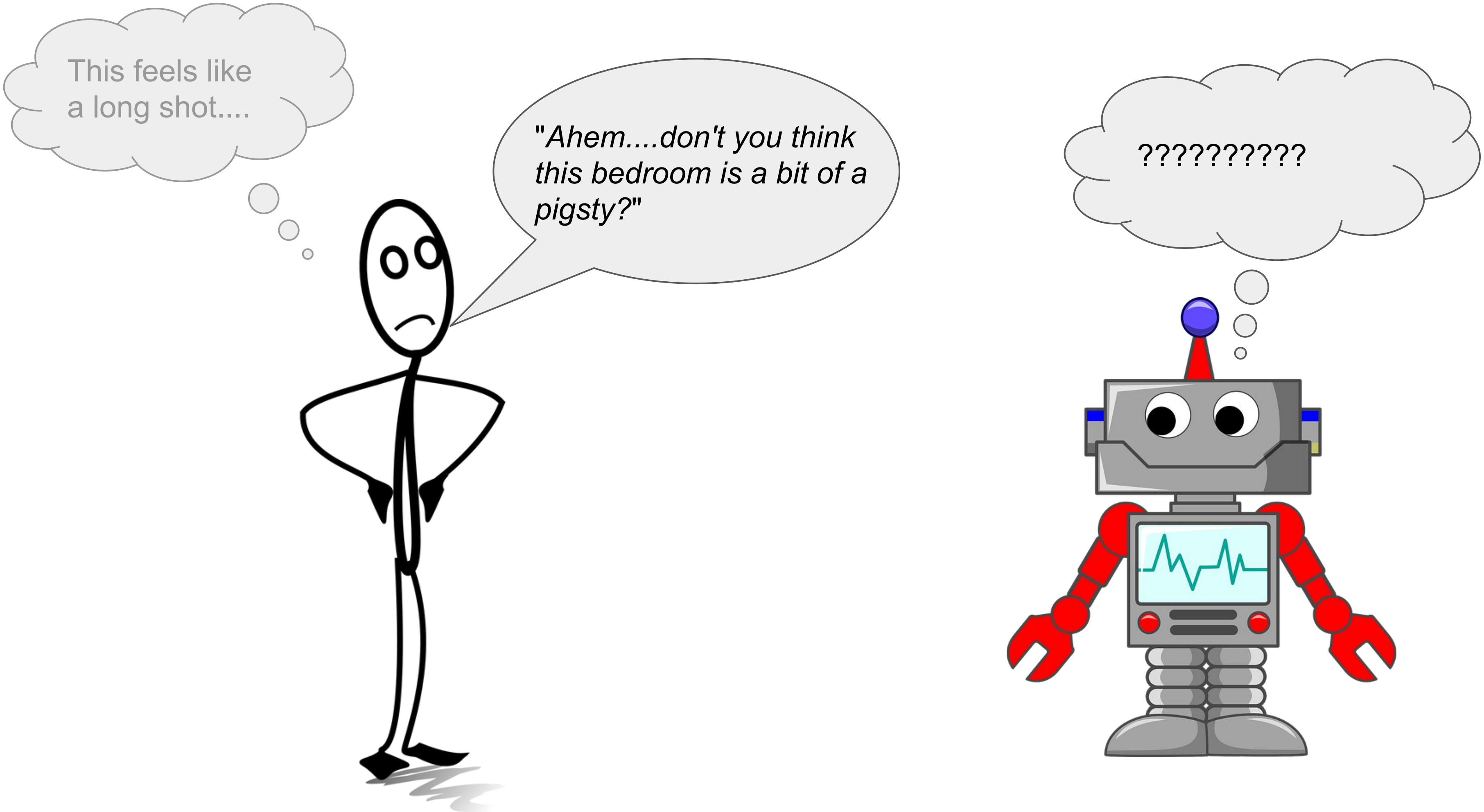
Situated Language Learning with Policy Gradients

L14. Felix Hill

DeepMind

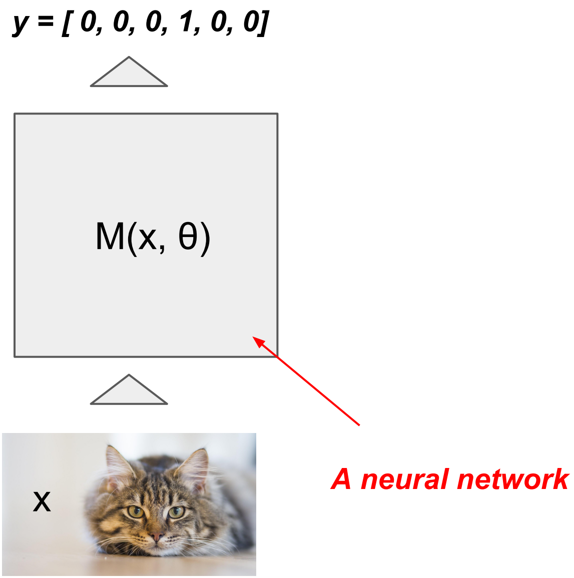

This feels like
a long shot....
"Ahem....don't you think this bedroom is a bit of a pigsty?"
??????????

Reinforcement learning for games

- General-purpose learning algorithm
- Works for many different problems
- Teaches us about the game

Mnih et al, 2015

Supervised learning



Supervised learning

Find weights θ to minimize e.g.

$$\sum_{x,y \in D} -\log(g(x,y,\theta))$$

A dataset D of (x,y) input-output pairs

where $$g(x,y,\theta) = M(x,\theta)|_y = P(y|x,\theta)$$

y = [0, 0, 0, 1, 0, 0]

M(x, θ)

Supervised learning

Find theta to minimize e.g.

$$\sum_{x,y \in D} -\log(g(x,y,\theta))$$ = C(x, y, θ)

The "cost" function

A dataset D of (x,y) input-output pairs

where $$g(x,y,\theta) = M(x,\theta)|_y = P(y|x,\theta)$$

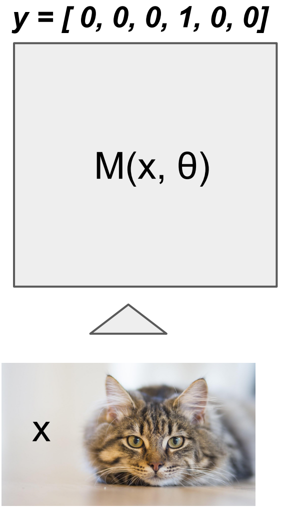
y = [0, 0, 0, 1, 0, 0]

M(x, θ)

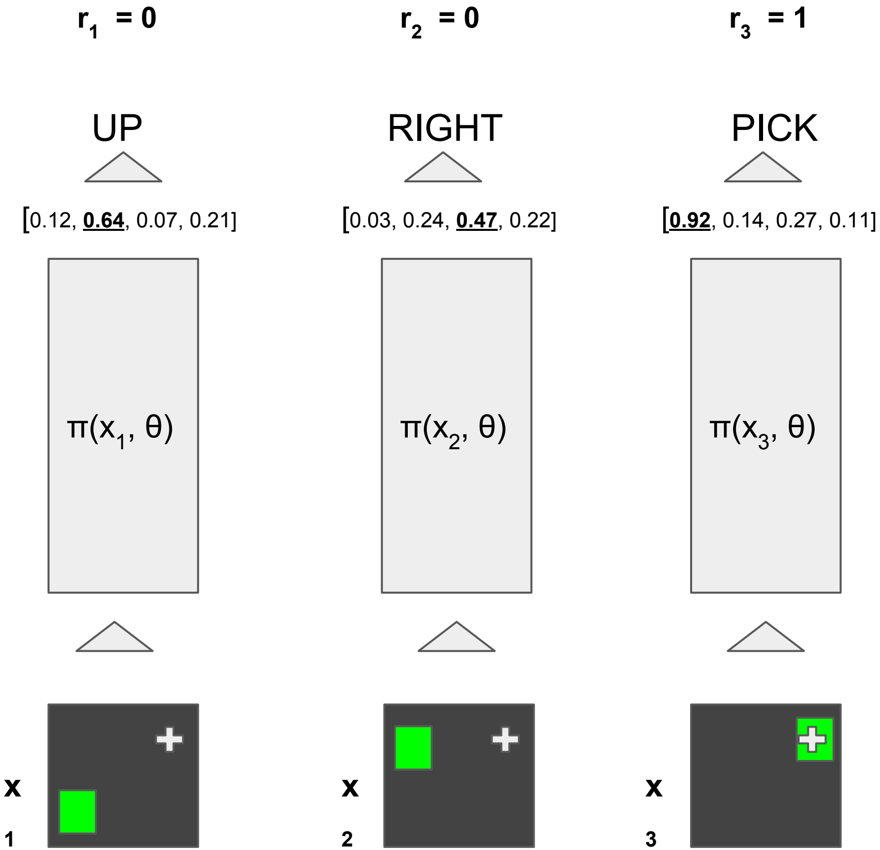
Improve M (θ values) by gradient descent

$$\theta \rightarrow \theta + \alpha \nabla_\theta C(x,y,\theta)$$

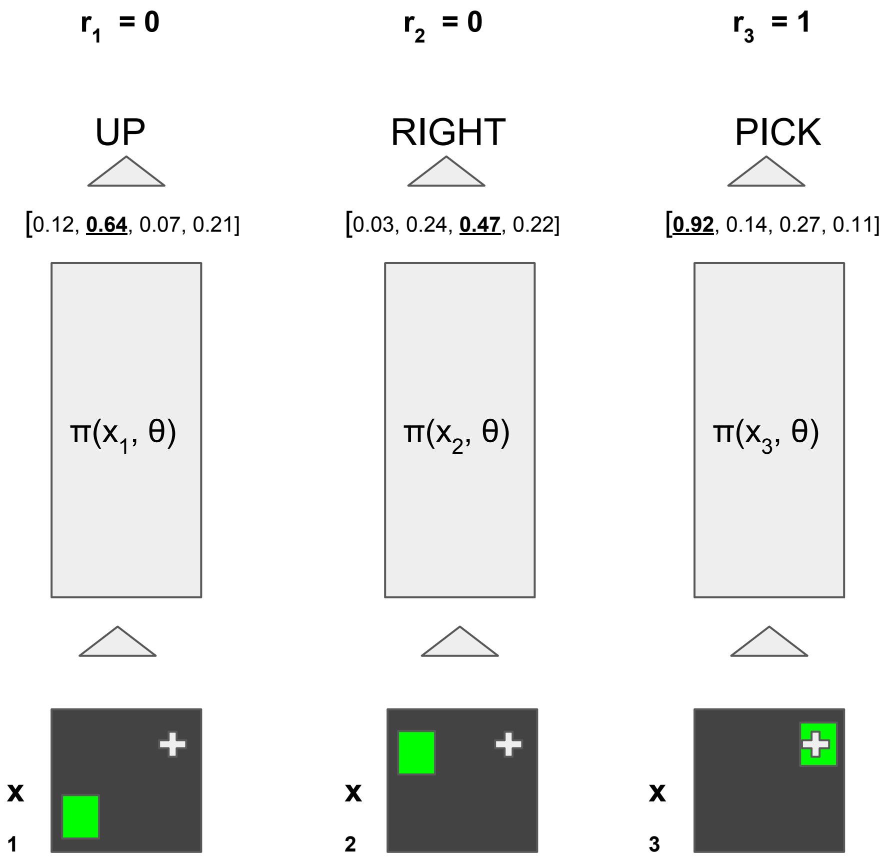
Supervised learning



Reinforcement learning

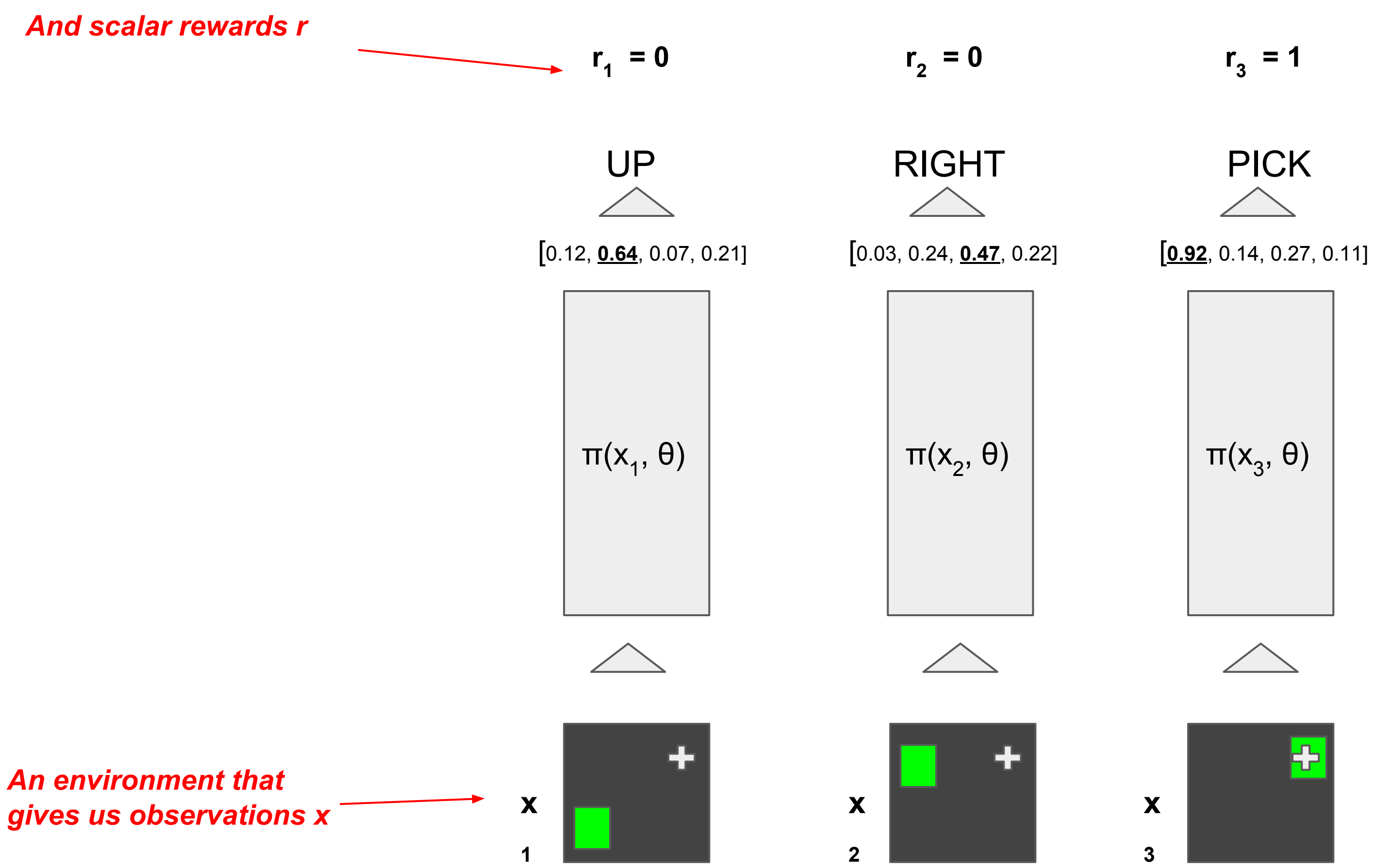


Reinforcement learning



An environment that gives us observations x

Reinforcement learning



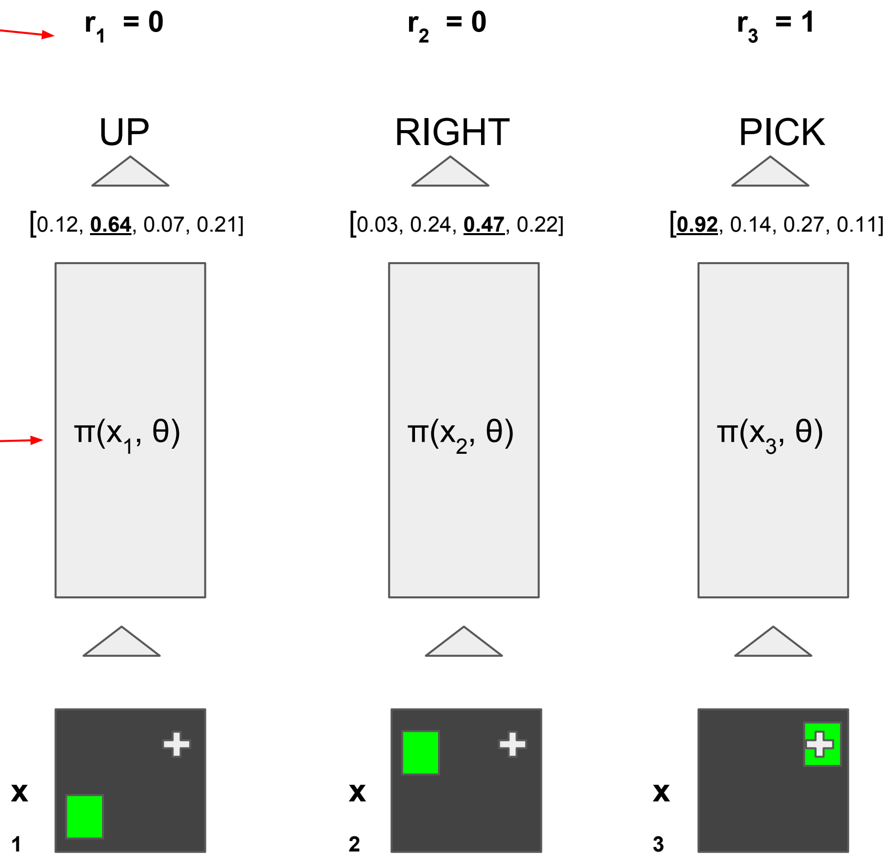
Reinforcement learning

And scalar rewards r

The "policy" π predicts action probabilities given observations

- ***Like the supervised M***

An environment that gives us observations x



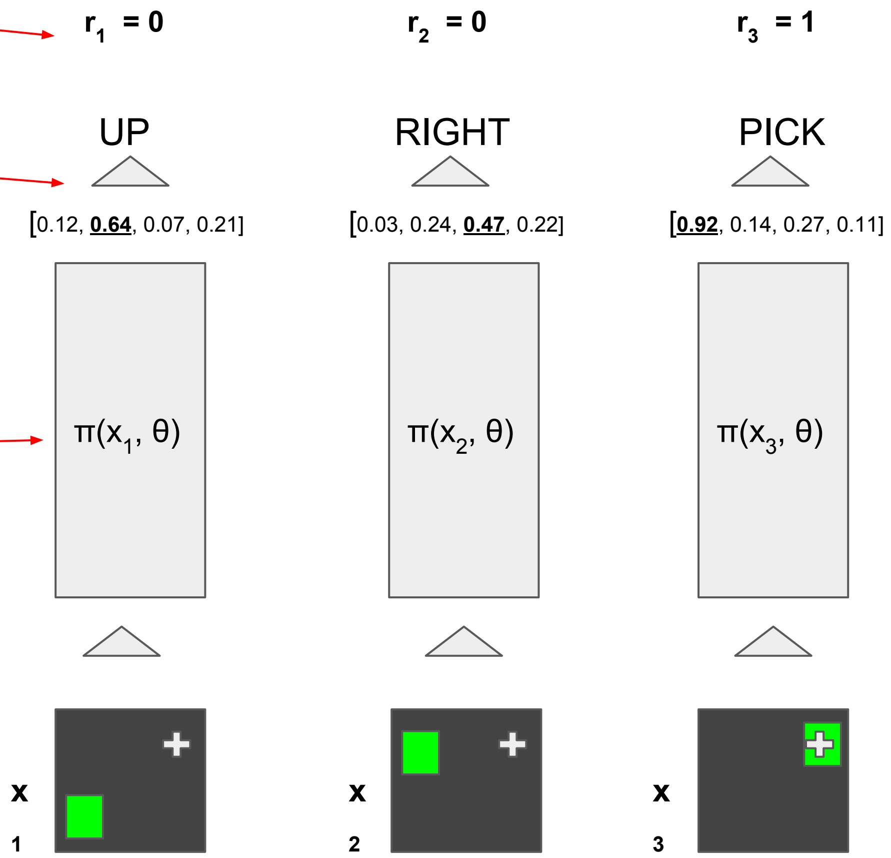
Reinforcement learning

And scalar rewards r

Only rewards to learn from - no guarantee 'actions' are correct

The "policy" π predicts actions given observations

An environment that gives us observations x



Gradient methods for Reinforcement Learning?

We want to optimise

$$J(\theta) = \mathbb{E}_{\pi(\theta)}(R)$$

where

$$R = \sum_{t=1}^{k} \gamma^t r_t$$

Find the policy weights that give us the highest expected return

All the reward I got from the environment

Gradient methods for Reinforcement Learning?

We want to optimise $$J(\theta) = \mathbb{E}_{\pi(\theta)}(R)$$ where $$R = \sum_{t=1}^{k} \gamma^t r_t$$

If we knew $\nabla_\theta J(\theta)$

we could just do gradient ascent!

$$\theta \to \theta + \alpha \nabla_\theta J(\theta)$$

Gradient methods for Reinforcement Learning?

We want to optimise

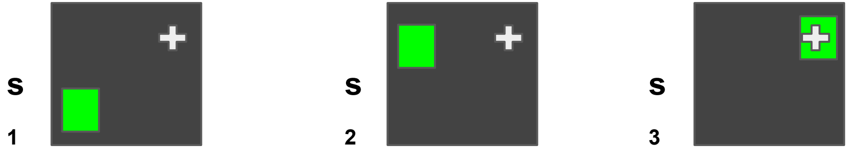
$$J(\theta) = \mathbb{E}_{\pi(\theta)}(R)$$

where

$$R = \sum_{t=1}^{k} \gamma^t r_t$$

Trajectory

$$\tau = \{s_1 \cdots s_k,\}$$



Gradient methods for Reinforcement Learning?

We want to optimise

$$J(\theta) = \mathbb{E}_{\pi(\theta)}(R)$$

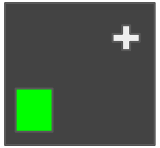
where

$$R = \sum_{t=1}^{k} \gamma^t r_t$$

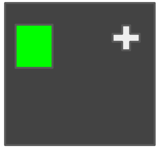
Trajectory

$$\tau = \{s_1 \cdots s_k, \}$$

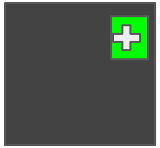
s_1



s_2



s_3



Return

$$R(\tau) = \sum_{t=1}^{k} \gamma^t r_t$$

For a stationary environment:

A given **trajectory** -> unique **well-defined return**

Gradient methods for Reinforcement Learning?

We want to optimise

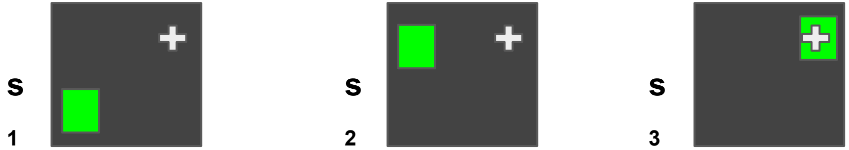
$$J(\theta) = \mathbb{E}_{\pi(\theta)}(R)$$

where

$$R = \sum_{t=1}^{k} \gamma^t r_t$$

Trajectory

$$\tau = \{s_1 \cdots s_k,\}$$



Return

$$R(\tau) = \sum_{t=1}^{k} \gamma^t r_t$$

So condition on trajectories

$$J(\theta) = \mathbb{E}_{\pi(\theta)}(R) = \mathbb{E}_{\tau \in T|\pi}(R(\tau))$$

Estimating a policy gradient in an environment

$$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{\tau|\pi(\theta)}(R(\tau))$$

Estimating a policy gradient in an environment

$$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{\tau|\pi(\theta)}(R(\tau))$$

$$= \nabla_\theta \sum_{\tau \in T} R(\tau) P(\tau|\pi_\theta)$$ ***Definition of expectation***

Space of all possible trajectories

Estimating a policy gradient in an environment

$$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{\tau|\pi(\theta)}(R(\tau))$$

$$= \nabla_\theta \sum_{\tau \in T} R(\tau) P(\tau|\pi_\theta)$$ ***Definition of expectation***

$$= \sum_{\tau \in T} R(\tau) \nabla_\theta P(\tau|\pi_\theta)$$ ***Only P depends on theta***

Estimating a policy gradient in an environment

$$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{\tau|\pi(\theta)}(R(\tau))$$

$$= \nabla_\theta \sum_{\tau \in T} R(\tau) P(\tau|\pi_\theta)$$ ***Definition of expectation***

$$= \sum_{\tau \in T} R(\tau) \nabla_\theta P(\tau|\pi_\theta)$$ ***Only P depends on theta***

$$= \sum_{\tau \in T} R(\tau) \nabla_\theta P(\tau|\pi_\theta) \frac{P(\tau|\pi_\theta)}{P(\tau|\pi_\theta)}$$ ***x 1***

Estimating a policy gradient in an environment

$$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{\tau|\pi(\theta)}(R(\tau))$$

$$= \nabla_\theta \sum_{\tau \in T} R(\tau) P(\tau|\pi_\theta)$$ ***Definition of expectation***

$$= \sum_{\tau \in T} R(\tau) \nabla_\theta P(\tau|\pi_\theta)$$ ***Only P depends on theta***

$$= \sum_{\tau \in T} R(\tau) \nabla_\theta P(\tau|\pi_\theta) \frac{P(\tau|\pi_\theta)}{P(\tau|\pi_\theta)}$$ ***x 1***

$$= \sum_{\tau \in T} R(\tau) \nabla_\theta \log P(\tau|\pi_\theta) P(\tau|\pi_\theta)$$ ***By chain rule*** $$\nabla_\theta \log P(\tau|\pi_\theta) \leftarrow \frac{\nabla_\theta P(\tau|\pi_\theta)}{P(\tau|\pi_\theta)}$$

Estimating a policy gradient in an environment

$$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{\tau|\pi(\theta)}(R(\tau))$$

$$= \nabla_\theta \sum_{\tau \in T} R(\tau) P(\tau|\pi_\theta)$$ ***Definition of expectation***

$$= \sum_{\tau \in T} R(\tau) \nabla_\theta P(\tau|\pi_\theta)$$ ***Only P depends on theta***

$$= \sum_{\tau \in T} R(\tau) \nabla_\theta P(\tau|\pi_\theta) \frac{P(\tau|\pi_\theta)}{P(\tau|\pi_\theta)}$$ ***x 1***

$$= \sum_{\tau \in T} R(\tau) \nabla_\theta \log P(\tau|\pi_\theta) P(\tau|\pi_\theta)$$ ***By chain rule*** $\nabla_\theta \log P(\tau|\pi_\theta) \leftarrow \frac{\nabla_\theta P(\tau|\pi_\theta)}{P(\tau|\pi_\theta)}$

$$= \mathbb{E}_{\tau \in T}(R(\tau) \nabla_\theta \log P(\tau|\pi_\theta))$$ ***Definition of expectation***

Estimating a policy gradient in an environment

$$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{\tau|\pi(\theta)}(R(\tau))$$

The gradient of the objective wrt. policy weights

$$= \nabla_\theta \sum_{\tau \in T} R(\tau) P(\tau|\pi_\theta)$$ **Definition of expectation**

$$= \sum_{\tau \in T} R(\tau) \nabla_\theta P(\tau|\pi_\theta)$$ **Only P depends on theta**

$$= \sum_{\tau \in T} R(\tau) \nabla_\theta P(\tau|\pi_\theta) \frac{P(\tau|\pi_\theta)}{P(\tau|\pi_\theta)}$$ **x 1**

$$= \sum_{\tau \in T} R(\tau) \nabla_\theta \log P(\tau|\pi_\theta) P(\tau|\pi_\theta)$$ **By chain rule** $\nabla_\theta \log P(\tau|\pi_\theta) \leftarrow \frac{\nabla_\theta P(\tau|\pi_\theta)}{P(\tau|\pi_\theta)}$

$$= \mathbb{E}_{\tau \in T}(R(\tau) \nabla_\theta \log P(\tau|\pi_\theta))$$ **Definition of expectation**

A quantity that I can compute by following a trajectory τ

The REINFORCE algorithm

To estimate $\nabla_\theta J(\theta)$ the **gradient of our objective** wrt. the parameters of the policy function.

We can estimate $\mathbb{E}_{\tau \in T}(R(\tau)\nabla_\theta \log P(\tau|\pi_\theta))$

Notice also that

The log probability of following a trajectory

$$\nabla_\theta log(P(\tau|\pi)) = \nabla_\theta \sum_{t=1}^{k} logP(a_t|\pi, s_t)) = \sum_{t=1}^{k} \nabla_\theta logP(a_t|\pi, s_t))$$

So - **act in the environment** (follow trajectories).

- At each time step, remember: $\nabla_\theta logP(a_t|\pi, s_t)$
- After each trajectory (episode), compute : $R(\tau) = \sum_{t=1}^{k} \gamma^t r_t$

The REINFORCE algorithm

Initialise θ randomly:

For episodes $\{s_1, a_1, r_1 \cdots s_k, a_k, r_k\} \sim \pi_\theta$

Compute $R(\tau) = \sum_{t=1}^{k} \gamma^t r_t$

For t = 1 .. T:

$$\theta \rightarrow \theta + \alpha R \nabla_\theta log \pi(x_t, a_t)$$

The gradient of the policy network, evaluated at a particular input / output pair

If action choices led to good rewards, move weights to follow gradient (scaled by R)

Supervised learning

- **Something went well** (high LL) AND you know how to make it even better
- **Something went badly** (low LL) - you know how to fix it

Reinforcement learning

- **Something went well** (high R), you know what you did - can reinforce
  - Not sure if you could have done better
- **Something went badly** (low R) - no idea what you should have done

Learning language in RL environments

Language can refer to the visual **world**

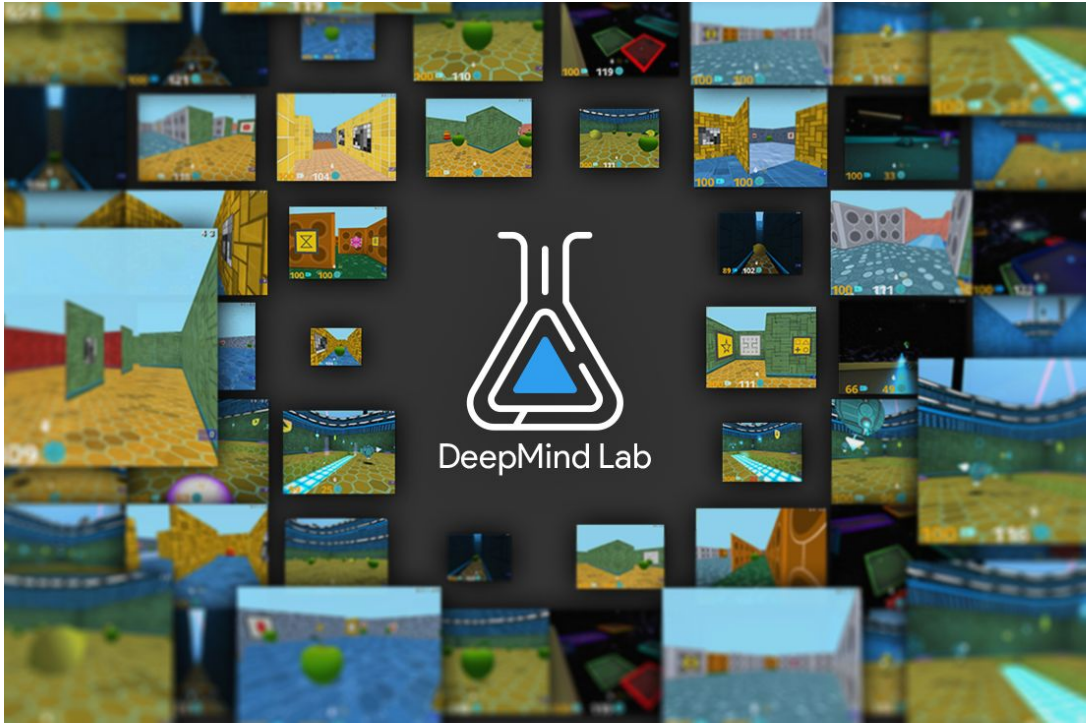
- Similar to image captioning / VQA

Language can refer to **actions** and / or **policies**

- Like a lot of natural language does!

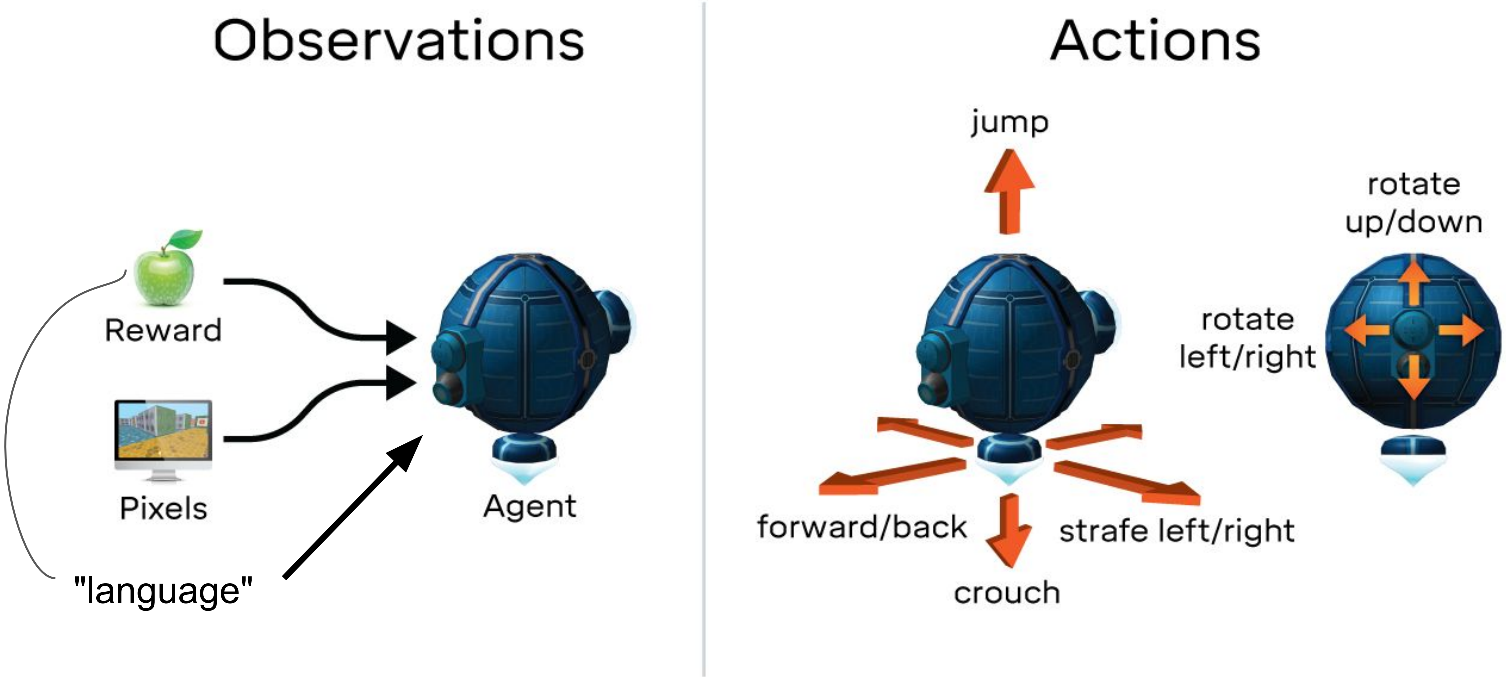
Where does **reward** come from?

DeepMind Lab



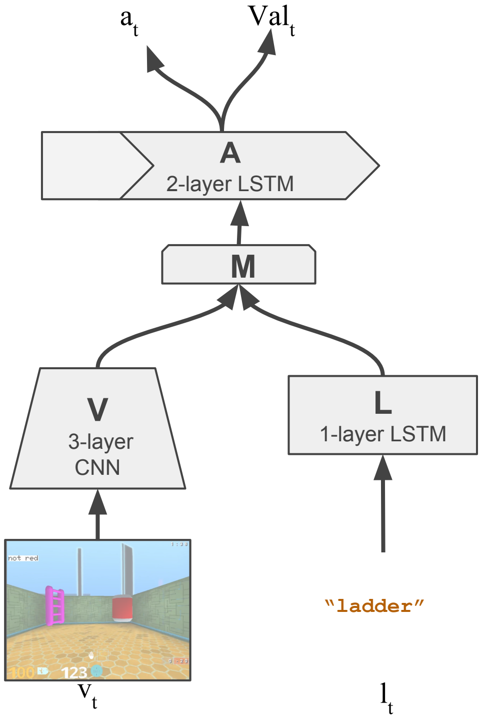
Beattie et al. DeepMind Lab. arXiv 2016. (https://github.com/deepmind/lab)

DeepMind Lab

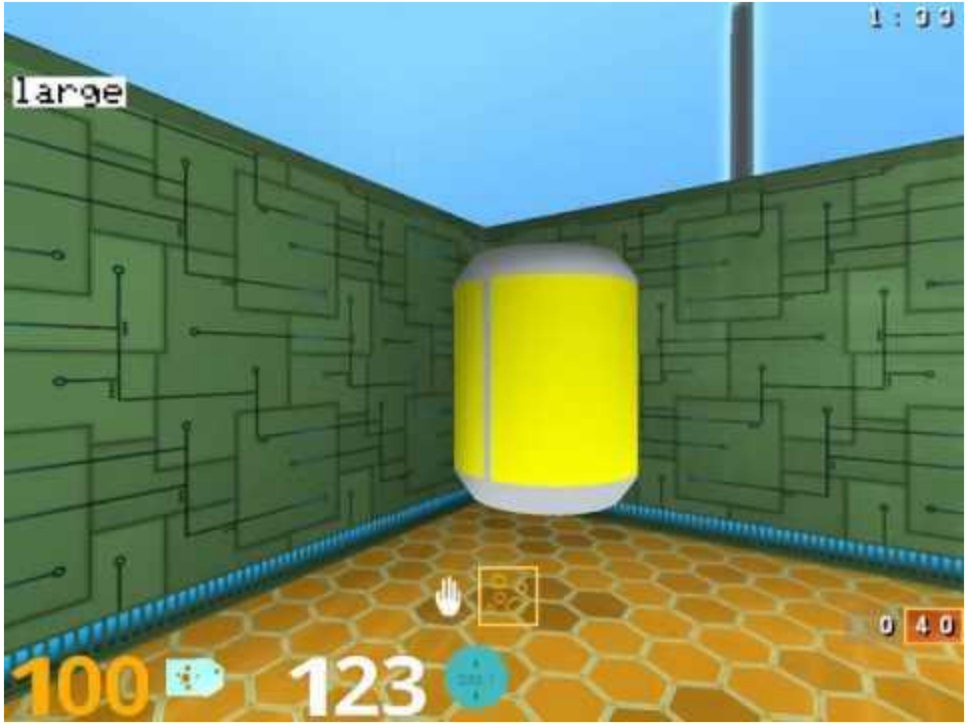


Beattie et al. DeepMind Lab. arXiv 2016. (https://github.com/deepmind/lab)

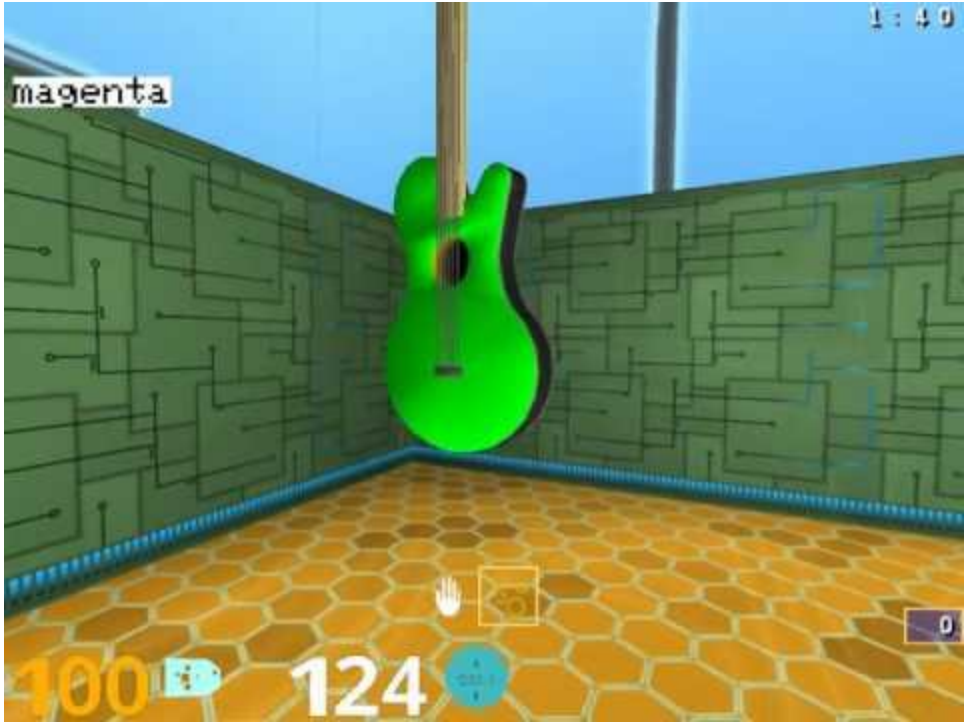
Deep RL alone (A3C) not enough



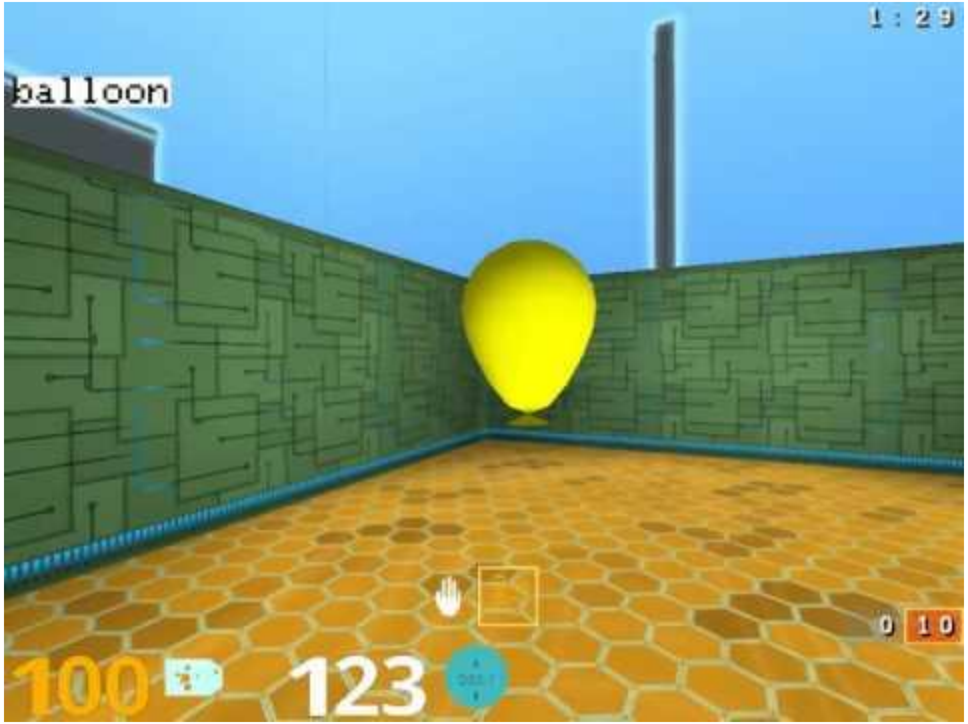
Start off small (or large)...



Colour words...



Shape words...



Language in DeepMind Lab: The Lexicon

Shapes (40) tv, ball, balloon, cake, can, cassette, chair, guitar, hairbrush, hat, ice lolly, ladder, mug, pencil, suitcase, toothbrush, key, bottle, car, cherries, fork, fridge, hammer, knife, spoon, apple, banana, cow, flower, jug, pig, pincer, plant, saxophone, shoe, tennis racket, tomato, tree, wine glass, zebra.

Colours (13) red, blue, white, grey, cyan, pink, orange, black, green, magenta, brown, purple, yellow.

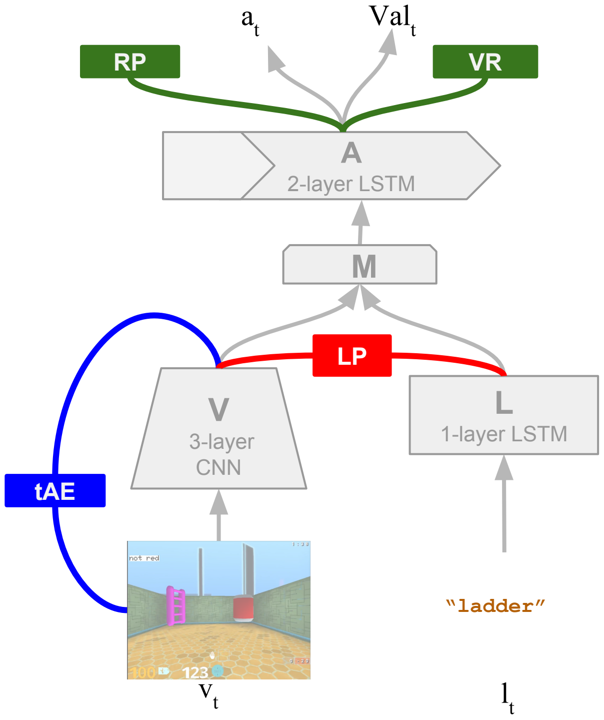
Patterns (9) plain, chequered, crosses, stripes, discs, hex, pinstripe, spots, swirls.

Shades (3) light, dark, neutral.

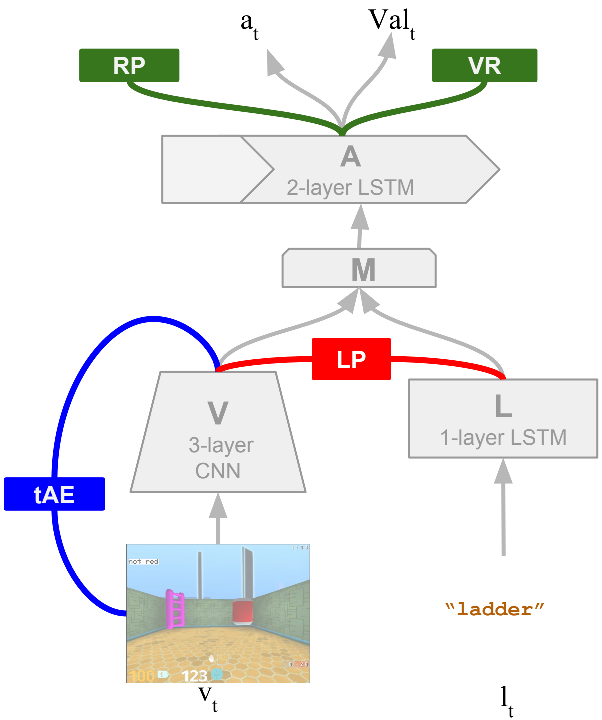
Sizes (3) small, large, medium.

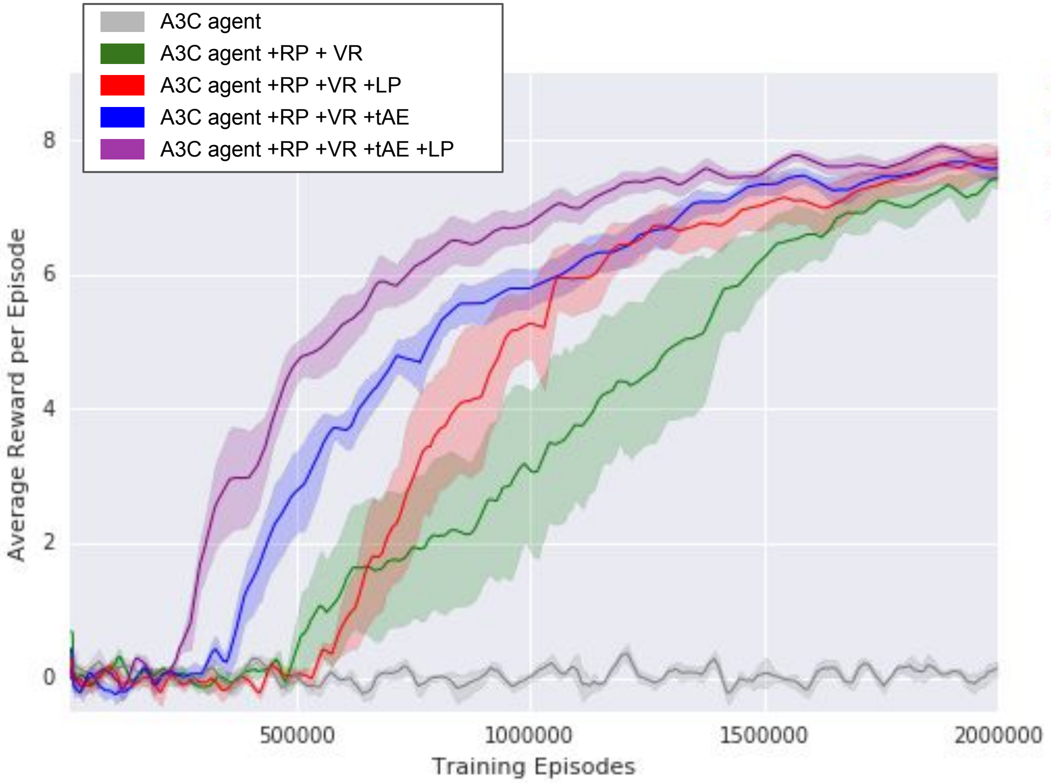
Hermann and Hill et al. Grounded Language Learning in a Simulated 3D World. arXiv 2017.

Auxiliary objectives



Unsupervised learning makes word learning possible

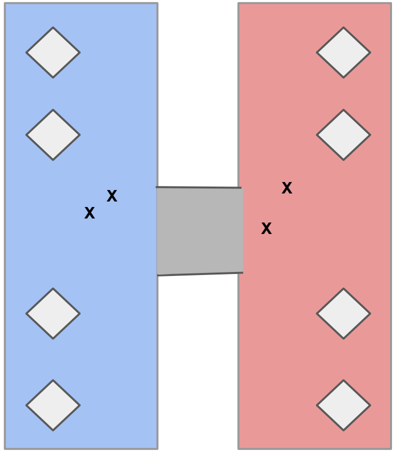


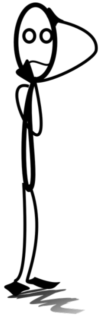


And provides insight into agents' 'thoughts'....

Combining exploration and language

Top-down view of the level



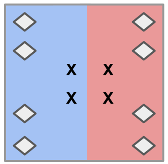


Curriculum is critical

single-room layout

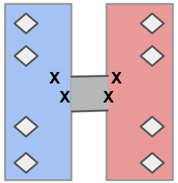
two room layout

1



two object words and room descriptors

2



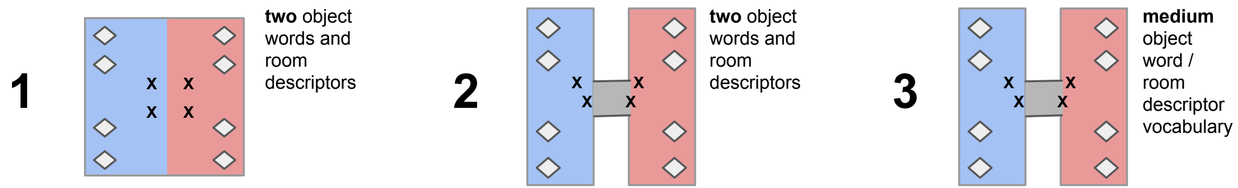
two object words and room descriptors

Curriculum is critical

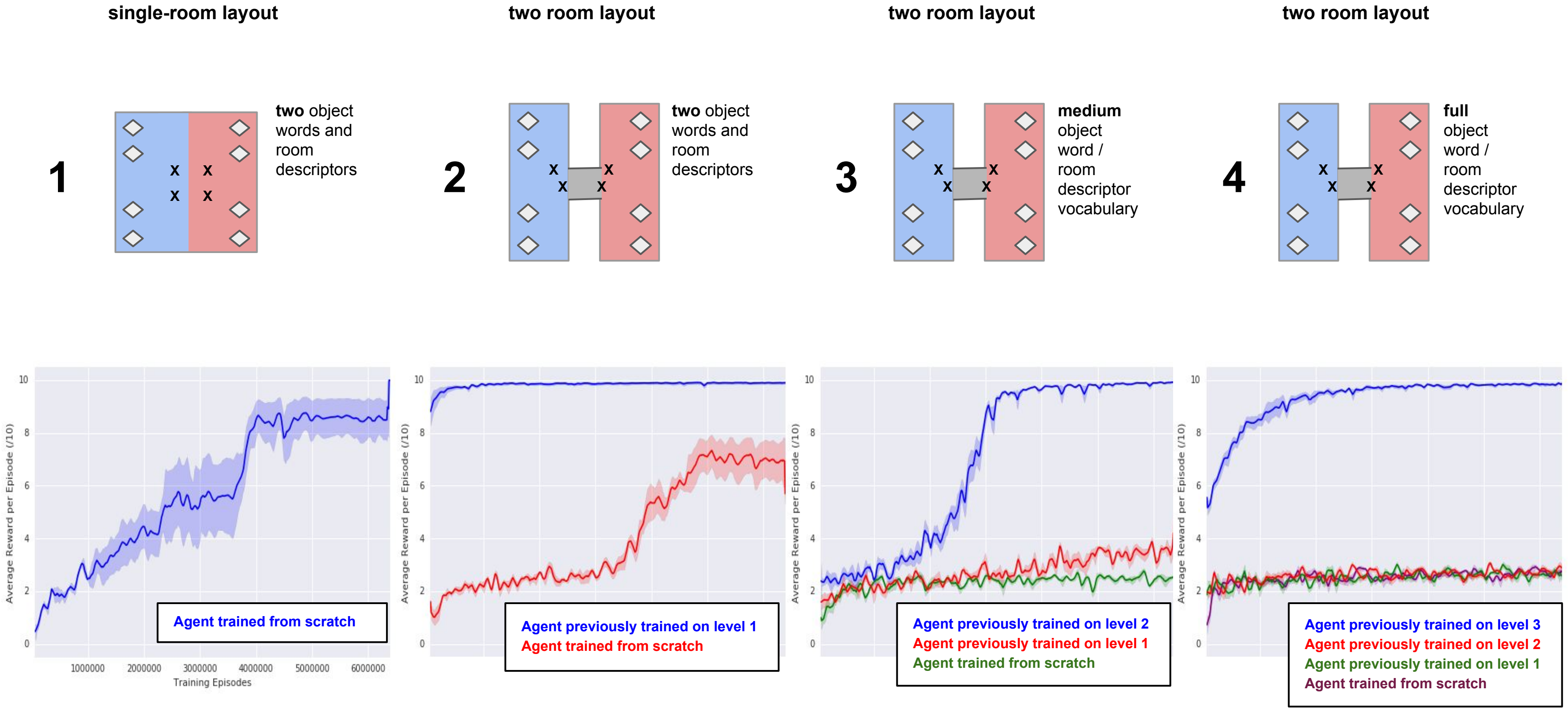
single-room layout

two room layout

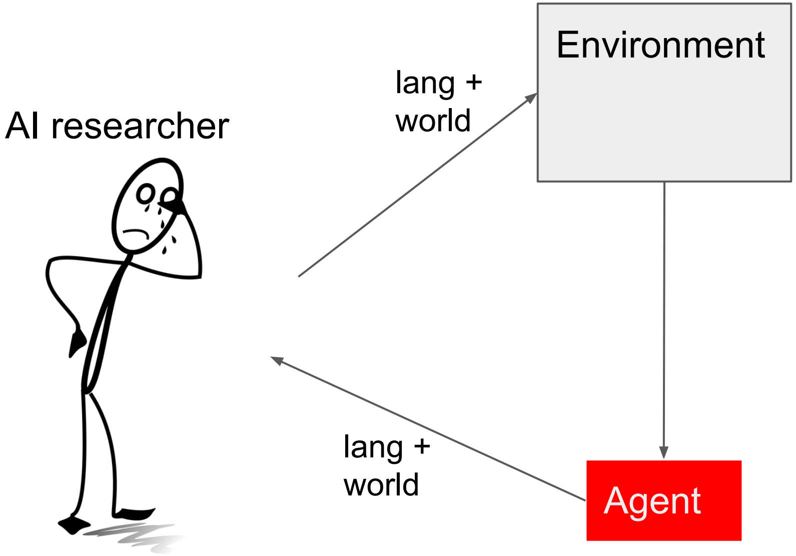
two room layout



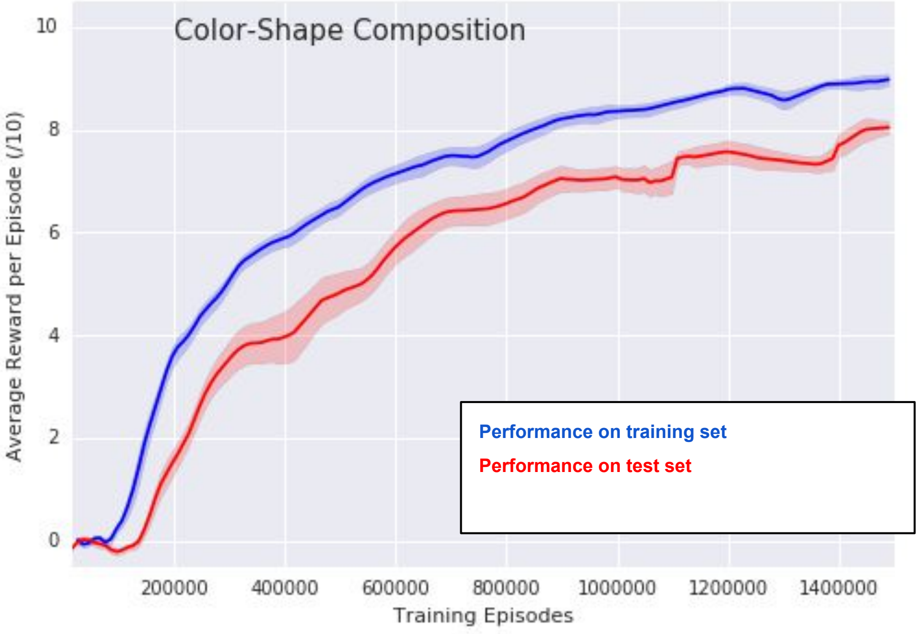
Curriculum is critical



Isn't this all a bit convoluted?



Agents naturally generalise word composition...



Training

"ladder"
"mug"
"pencil"
"suitcase"
"toothbrush"
x 40

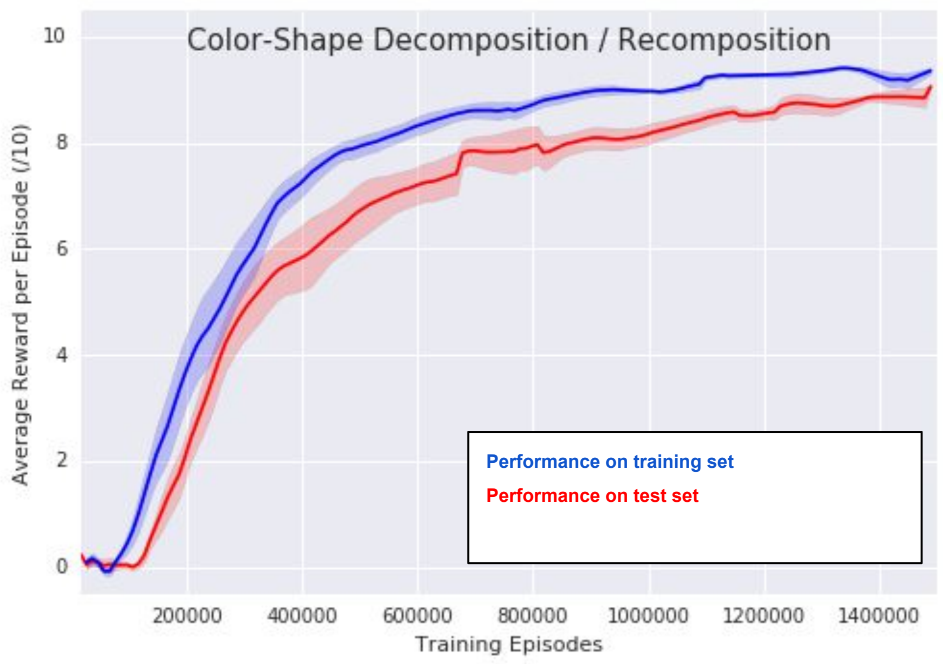
"red"
"green"
"blue"
"pink"
x 13

"red ladder"
"green mug"
"blue pencil"
x 400

Test

"pink ladder"
"yellow mug"
"green pencil"
x 120

Decompose before re-compose



Training

"red ladder"
"**green** mug"
"blue pencil"

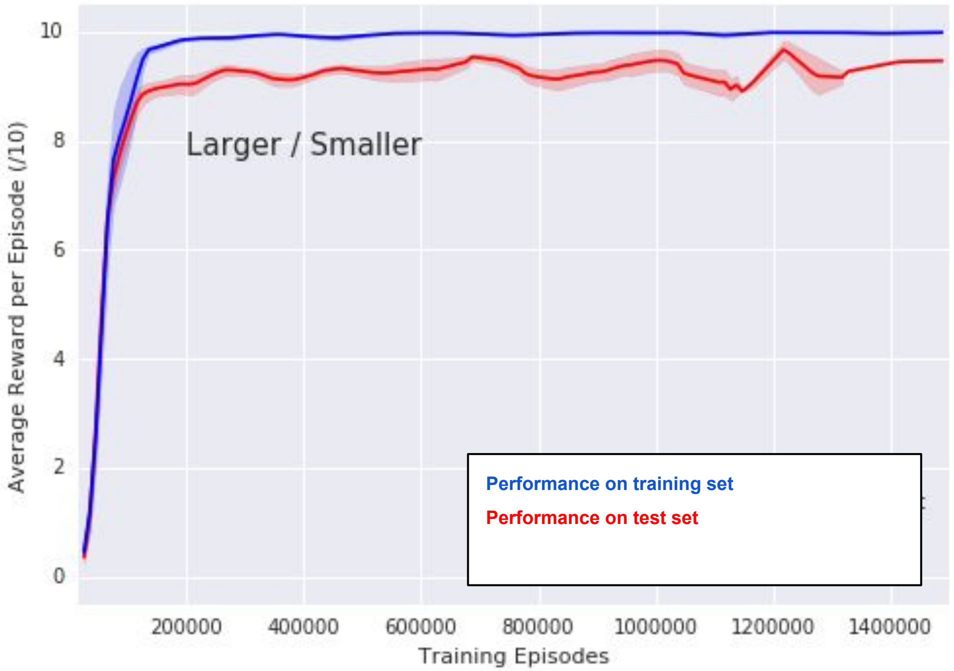
x 400

Test

"pink ladder"
"yellow mug"
"**green** pencil"

x 120

Apply modifiers and predicates to novel objects



Training

"larger" (ball)
"smaller" (ball)

"larger" (mug)
"smaller" (mug)

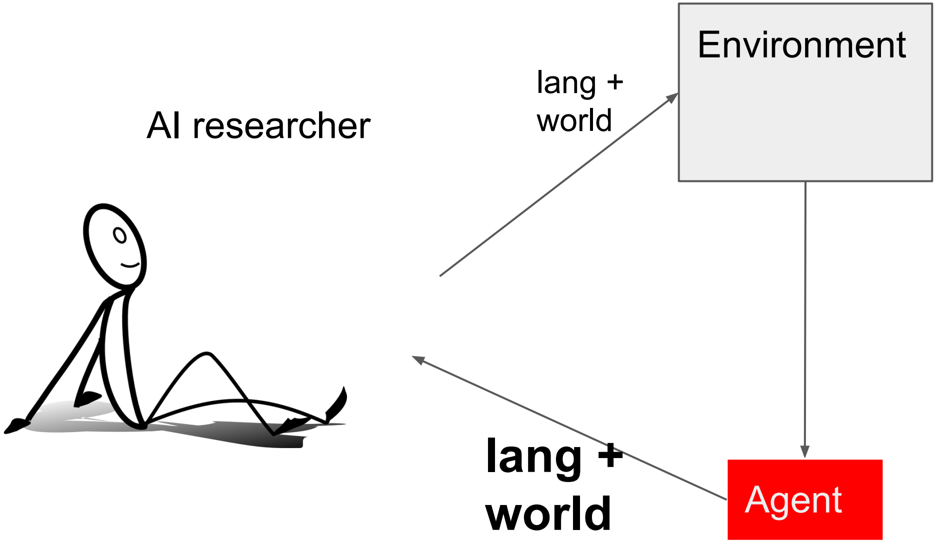
x 30

Test

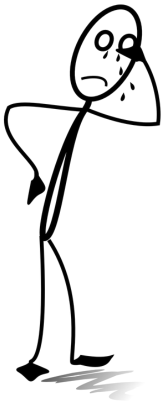
"larger" (pencil)
"smaller" (pencil)

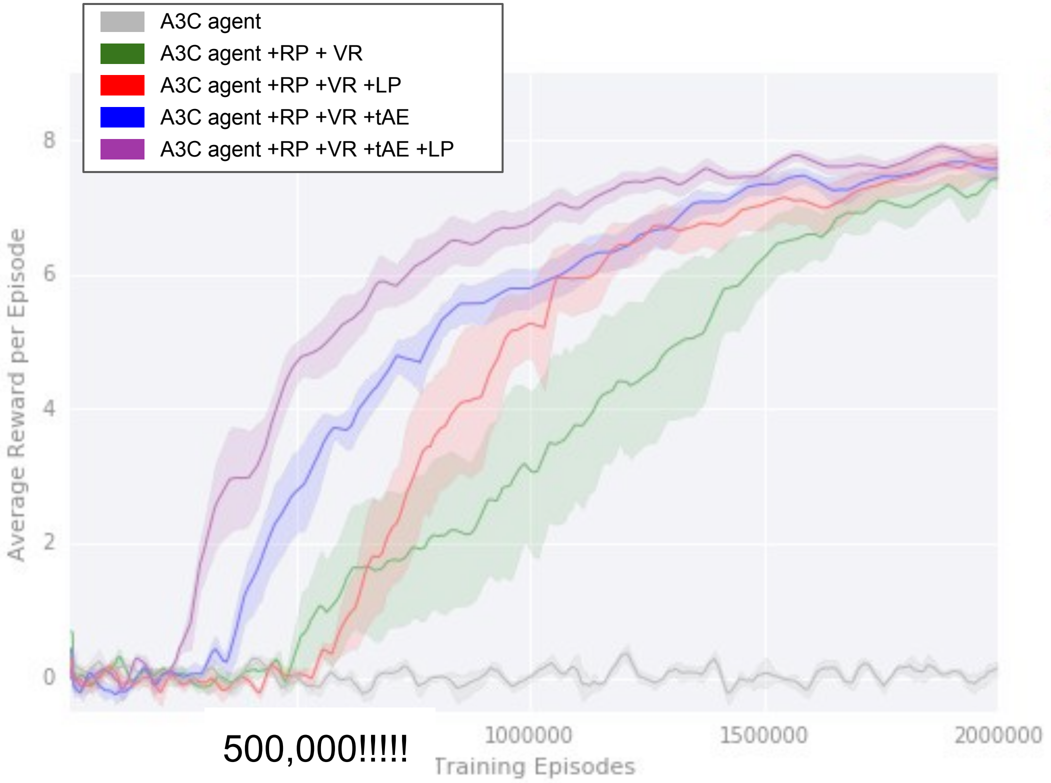
x 10

Generalisation (zero-shot etc...)

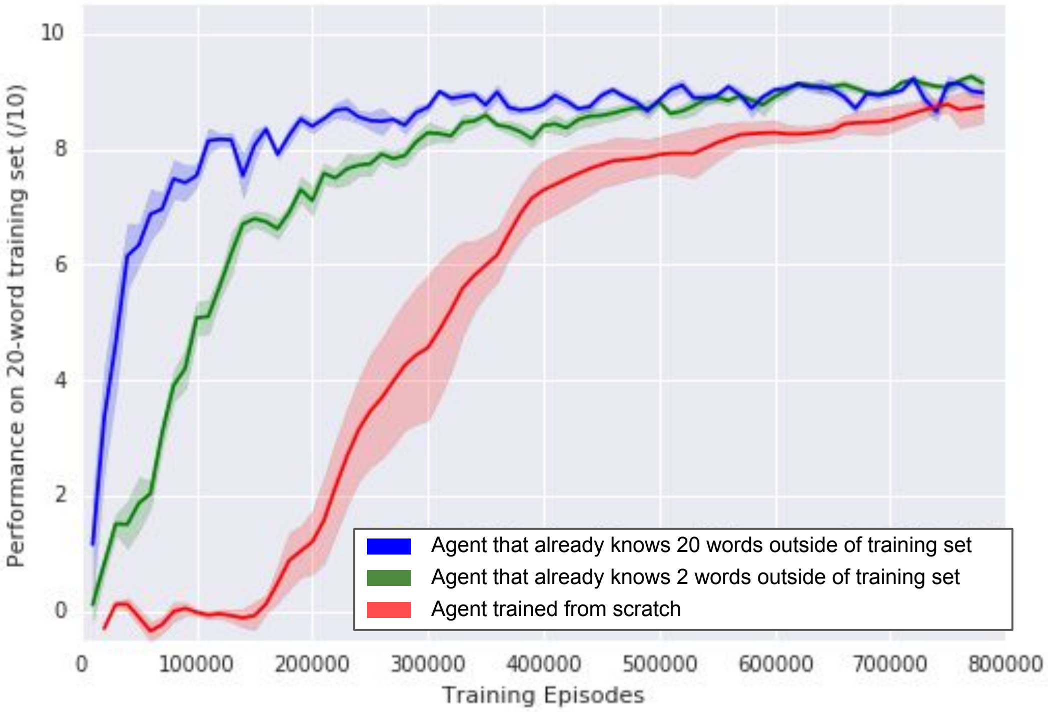


Isn't learning slow?

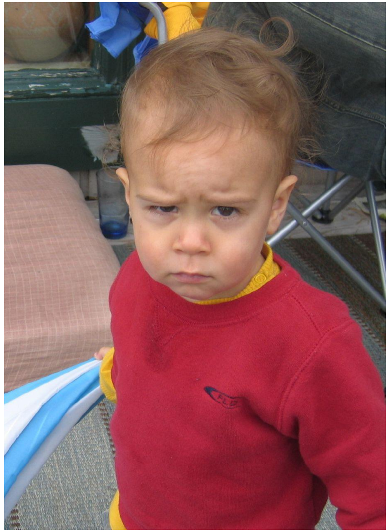




Word learning gets quicker the more the agent 'knows'



Much like little people



296 *K. Plunkett* et al.

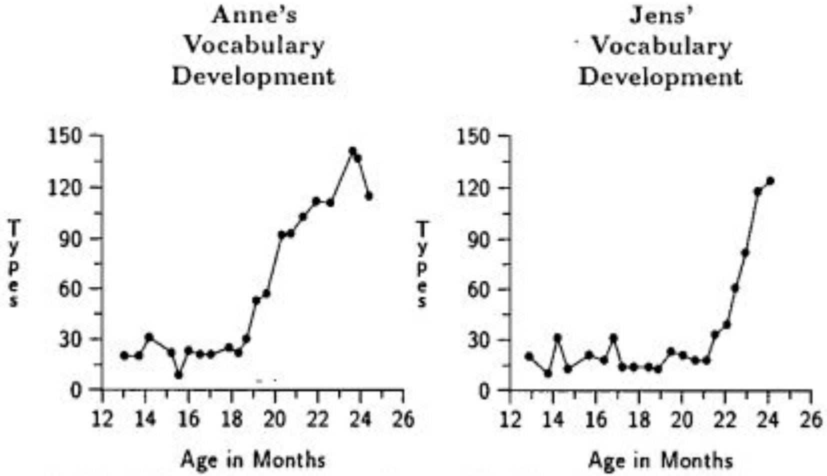
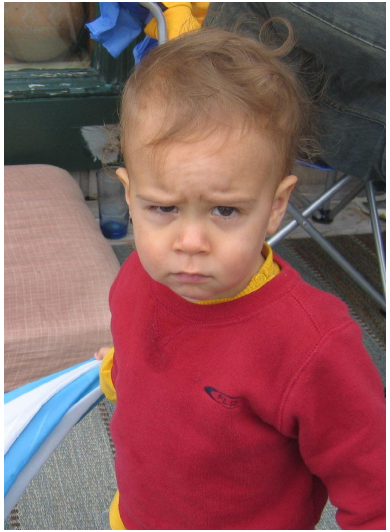


Figure 1. Vocabulary development in two Danish children: the plateau period is followed by a period of accelerated growth referred to as the vocabulary spurt.

Much like little people



296 *K. Plunkett* et al.

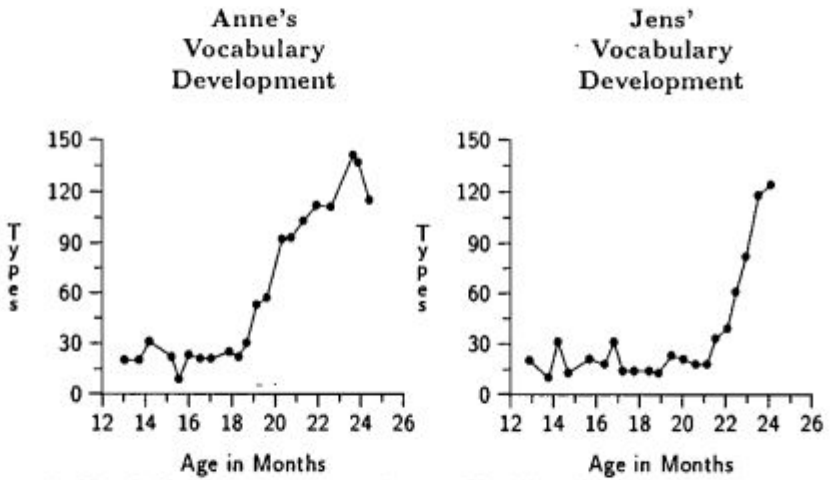
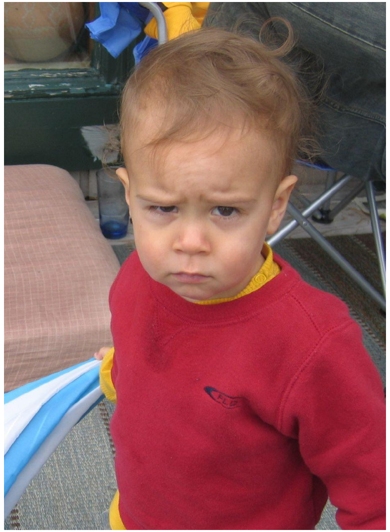
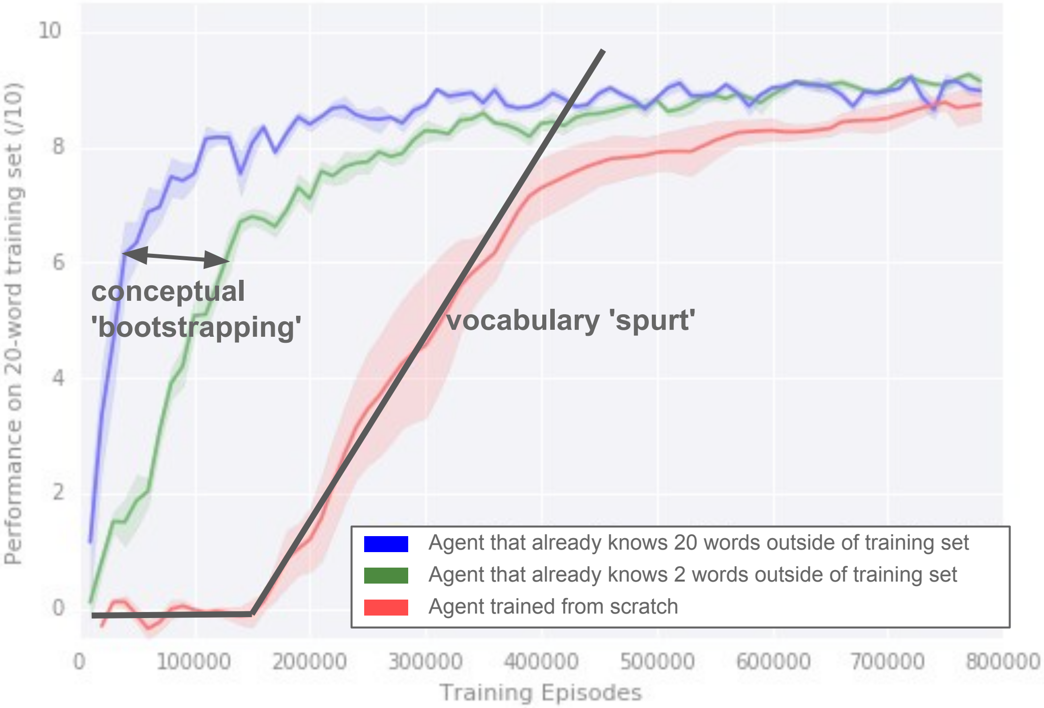


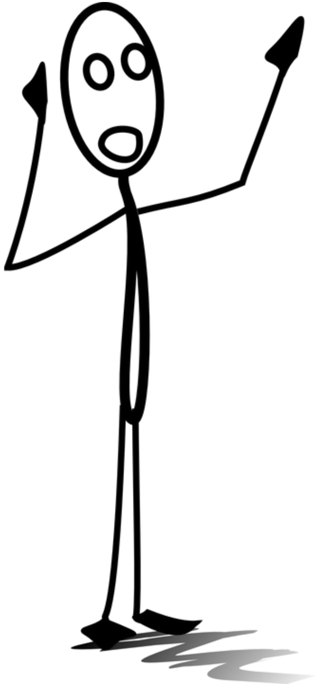
Figure 1. Vocabulary development in two Danish children: the plateau period is followed by a period of accelerated growth referred to as the vocabulary spurt.

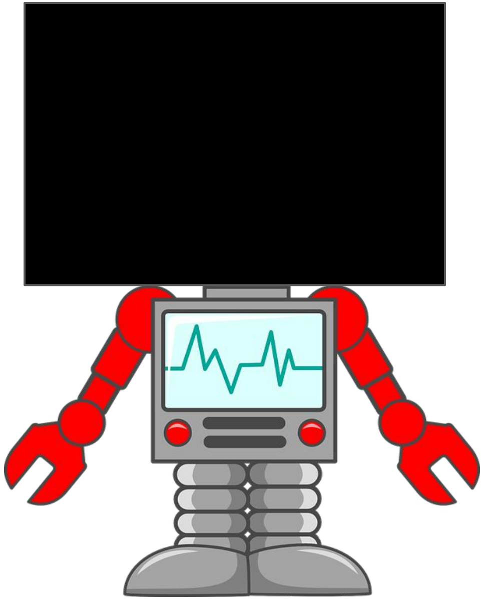
Much like little people



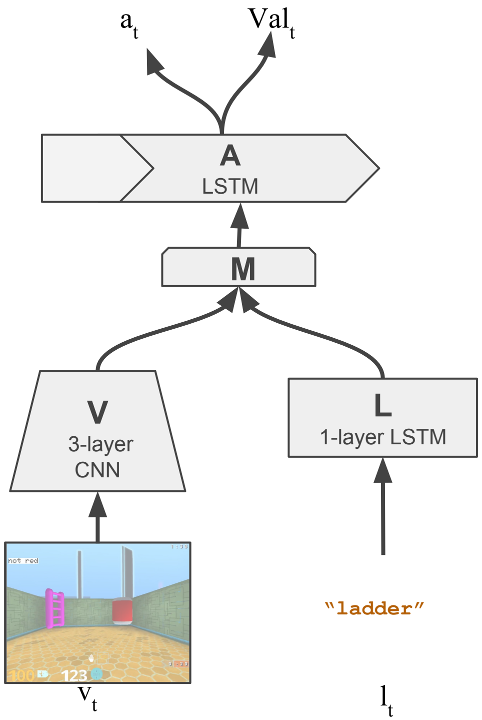


How does the agent represent its knowledge?

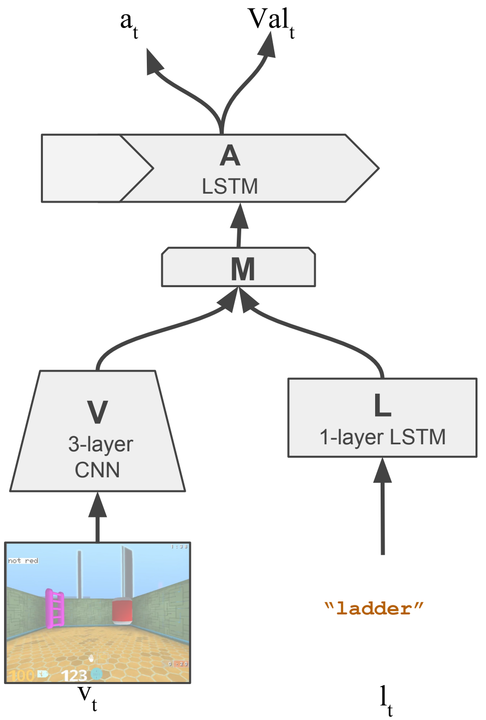


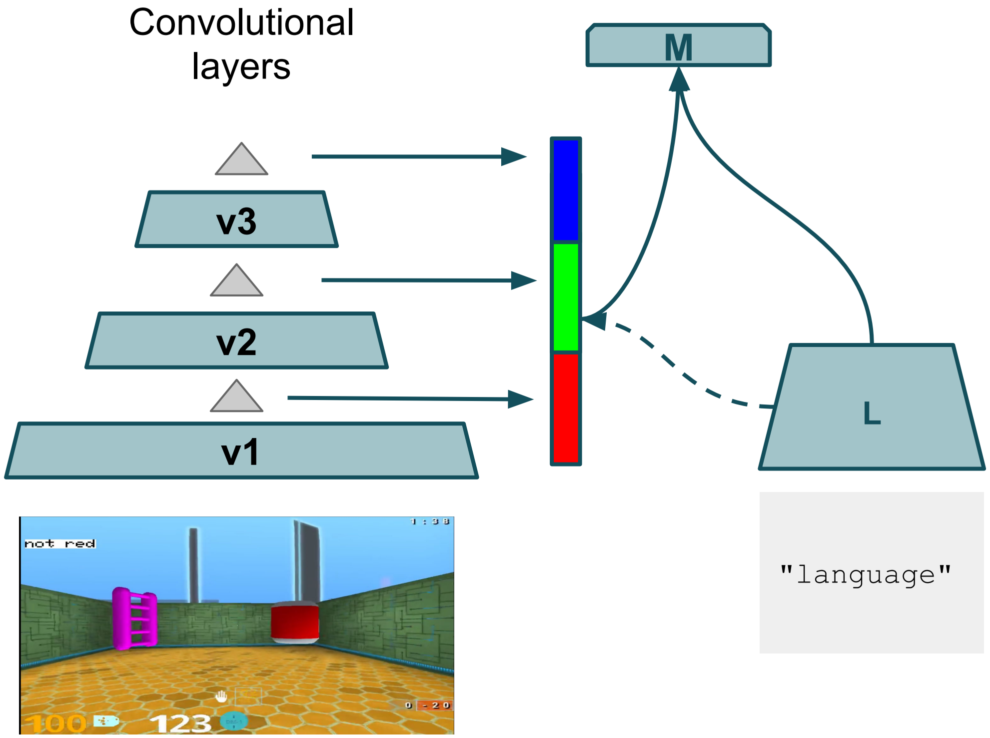


Layerwise attention

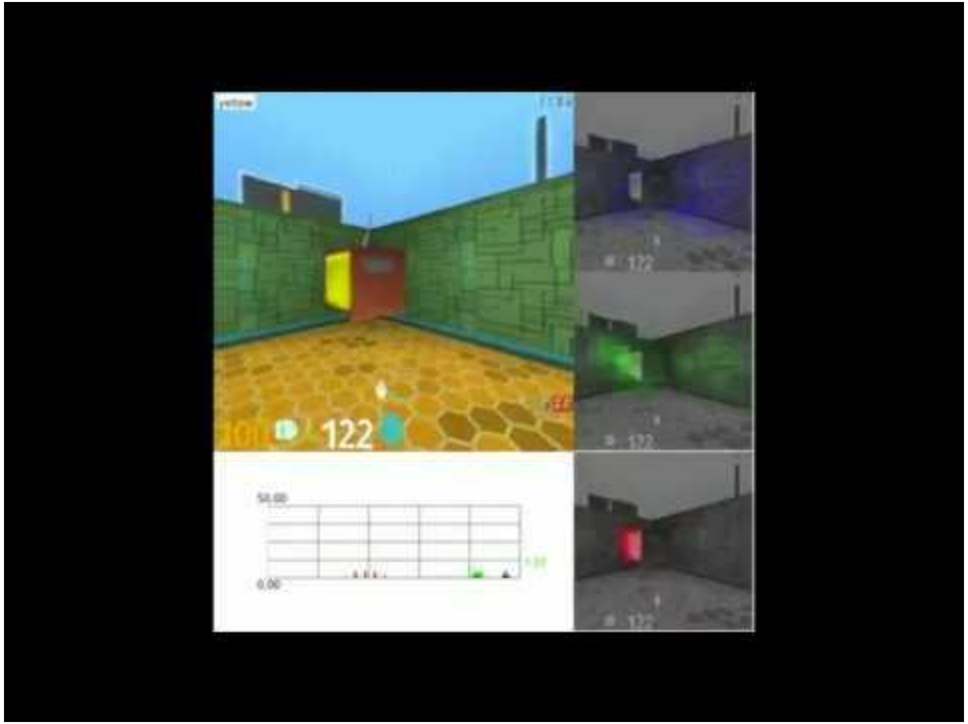


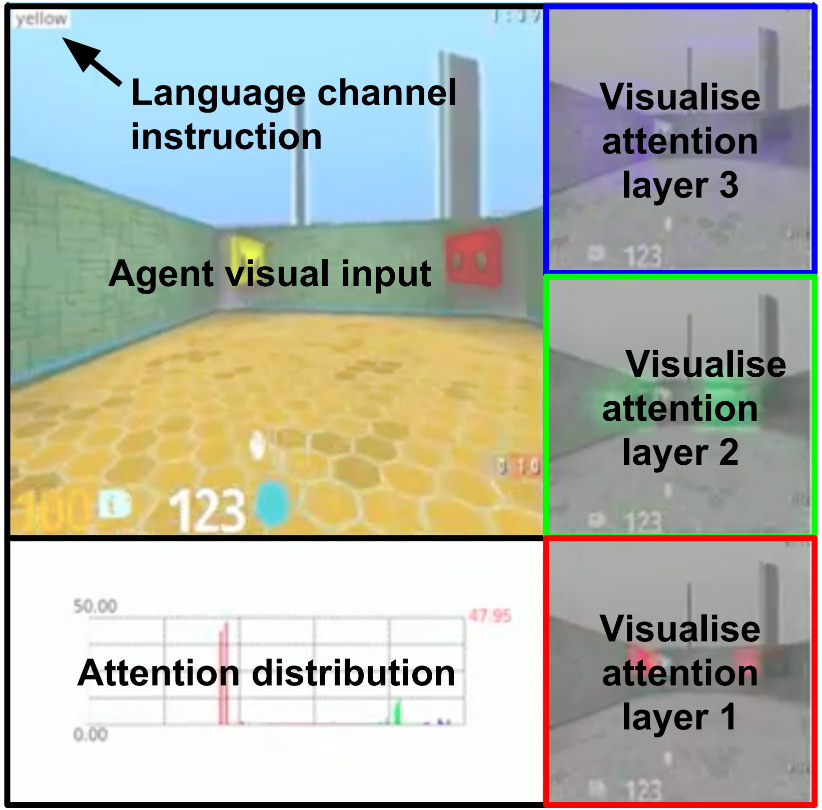
Layerwise attention



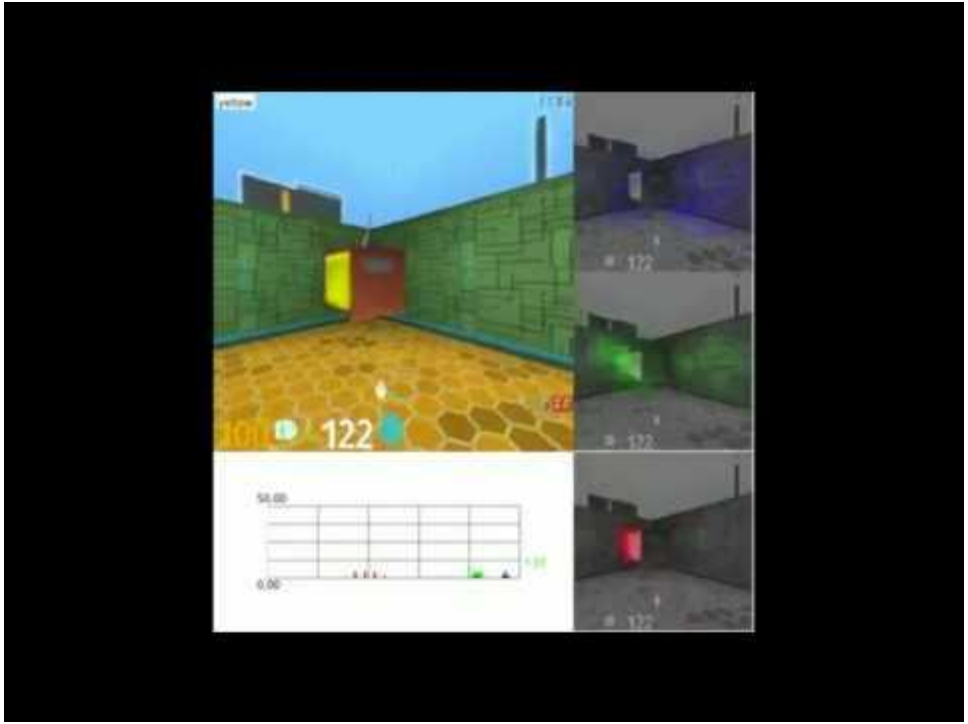


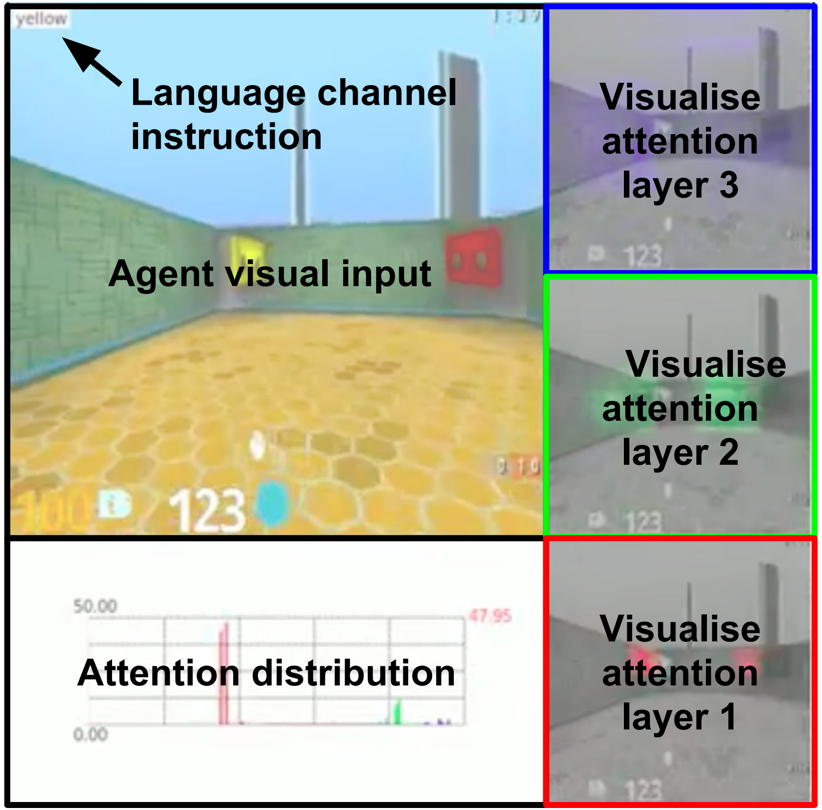
Processing colour words





Processing shape words





Conclusions

- Using RL we can ground language in **vision**, **actions** and **policies**
- Neural network policies enable natural **generalisation** and **composition**
- Ongoing work to scale approaches to **full sentences** and **natural commands**
    - What aspects of language are not covered by this approach?
    - What challenges do we face extending this?