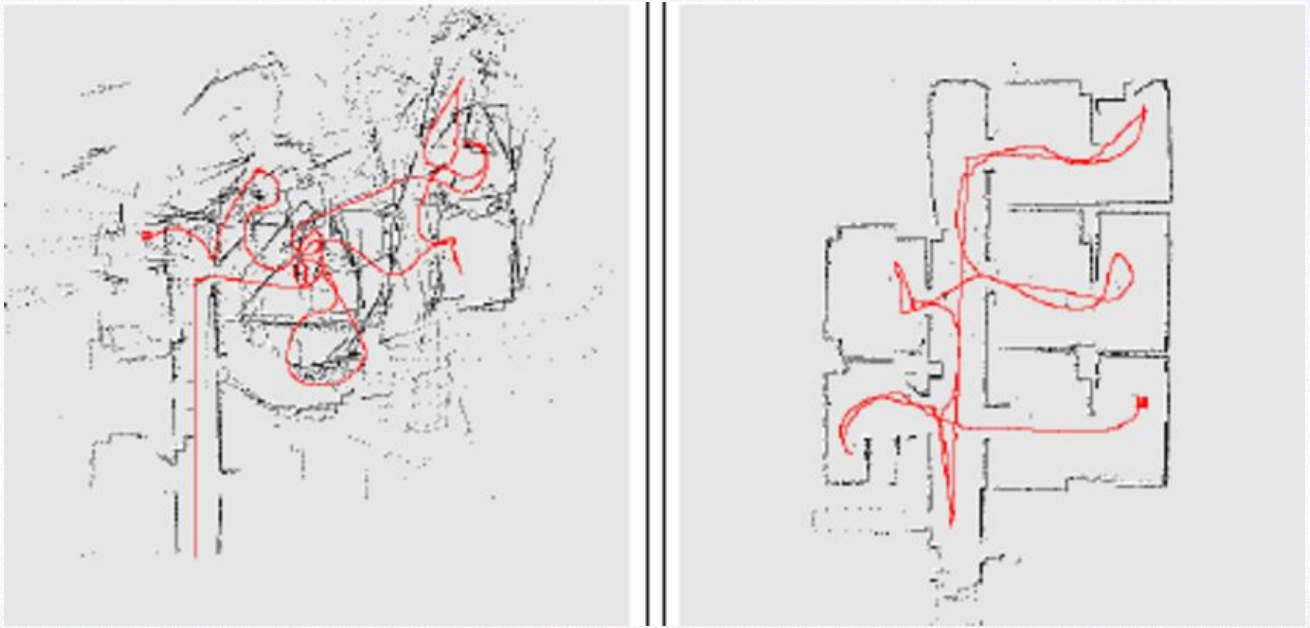
- challenges: rewards, data efficiency, task scaling and complexity

Quick survey: Which direction is the most important/powerful/profitable?

Thank you!

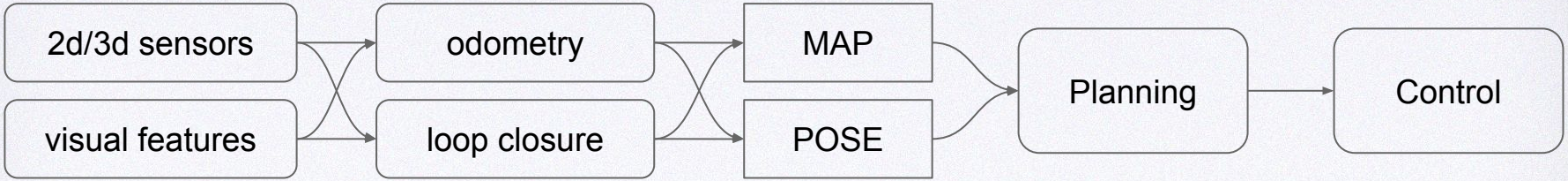
SLAM - Simultaneous Localisation and Mapping

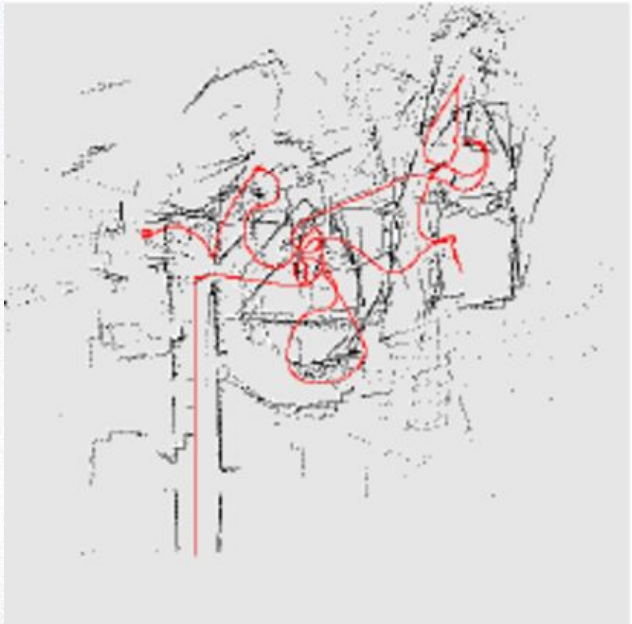


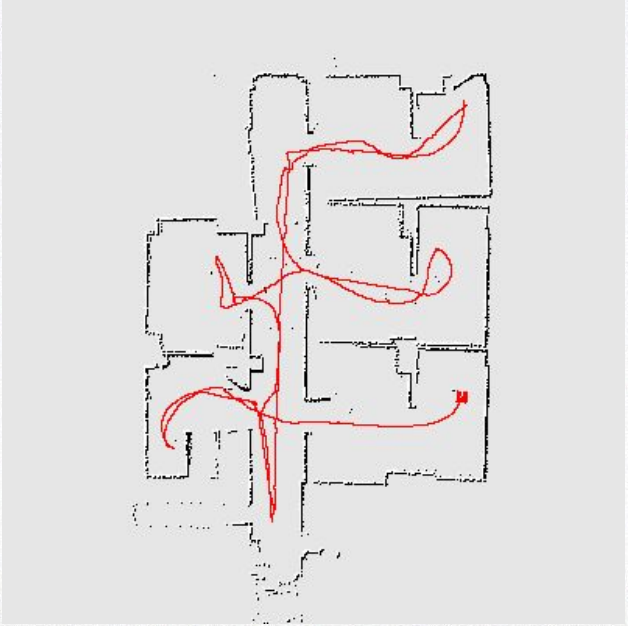


A Real-Time Algorithm for Mobile Robot Mapping, Thrun et al, 2000

SLAM - Simultaneous Localisation and Mapping







A Real-Time Algorithm for Mobile Robot Mapping, Thrun et al, 2000