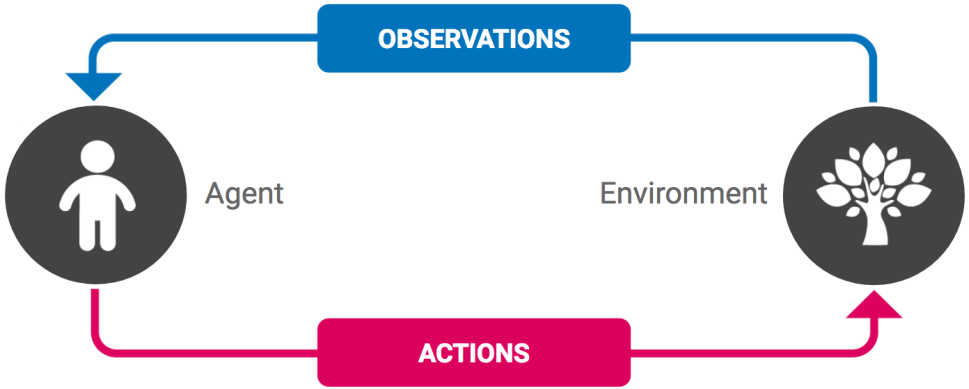
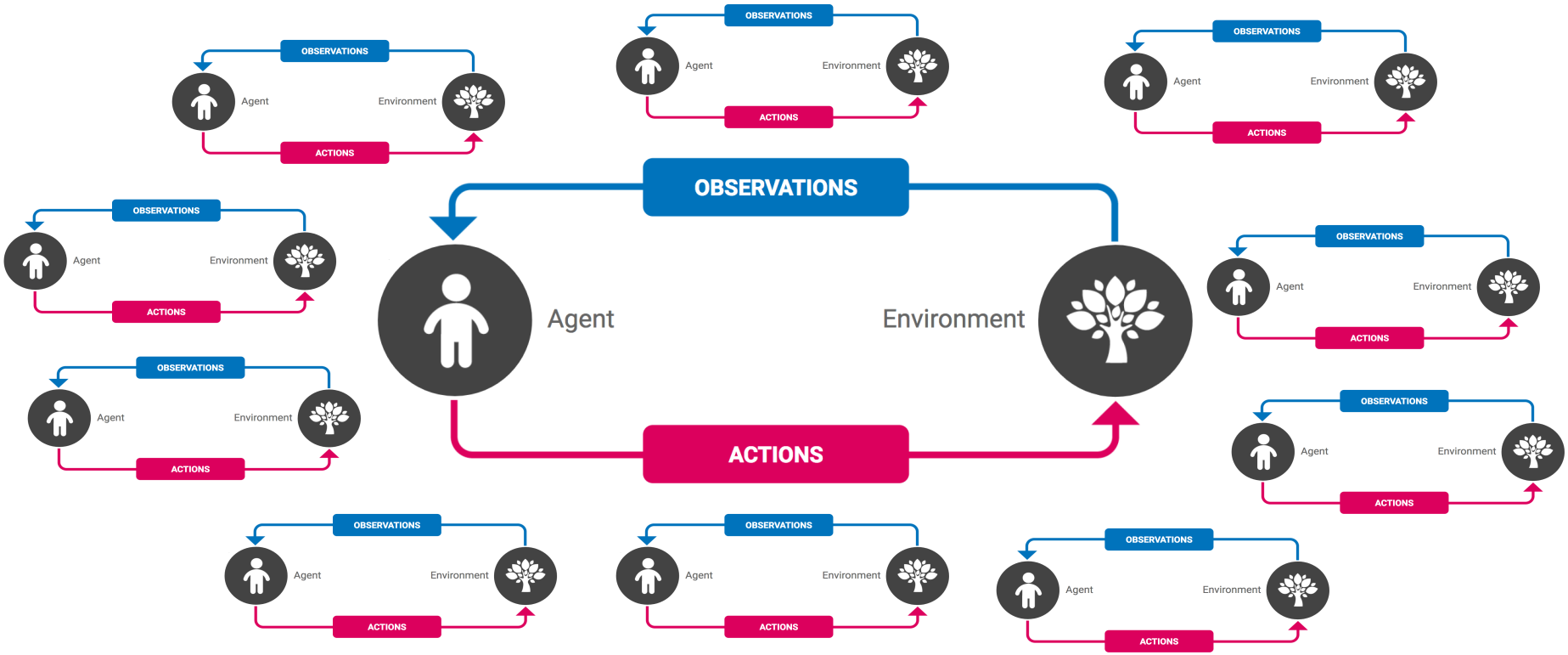
Probabilistic Perspectives on Meta and Reinforcement Learning

Yee Whye Teh
Statistics @ Oxford
DeepMind
http://csml.stats.ox.ac.uk/people/teh/

Reinforcement Learning



Efficient Learning from Multiple Tasks

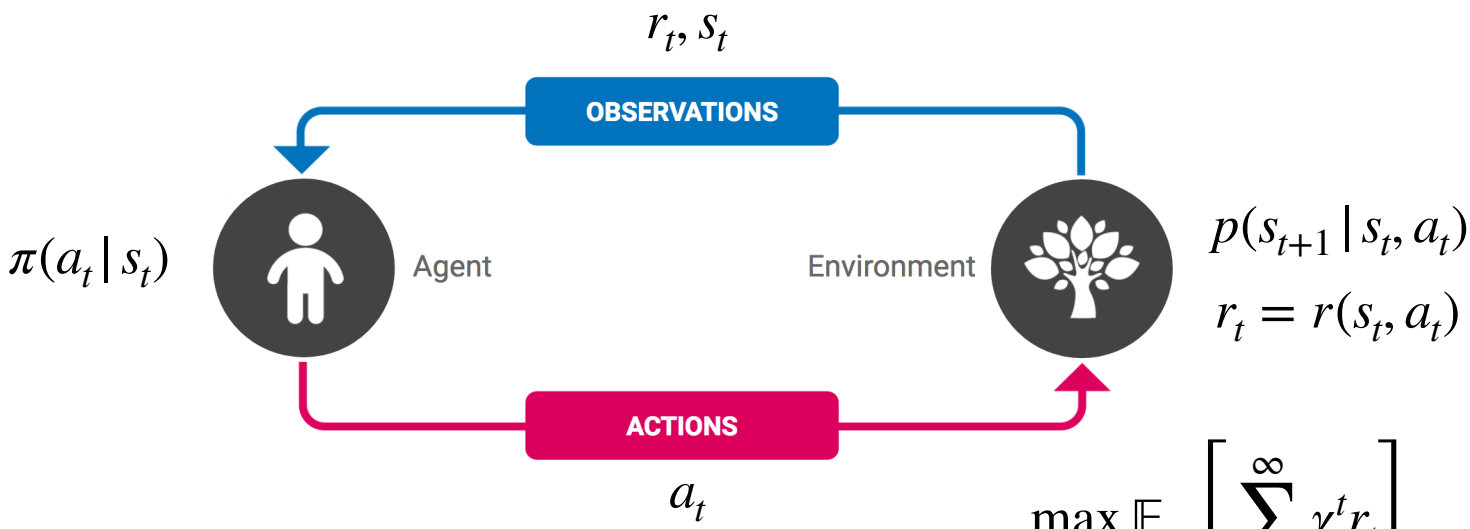


Transferrable Structures

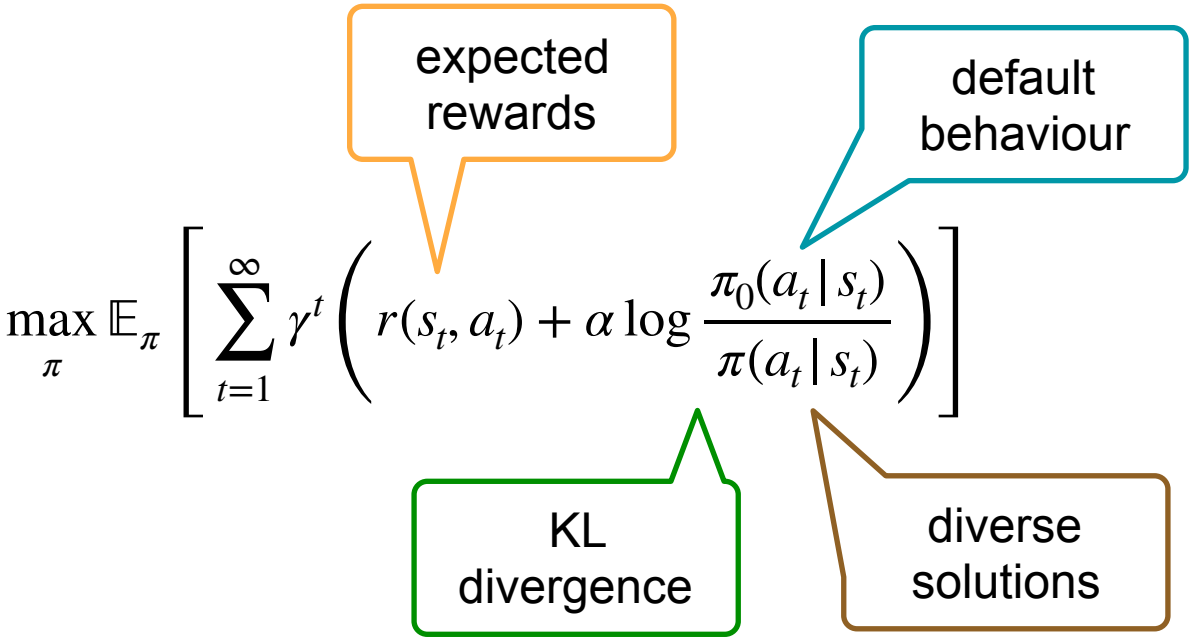
- Transferrable structure in policies/solutions
  - Learning prior policies using KL-regularised RL:
    - [Teh et la NeurIPS 2017] [Galashov et al ICLR 2019] [Tirumala, Noh et al ArXiv 2019]
  - Neural probabilistic motor primitives
    - [Merel, Hasenclever et al ICLR 2019]

- Transferrable structure in environments
  - Meta-learning with neural processes:
    - [Garnelo et al 2018 ICML] [Garnelo et al 2018 ICML WS] [Kim et al ICLR 2019] [Galashov et al ArXiv 2019]

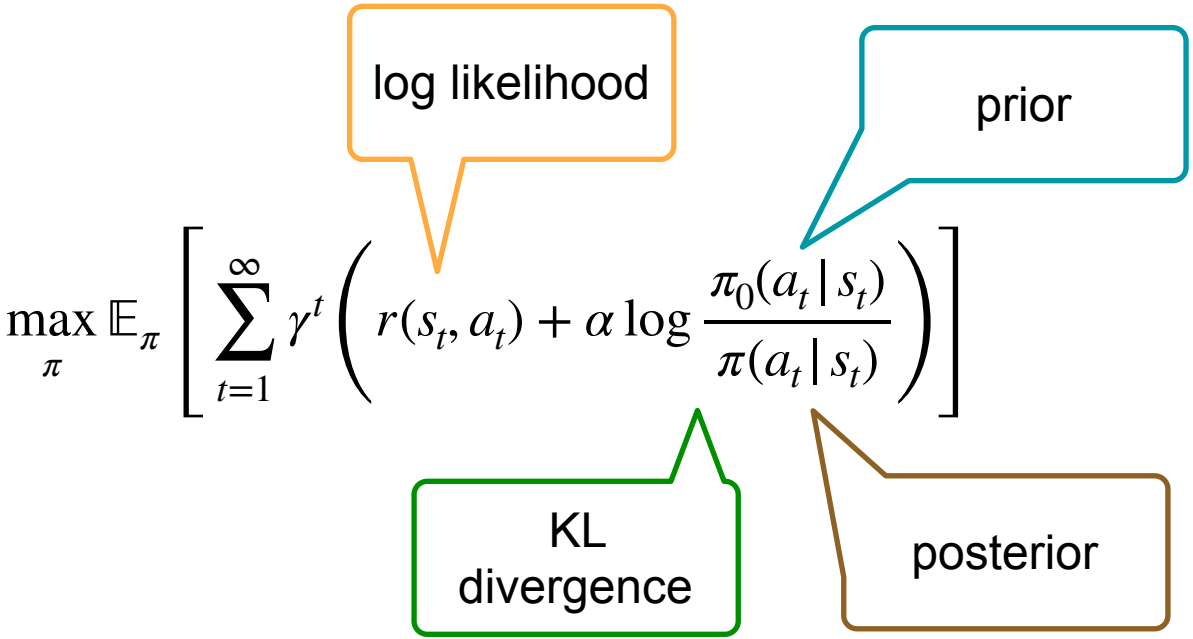
Markov Decision Processes



KL-regularised Reinforcement Learning



KL-regularised Reinforcement Learning



KL-regularised Reinforcement Learning

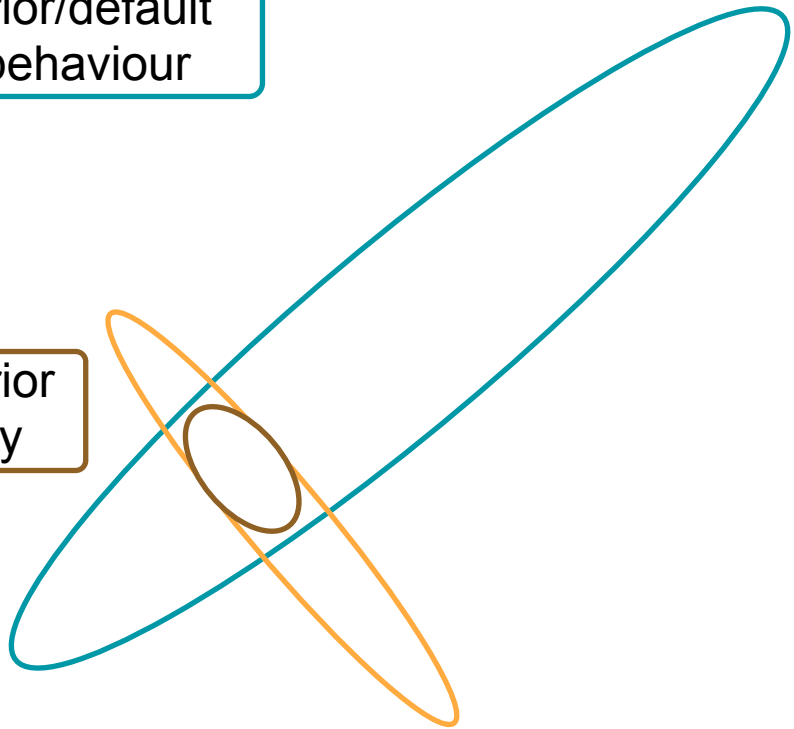
log likelihood/ rewards

prior/default behaviour

$$\max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=1}^{\infty} \gamma^t \left(r(s_t, a_t) + \alpha \log \frac{\pi_0(a_t \mid s_t)}{\pi(a_t \mid s_t)} \right) \right]$$

$$\pi^*(a_t \mid s_t) \propto \pi_0(a_t \mid s_t) \exp(Q^*(a_t \mid s_t))$$

posterior policy



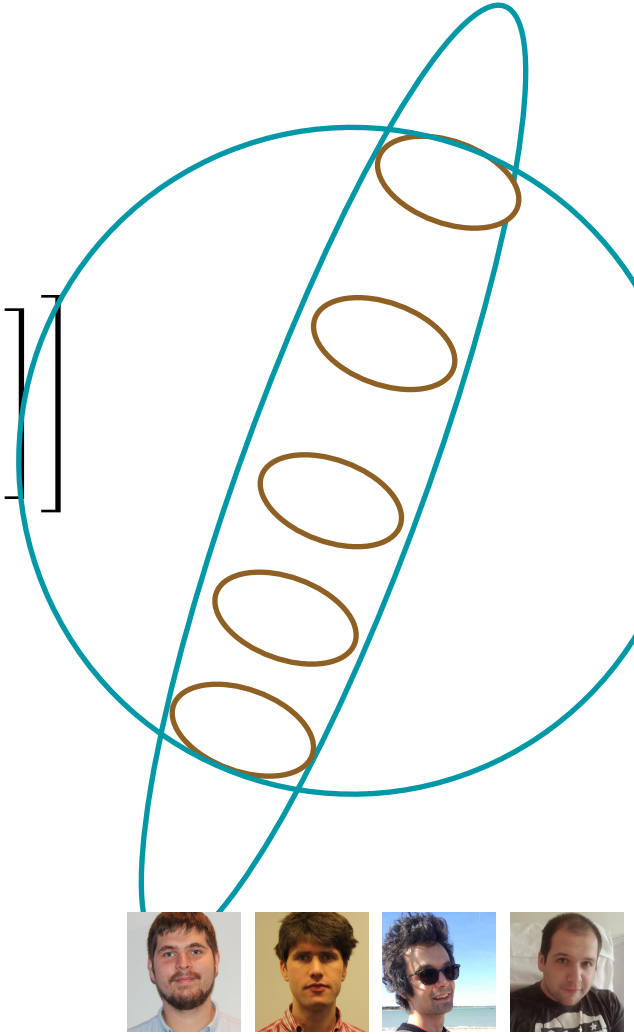
KL-regularised Reinforcement Learning

- Distribution p(w) over tasks w:

$$\max_{\pi} \sum_{w} p(w) \left[\mathbb{E}_{\pi_w} \left[\sum_{t=1}^{\infty} \gamma^t \left(r_w(s_t, a_t) + \alpha \log \frac{\pi_0(a_t \mid s_t)}{\pi_w(a_t \mid s_t)} \right) \right] \right]$$

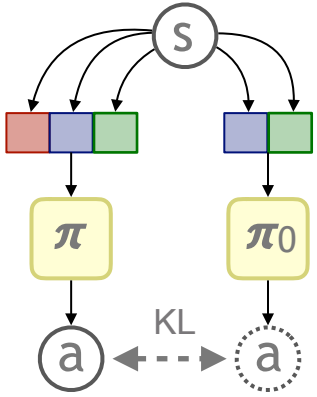
$$\pi_w^*(a_t \mid s_t) \propto \pi_0(a_t \mid s_t) \exp(Q_w^*(a_t \mid s_t))$$

$$\pi_0^*(a_t \mid s_t) = \arg\max_{\pi_0} \sum_{w} p(w) \mathbb{E}_{\pi_w} \left[\sum_{t} \gamma^t \log \pi_0(a_t \mid s_t) \right]$$

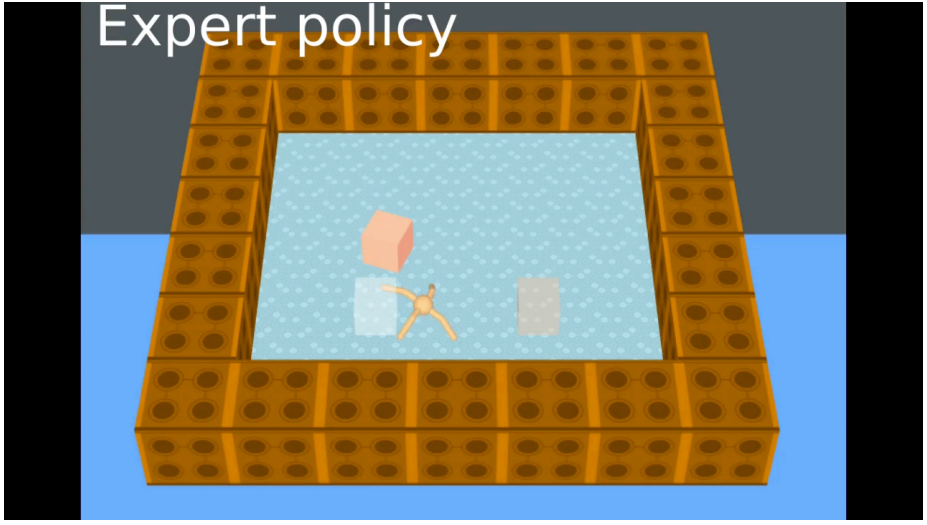


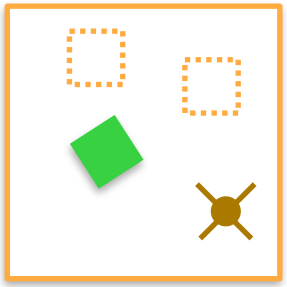
Information asymmetry in KL-regularized RL

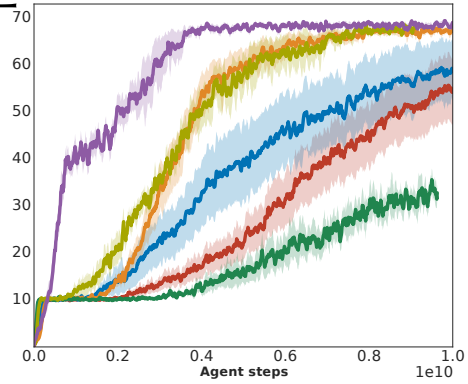
Task w
Exteroceptive
Proprioceptive



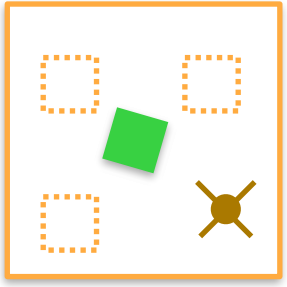
- Default policy trained alongside policy
- Default policy sees partial information
- "Information hiding" forces generalization

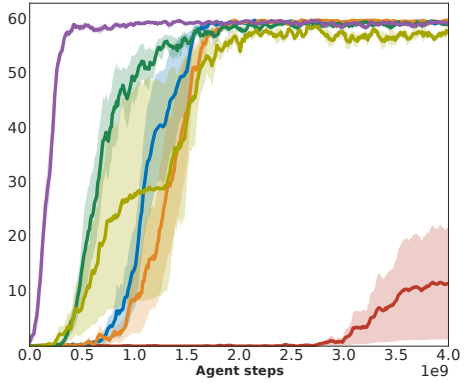


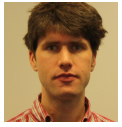




Baseline
Proprio
Proprio+Box
Transfer Proprio
Transfer Proprio+Box





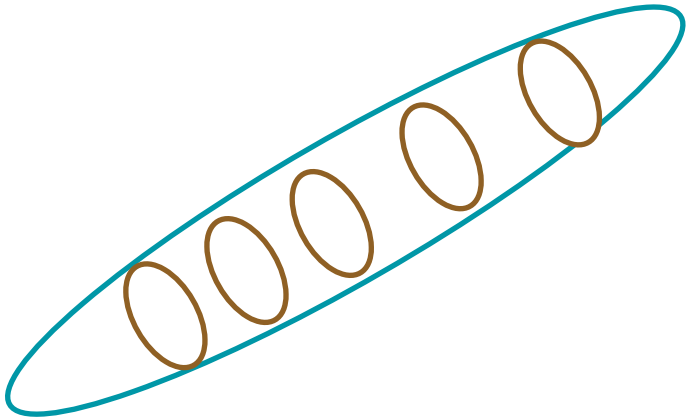


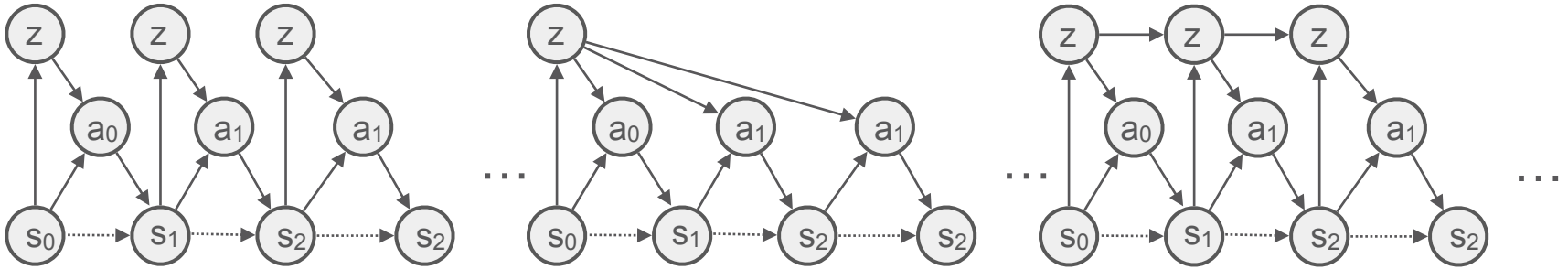
Introducing Structure via Latent Variables

trajectory

$$\pi(\tau) = \int \pi(\tau \mid z) dz$$

latent variables





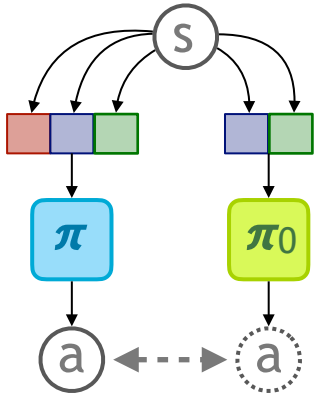
richer action distribution

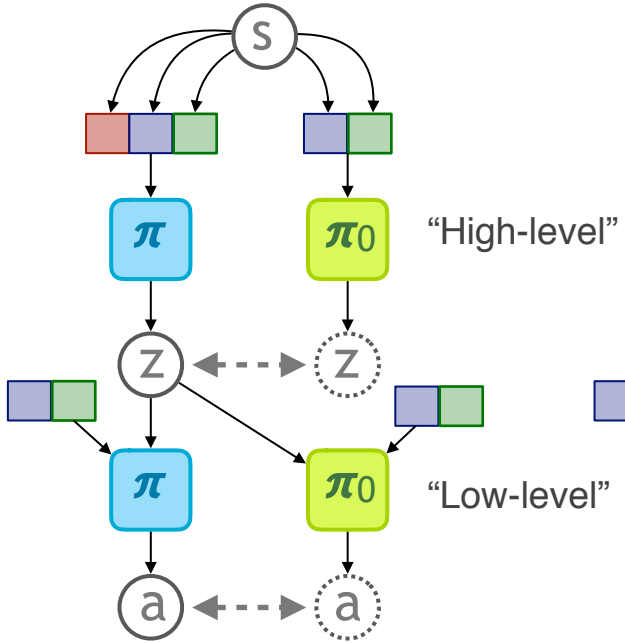
goal conditional policy

temporal correlations — e.g. options

Hierarchical Structure in Policy and Prior

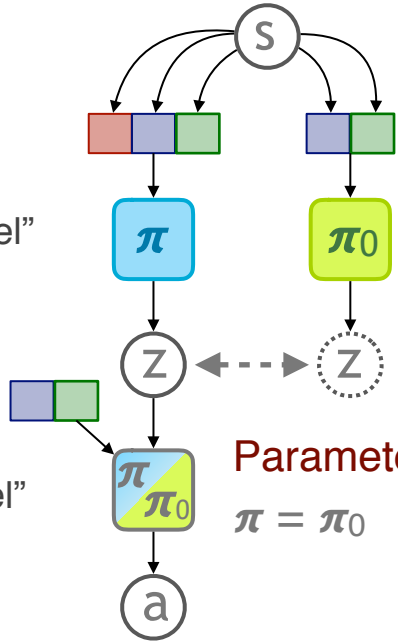
Task
Exteroceptive
Proprioceptive





"High-level"

"Low-level"



Parameter sharing

$\pi = \pi_0$

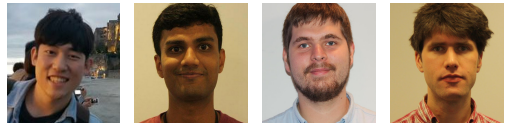
"Reusable LL controller" / "Skills"

$$\mathbb{E}_\pi(\tau)[\textstyle\sum_t r_t] - \mathrm{KL}[\pi(\tau)||\pi_0(\tau)]$$

high-level

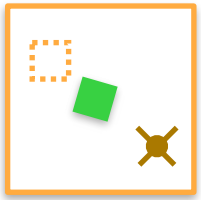
low-level

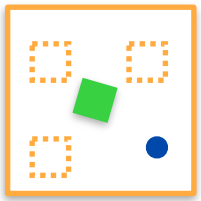
$$\geq \mathbb{E}_\pi(\tau)[\textstyle\sum_t r_t] - \mathrm{KL}[\pi(z)||\pi_0(z)] - \mathrm{KL}[\pi(\tau|z)||\pi_0(\tau|z)]$$

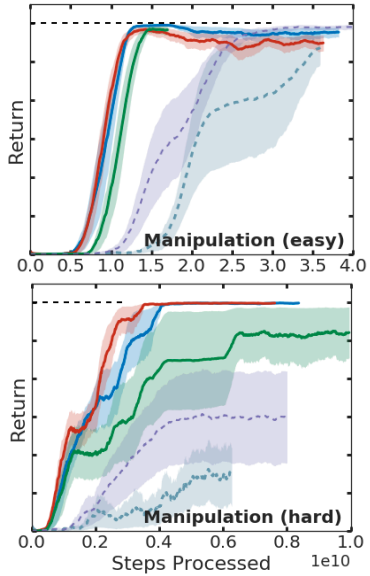


Yee Whye Teh

Hierarchical Structure in Policy and Prior

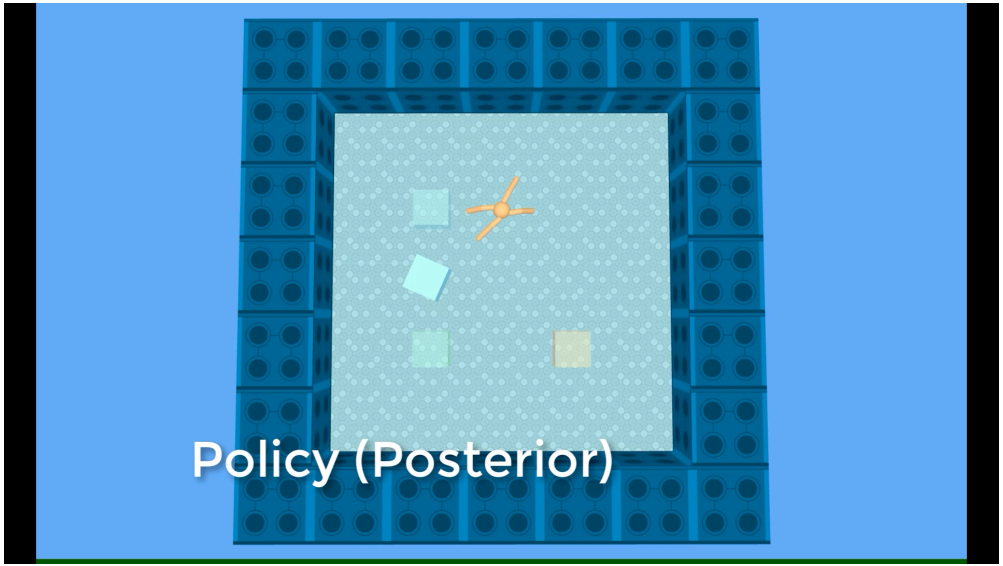




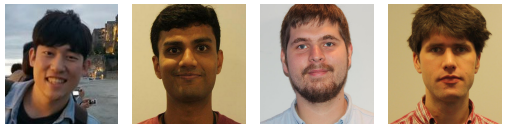


Prior
Hierarchical

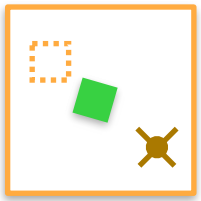
Unstructured
None

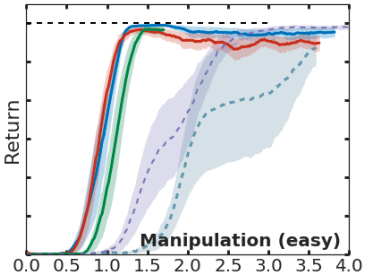


Yee Whye Teh

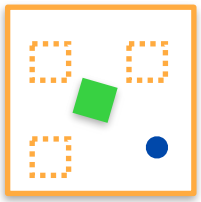


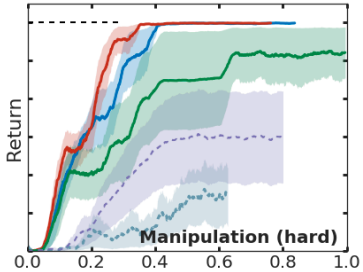
Hierarchical Structure in Policy and Prior

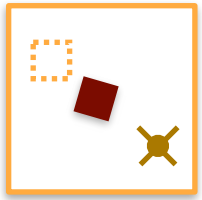


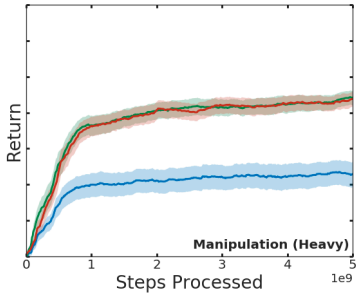


Prior
Hierarchical
Unstructured
None

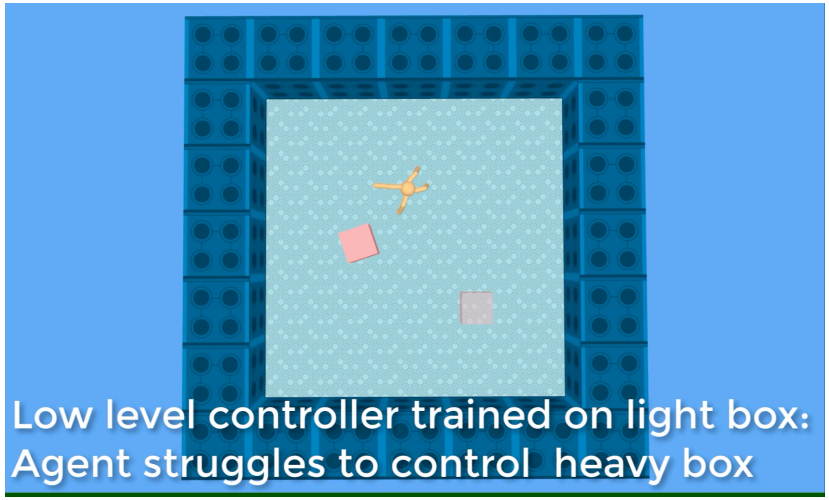


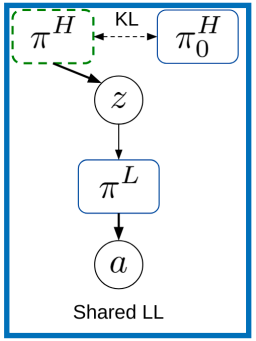


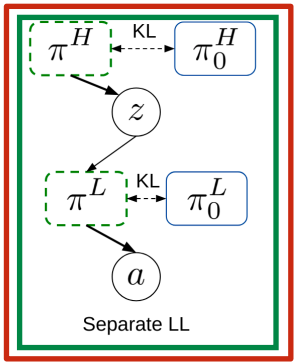


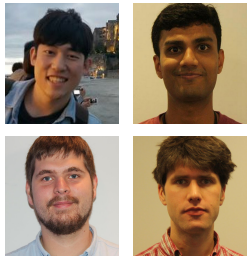


Separate LL
Shared LL



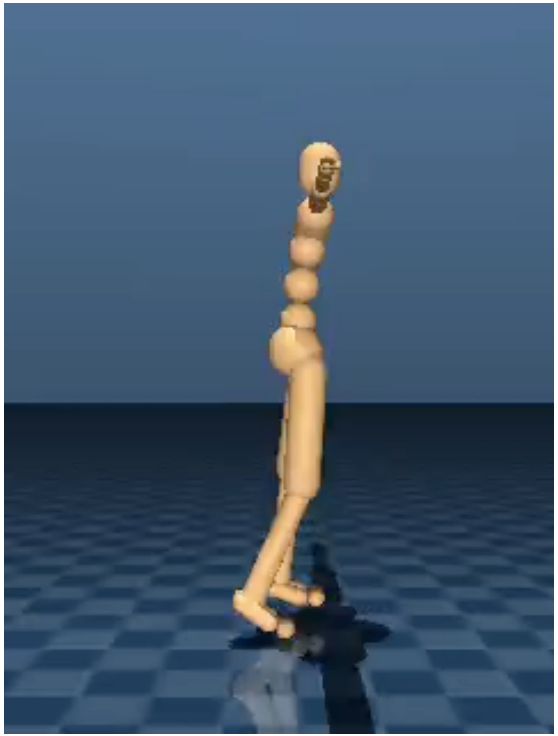




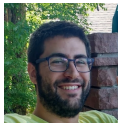


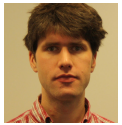
Yee Whye Teh

Neural Probabilisitic Motor Primitives

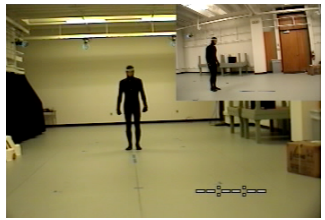


Yee Whye Teh



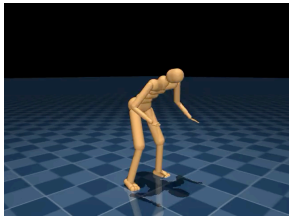


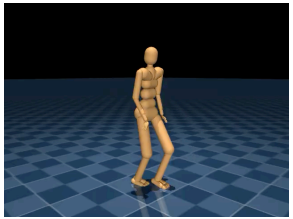
Demonstration

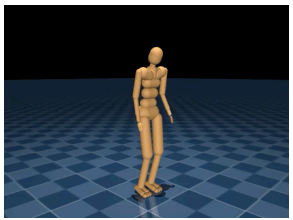


RL

Individual skill



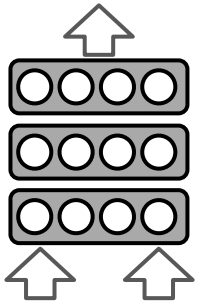




General purpose
"motor module"

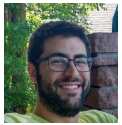
Supervised Learning

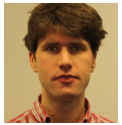
Action a
(Joint torque)



Proprioception
(body pose, s)

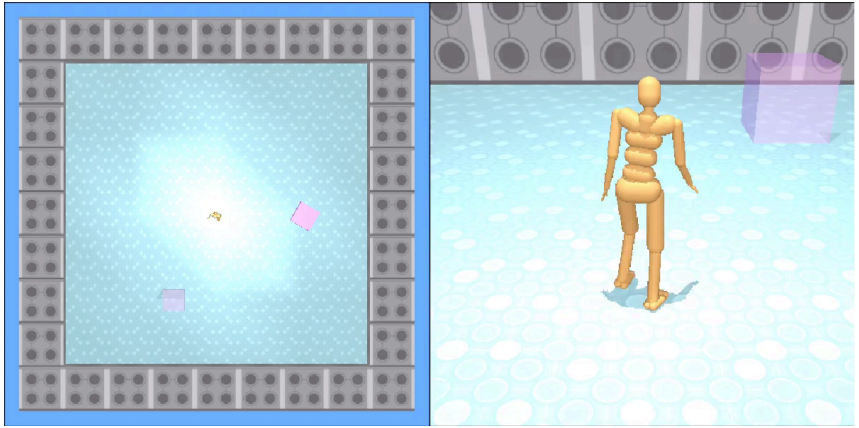
Motor
intention z

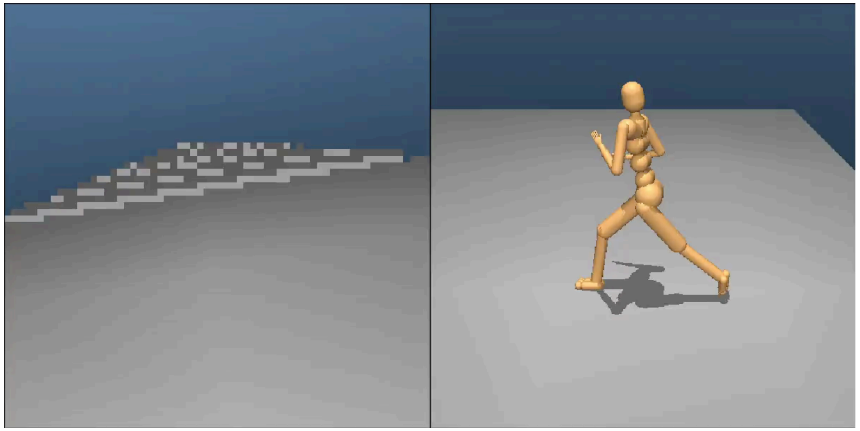


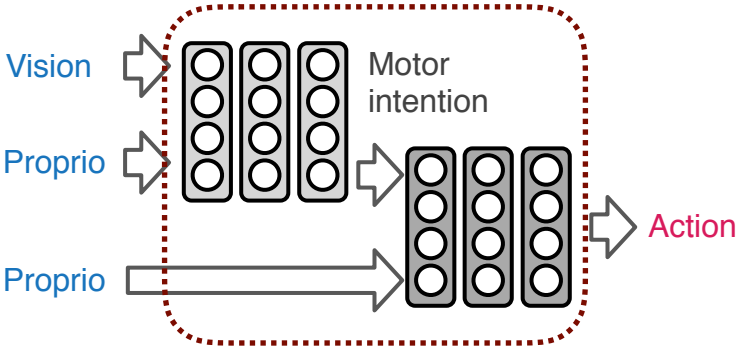


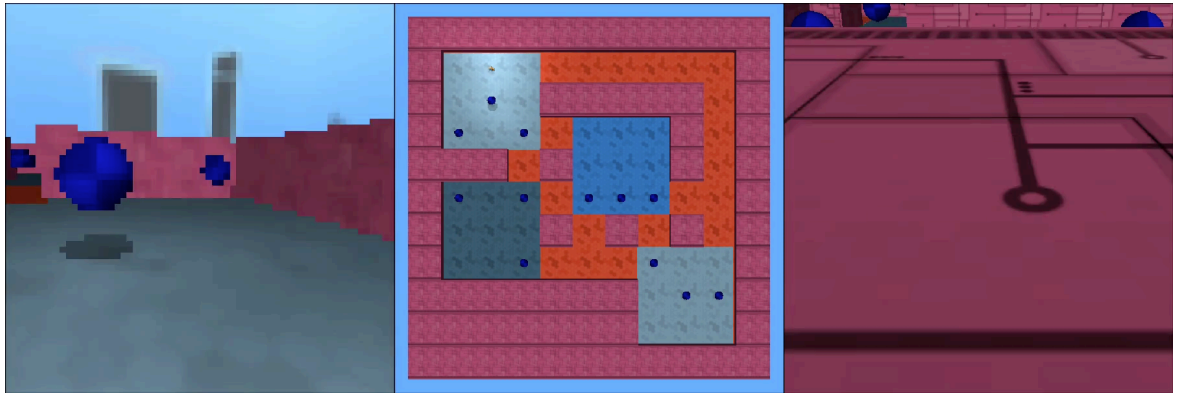
$$\log \pi_0(a_{1:T}|s_{1:T}) = \int \pi_0(a_{1:T}|s_{1:T}, z_{1:T}) p_z(z_{1:T}) dz_{1:T}$$

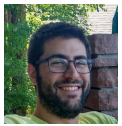
$$\geq \mathbb{E}_q \left[\sum_{t=1}^{T} \log \pi_0(a_t|s_t, z_t) + \beta \left(\log p_z(z_t|z_{t-1}) - \log q(z_t|z_{t-1}, x_t) \right) \right]$$











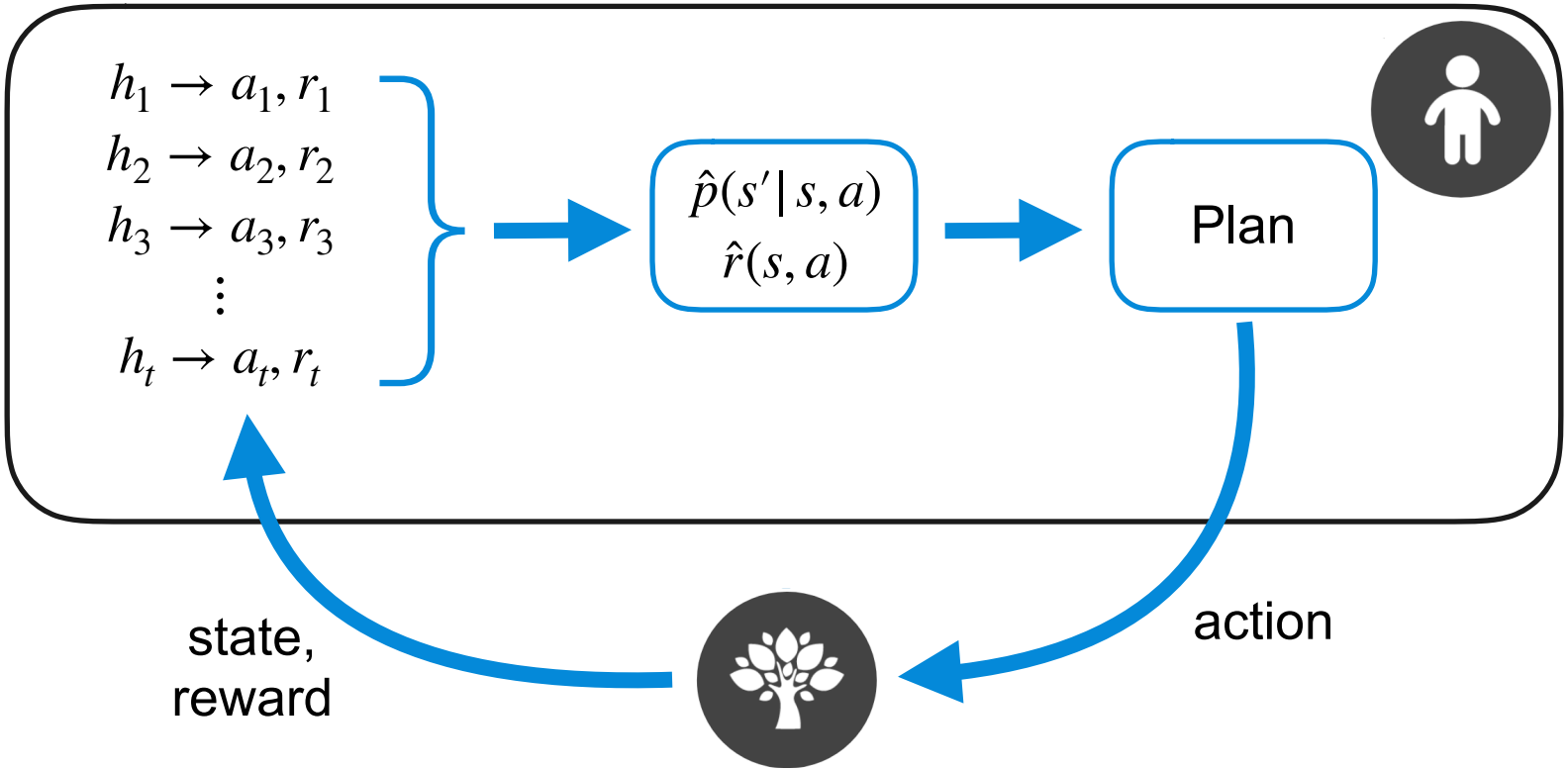
Yee Whye Teh

Transferrable Structures

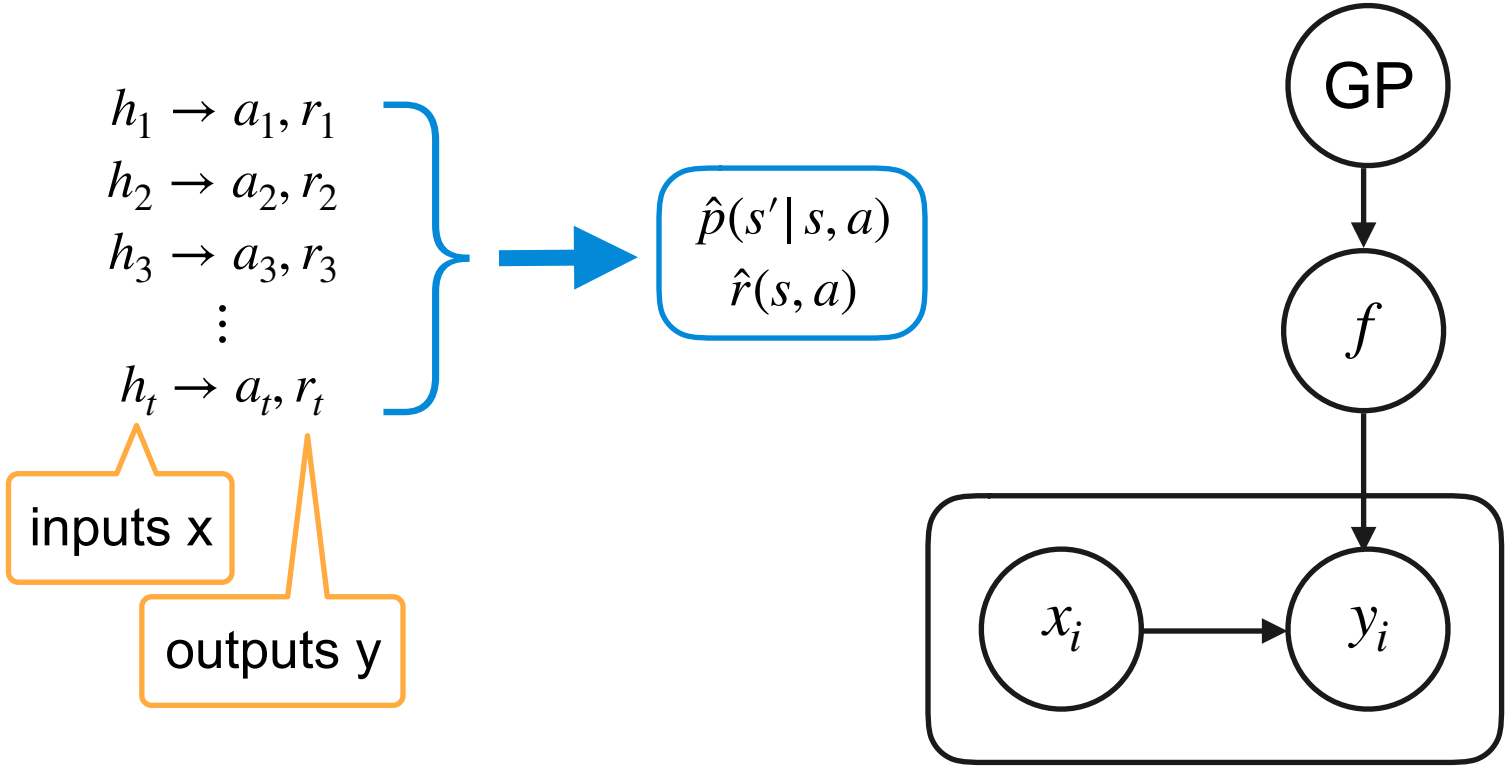
- Transferrable structure in policies/solutions
  - Learning prior policies using KL-regularised RL:
    - [Teh et la NeurIPS 2017] [Galashov et al ICLR 2019] [Tirumala, Noh et al ArXiv 2019]
  - Neural probabilistic motor primitives
    - [Merel, Hasenclever et al ICLR 2019]

- Transferrable structure in environments
  - Meta-learning with neural processes:
    - [Garnelo et al 2018 ICML] [Garnelo et al 2018 ICML WS] [Kim et al ICLR 2019] [Galashov et al ArXiv 2019]

Model-based Reinforcement Learning

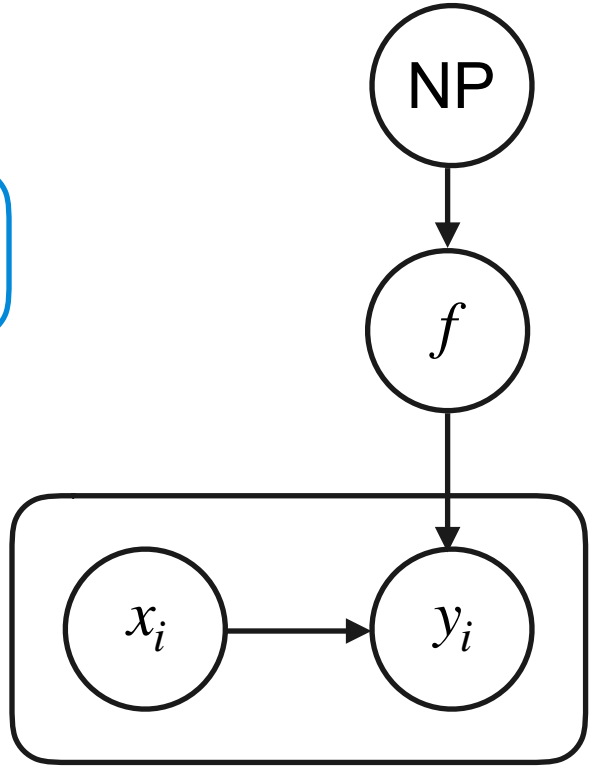


Model-based Reinforcement Learning



Model-based Reinforcement Learning

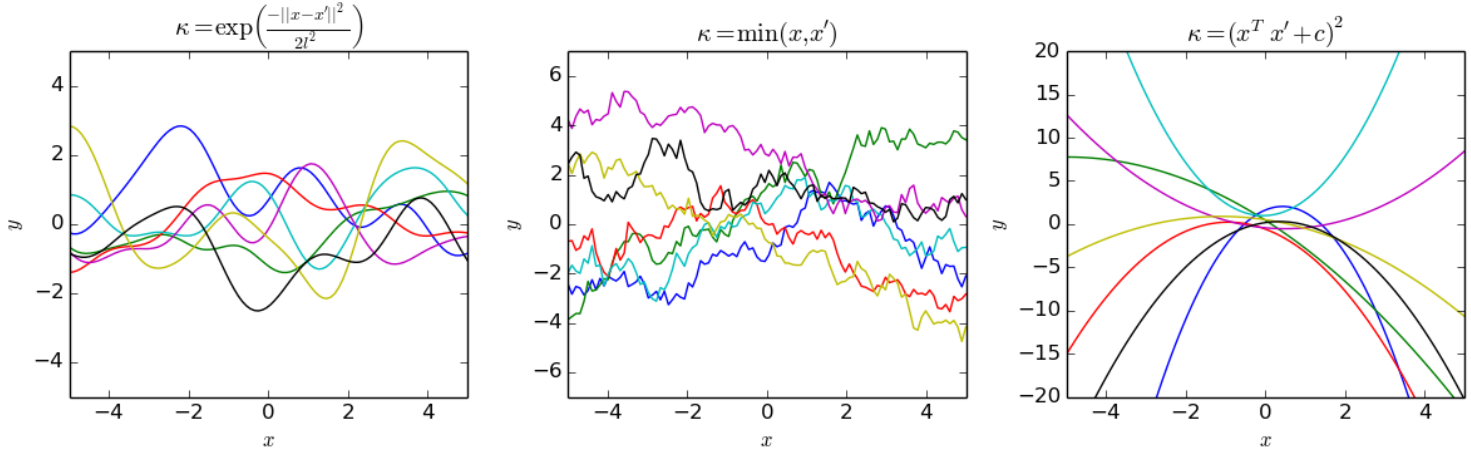
$$
\begin{aligned}
h_1 &\rightarrow a_1, r_1 \\
h_2 &\rightarrow a_2, r_2 \\
h_3 &\rightarrow a_3, r_3 \\
&\vdots \\
h_t &\rightarrow a_t, r_t
\end{aligned}
\quad \Rightarrow \quad
\begin{aligned}
&\hat{p}(s' \mid s, a) \\
&\hat{r}(s, a)
\end{aligned}
$$



Specifying Stochastic Processes

- Gaussian processes are typically described via marginal distributions:

$$\begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_t) \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu(x_1) \\ \mu(x_2) \\ \vdots \\ \mu(x_t) \end{pmatrix}, \begin{pmatrix} K(x_1,x_1) & K(x_1,x_2) & \cdots & K(x_1,x_t) \\ K(x_2,x_1) & K(x_2,x_2) & \cdots & K(x_2,x_t) \\ \vdots & & \ddots & \vdots \\ K(x_t,x_1) & K(x_t,x_2) & \cdots & K(x_t,x_t) \end{pmatrix} \right)$$



Specifying Stochastic Processes

- Gaussian processes can equivalently be described via its conditional distributions:

$$f(x_{t+1})|f(x_1) = y_1, \ldots, f(x_t) = y_t$$
$$\sim \mathcal{N}\left(\mu(x_{t+1}) + K_{t+1,1:t}K^{-1}_{1:t,1:t}y_{1:t},\ K_{t+1,t+1} - K_{t+1,1:t}K^{-1}_{1:t,1:t}K_{1:t,t+1}\right)$$

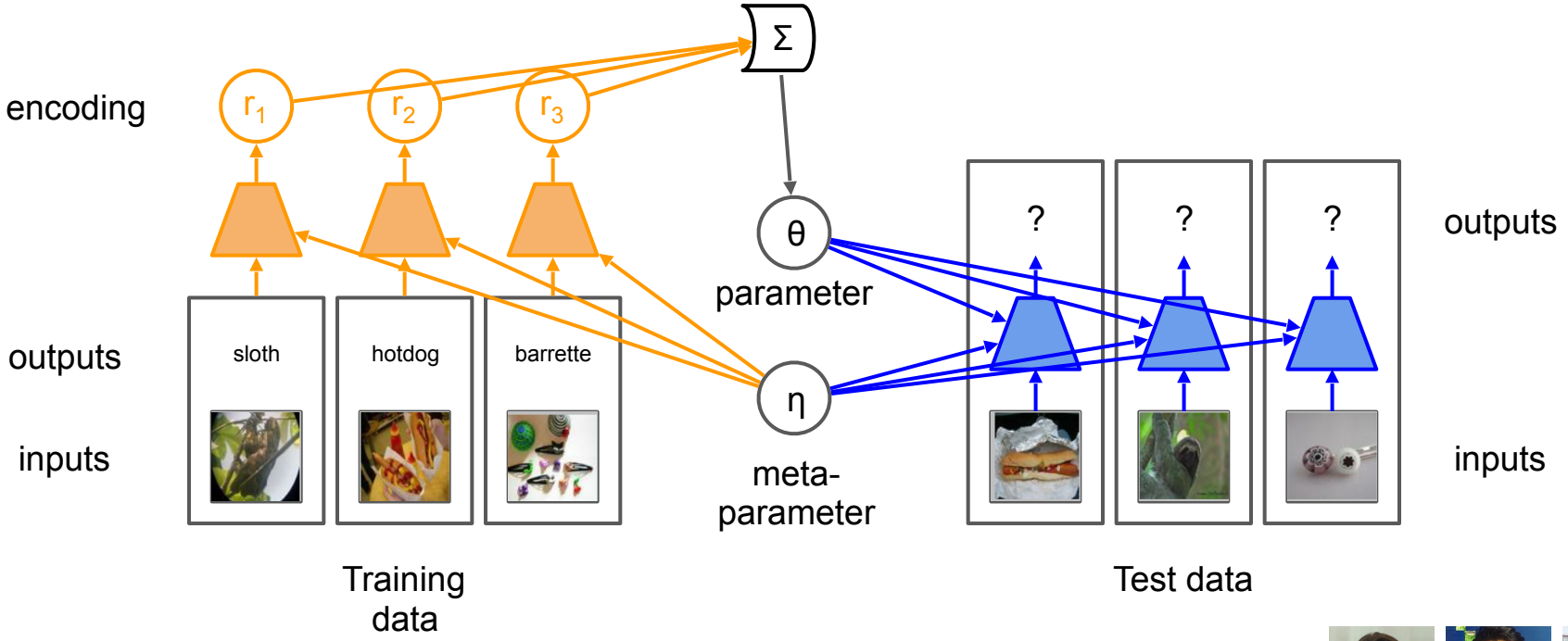
- In general, stochastic processes can also be described using a consistent family of conditional distributions:

$$\mathbb{P}(f(x_{t+1}) = y_{t+1}|f(x_1) = y_1, \ldots, f(x_t) = y_t)$$

for training sets $\{x_{1:t}, y_{1:t}\}$ and test sets $\{x_{t+1}, y_{t+1}\}$.

Learning Neural Stochastic Processes

- Use a neural network to parameterise the conditional distributions.



Learning Neural Stochastic Processes

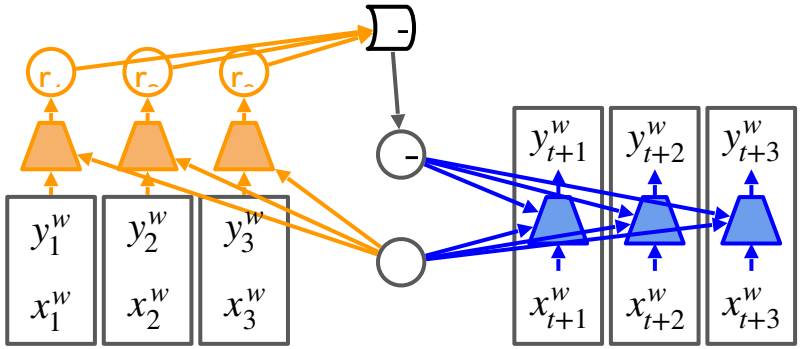
$$x_1^1 \to y_1^1 \quad \vdots \quad x_t^1 \to y_t^1 \quad x_{t+1}^1 \to y_{t+1}^1 \quad \vdots \quad x_{t+s}^1 \to y_{t+s}^1$$

$$x_1^2 \to y_1^2 \quad \vdots \quad x_t^2 \to y_t^2 \quad x_{t+1}^2 \to y_{t+1}^2 \quad \vdots \quad x_{t+s}^2 \to y_{t+s}^2$$

$$x_1^3 \to y_1^3 \quad \vdots \quad x_t^3 \to y_t^3 \quad x_{t+1}^3 \quad x_{t+s}^3$$

$$x_1^4 \to y_1^4 \quad \vdots \quad x_t^4 \to y_t^4 \quad x_{t+1}^4 \to y_{t+1}^4 \quad \vdots \quad x_{t+s}^4 \to y_{t+s}^4$$

$$x_1^w \to y_1^w \quad \vdots \quad x_t^w \to y_t^w \quad x_{t+1}^w \to y_{t+1}^w \quad \vdots \quad x_{t+s}^w \to y_{t+s}^w$$



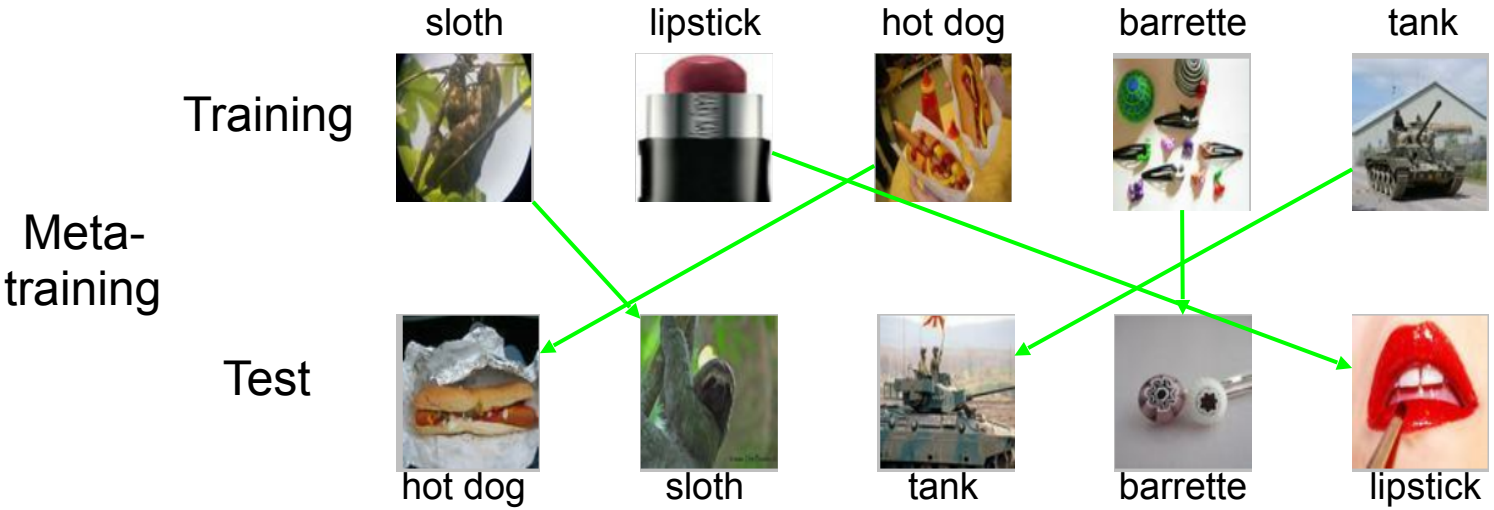
$$\max_{\eta} \sum_{w} p(w) \sum_{j=1}^{s} \log p_\eta(y_{t+j}^w \mid x_{t+j}^w, \{x_i^w, y_i^w\}_{i=1}^t)$$

- A probabilistic perspective on meta-learning.

Few Shot Image Classification

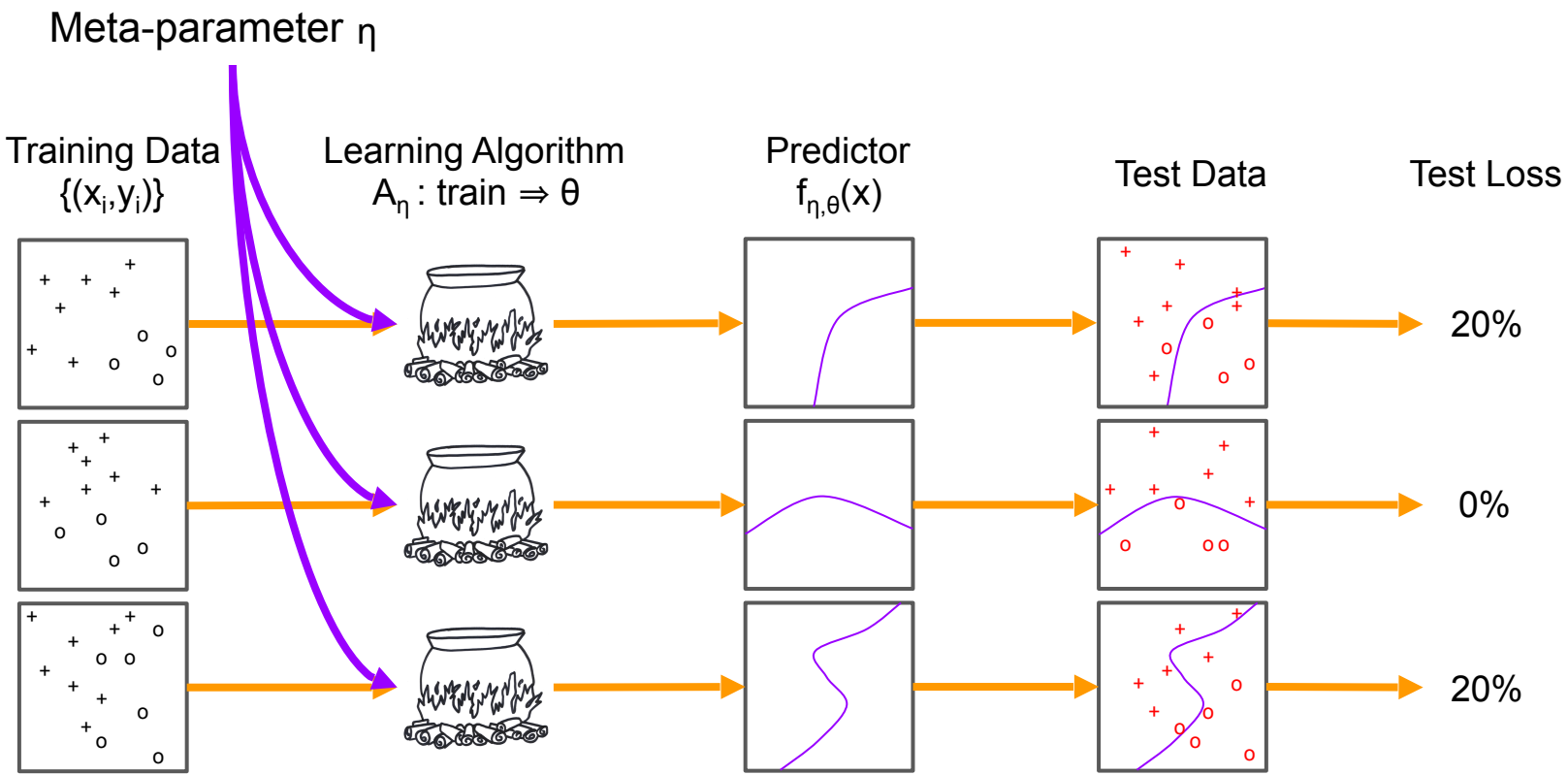
Credit: Andrei Rusu, ImageNet

Few Shot Image Classification via Meta-Learning

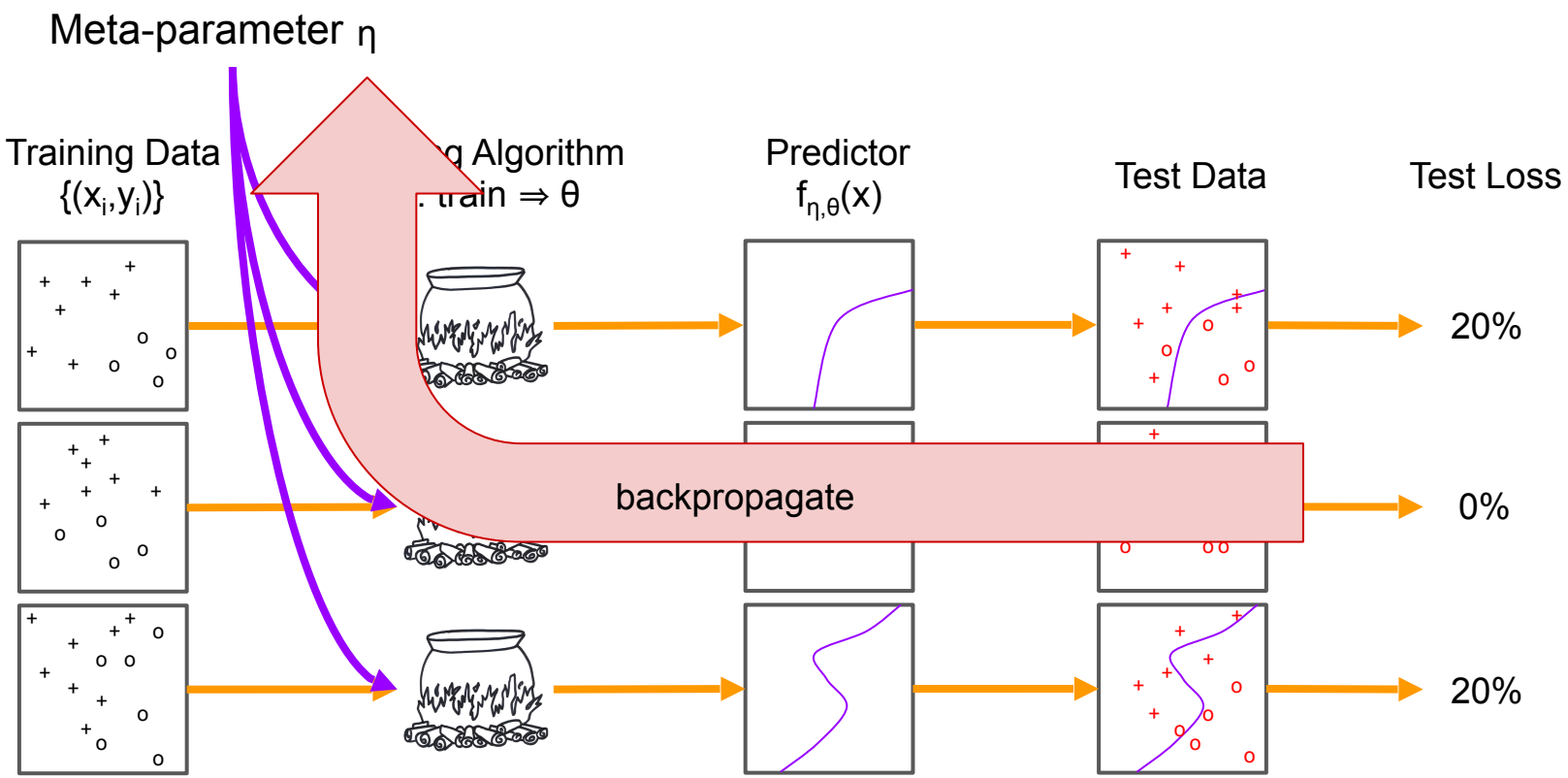


Credit: Andrei Rusu, miniImageNet

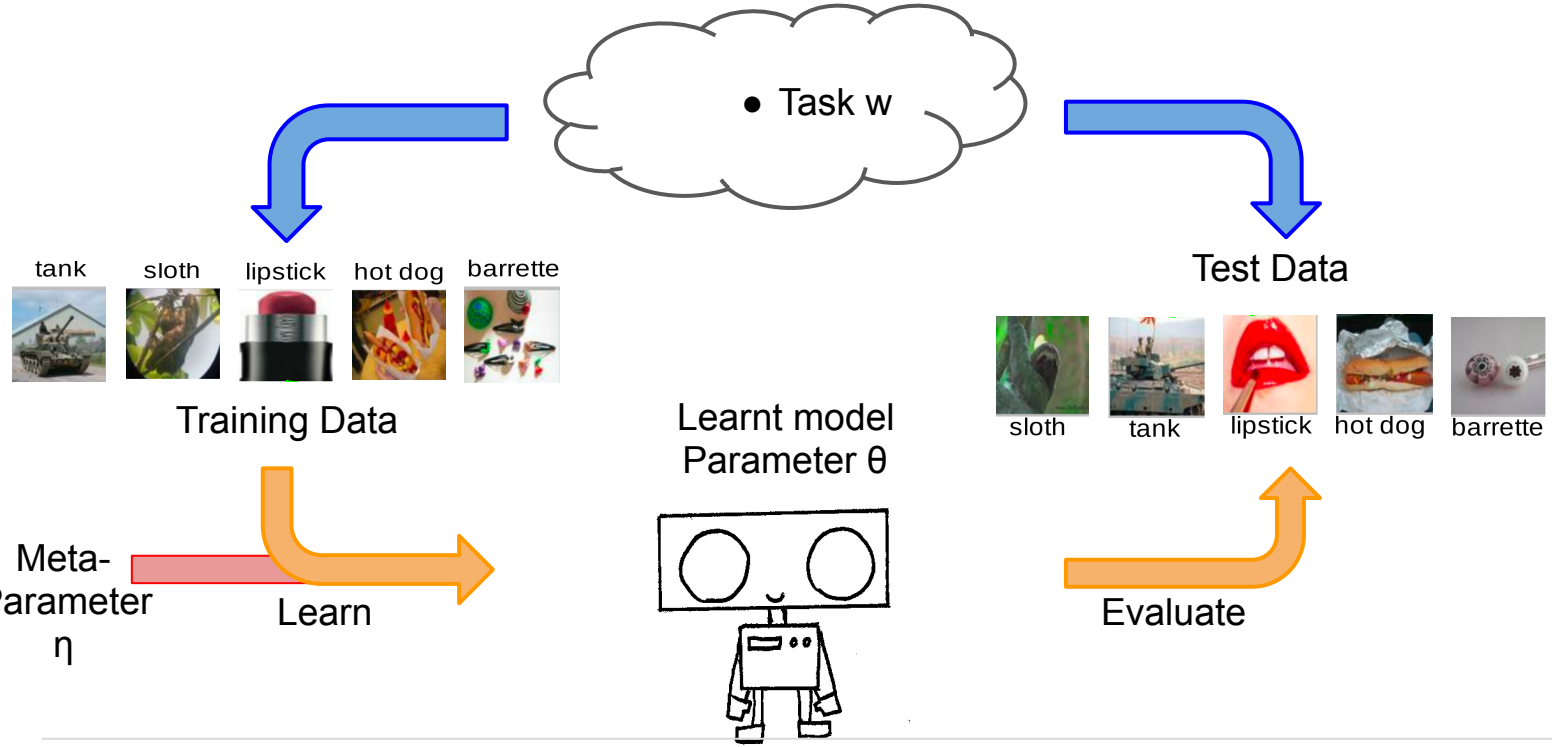
Optimization Perspective on Meta-Learning



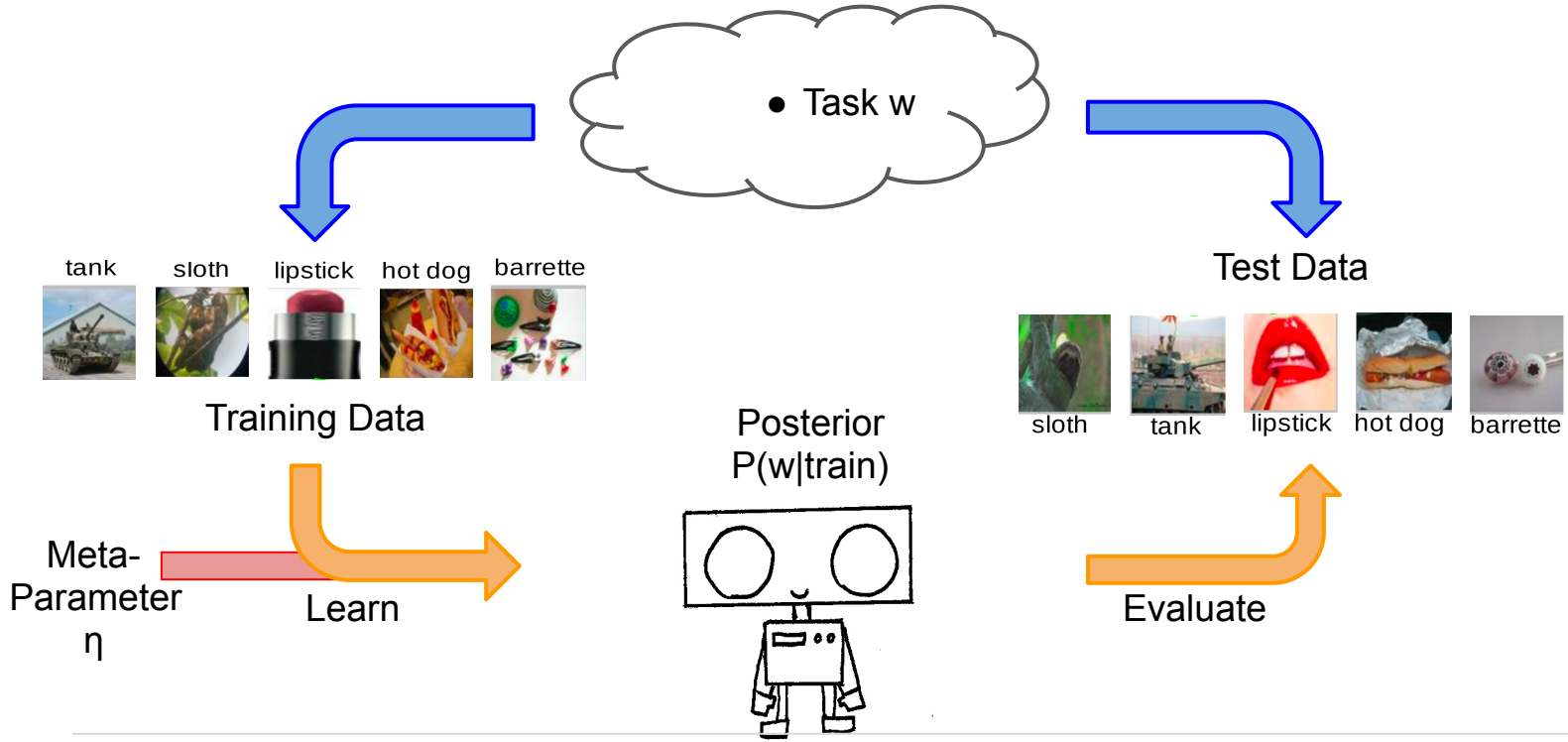
Optimization Perspective on Meta-Learning



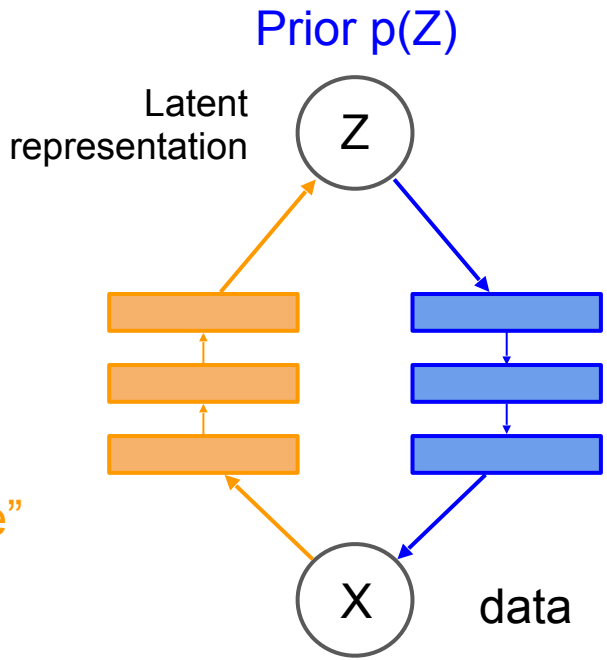
Optimization Perspective on Meta-Learning



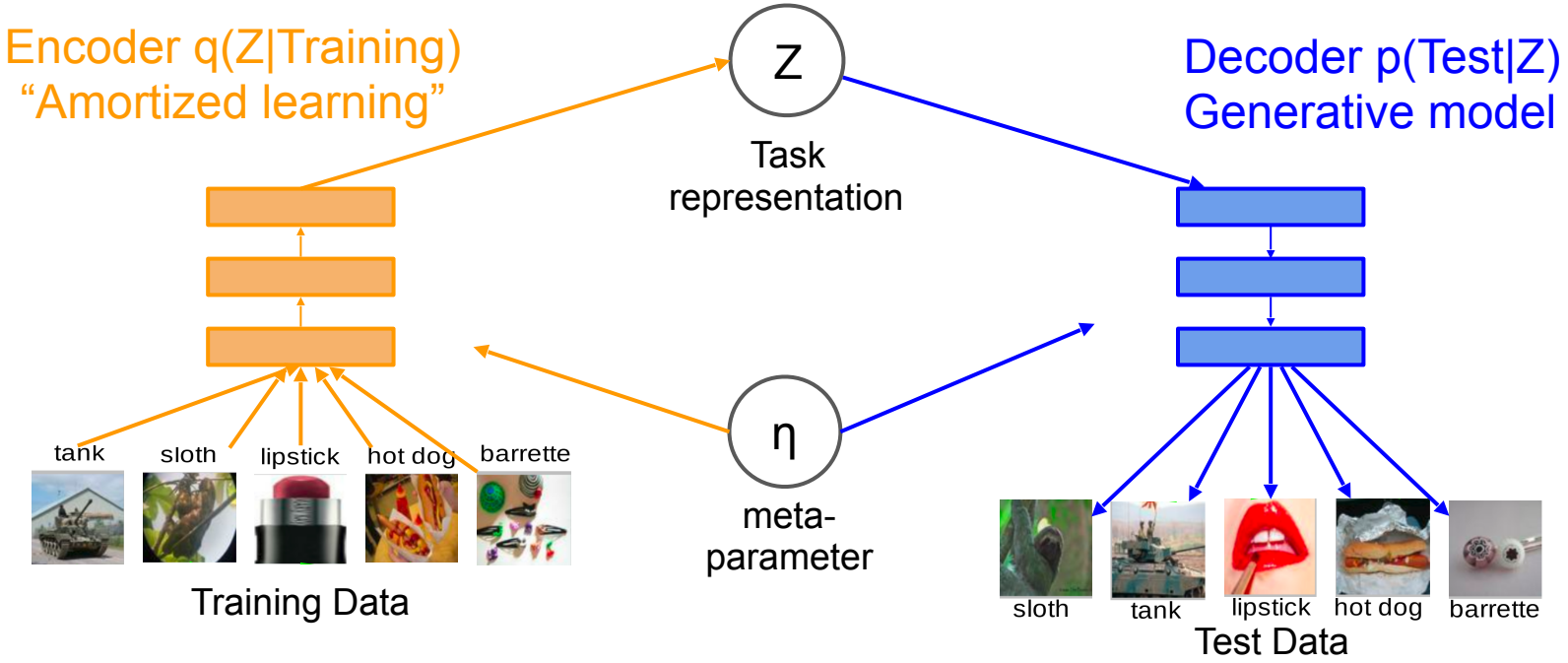
Probabilistic Perspective on Meta-Learning



Variational Auto-Encoders

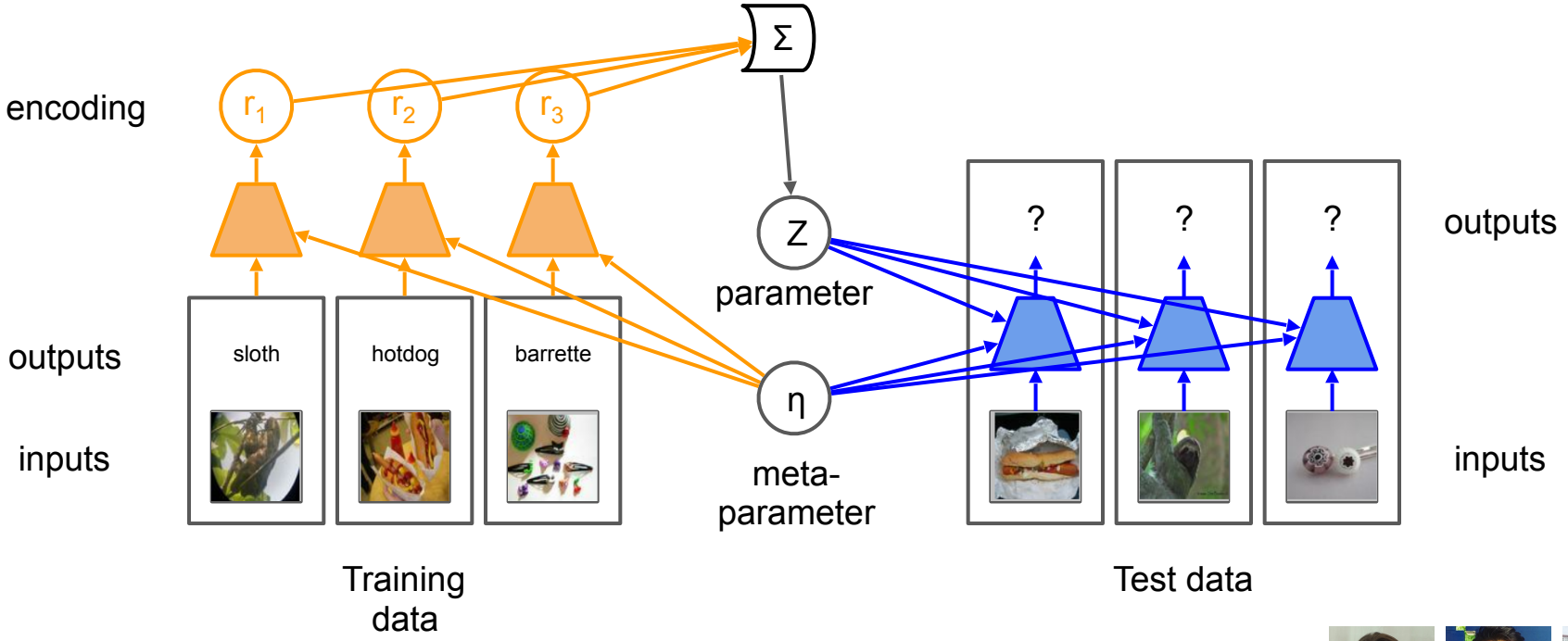


Probabilistic Model for Meta-Learning



Yee Whye Teh

Neural Processes

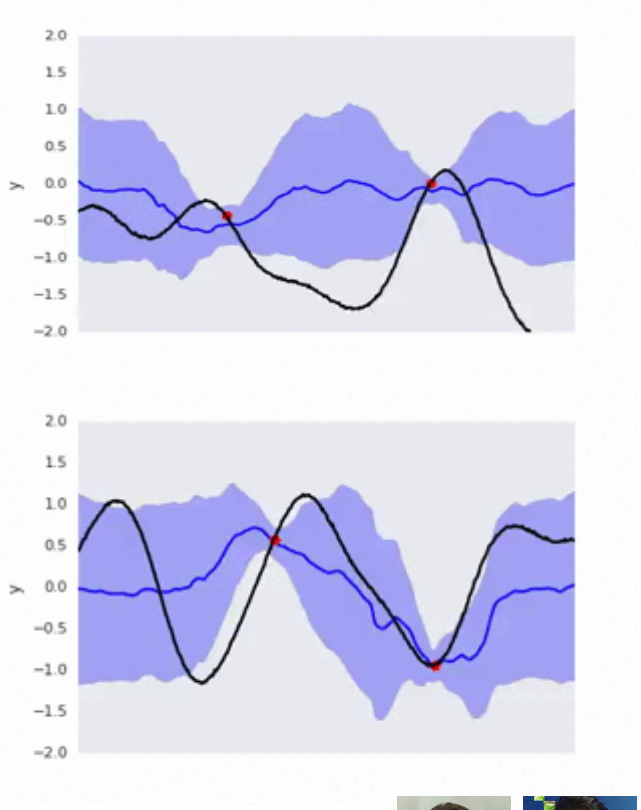


Yee Whye Teh

Neural Processes

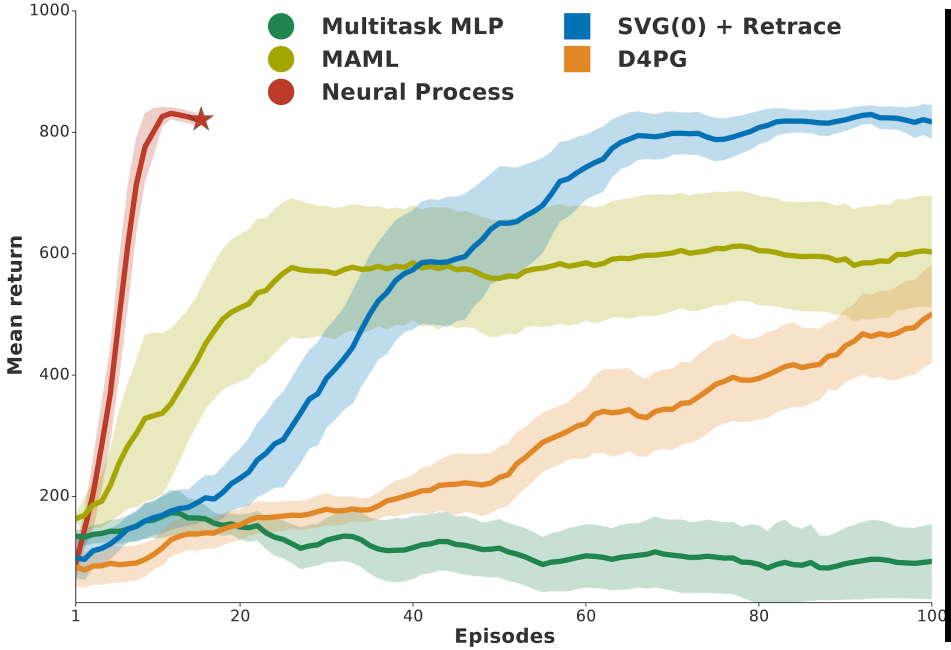
Task = Function on 1D space.

Given training points, use neural processes to predict mean and std of function values at other locations.



Yee Whye Teh

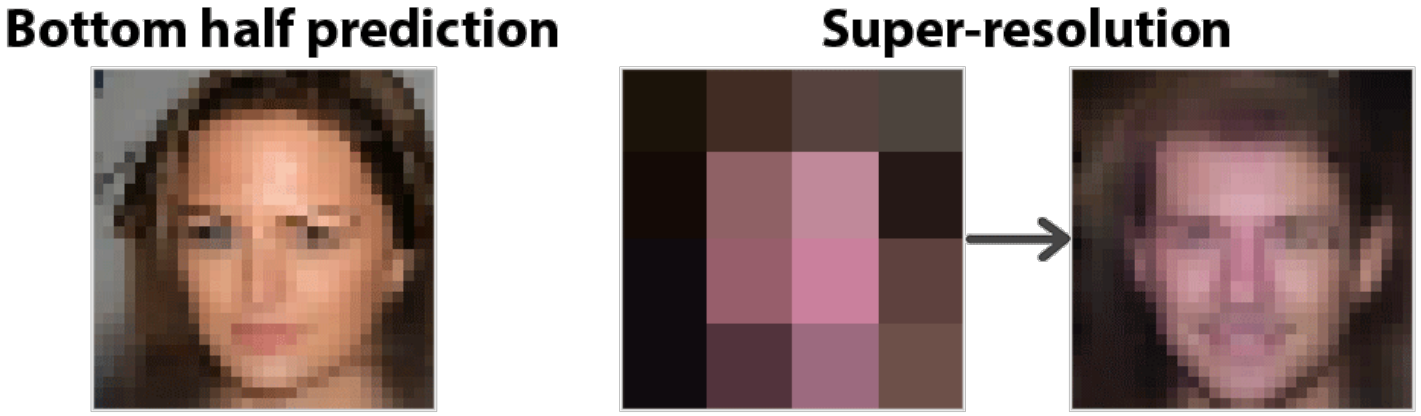
Cart Pole



Yee Whye Teh

Image Super-resolution

Task = Image = Function on 2D space.



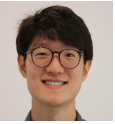
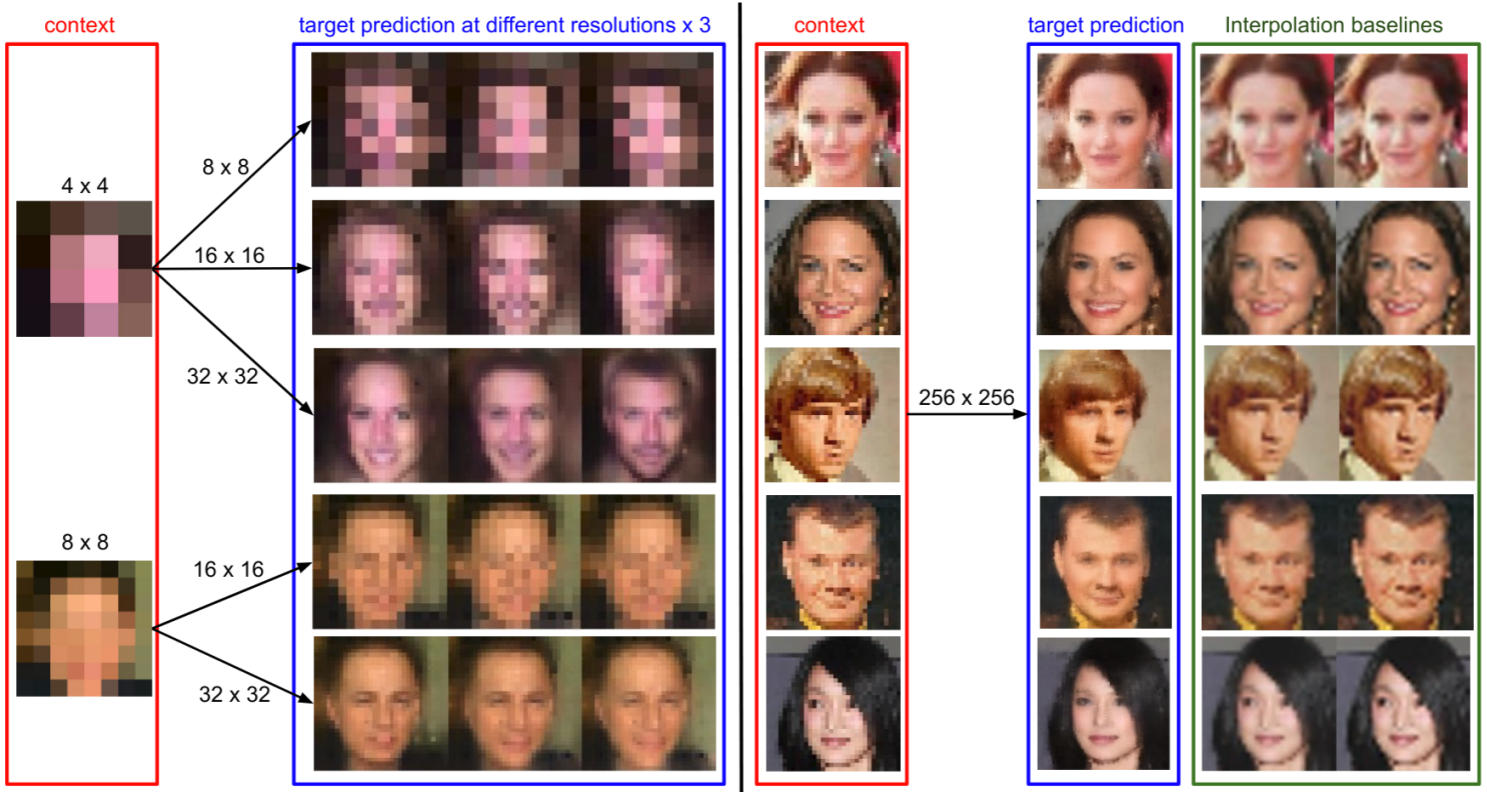
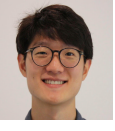
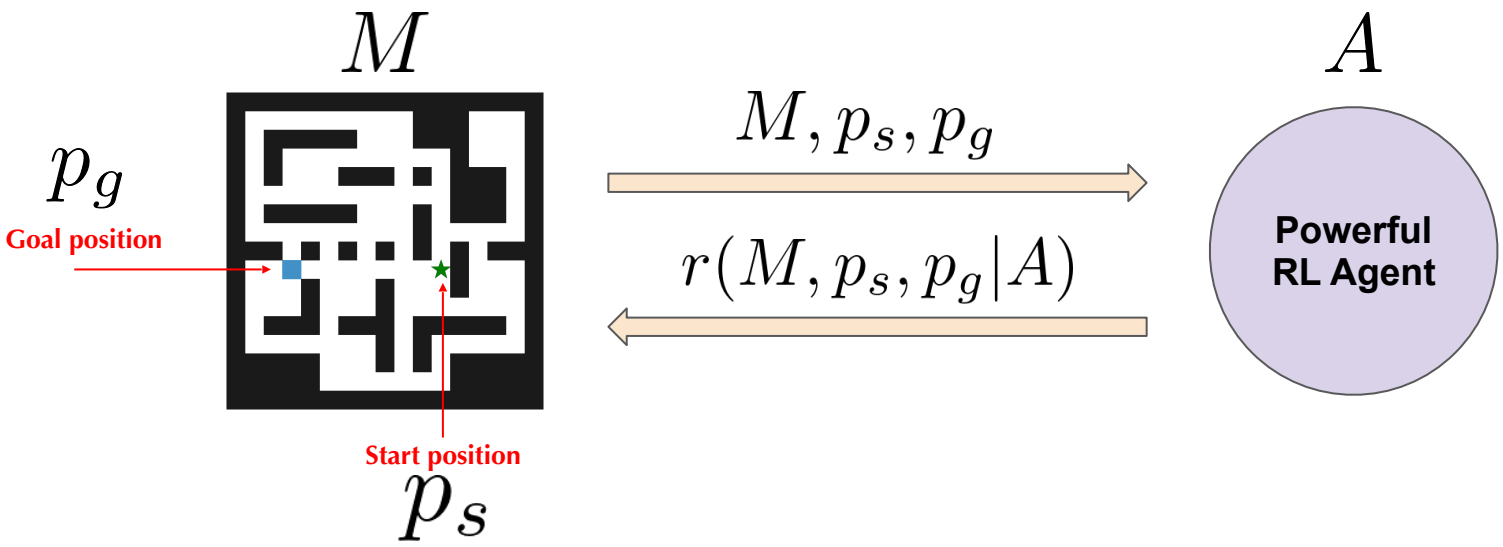


Image Super-resolution

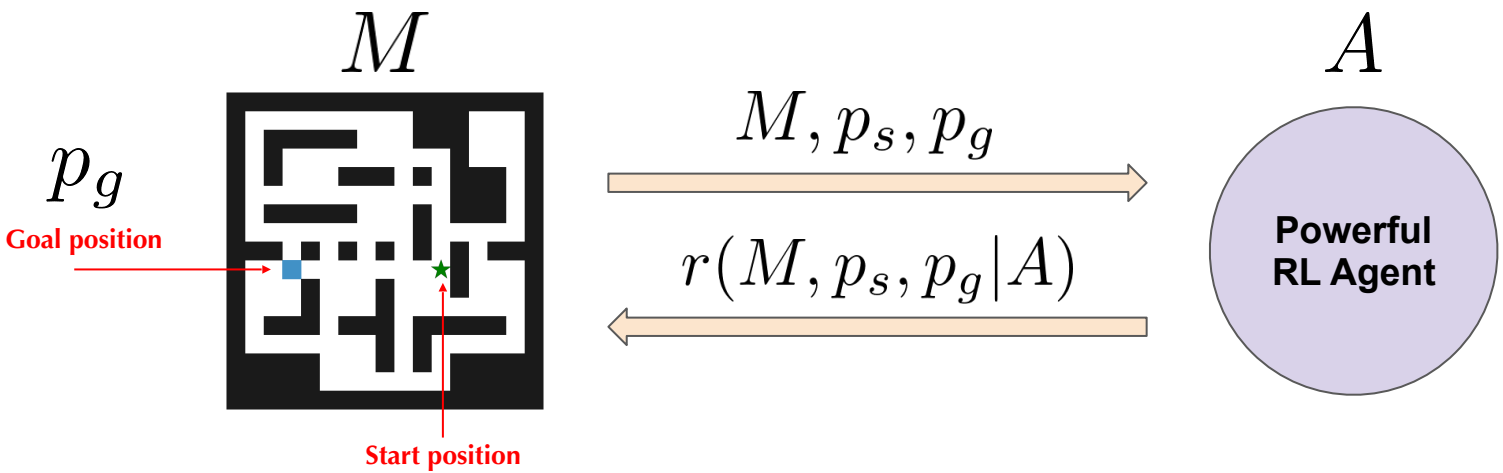




Adversarial testing of RL agents



Adversarial testing of RL agents

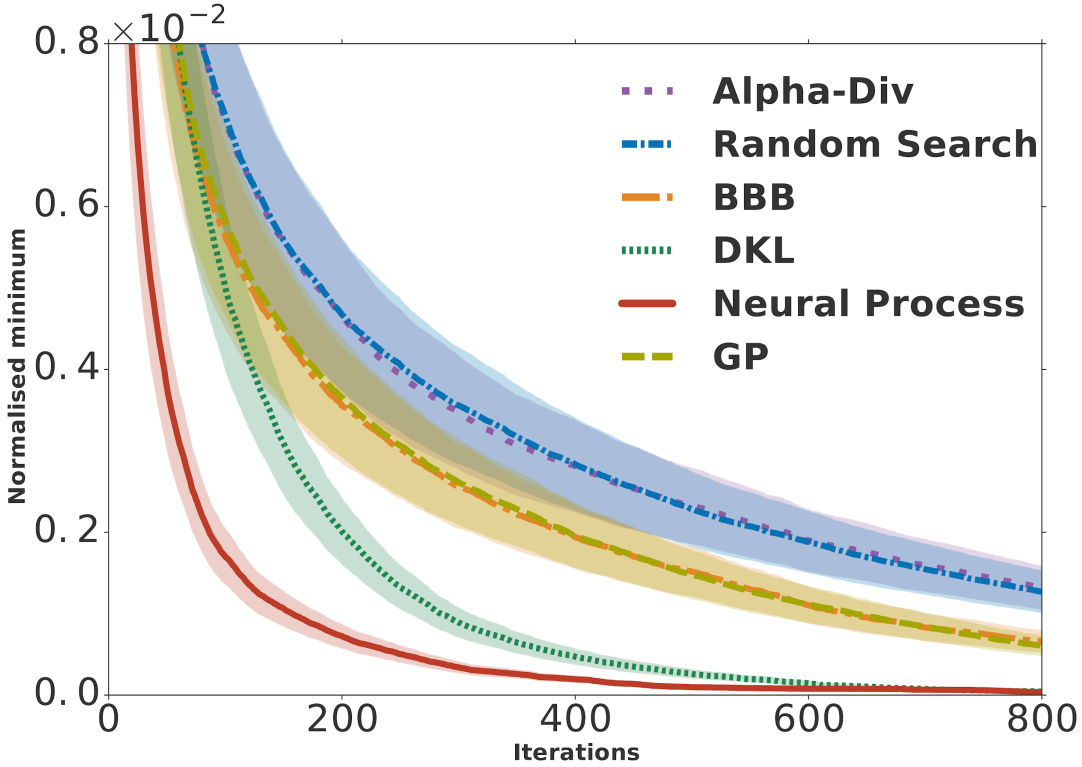


Bayesian Optimizaion

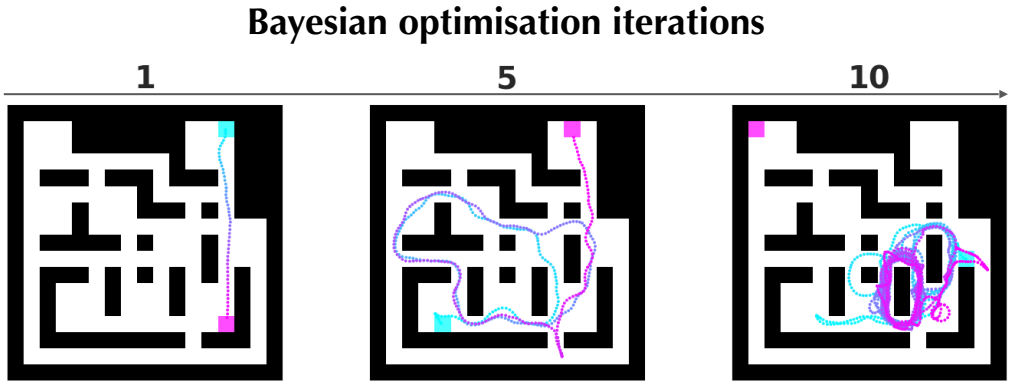
$$\min_{M, p_s, p_g} \; r(M, p_s, p_g|A)$$

$(M, p_s, p_g, A) \sim p(\mathcal{T})$ - training & holdout samples (agents, mazes, positions)

Bayesian Optimization performance



Examples of performance decrease



Summary

- Prior knowledge/inductive biases are necessary for fast learning.
  - Knowledge about what is in the environment
  - Knowledge about how to solve tasks
- Sources of prior knowledge:
  - Features, losses, architectures
  - Data augmentation
  - Data from other modalities
  - Related tasks

Thank You!

- Distral: Robust Multitask Reinforcement Learning. Teh et al NeurIPS 2017.
- Information asymmetry in KL-regularized RL. Galashov et al ICLR 2019. arXiv:1905.01240
- Exploiting Hierarchy for Learning and Transfer in KL-regularized RL. Tirumala, Noh et al 2019. arXiv:1903.07438
- Neural Probabilistic Motor Primitives for Humanoid Control. Merel, Hasenclever et al ICLR 2019. arXiv:1811.11711
- Conditional Neural Processes. Garnelo et al. ICML 2018. arXiv:1807.01613
- Neural Processes. Garnelo et al. ICML 2018 Workshop on Deep Generative Models. arXiv: 1807.01622
- Attentive Neural Processes. Kim et al. ICLR 2019. arXiv:1901.05761
- Empirical Evaluation of Neural Process Objectives. Le et al. NeurIPS 2018 Workshop on Bayesian Deep Learning.
- Meta-Learning Surrogate Models for Sequential Decision Making. Galashov et al. arXiv: 1903.11907
- Meta-learning of Sequential Strategies. Ortega et al. arXiv:1905.03030

Yee Whye Teh

Recommender systems

Objective:

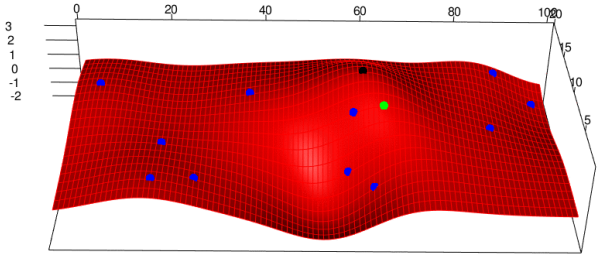
For given user $\mathcal{U}$, approximate its rating function $f_u : \mathcal{I} \to \mathbb{R}$ given the observed context $\mathcal{I}_u \subset I$

Recommender systems

Objective:

For given user $\mathcal{U}$, approximate its rating function $f_u : \mathcal{I} \to \mathbb{R}$ given the observed context $\mathcal{I}_u \subset I$

$(\mathcal{I}_u, \mathcal{R}_u) \sim p(\mathcal{T})$ - training & holdout samples



$$\mathcal{IG}(\mathcal{I}_i) := \mathcal{H}\big(p(\mathbf{r}_{\setminus i}|\mathcal{C})\big) - \mathbb{E}_{p(\mathbf{r}_{\setminus i}|\mathcal{C})}\big[\mathcal{H}\big(p(\mathbf{r}_{\setminus i}|\mathcal{C}')\big)\big] \qquad \mathcal{C}' = \mathcal{C} \cup \{\mathcal{I}_i, \hat{r}_i\}$$

Bootstrapping from the model's predictions

Results on MovieLens: RMSE

	MovieLens 100k		
Model	**20% of user data**	**50%**	**80%**
SVD++	1.0517	1.0217	1.0124
Multitask MLP	0.9831	0.9679	0.9507
MAML	0.9593	0.9441	0.9295
NP (random)	0.9359	0.9215	0.9151
NP (Info gain)	0.9288	0.8829	0.8557

	MovieLens 20m		
Model	**20%**	**50%**	**80%**
SVD++	0.9454	0.9454	0.9452
Multitask MLP	0.8570	0.8401	0.8348
MAML	0.8142	0.7852	0.7780
NP (random)	0.7982	0.7684	0.7570
NP (Info gain)	0.7932	0.7366	0.6857