Reinforcement Learning

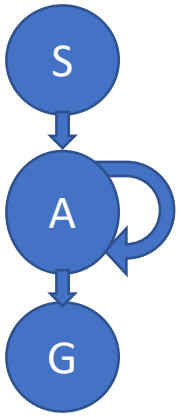
Caus Danu

Plan

Start state: Assume **basic knowledge** of RL from the lectures in this course

Actions: Iterate and **exploit prior** knowledge to **explore** new **RL concepts**

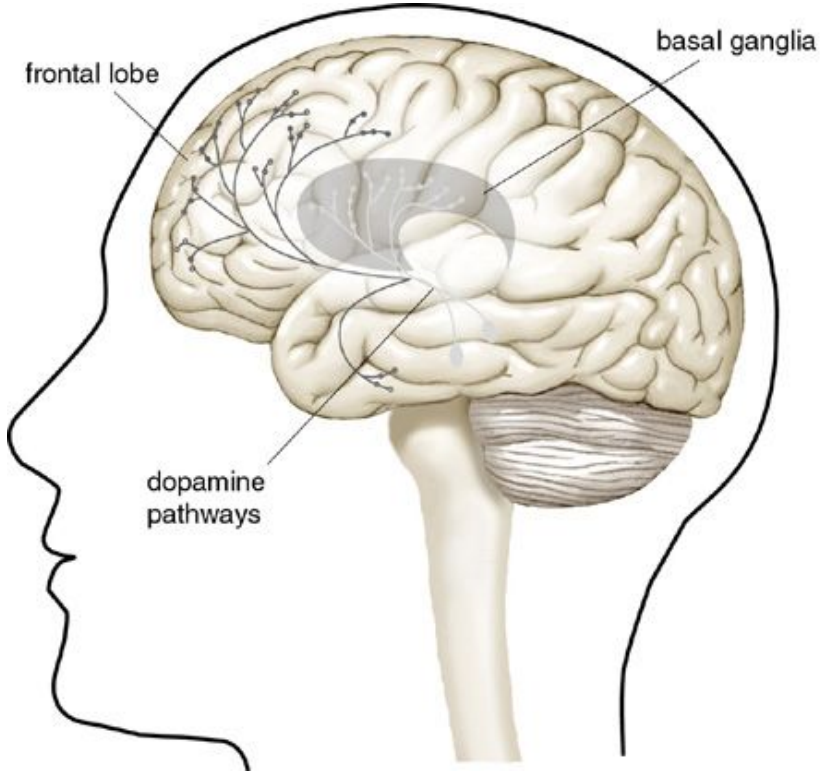
Goal state: Have a **comprehensive view** of the field of RL



- RL in the Brain
- Model Free RL
- Model Based RL
- Comparisons
- Conclusion

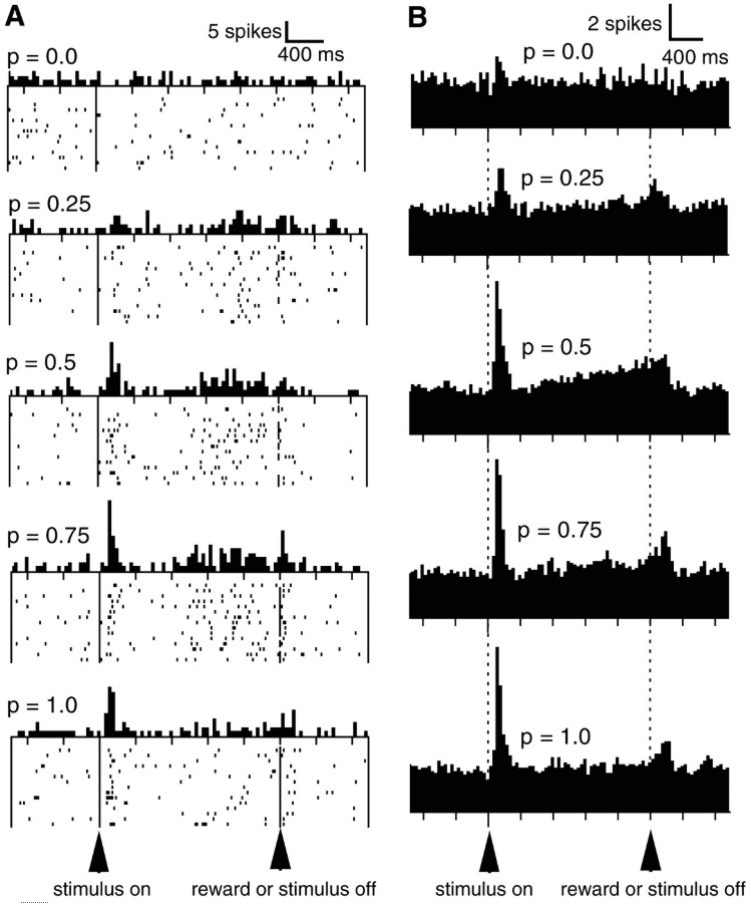
RL in the Brain

Dopamine Neurons



Encode
Prediction
Error

Dopamine Neuron Spikes

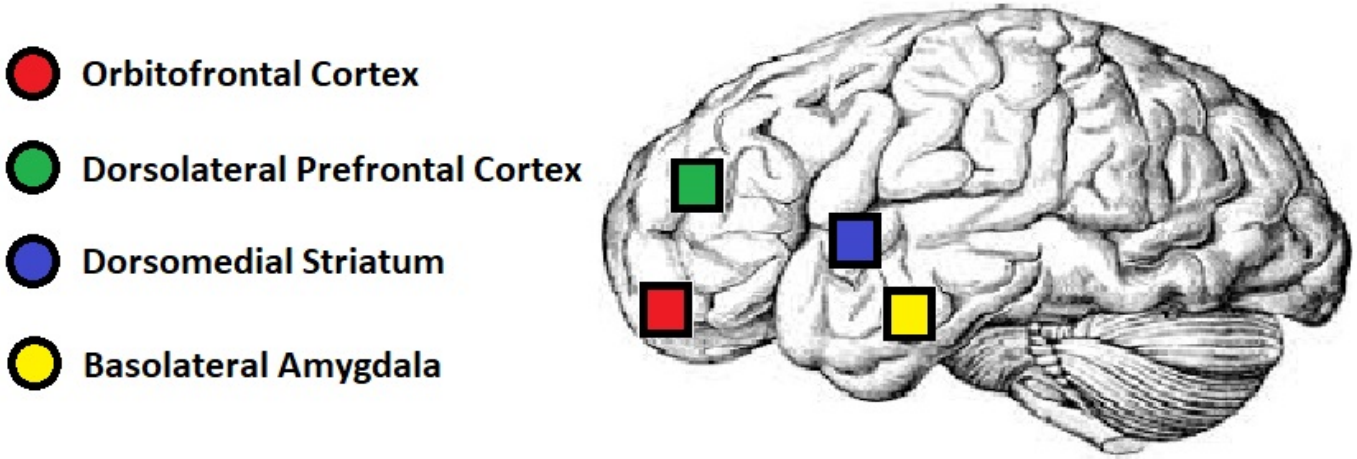


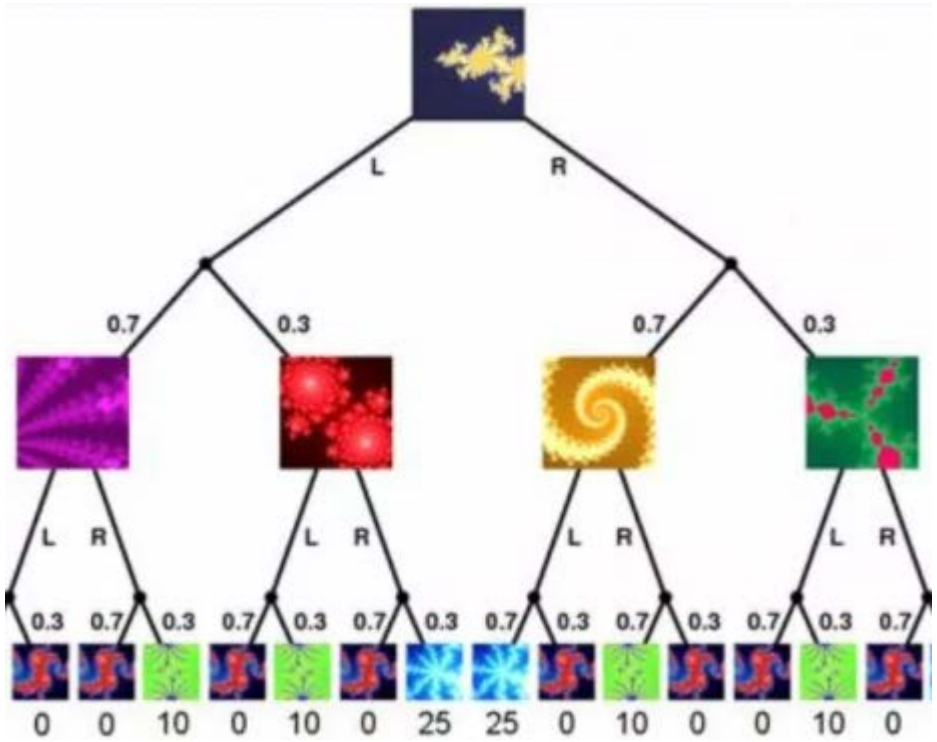
Fiorillo et al.

RL in the Brain

Frontal Cortex: involved in logic
Amygdala: constrains decisions based on model

Build a mental **decision tree** based on model





Model Free RL

- You already know it from every RL lecture!
- Typical Examples: **Q-Learning** and **Sarsa**
- Very statistically **inefficient** at **train** time
- **Quick** at **test** time

Q-Table

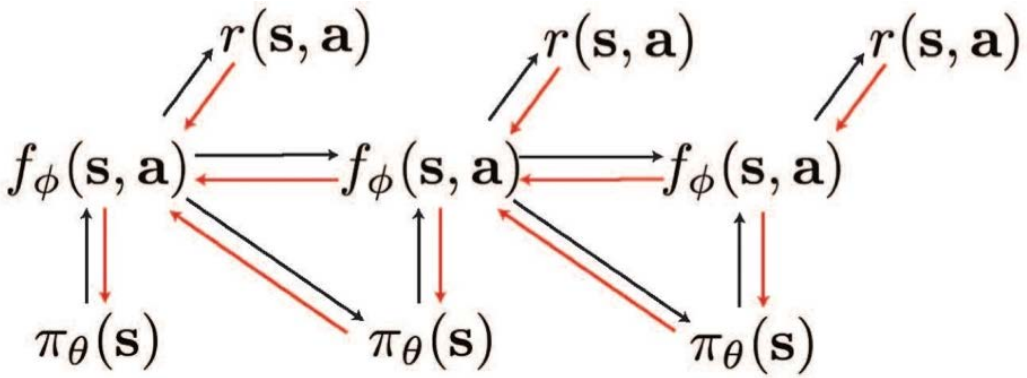
0.812	0.868	0.918	+1
0.762		0.660	-1
0.705	0.655	0.611	0.388

Gist: Compute a lookup **table** of state **values**

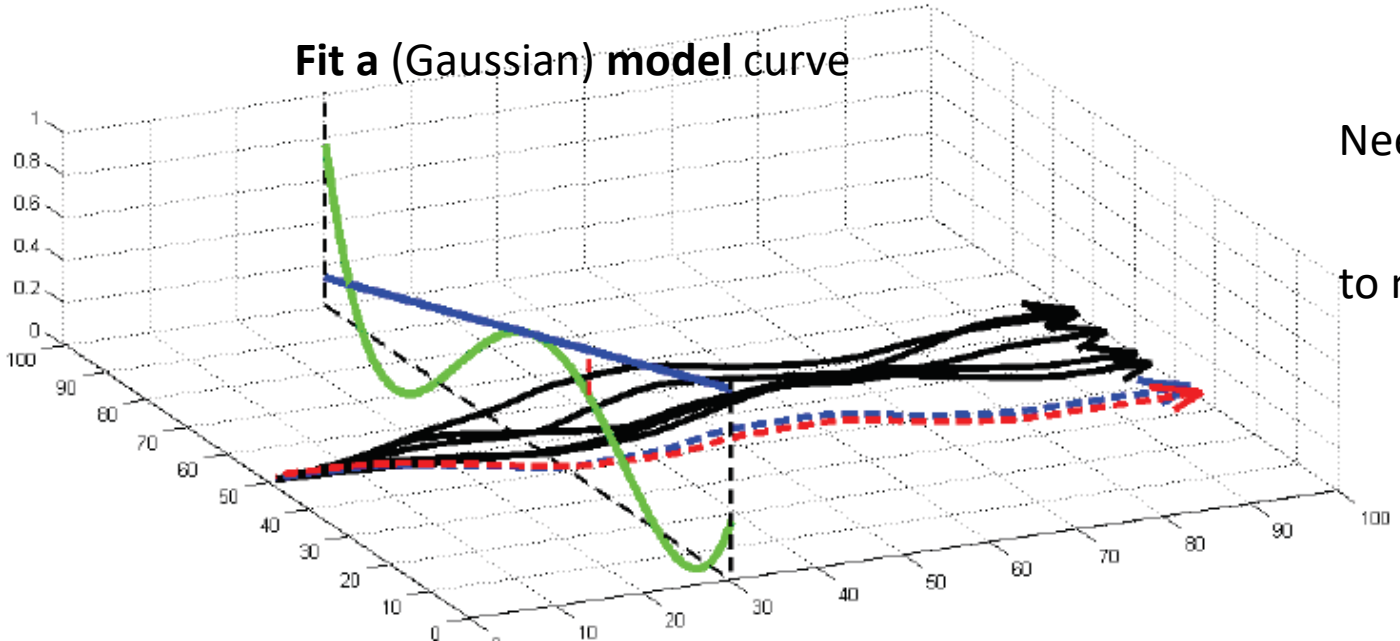
Model Based RL

- Often overlooked in standard RL literature
- Typical Examples: **Guided Policy Search**
- Very statistically **efficient** at **train** time
- **Slow** at **test** time

Backpropagate "through the model into the policy"



Fit a (Gaussian) **model** curve

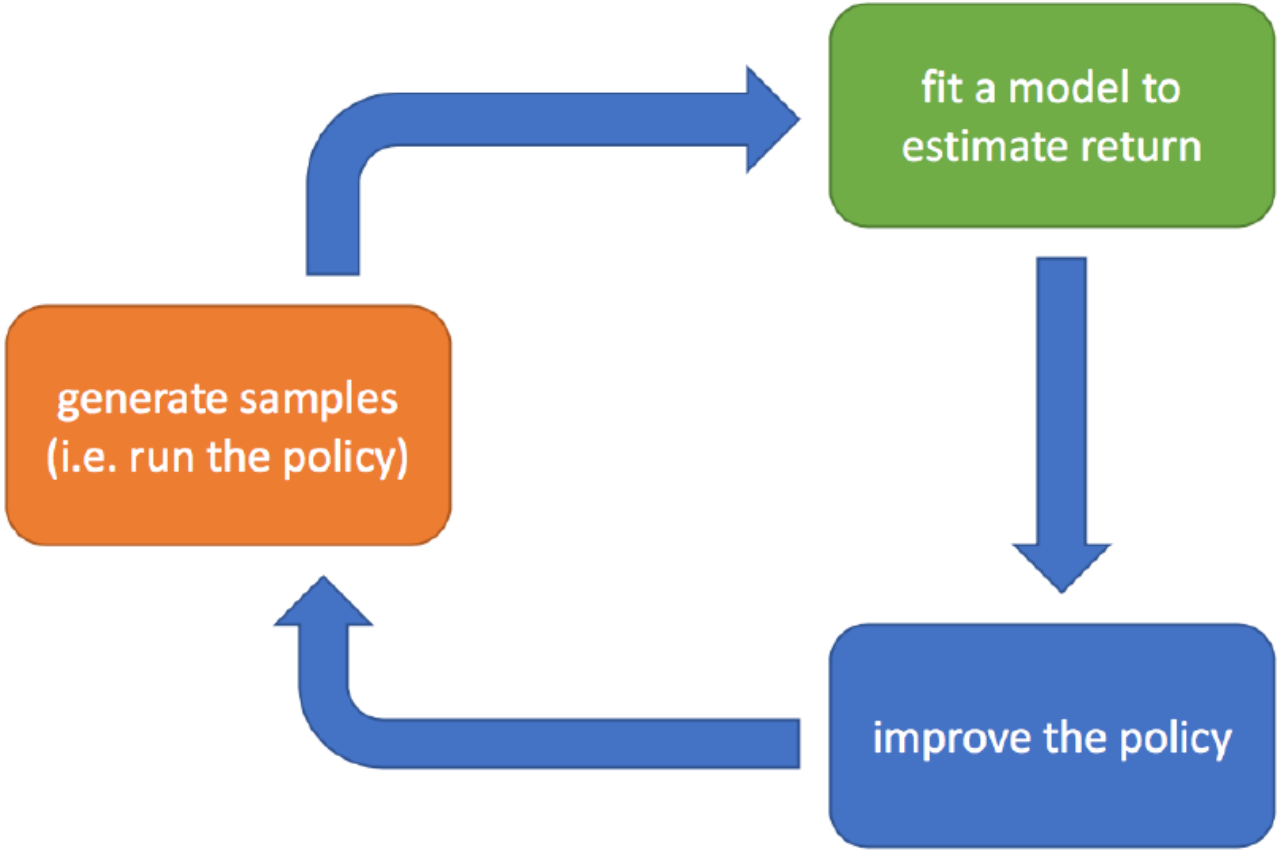


Need: $\frac{\mathrm{d}f}{\mathrm{d}a_t}, \frac{\mathrm{d}f}{\mathrm{d}s_t}, \frac{\mathrm{d}r}{\mathrm{d}a_t}, \frac{\mathrm{d}r}{\mathrm{d}s_t}$

to maximize **r(s, a)**

[S. Levine]

Structure of RL Problem

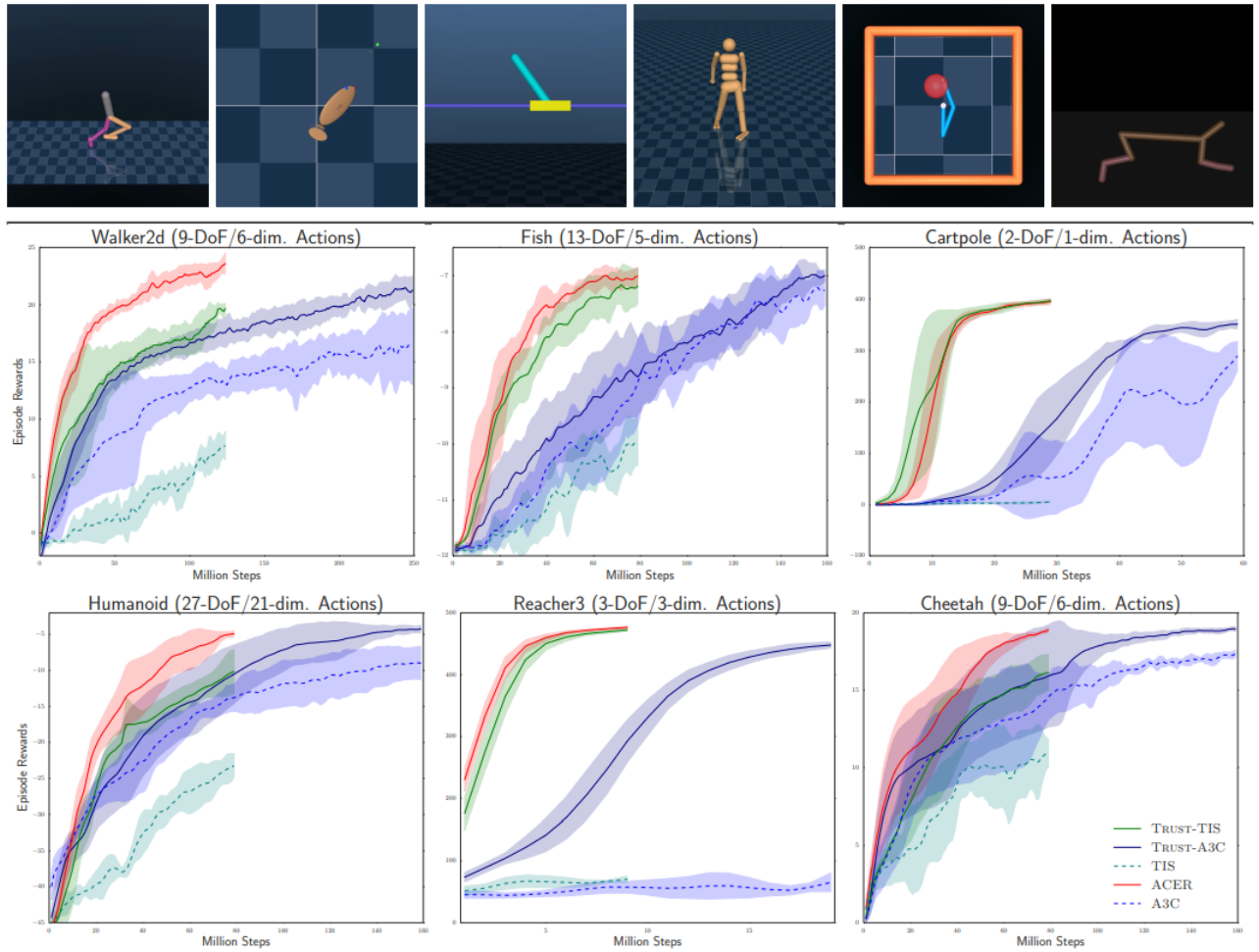


compute $\hat{Q} = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$ (MC policy gradient)
fit $Q_\phi(\mathbf{s}, \mathbf{a})$ (actor-critic, Q-learning)
estimate $p(\mathbf{s}'|\mathbf{s}, \mathbf{a})$ (model-based)

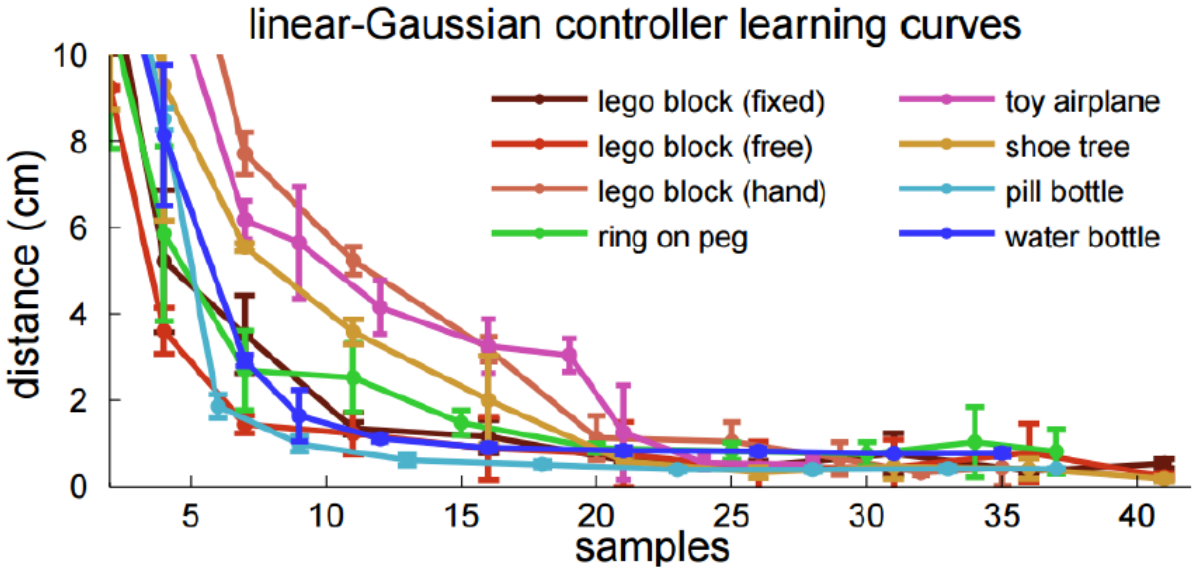
$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$ (policy gradient)
$\pi(\mathbf{s}) = \arg\max Q_\phi(\mathbf{s}, \mathbf{a})$ (Q-learning)
optimize $\pi_\theta(\mathbf{a}|\mathbf{s})$ (model-based)

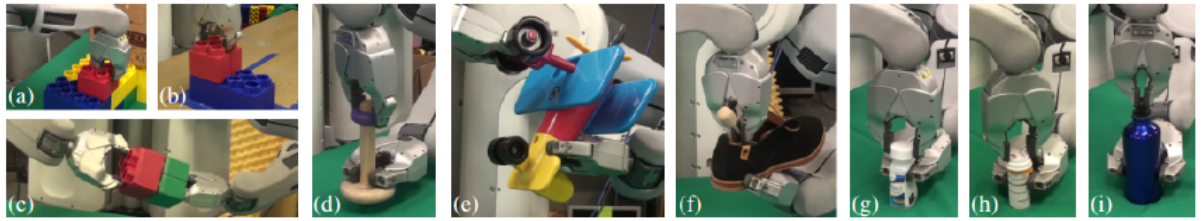
[S. Levine]

Model Free vs Model Based Efficiency



Wang et al.



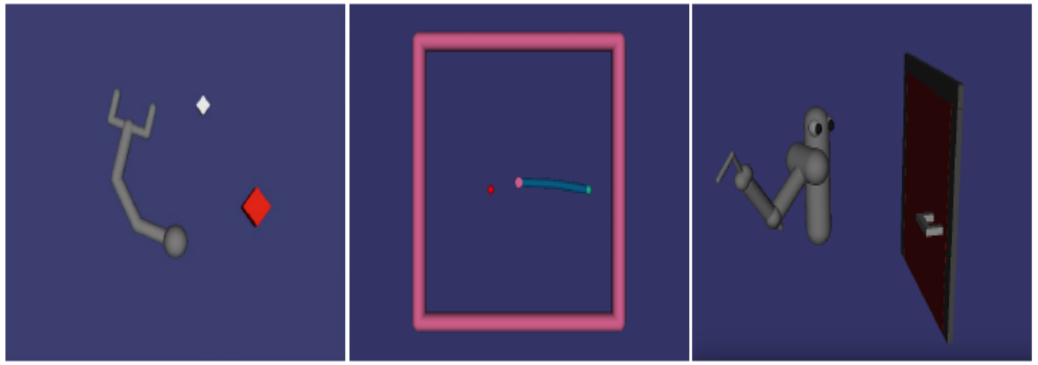


Levine et al.

Concluding Remarks

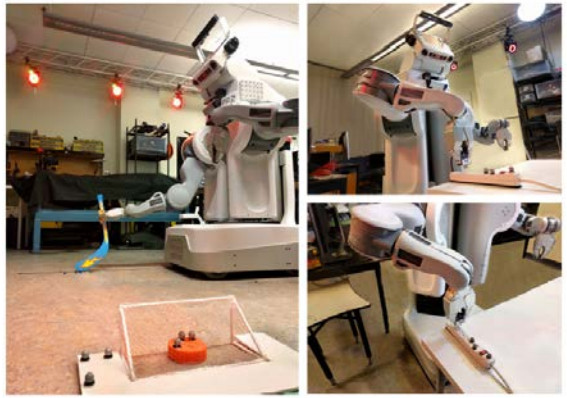
- Use **Bayesian** Theory to concatenate both approaches: **Hybrid RL**
- Employ **Model Free** approach to compute a **Prior** distribution in a **simulator**
- Use the Prior distribution with a **Model Based** approach in order to compute a **Posterior** distribution on **hardware**
- Consider the **posterior** distribution function as the **model**

Prior Distribution



Transfer Prior

Learn Posterior Model



Chebotar et al.

Did you know that:
“Questions reinforce what we’ve learned”

Anonymous Psychologist

References

- [1] C. D. Fiorillo, P. N. Tobler, and W. Schultz, “Discrete coding of reward probability and uncertainty by dopamine neurons” Science, vol. 299, no. 5614, pp. 1898{1902, 2003.
- [2] Z. Wang, V. Bapst, N. Heess, V. Mnih, R. Munos, K. Kavukcuoglu, and N. de Freitas, “Sample efcient actor-critic with experience replay” arXiv preprint arXiv:1611.01224, 2016.
- [3] S. Levine, N. Wagener, and P. Abbeel, “Learning contact-rich manipulation skills with guided policy search” in Robotics and Automation (ICRA), 2015 IEEE International Conference on, pp. 156{163, IEEE, 2015.
- [4] Y. Chebotar, K. Hausman, M. Zhang, G. Sukhatme, S. Schaal, and S. Levine, “Combining model-based and model-free updates for trajectory-centric reinforcement learning” arXiv preprint arXiv:1703.03078, 2017.