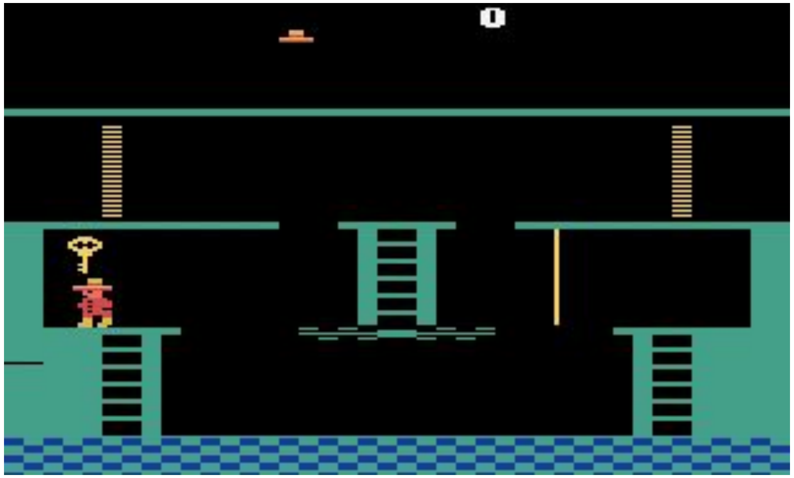
Lifelong / Meta / Transfer Learning

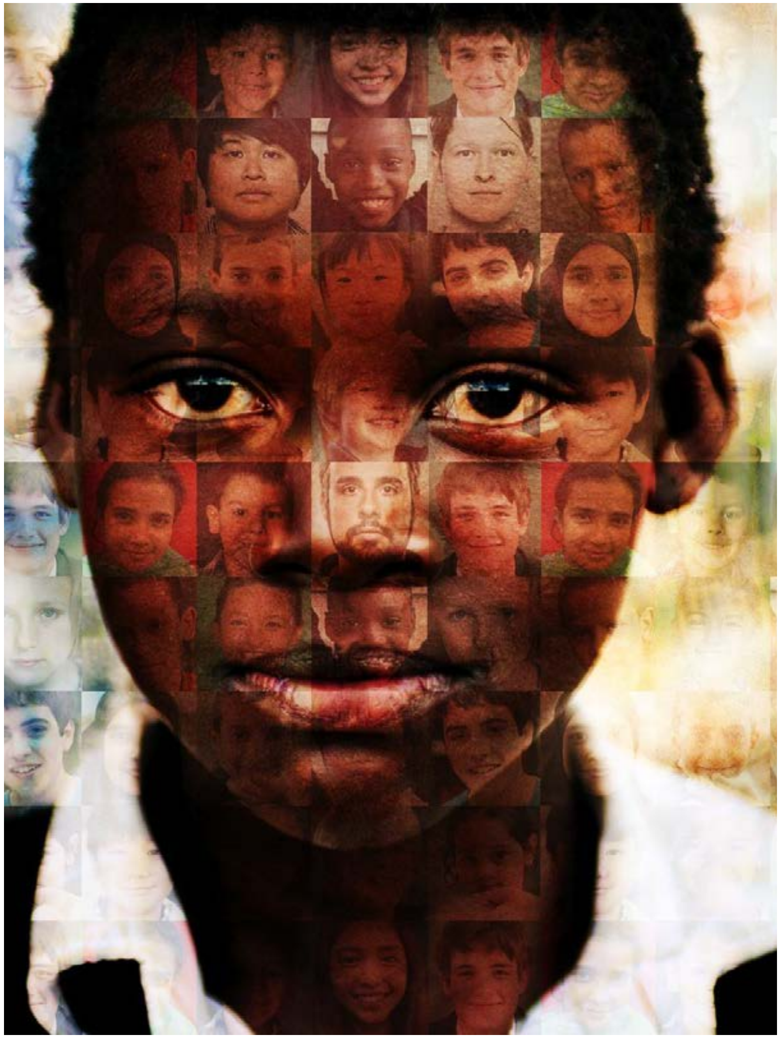
Emma Brunskill

Stanford

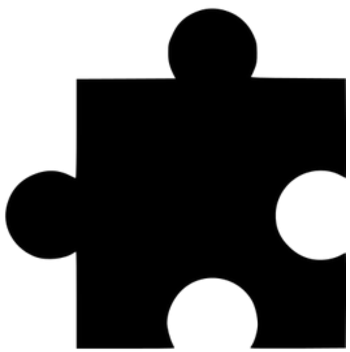
RL Summer School 2018



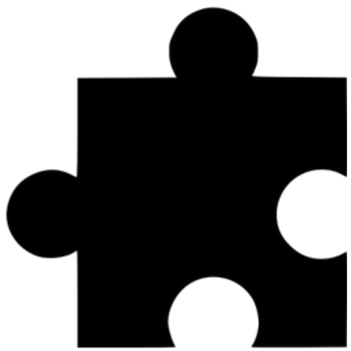
vs

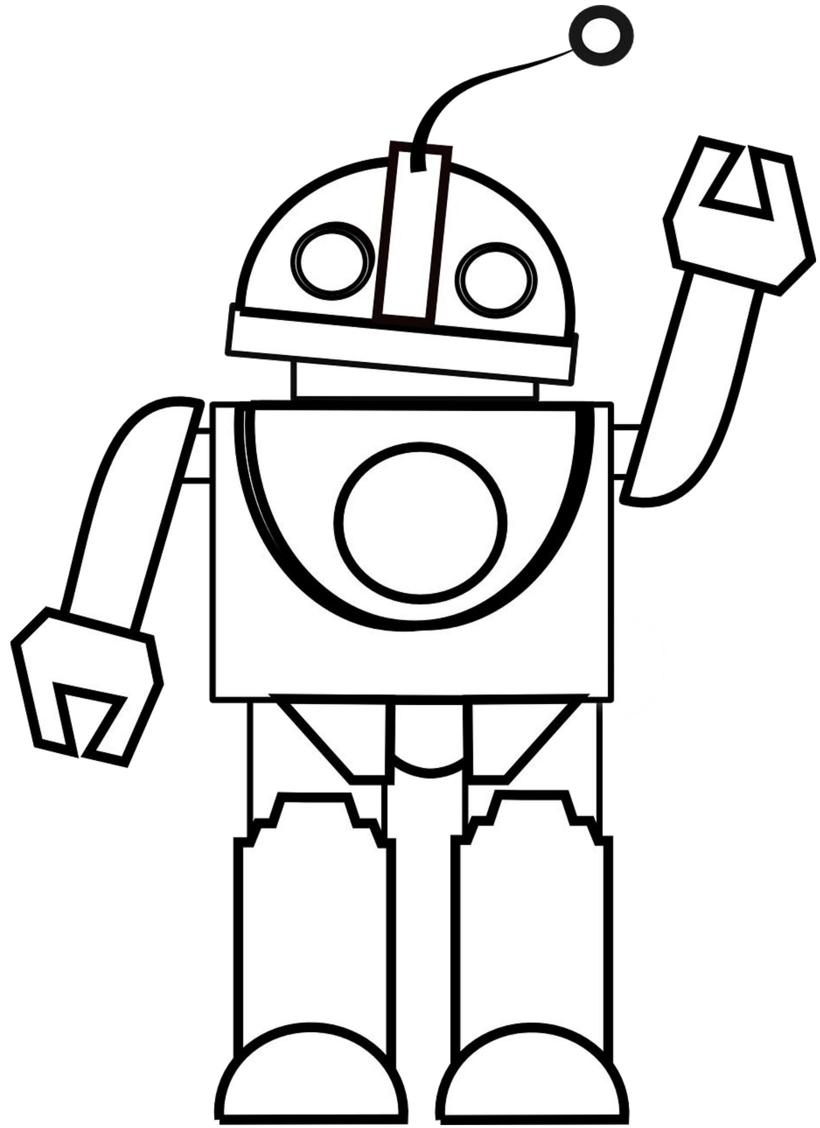


Learning to Solve a New (RL) Task

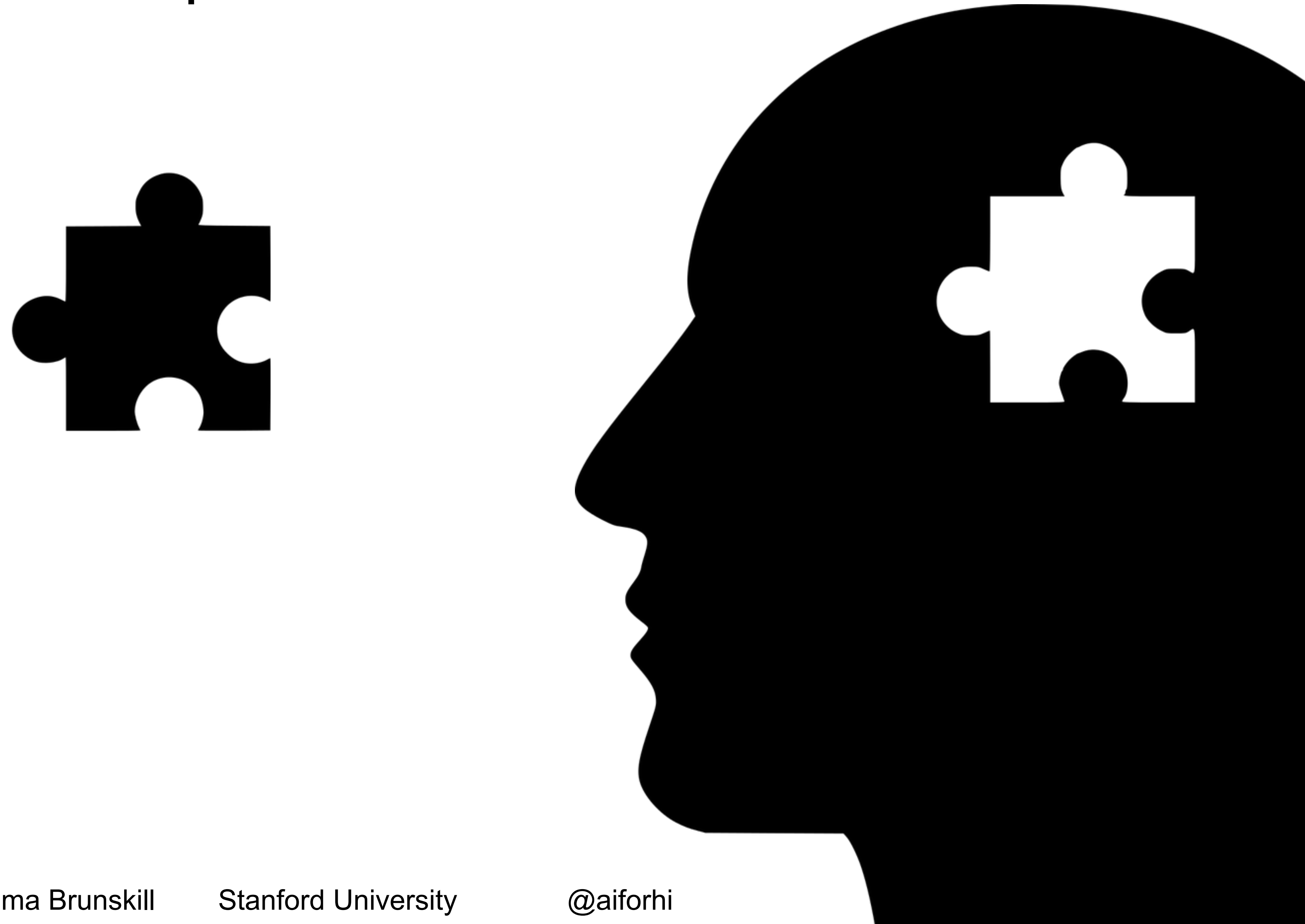


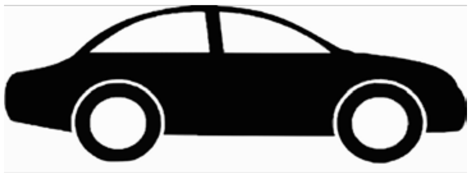
Most RL Agents Start From Scratch

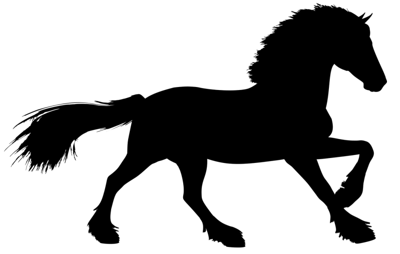


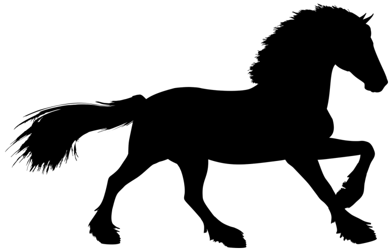
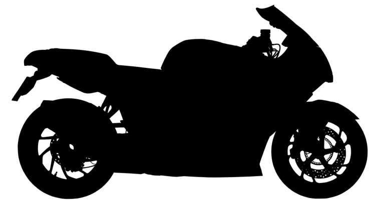
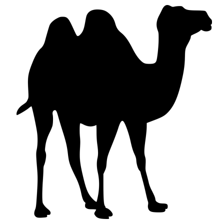


Cornerstone of Intelligence Behavior: Use Prior Experience To Solve New Tasks



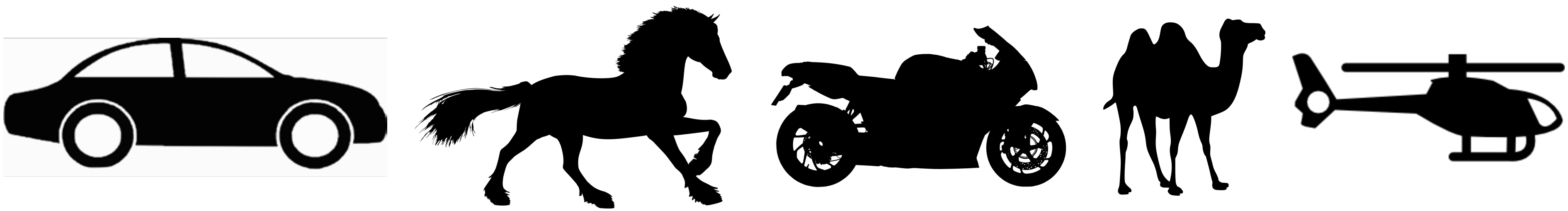


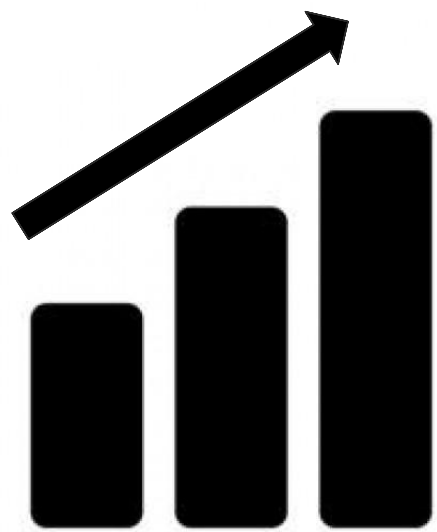


Helicopter made by Freepik
from www.flaticon.com

Transfer / Multi-task / Meta RL



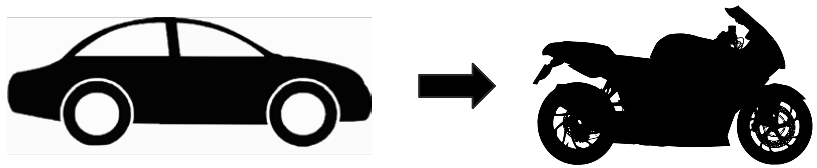


Helicopter made by Freepik
from www.flaticon.com

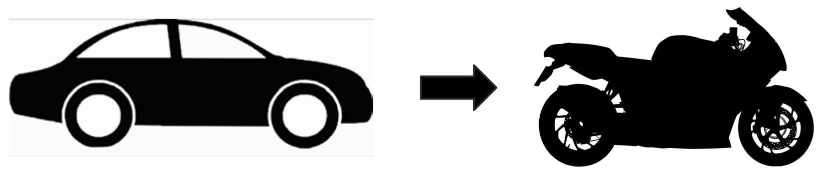
Emma Brunskill Stanford University @aiforhi https://cs.stanford.edu/people/ebrun/

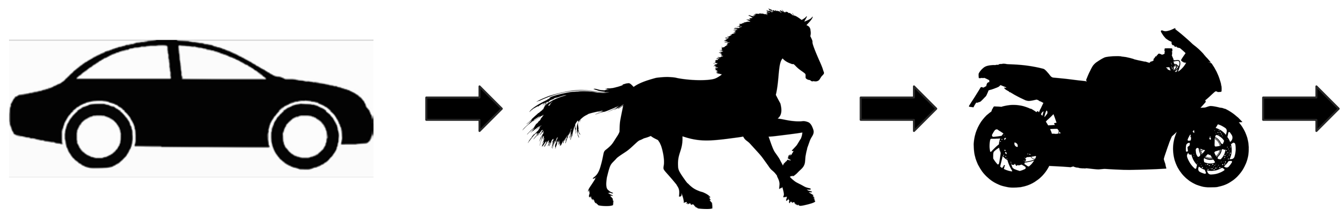
Common Settings

Transfer:



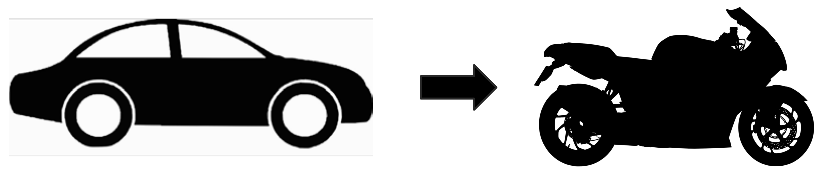
Common Settings

Transfer: 

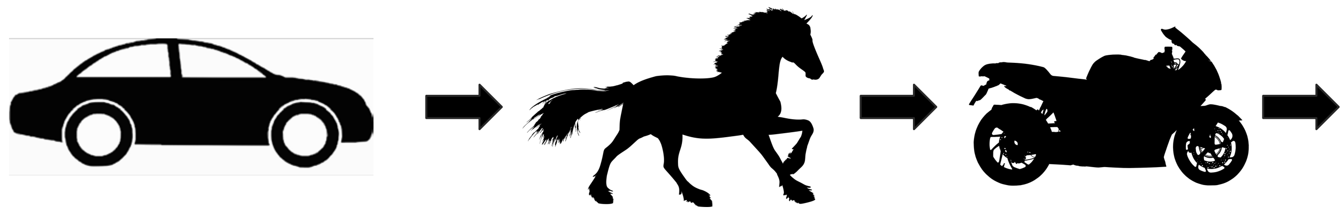
Lifelong: 

Common Settings

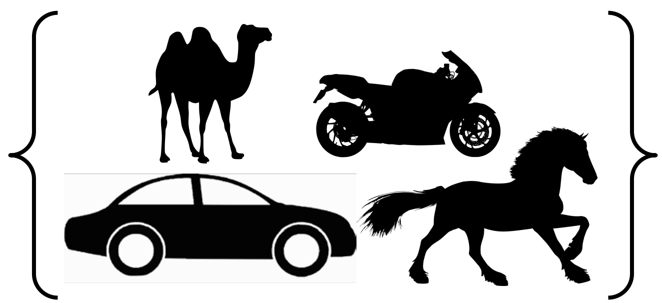
Transfer:



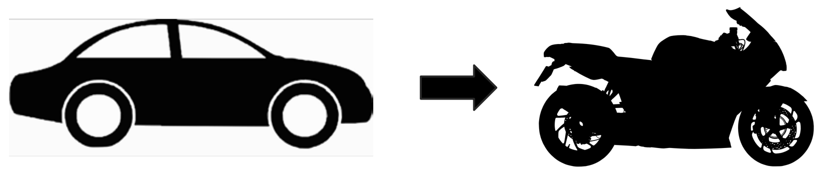
Lifelong:

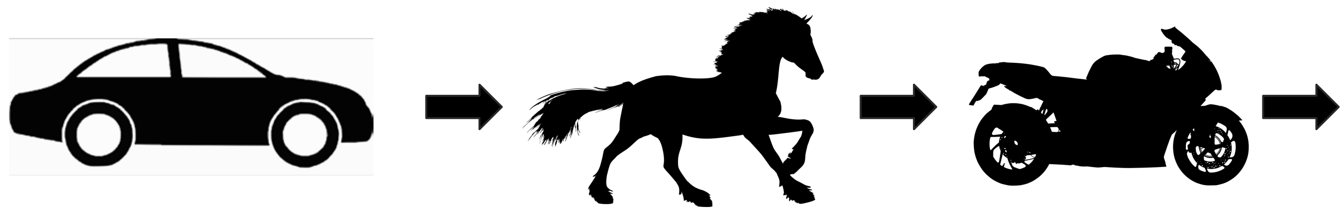


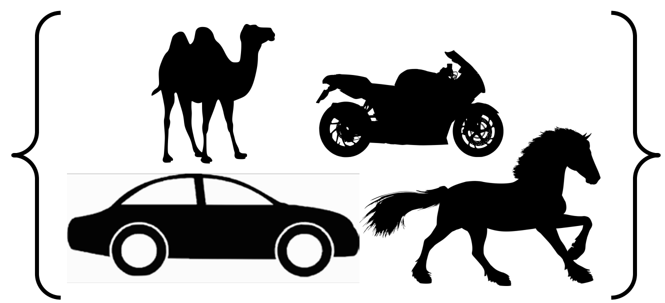
Multitask:

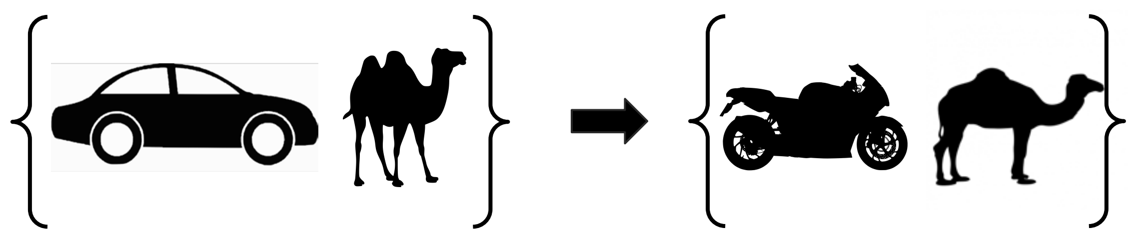


Common Settings

Transfer: 

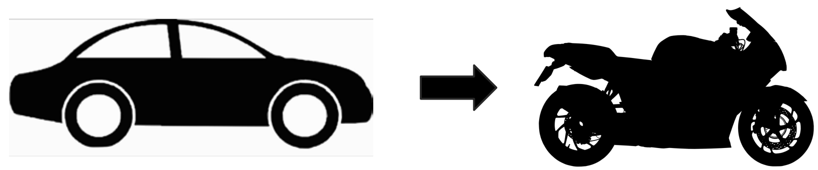
Lifelong: 

Multitask: 

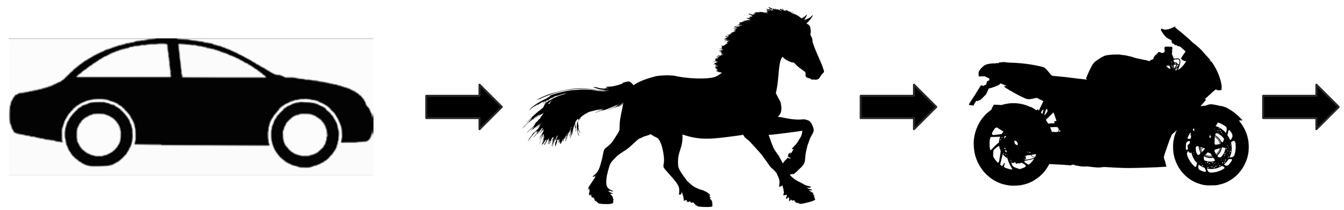
Many → Many: 

Common Settings

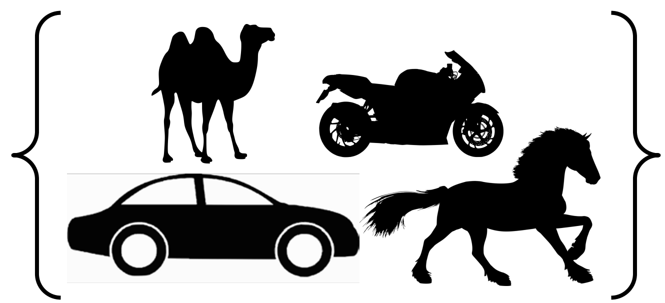
Transfer:



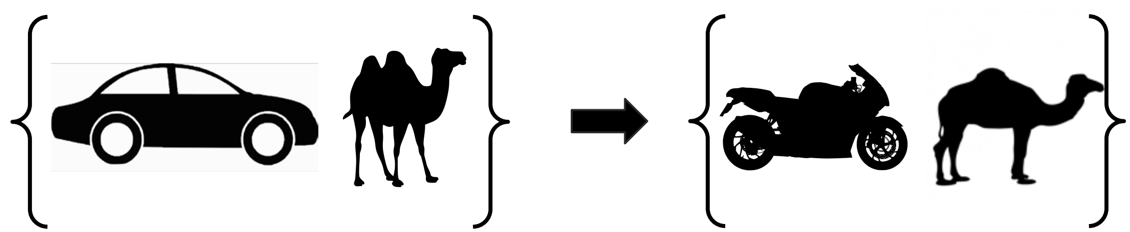
Lifelong:



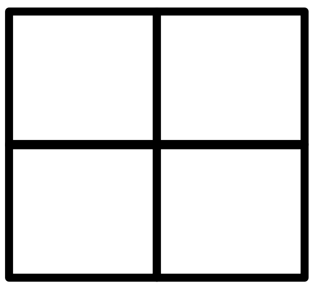
Multitask:

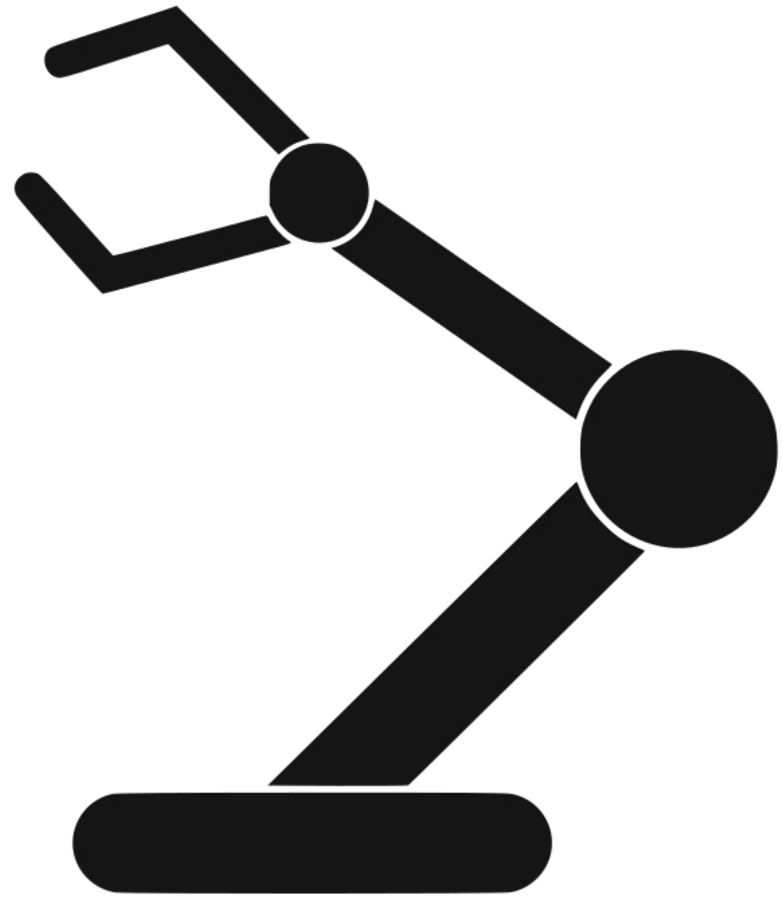


Many → Many:

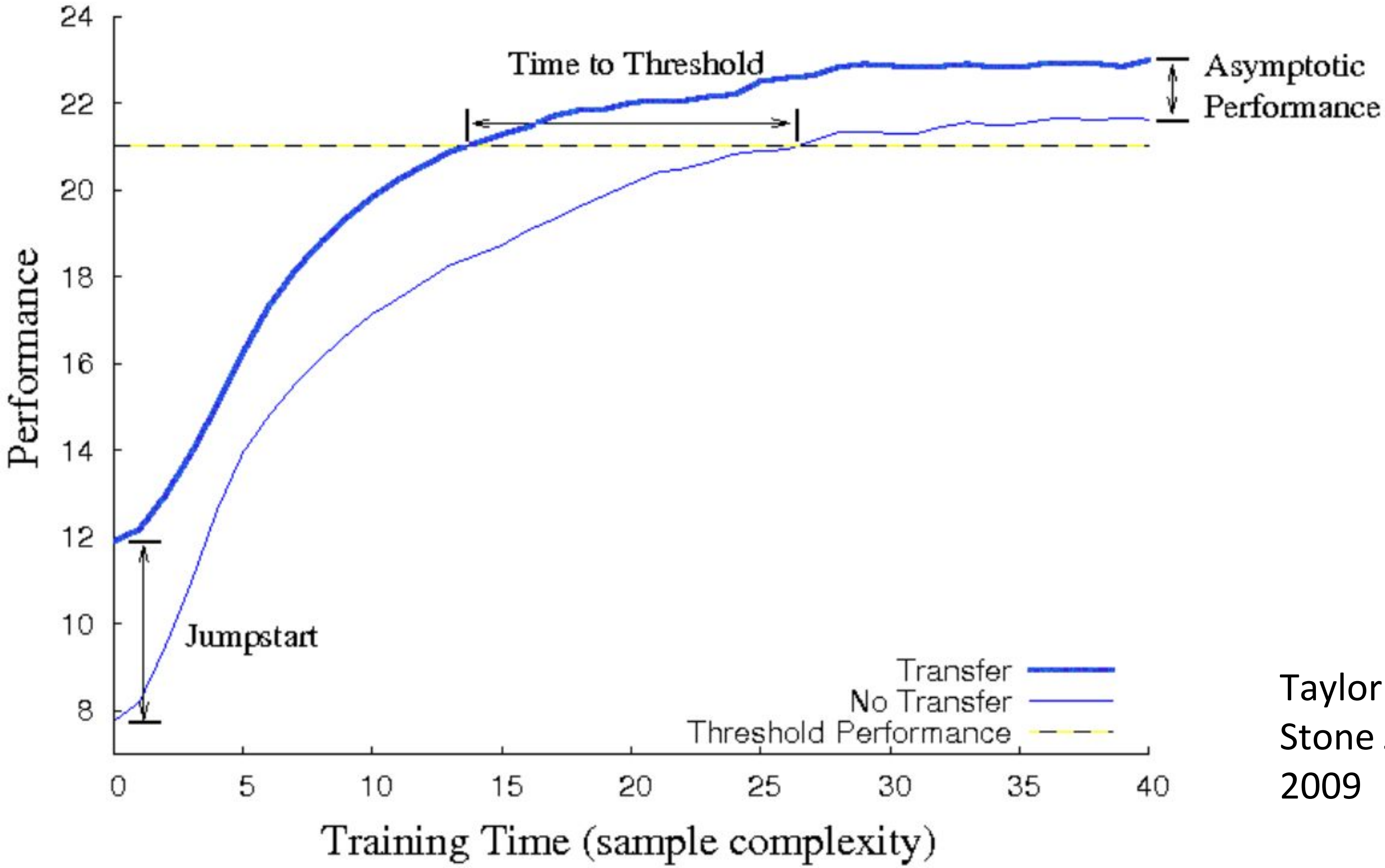


Tabular vs Function Approximation



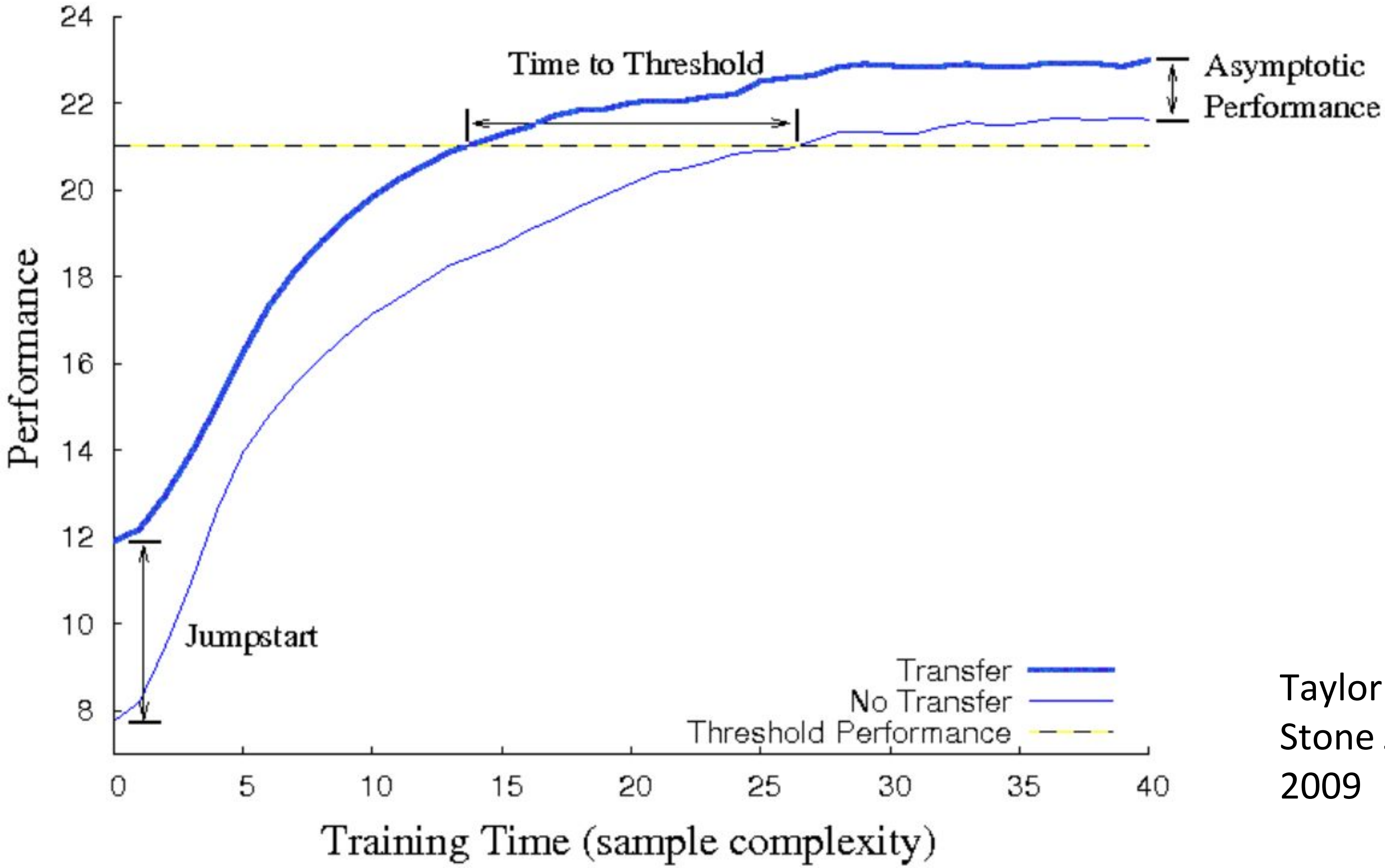


Evaluating Success in Transfer RL



Taylor & Stone JMLR 2009

Also, Provably Better Learning?



Taylor & Stone JMLR 2009

Two Core Parts of Multi-Task / Meta RL

- Summarize experience across tasks
- Use summary to improve learning in new task

Two Core Parts of Multi-Task / Meta RL

- Summarize experience across tasks
  - As dynamics / rewards models?
  - As value functions?
  - As policies?
- Use summary to improve learning in new task

Two Core Parts of Multi-Task / Meta RL

- Summarize experience across tasks
- Use summary to improve learning in new task

Rest of This Talk

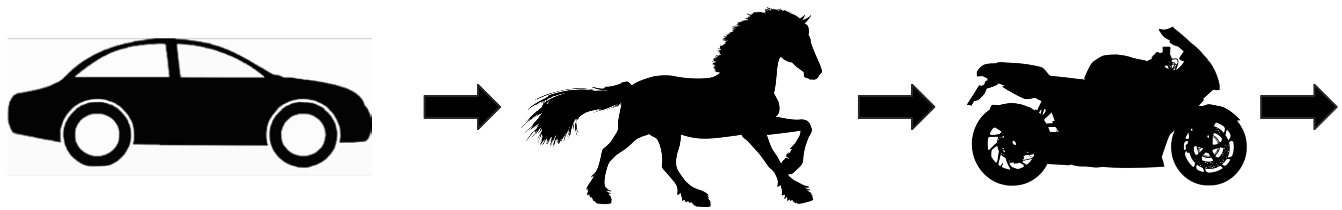
- Summarize experience across tasks
  - As a finite set of tasks (clustering)
  - As a low dimensional subspace
  - As a set of parameters near to desired set
- Use summary to improve learning in new task
  - As initialization to standard RL algorithm
  - To new RL algorithm to direct exploration

Rest of This Talk

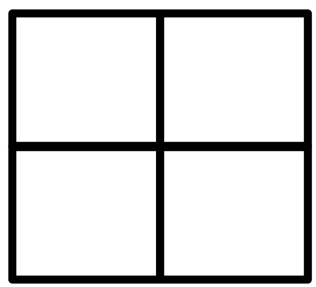
- Summarize experience across tasks
  - As a finite set of tasks (clustering)
  - As a low dimensional subspace
  - As a set of parameters near to desired set
- Use summary to improve learning in new task
  - As initialization to standard RL algorithm
  - To new RL algorithm to direct exploration

Setting

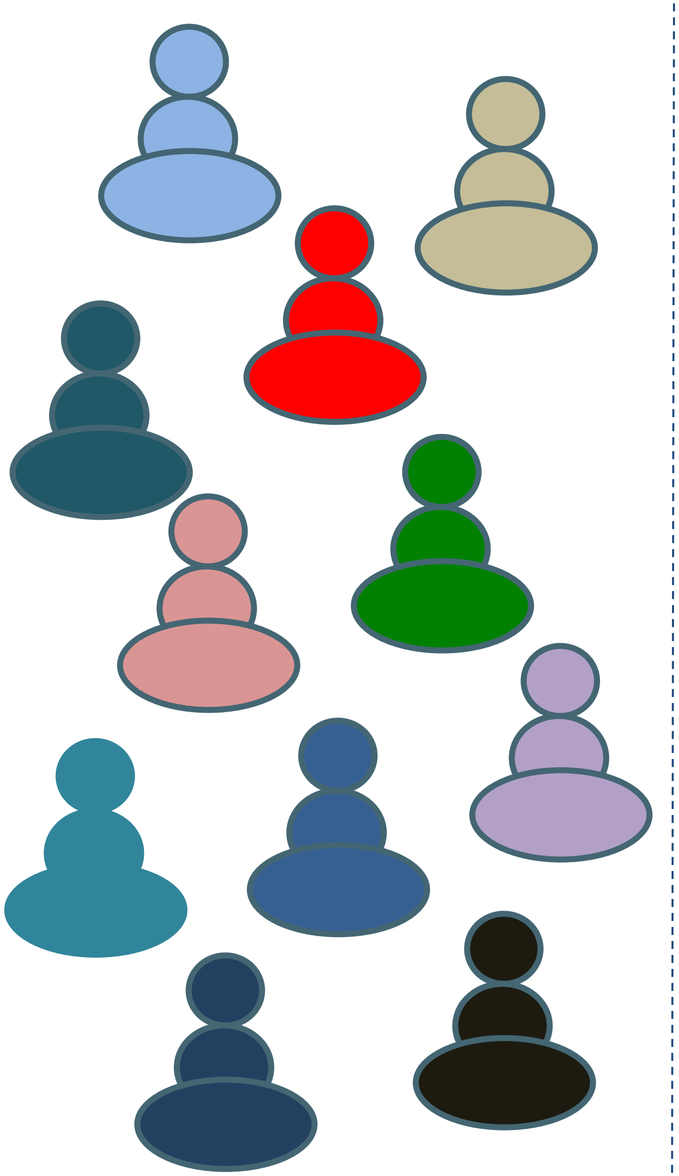
Lifelong



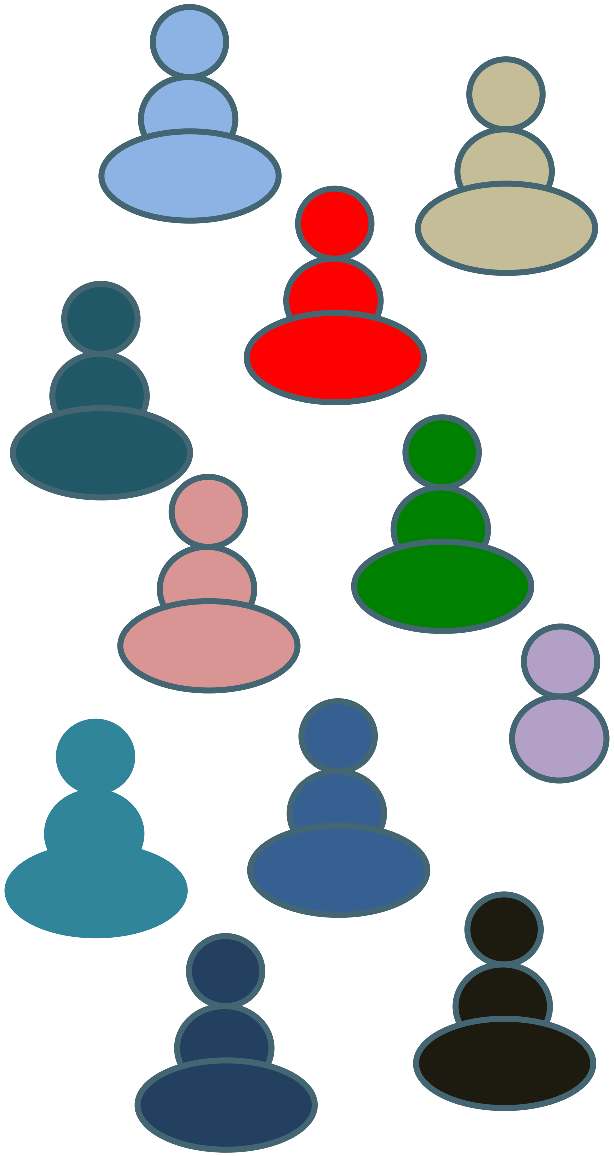
Tabular

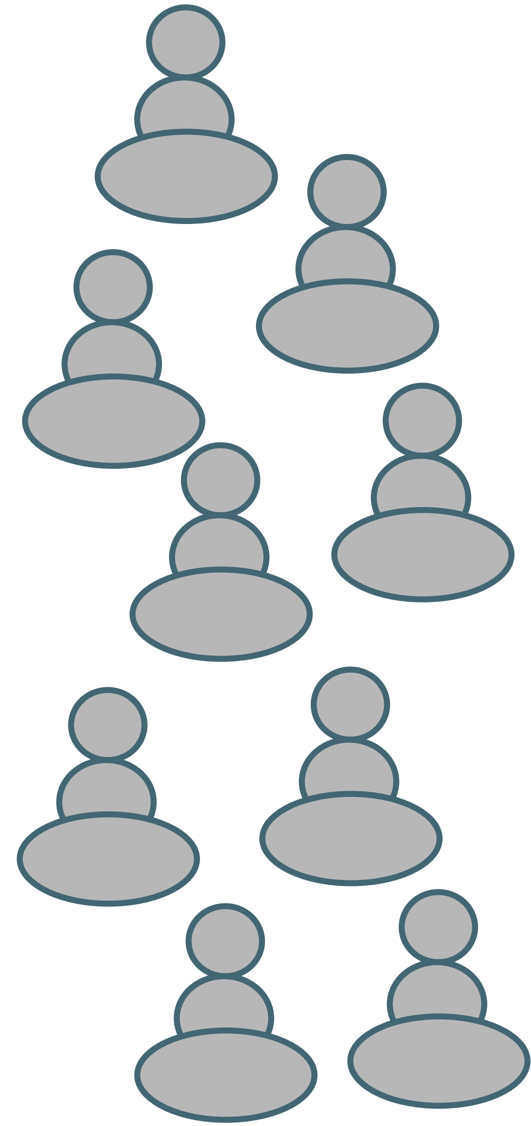


All Tasks Very Different

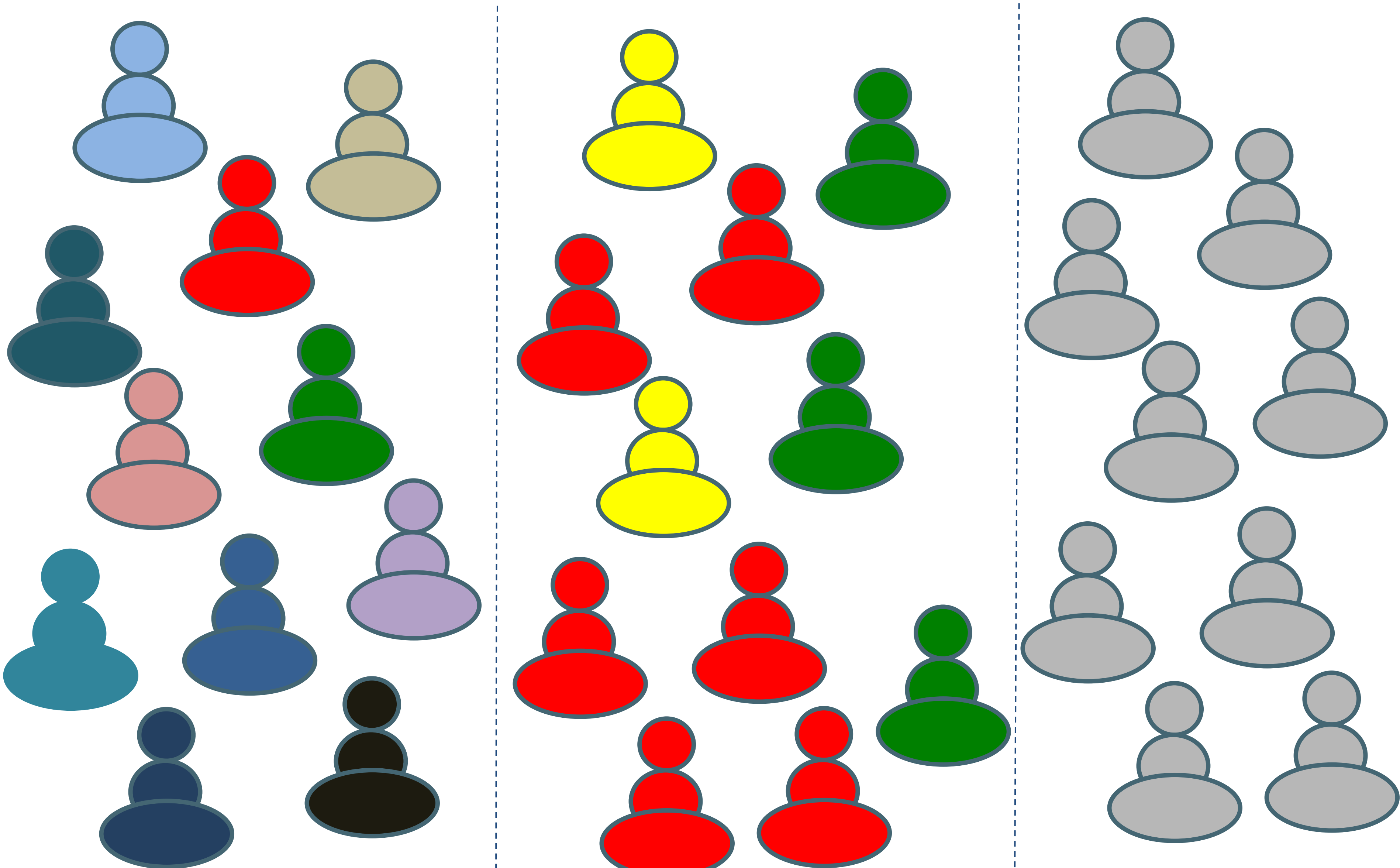


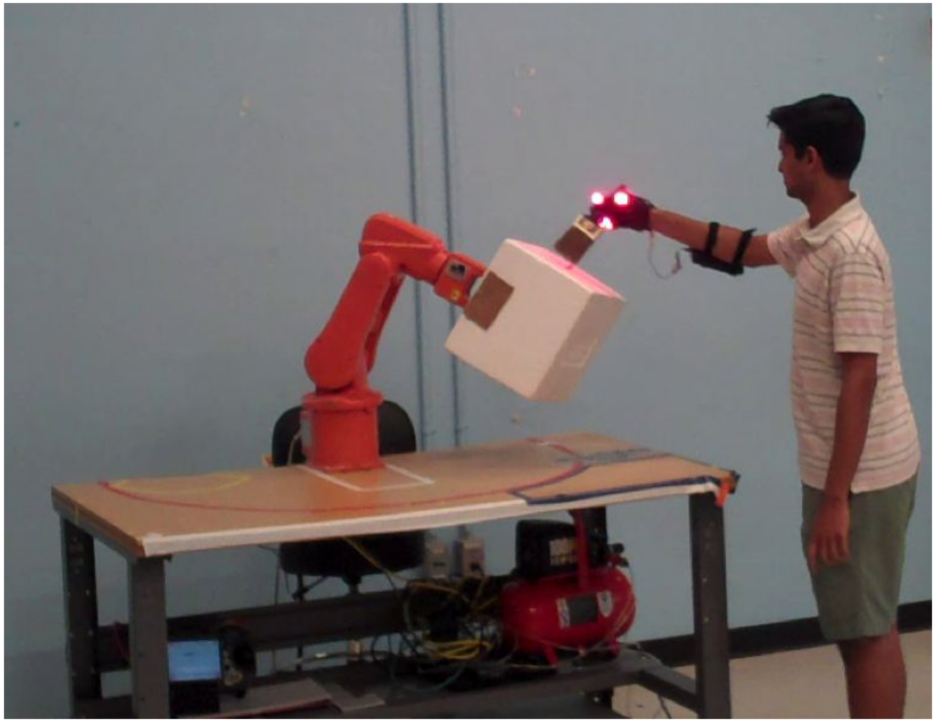
All Tasks Identical



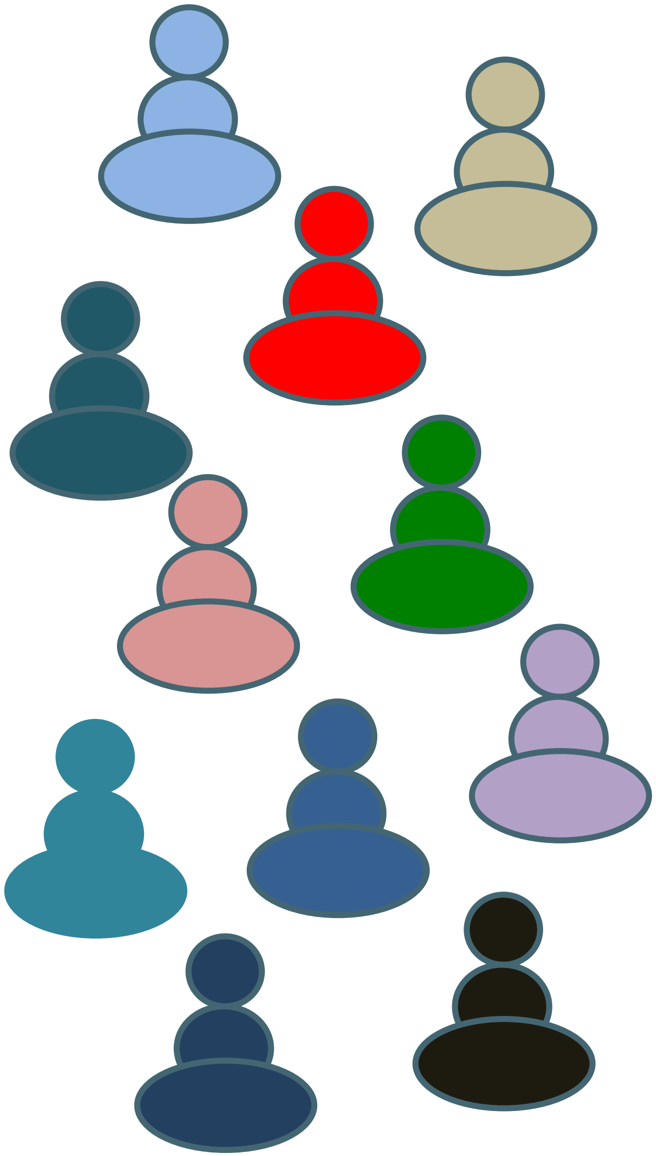


Finite Set of Tasks





Nikolaidis et al. HRI 2015



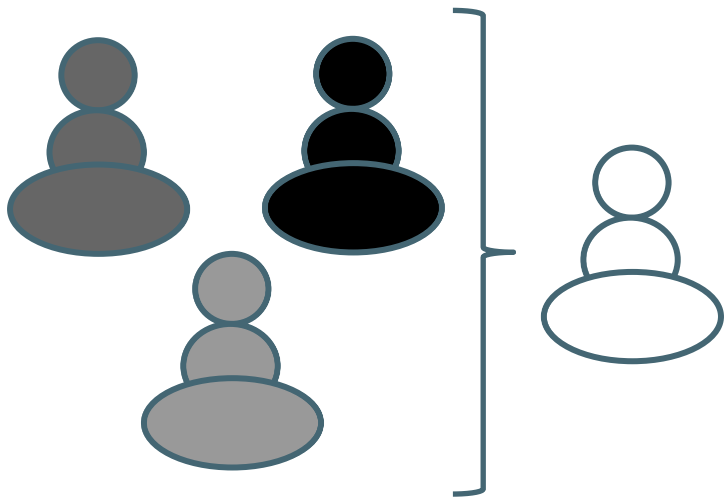
No apriori "labels" of similarity

Before try to learn this, if we knew the set of tasks, does it improve RL?

Two Core Parts of Multi-Task / Meta RL

- Summarize experience across tasks
  - As a finite set of tasks (clustering)
  - As a low dimensional subspace
  - As a set of parameters near to desired set
- Use summary to improve learning in new task
  - As initialization to standard RL algorithm
  - **To new RL algorithm to direct exploration**

If Know New Task is 1 of M Known Tasks, Can That Provably Improve Performance? (Spoiler: Yes!)



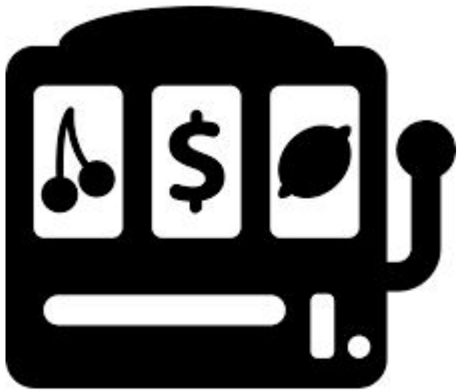
RL with Policy Advice

Azar, Lazaric, Brunskill, ECML 2013

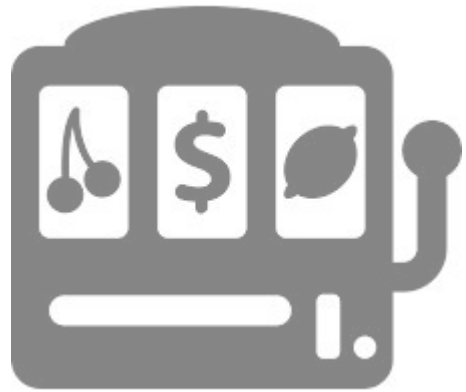
- Assumptions: New task sampled from M tasks
- Evaluation goal: Provably improve performance
- Approach: Leverage known M set **of policies**

RL with Policy Advice

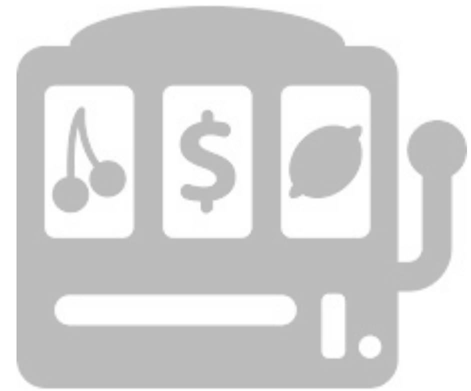
Azar, Lazaric, Brunskill, ECML 2013



π_1

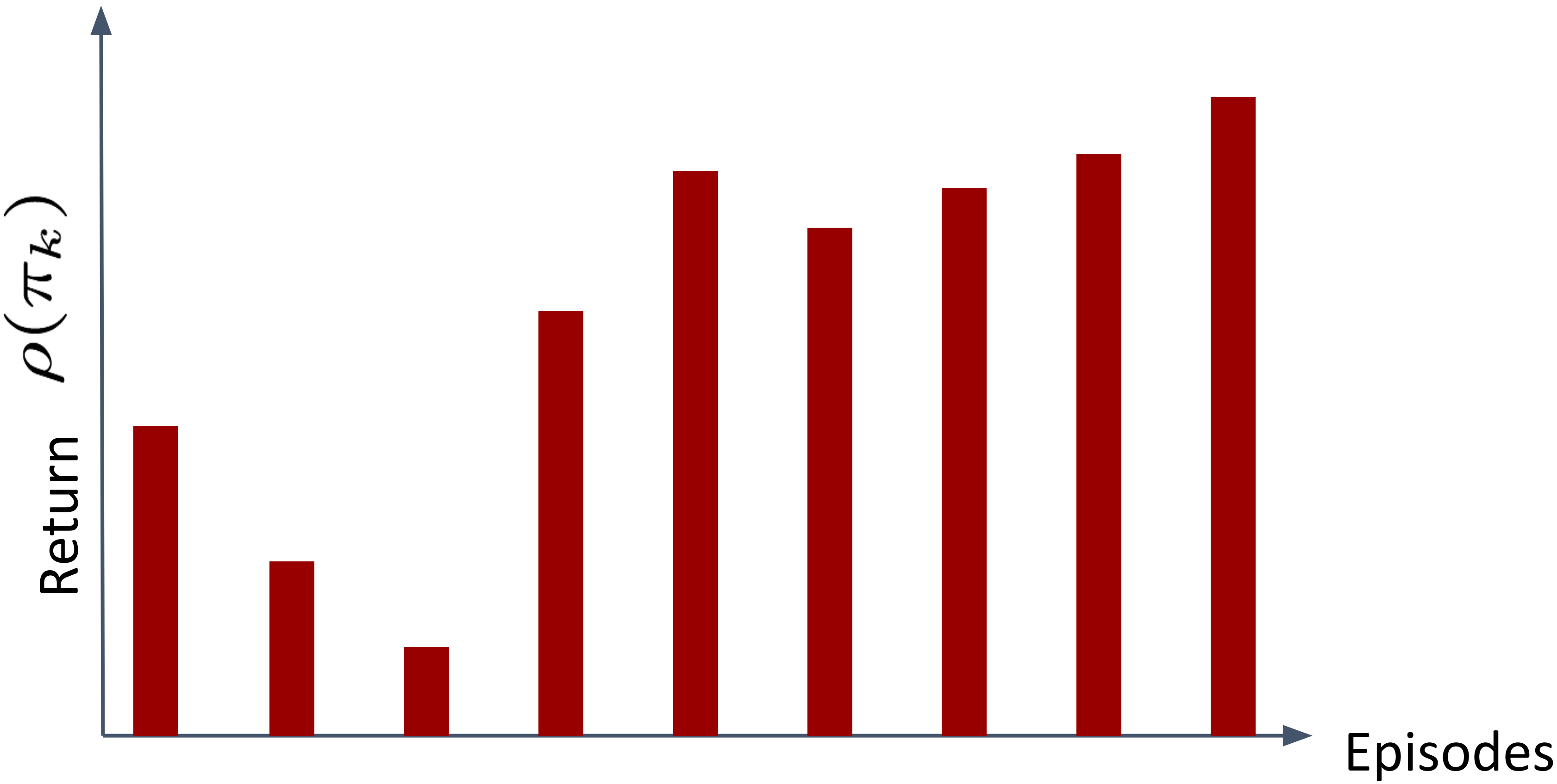


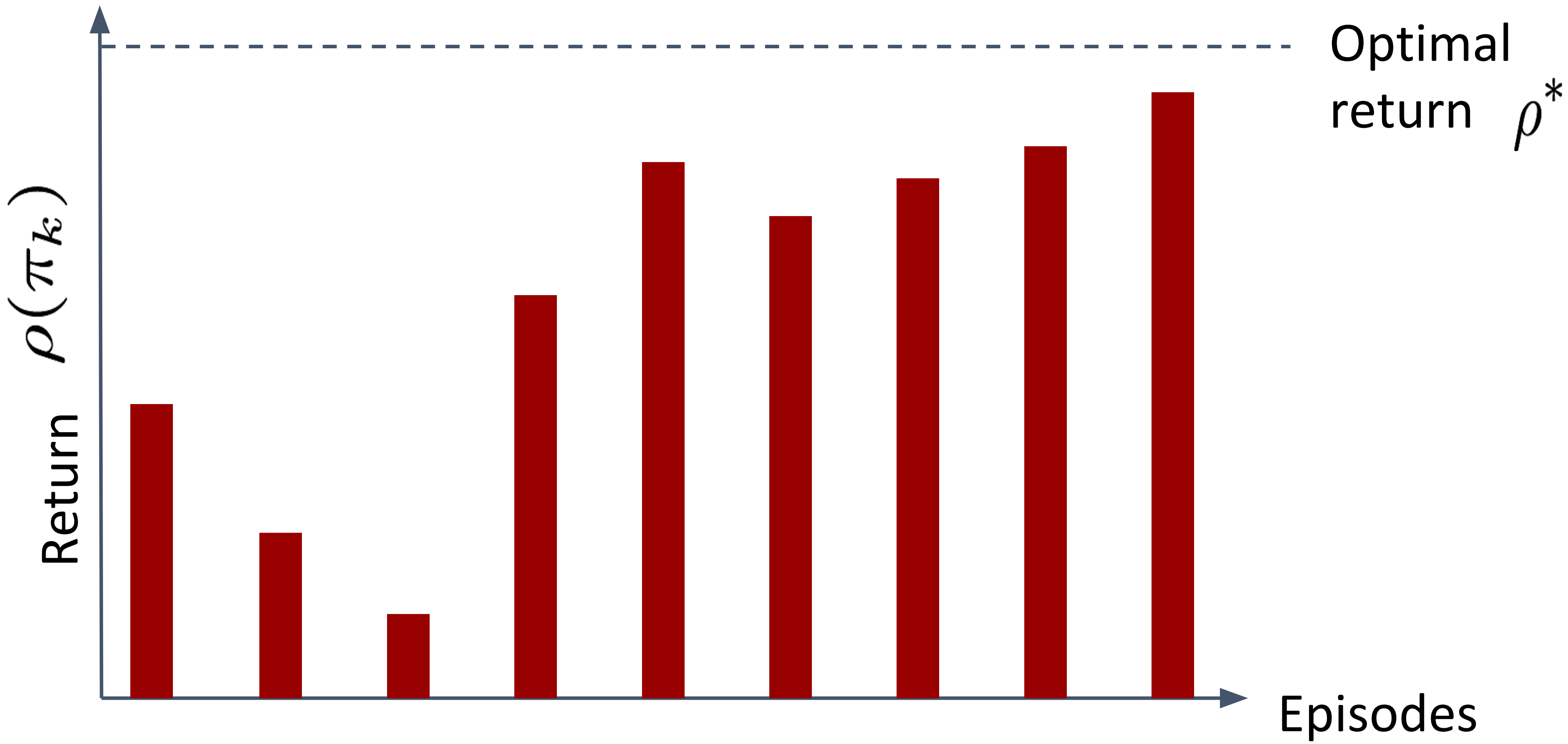
π_2



π_3

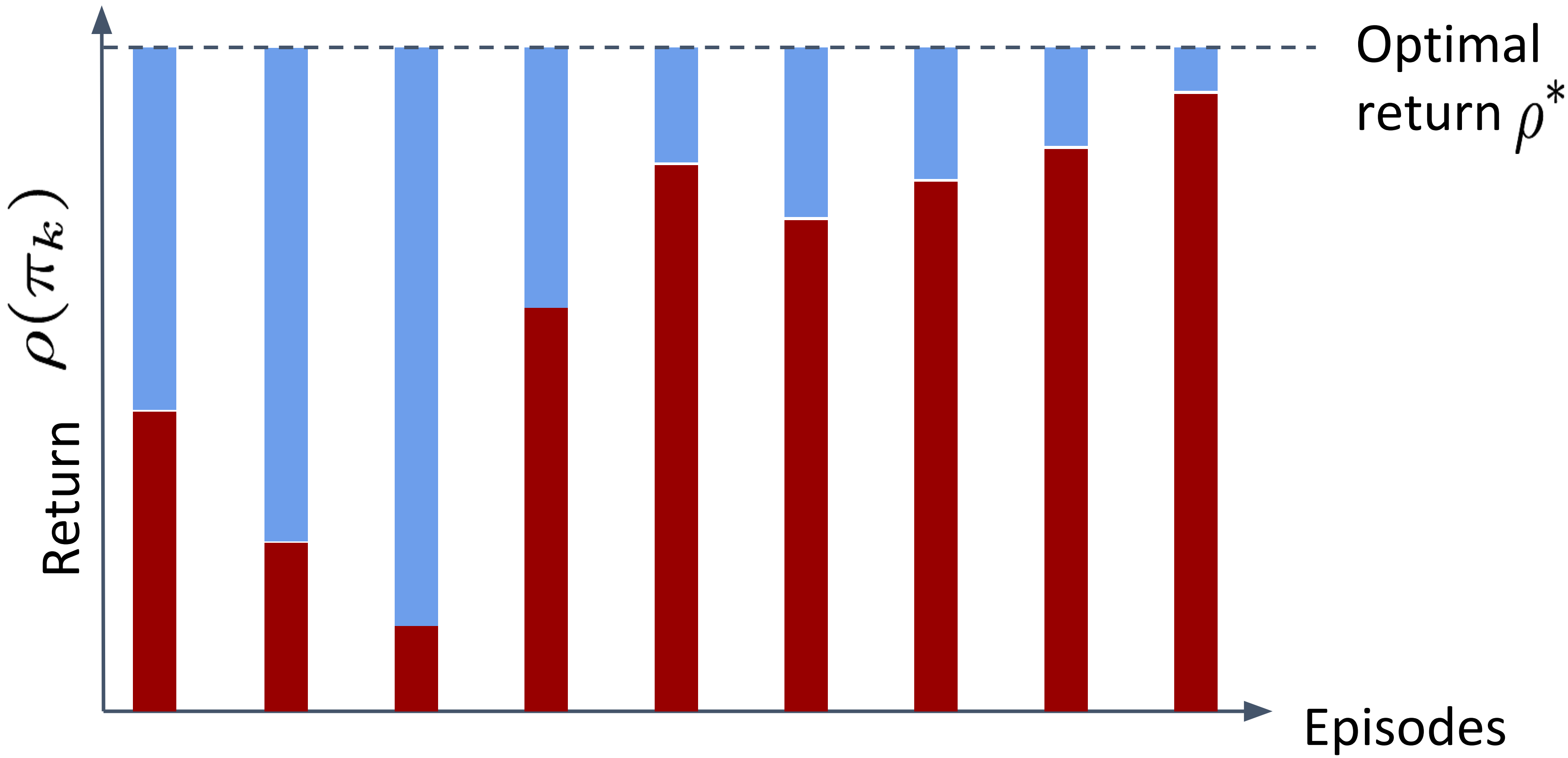
Quick Recap: Evaluating Performance



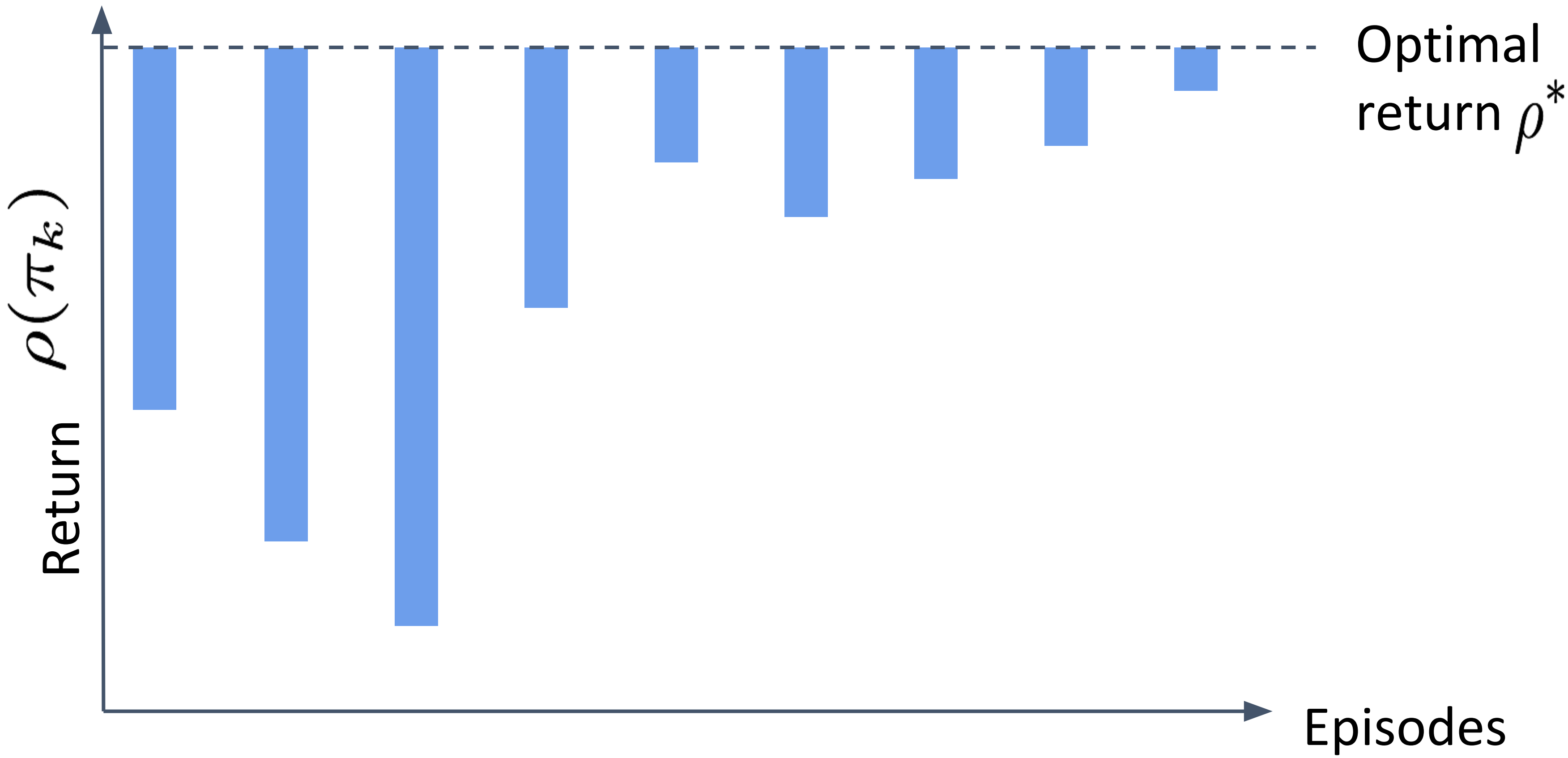


Optimal return ρ^*
Return $\rho(\pi_k)$
Episodes

Regret Bounds

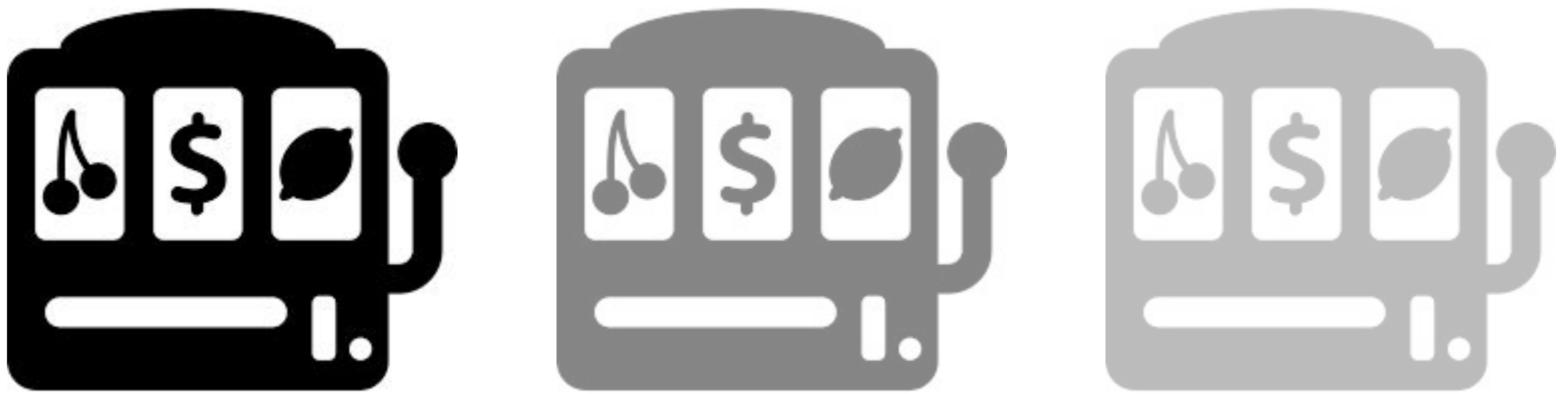


Regret Bounds: $R(T) = T\rho^* - \sum_{k=1}^{T} \rho(\pi_k)$



Provably Better Learning w/M Policies

Azar, Lazaric, Brunskill, ECML 2013

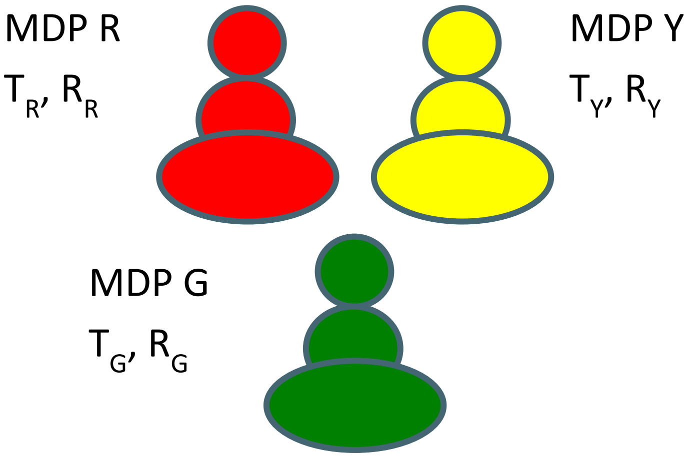


π_1 π_2 π_3

- Regret $\propto \sqrt{M}$ (independent of domain size)
- Related work: Talvitie & Singh IJCAI 2007; Dyagilev et al EWRL 2008; Maillard et al ICML 2013; Ortner et al ALT 2012

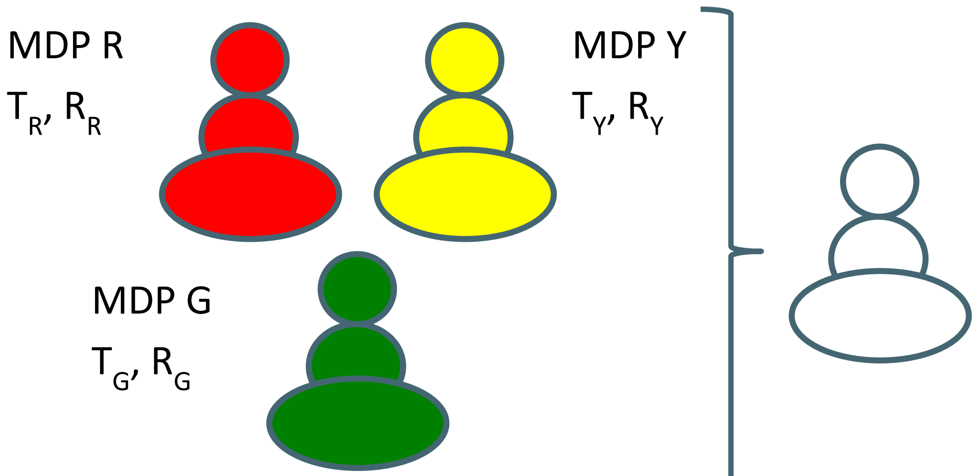
Sequential Transfer

- Assumptions: New task sampled from M tasks
- Evaluation criteria: Provably speed learning
- Approach: Leverage known M set **of models**


MDP R
T_R, R_R
MDP Y
T_Y, R_Y
MDP G
T_G, R_G

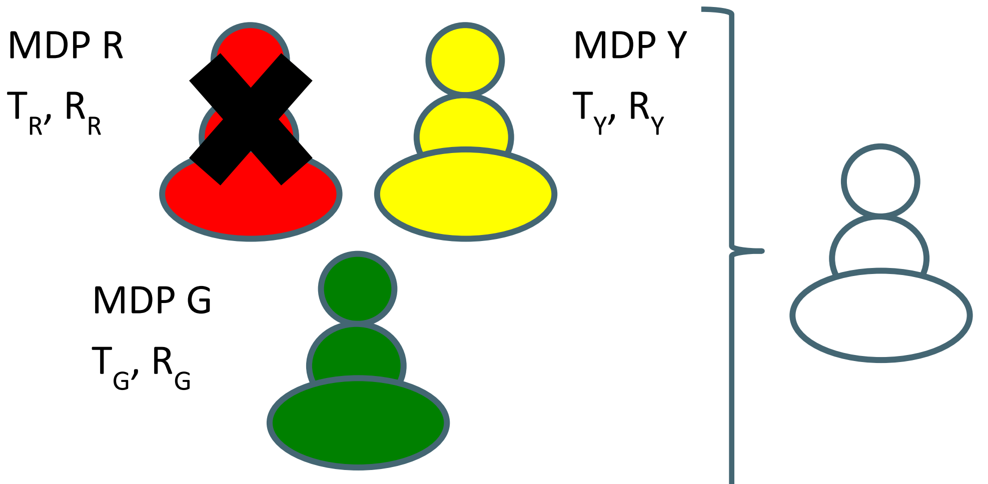
RL $\longrightarrow$ (Active) Classification

Brunskill & Li, UAI 2013



Maintain Hypothesis Set of Potential Identity of Current Task

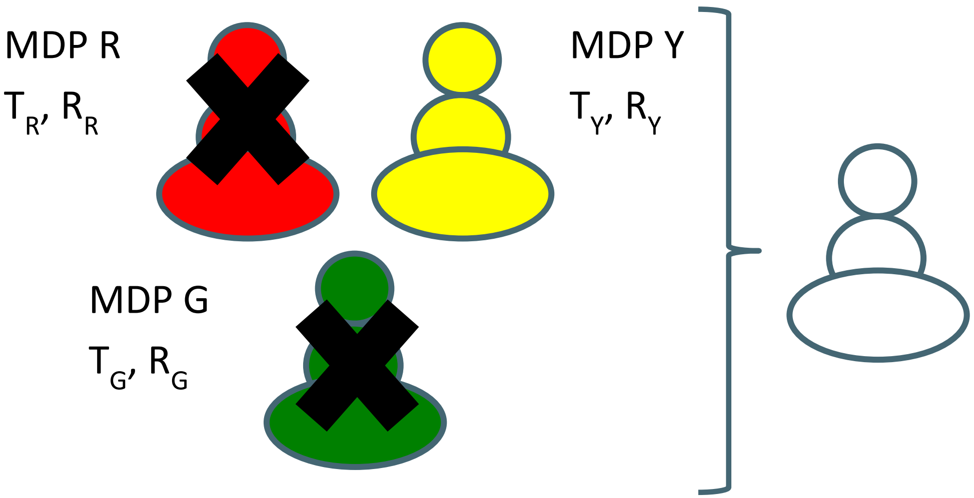
Brunskill & Li, UAI 2013



Act in current task
$<s_1,a_1,r_1,s_2,a_2,r_2,\ldots$

Direct Exploration to Quickly Identify Task*

Brunskill & Li, UAI 2013

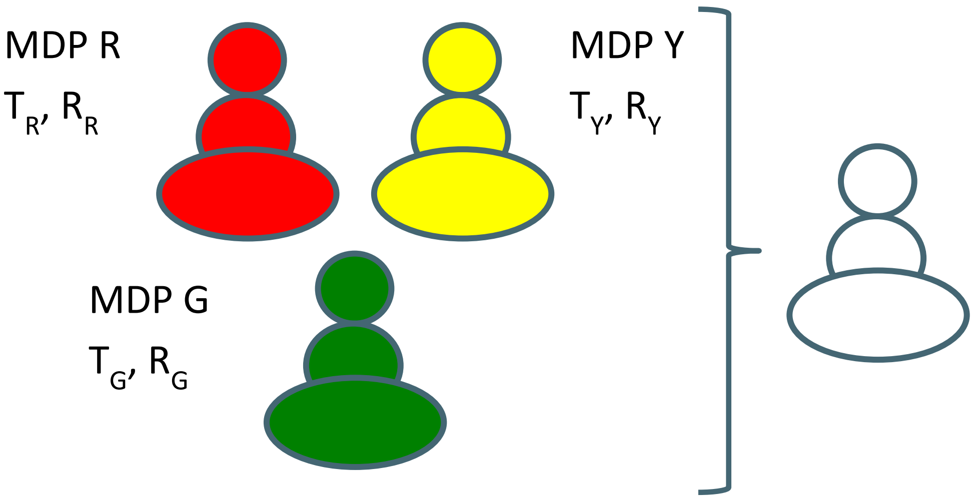


Act in current task

$<s_1,a_1,r_1,s_2,a_2,r_2,s_3,a_3,r_3,$

Grid World Example: Directed Exploration

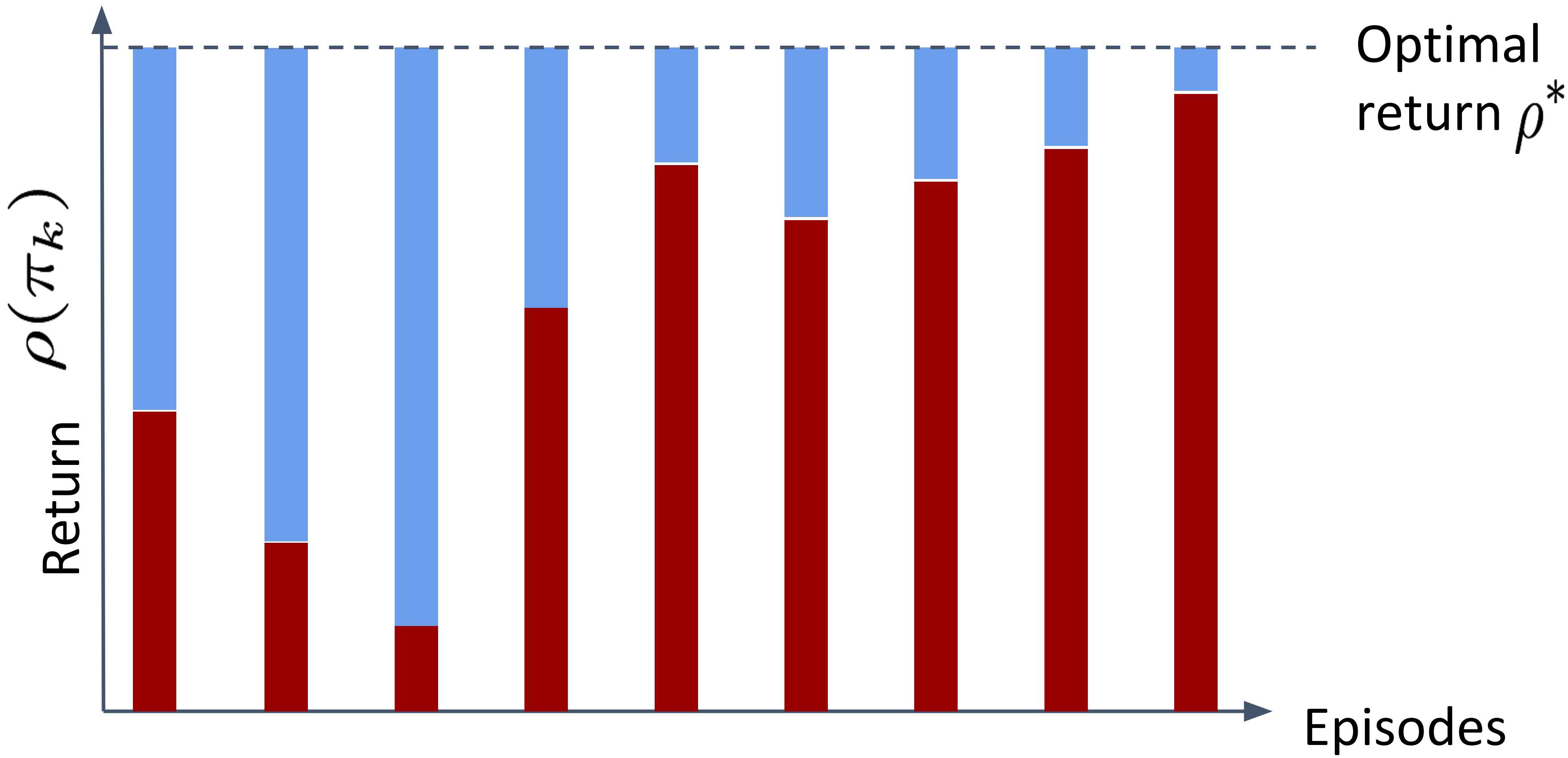
Intuition: Why Should This Speed Learning?



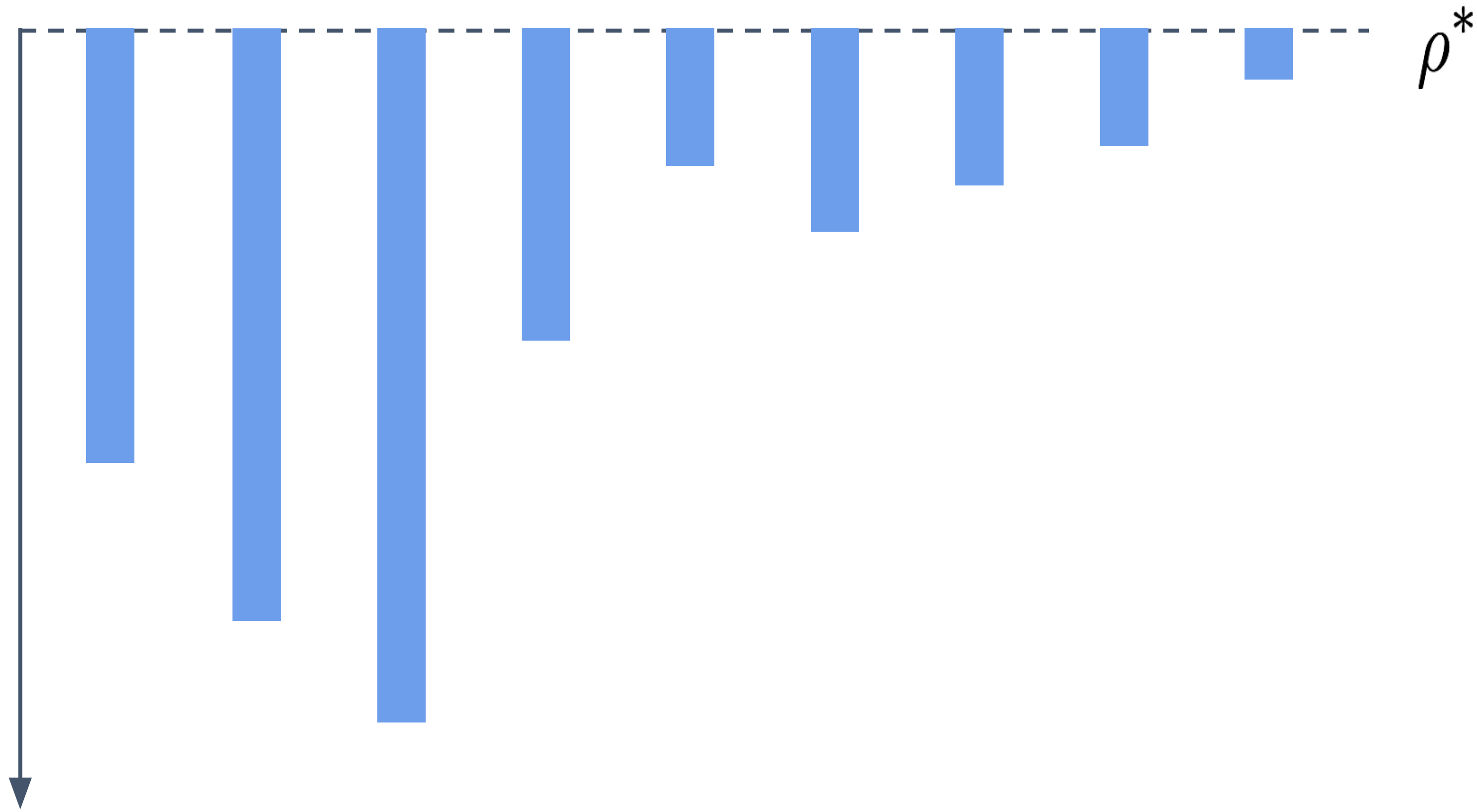
Act in it for H steps
$<s_1,a_1,r_1,s_2,a_2,r_2,s_3,a_3,\ldots s_H>$

- If MDPs agree (have same model parameters) for most (s,a) pairs, only a few (s,a) pairs need to visit
  - To classify task
  - To learn parameters (all others are known)
- If MDPs differ in most (s,a) pairs, easy to classify task

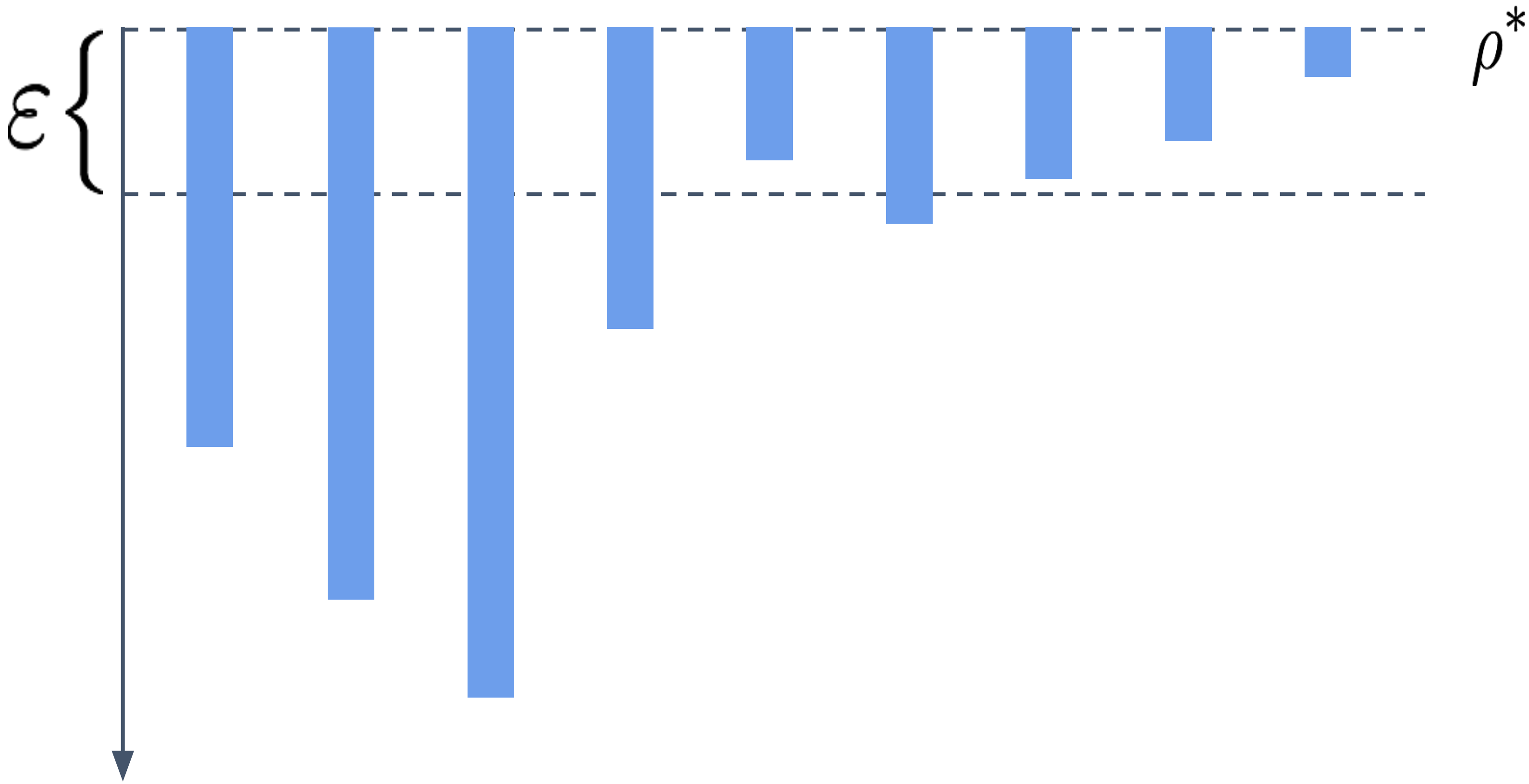
Formalizing RL Learning Speed



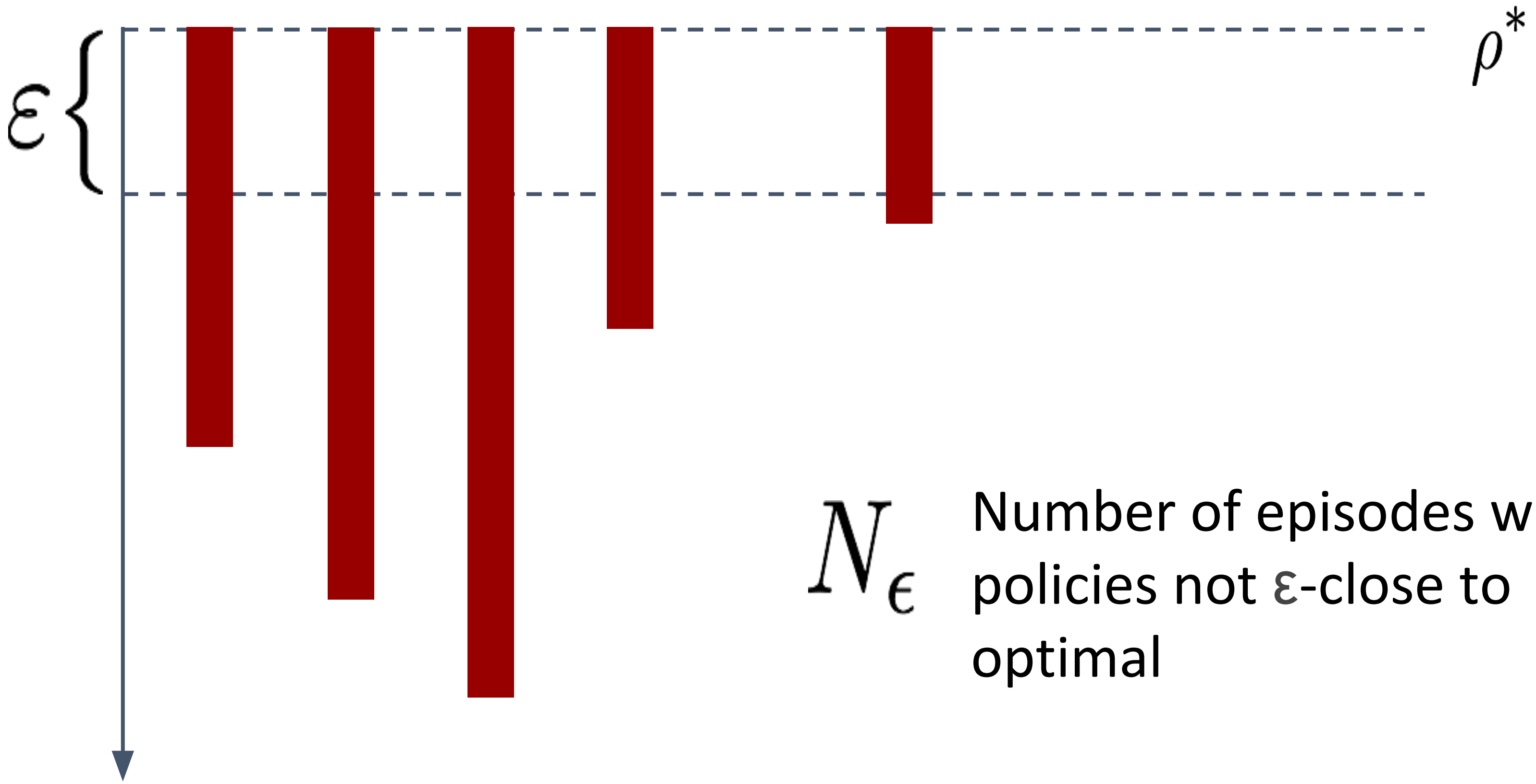
Formalizing RL Learning Speed



Formalizing RL Learning Speed

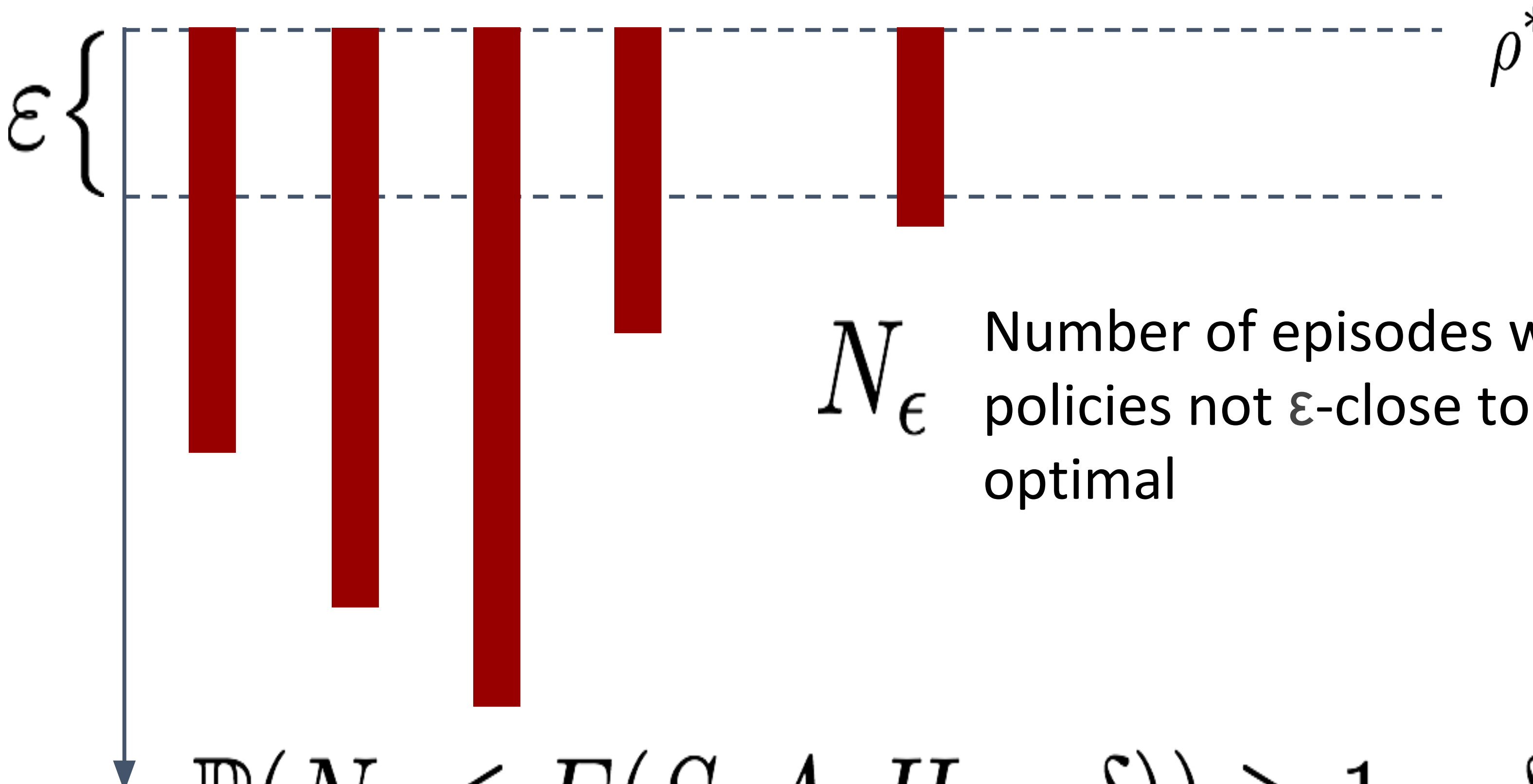


Only Count Big Mistakes



N_ϵ Number of episodes with policies not ε-close to optimal

Probably Approximately Correct RL

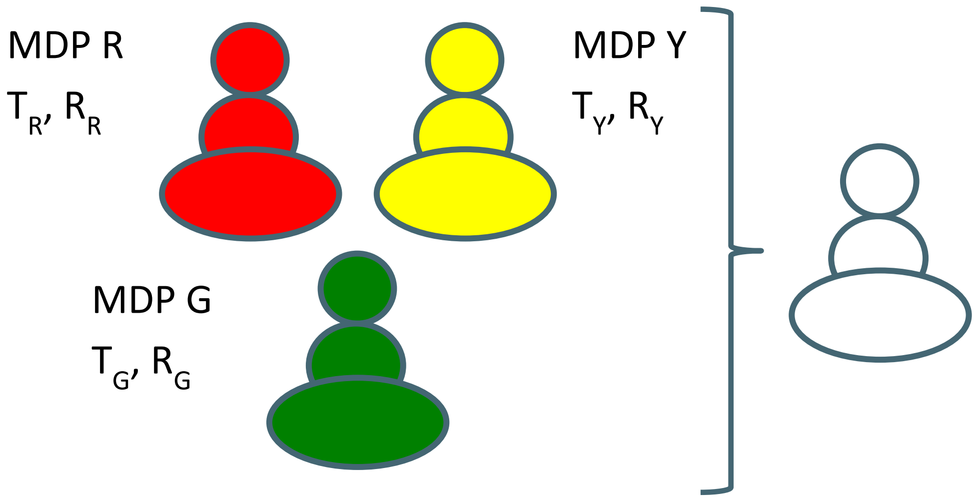


N_ϵ Number of episodes with policies not ε-close to optimal

$$\mathbb{P}(N_\epsilon \leq F(S, A, H, \epsilon, \delta)) \geq 1 - \delta$$

Provably Faster Learning Through Transfer

Brunskill & Li, UAI 2013



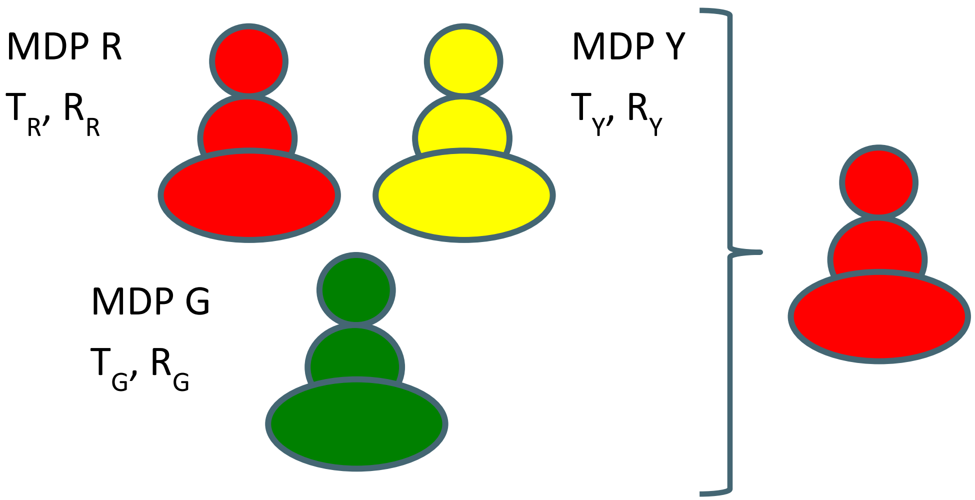
Theorem 1 *Given any* ϵ *and* δ, *run Algorithm 1 for* T *tasks, each for* $H = O\left(DSA(\max(\frac{1}{\Gamma^2}\log\frac{T}{\delta}, SD^2)\right)$ *steps. Then, the algorithm will select an* ϵ*-optimal policy on all but at most* $\tilde{O}\left(\frac{\zeta V_{\max}}{\epsilon(1-\gamma)}\right)$ *steps, with probability at least* $1-\delta$, *where*

$$\zeta = O\left(T_1\zeta_s + \bar{C}\zeta_s + (T - T_1)\frac{\bar{C}D}{\Gamma^2}\right),$$

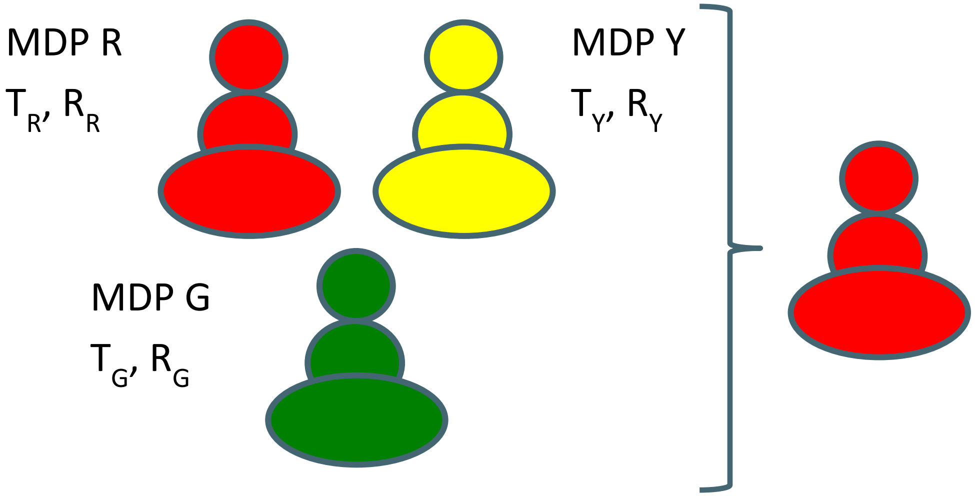
How Learn These Clusters?

- Summarize experience across tasks
  - **As a finite set of tasks (clustering)**
  - As a low dimensional subspace
  - As a set of parameters near to desired set
- Use summary to improve learning in new task
  - As initialization to standard RL algorithm
  - To new RL algorithm to direct exploration

Sequential Multitask Learning Across Finite Set of Markov Decision Processes

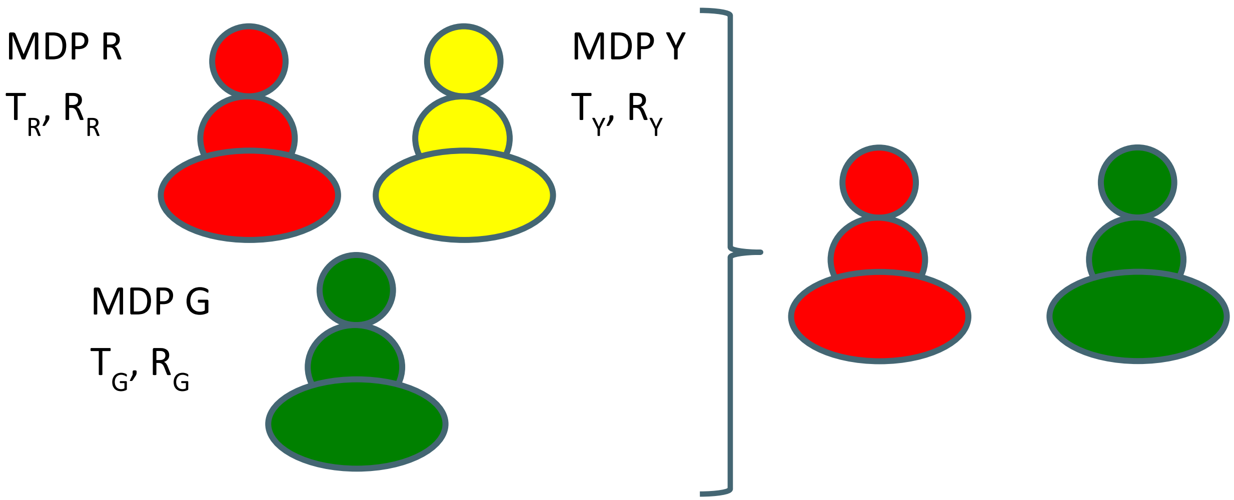


Sample a task from finite set of MDPs

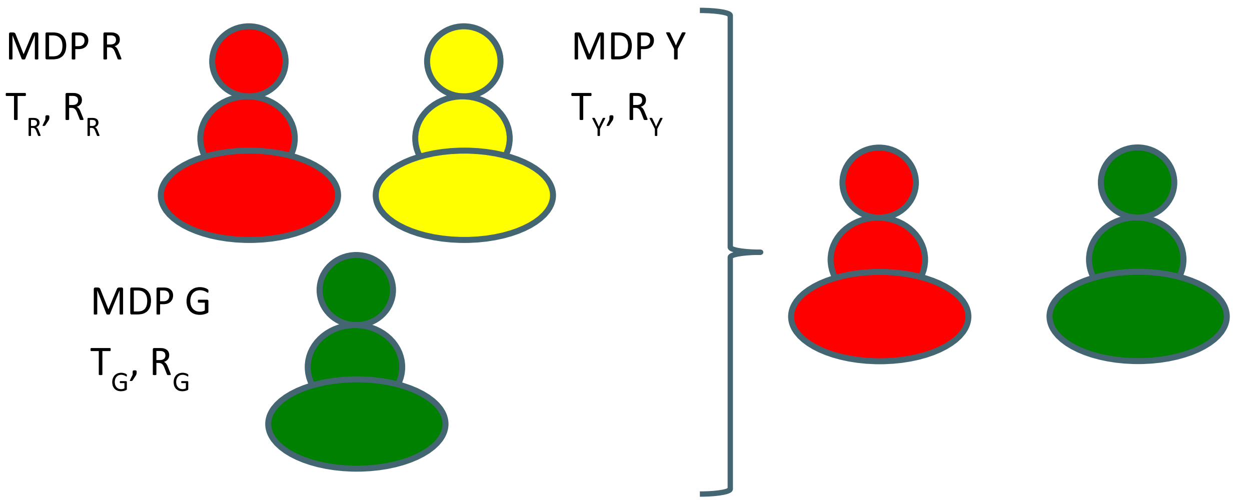


Act in it for H steps

$<s_1,a_1,r_1,s_2,a_2,r_2,s_3,a_3,\ldots s_H>$

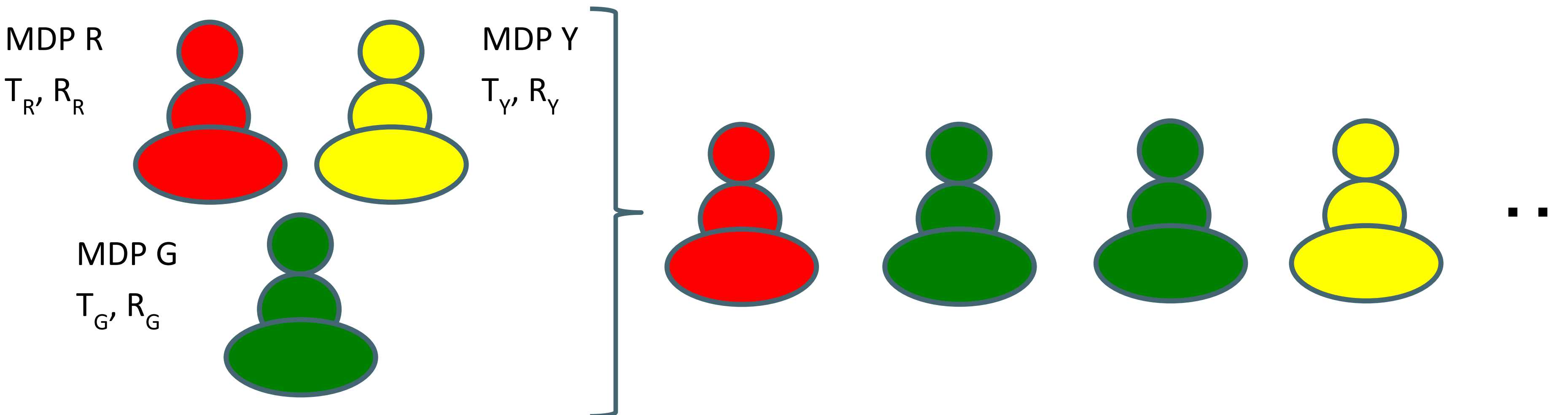


Again sample a MDP…



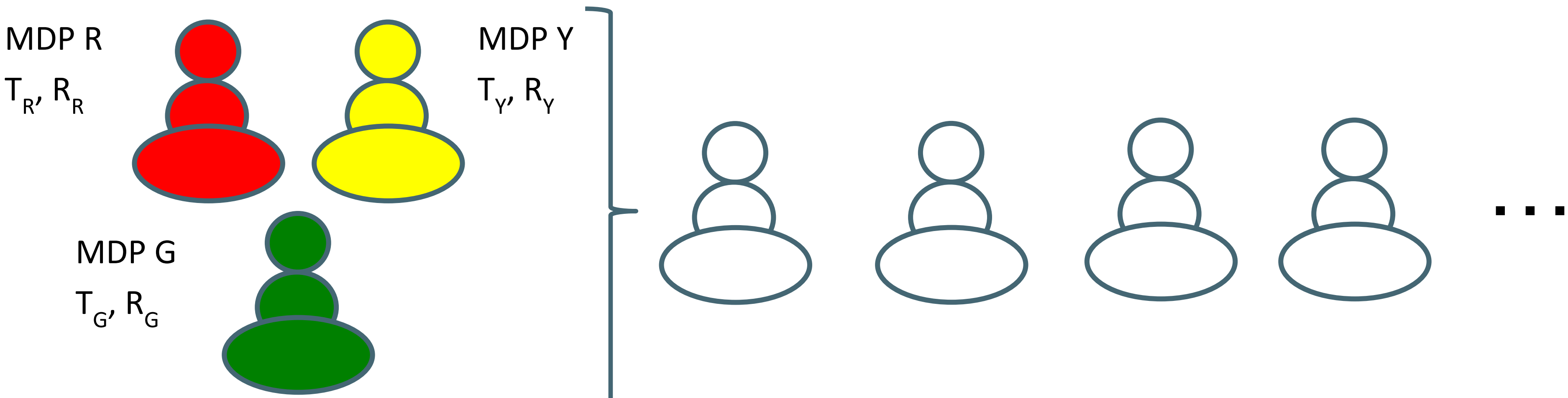
Act in it for H steps

$\langle s_1,a_1,r_1,s_2,a_2,r_2,s_3,a_3,\ldots s_H \rangle$

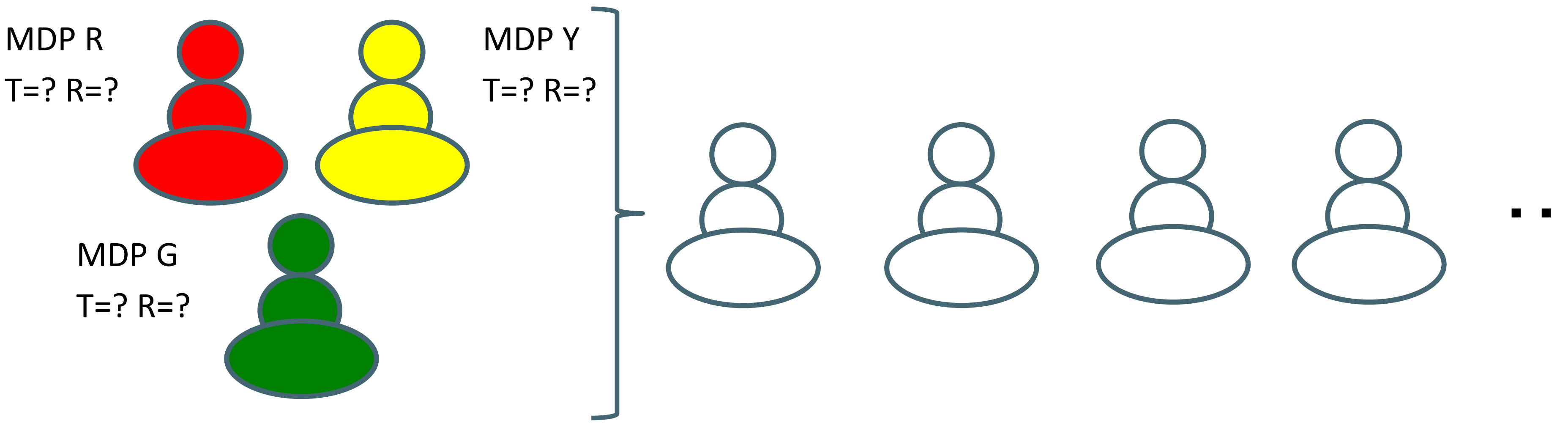


Series of tasks

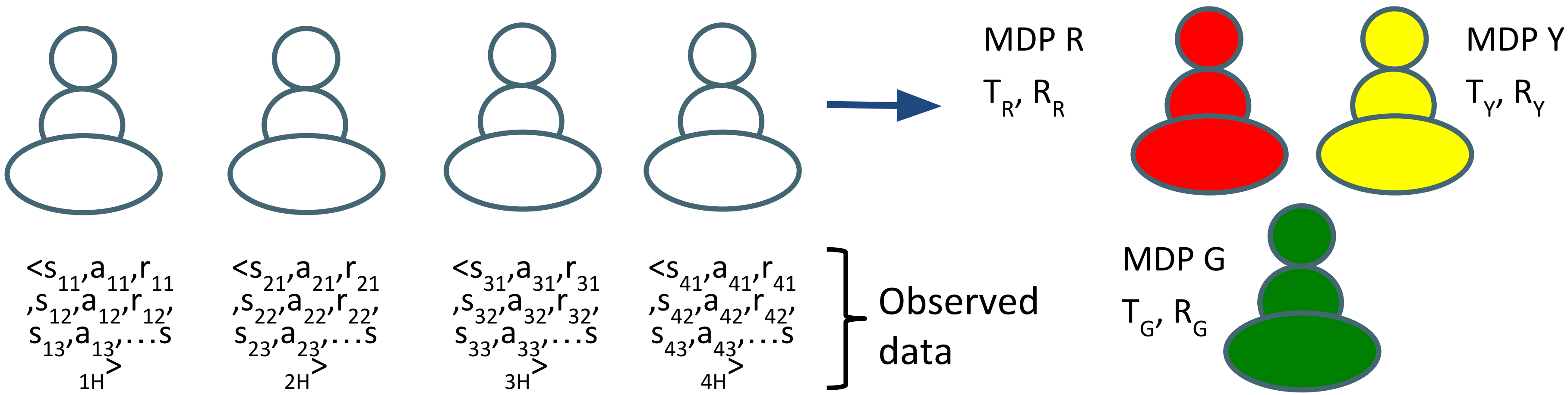
Act in each task for H steps


MDP R
T_R, R_R
MDP Y
T_Y, R_Y
MDP G
T_G, R_G
...

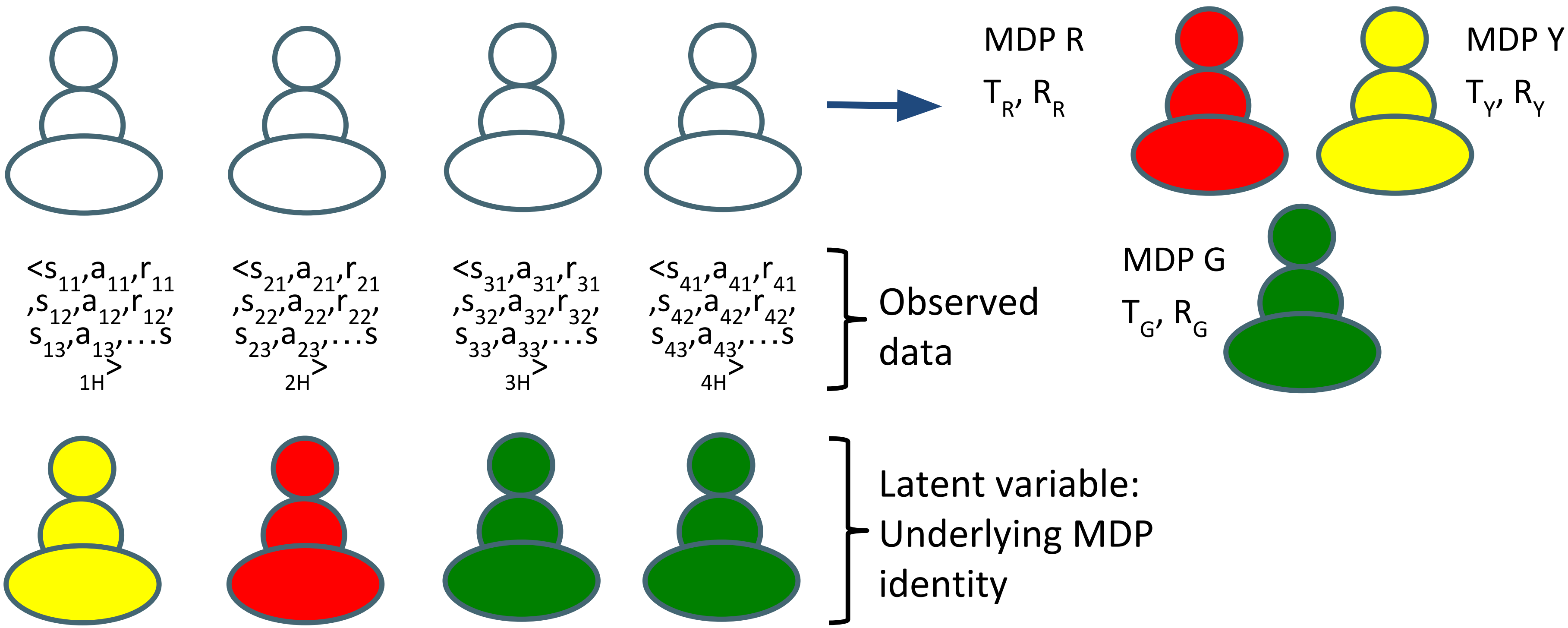
Latent Variable Modeling



Latent Variable Modeling



Latent Variable Modeling



Latent Variable Modeling

- Formally hard problem
- Expectation Maximization has weak theoretical guarantees
- Recent finite sample bounds on learned parameter estimates

Latent Variable Modeling

Assume for any 2 finite state—action MDPs M_i & M_j, there exists at least one state—action pair such that

$$\|\theta_i(\cdot|s,a) - \underbrace{\theta_j(\cdot|s,a)}_{\text{Vector of transition \& reward parameters for (s,a) for MDP } M_j}\| > \Gamma$$

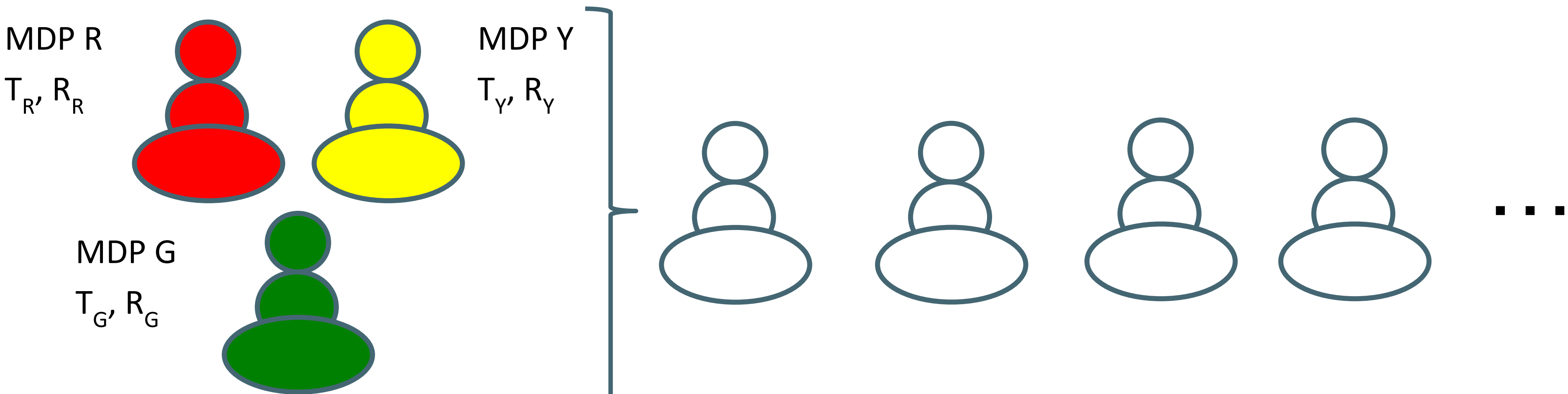
Note: to guarantee ε-optimal performance, very small differences in models are irrelevant. *Implies above property always holds in discrete MDPs for some Γ= f(ε)*

Implications

- Assume can visit any part of the decision making task an unbounded number of times
- If time horizon per task sufficiently long, can learn $O(\Gamma)$-accurate task parameters with high probability

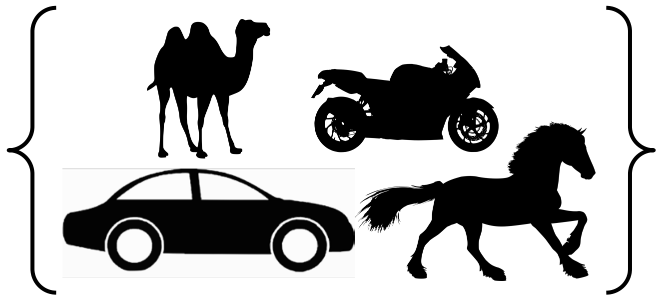
$\rightarrow$ Can correctly cluster tasks

Enables Provably Faster Learning in Finite Set of Tasks

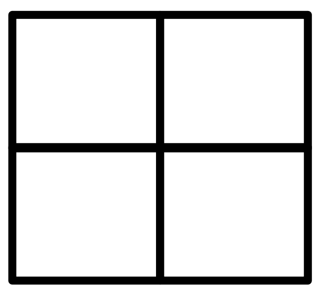


Setting

Multitask:



Tabular



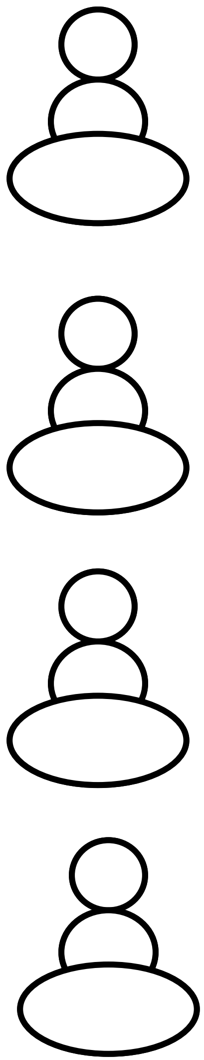
Multi-task RL

Or all customers using Amazon, or patients, or robot farm...

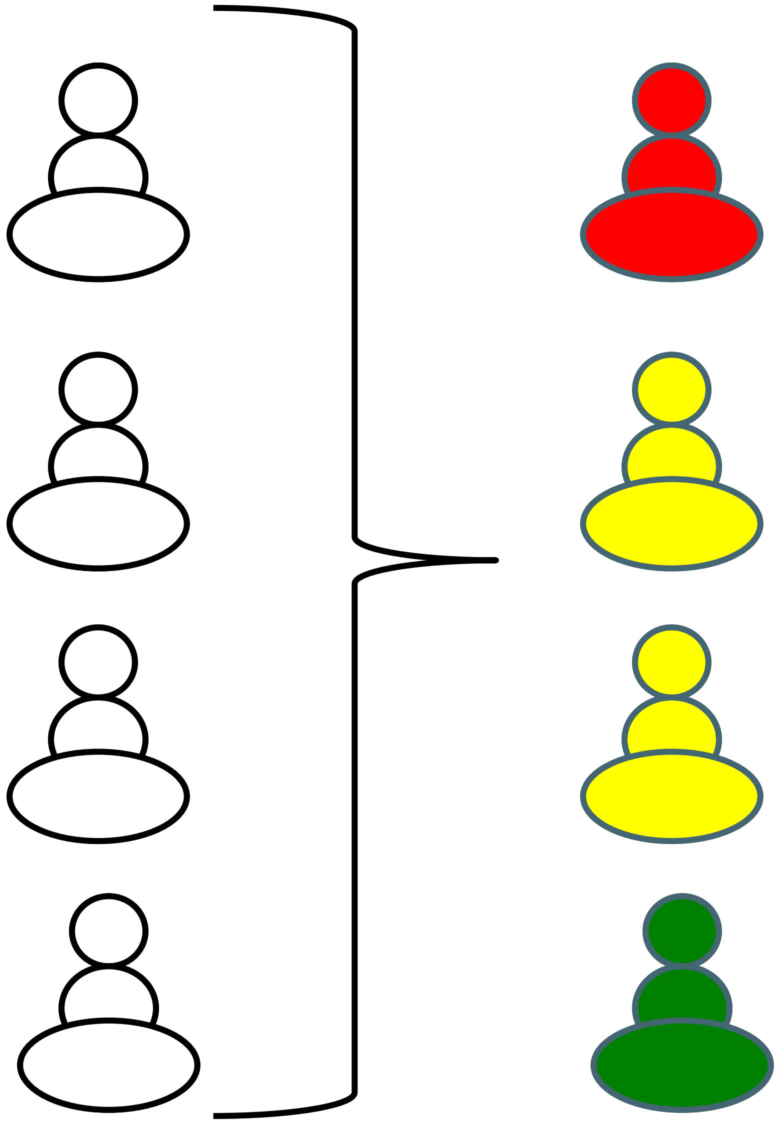
Provably Speeding Multitask RL

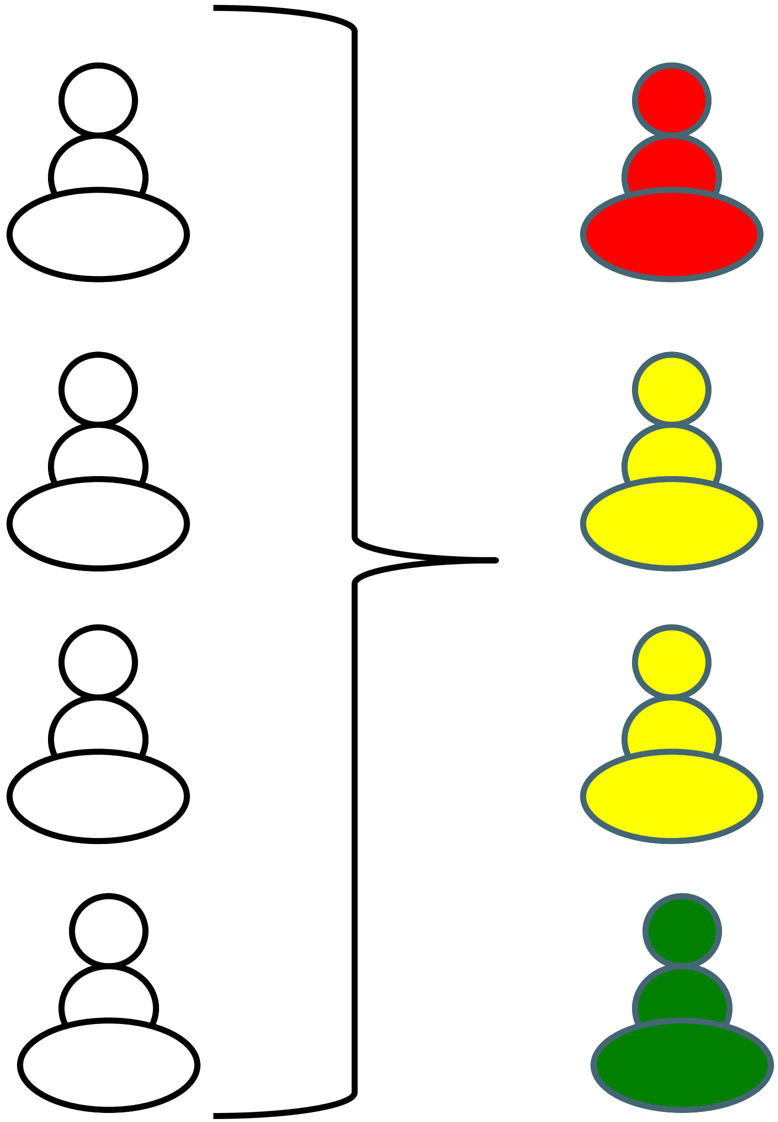
Guo and Brunskill, AAAI 2015

- Assumptions: K tasks sampled from M tasks
- Evaluation goal: Provably improve performance
- Approach: Quickly cluster and then share

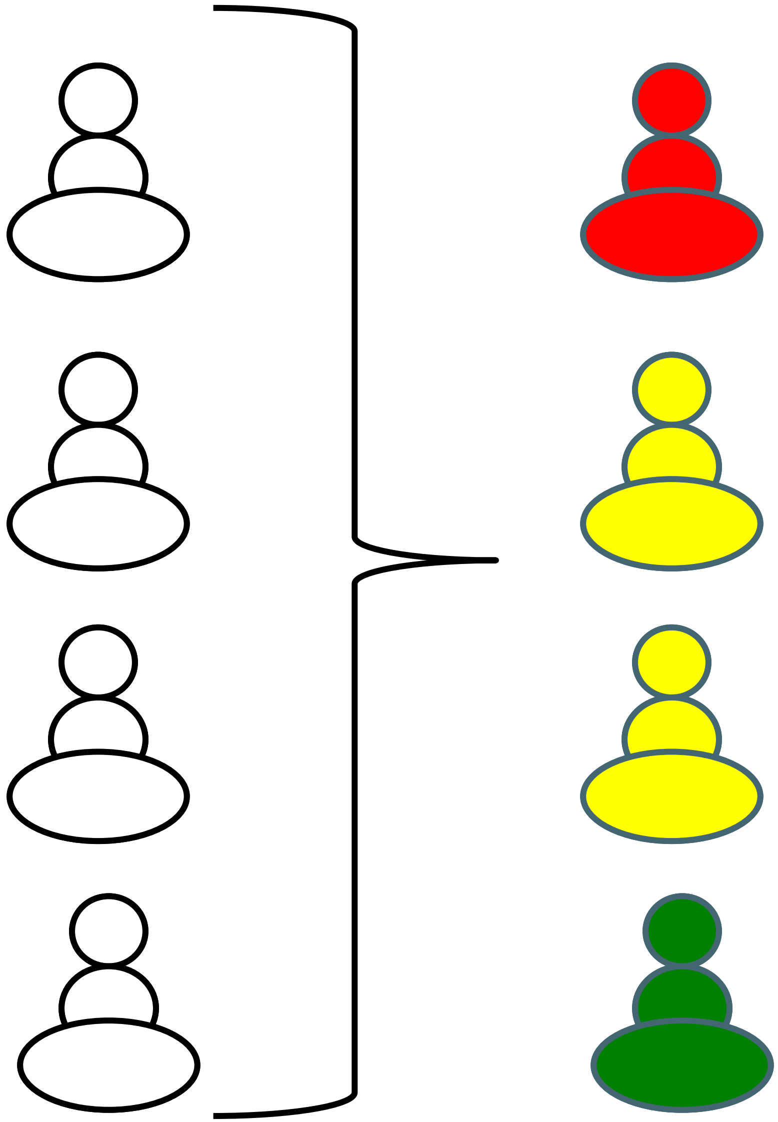


Cluster Tasks





Going Forward Share Data Across Similar Tasks



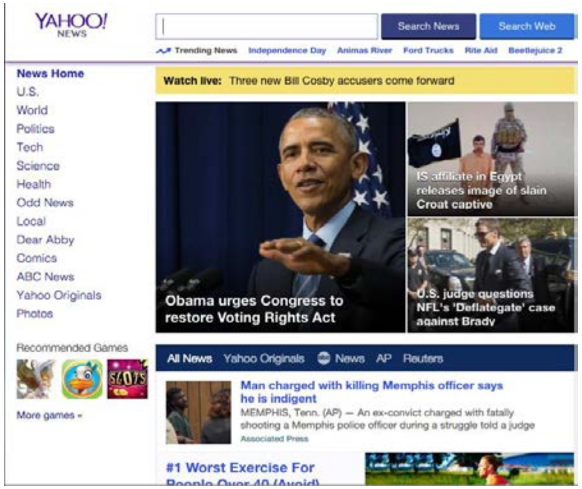
If Clusters are Well Separated, → Cluster Quickly and Provably Speed Learning

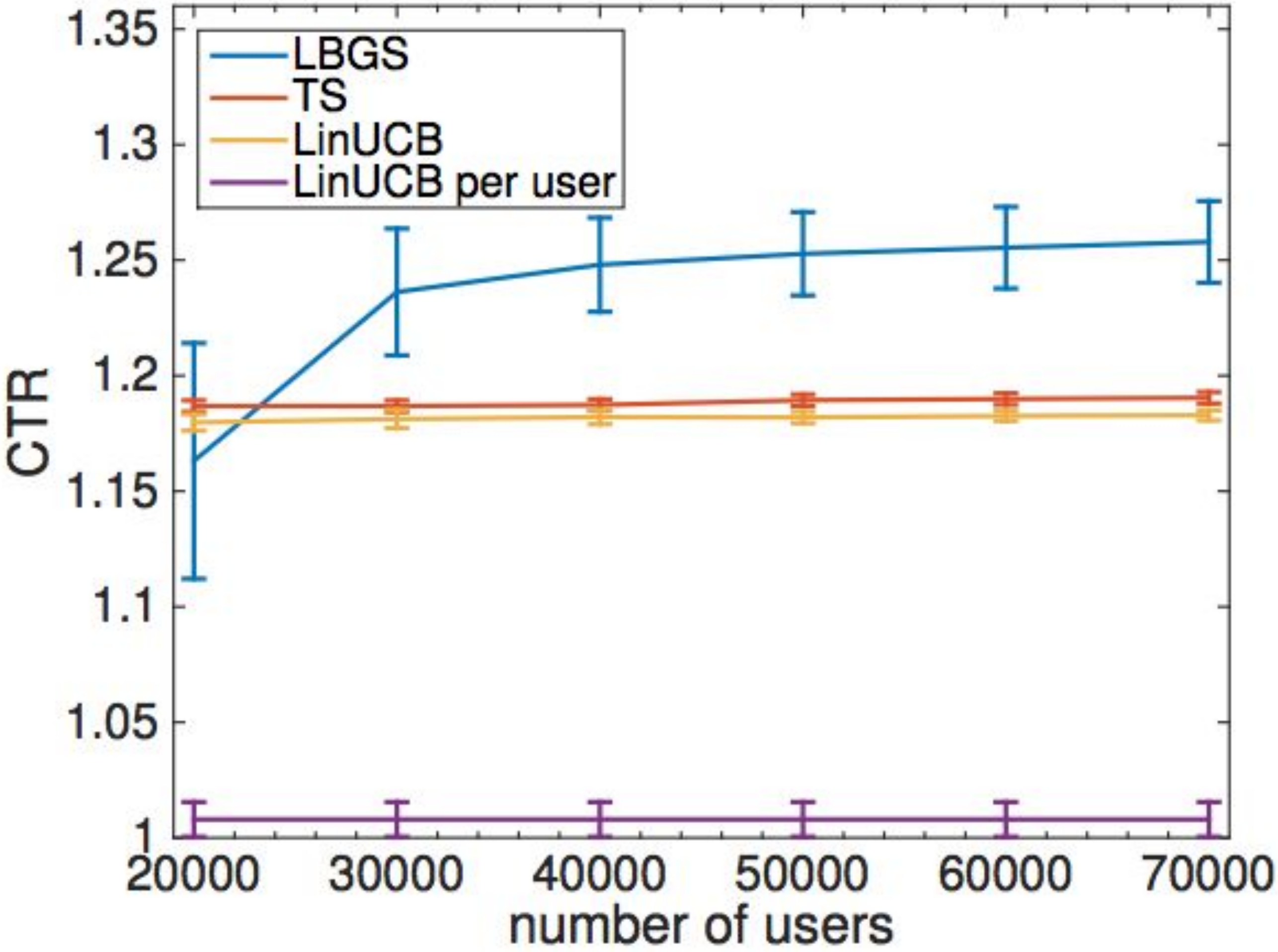
Latent Variable Modeling for Provably Improved RL

- Separability assumptions
  - Concurrent RL (Guo & B., AAAI 2015)
  - Multi-task RL options learning (Li & B. ICML 2014)
  - Continuous-state multi-task RL (Liu, Guo & B. AAMAS 2016 16)
- Method of moments
  - Multi-task bandits (Azar, Lazaric and B NIPS 2013)
  - Multi-task Contextual latent bandits (Zhou and B, IJCAI 2016)

Offline Evaluation of Online Latent Contextual Bandit for News Personalization

Zhou and Brunskill IJCAI 2016



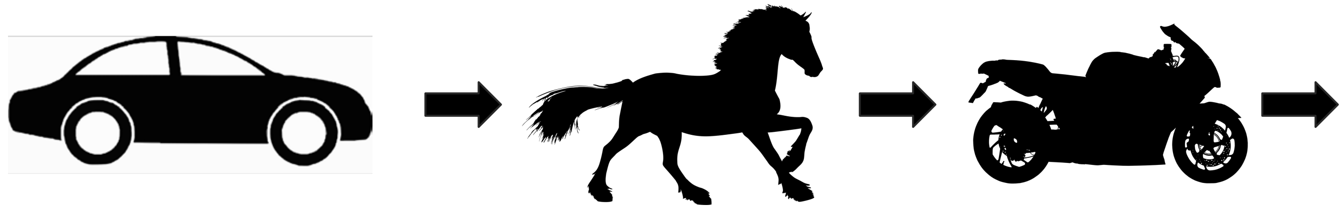


Two Core Parts of Multi-Task / Meta RL

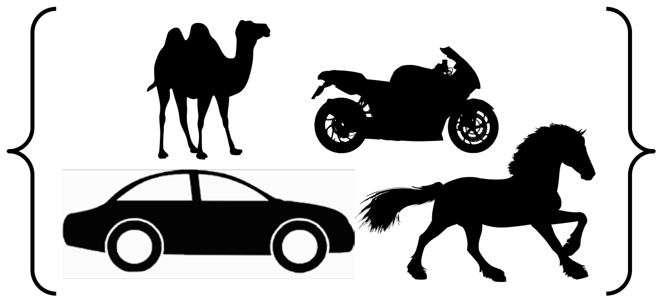
- Summarize experience across tasks
  - As a finite set of tasks (clustering)
  - **As a low dimensional subspace**
  - As a set of parameters near to desired set
- Use summary to improve learning in new task
  - As initialization to standard RL algorithm
  - **To new RL algorithm to direct exploration**

Settings

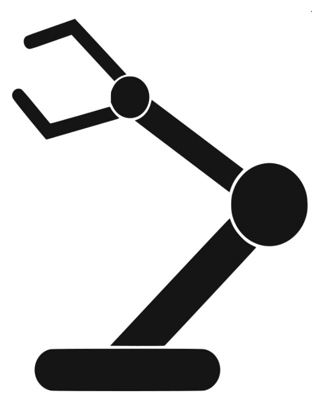
Lifelong:



Multitask:



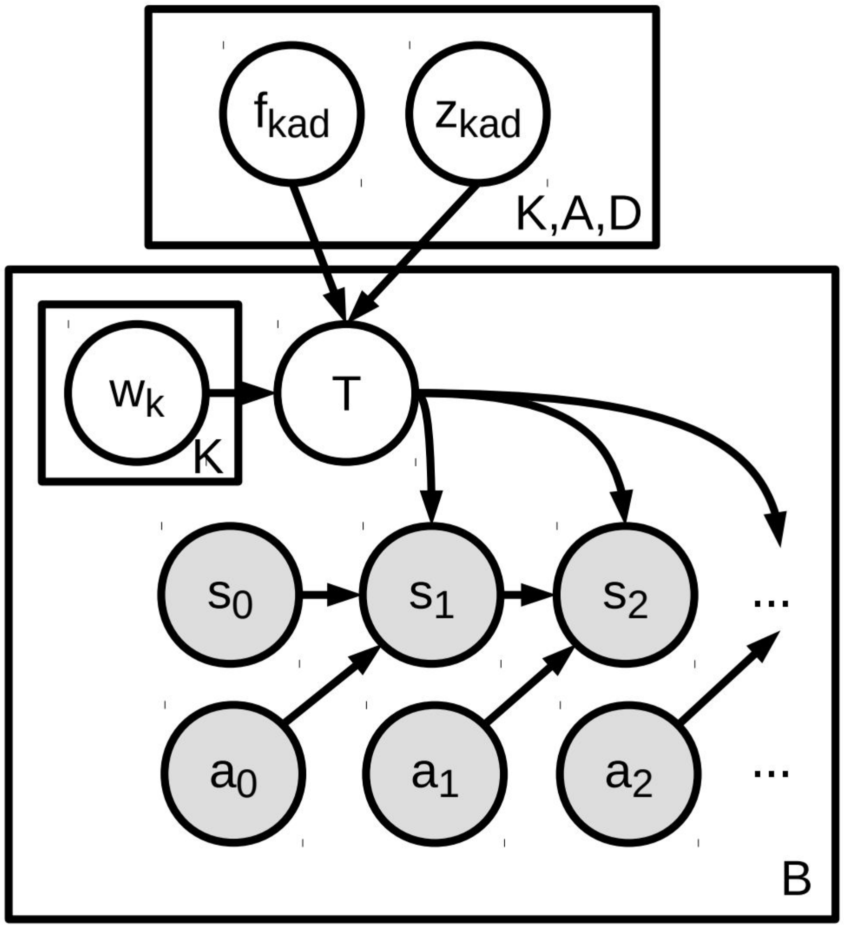
Function Approximation



Emma Brunskill Stanford University @aiforhi https://cs.stanford.edu/people/ebrun/

Hidden Parameter MDPs: Smooth Latent Space **Over Models**

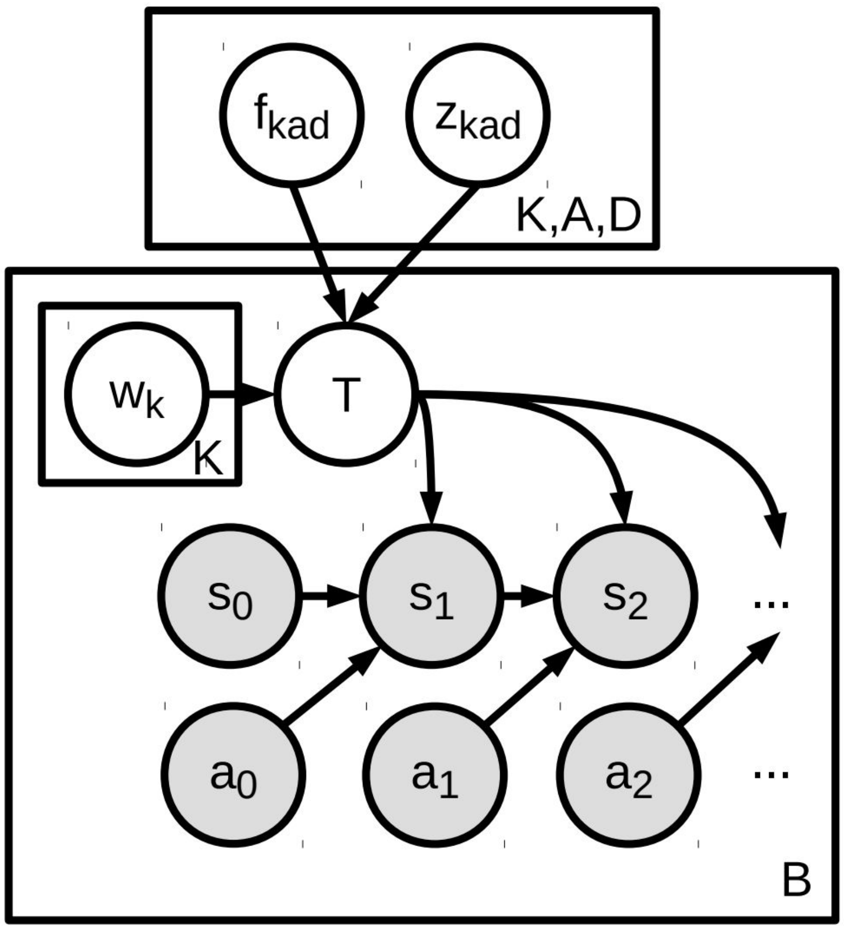
Doshi-Velez and Konidaris IJCAI 2016



$$
\begin{aligned}
(s'_d - s_d) &\sim \textstyle\sum_k^K z_{kad} w_{kb} f_{kad}(s) + \epsilon \\
\epsilon &\sim N(0, \sigma^2_{nad}),
\end{aligned}
$$

More Robust Hidden Parameter MDPs

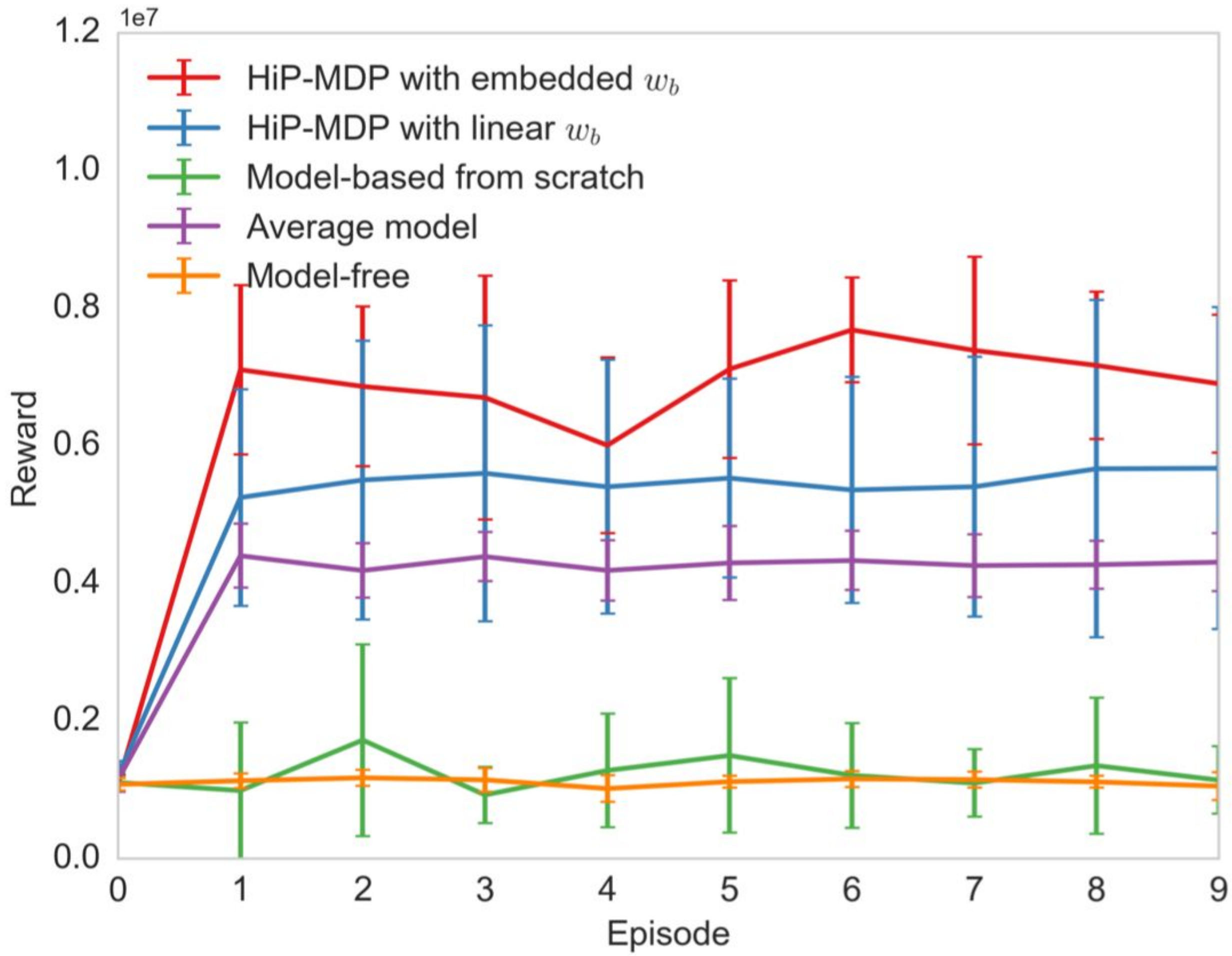
Killian, Konidaris, Doshi-Velez. NIPS 2017



→ Use Bayesian Neural Networks for modeling the dynamics

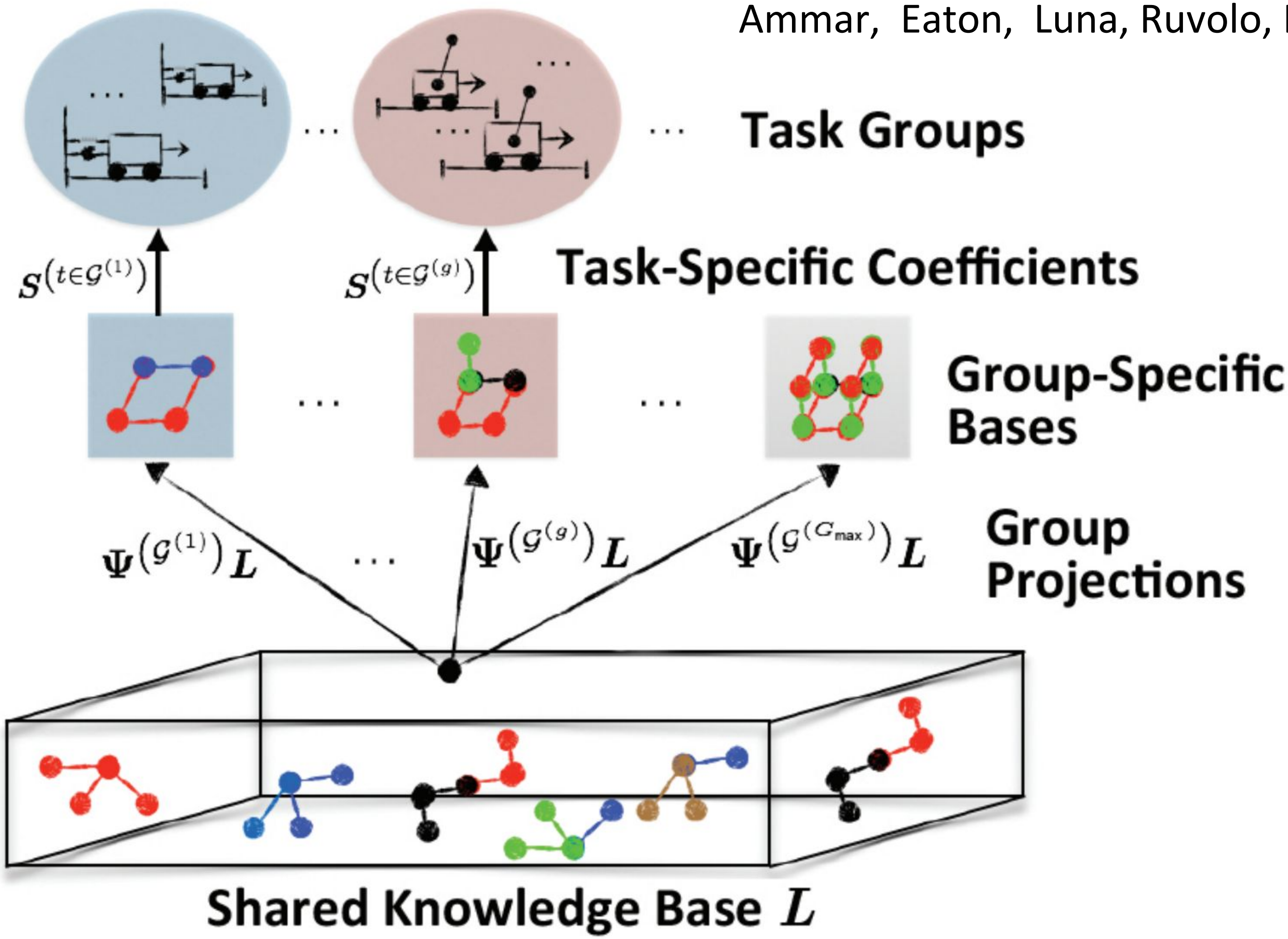
Better Transfer on HIV Simulator Across Patients

Killian, Konidaris, Doshi-Velez. NIPS 2017



Smooth Latent **Policy Space**

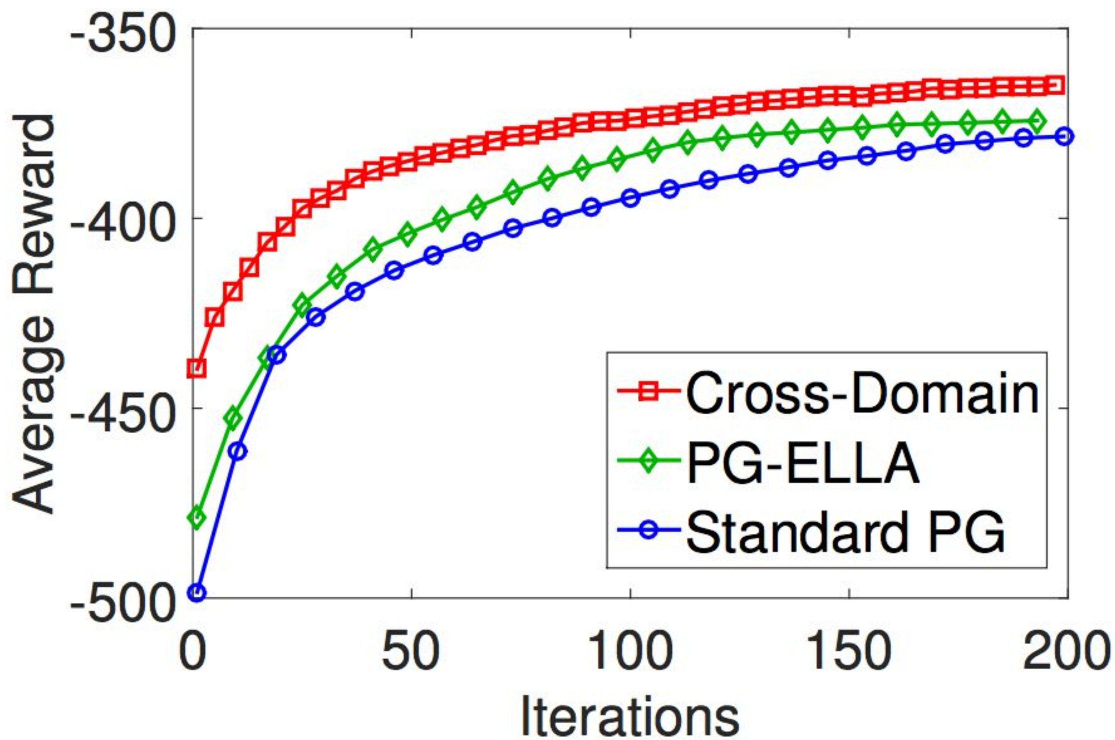
Ammar, Eaton, Luna, Ruvolo, IJCAI 2015



Smooth Latent **Policy Space for Cross Domain Transfer**

Ammar, Eaton, Luna, Ruvolo, IJCAI 2015

- Set of policies with shared basis set of parameters
- Can be used to do cross domain transfer (different state & actions)

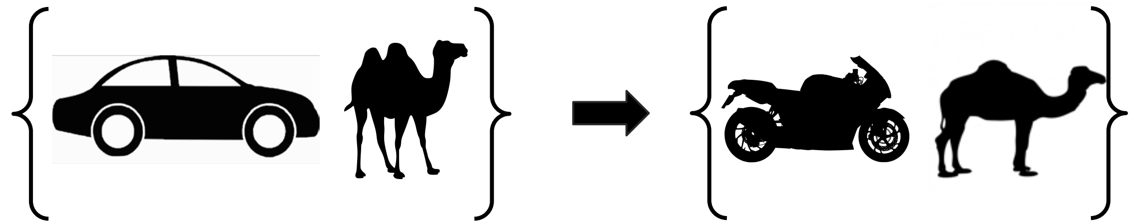


Two Core Parts of Multi-Task / Meta RL

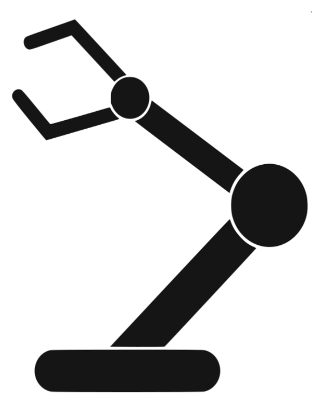
- Summarize experience across tasks
  - As a finite set of tasks (clustering)
  - As a low dimensional subspace
  - **As a set of parameters near to desired set**
- Use summary to improve learning in new task
  - As initialization to standard RL algorithm
  - **To new RL algorithm to direct exploration**

Setting

Many $\rightarrow$ Many:

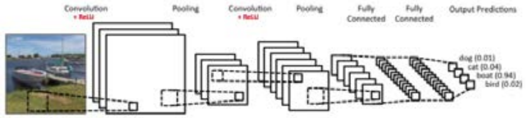


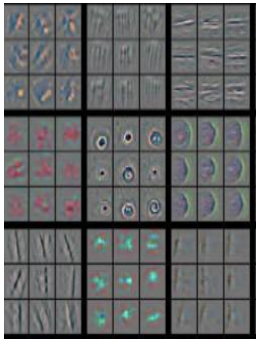
Function Approximation

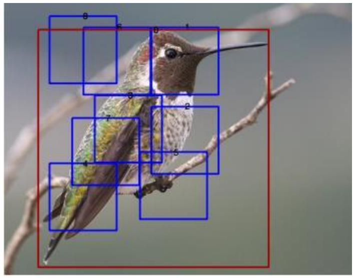


Inspiration: Pretraining

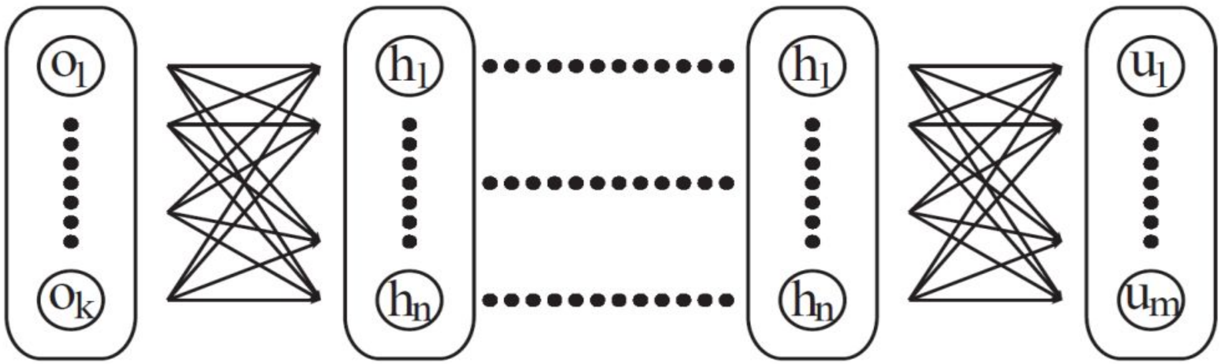
Slide from Sergey Levine







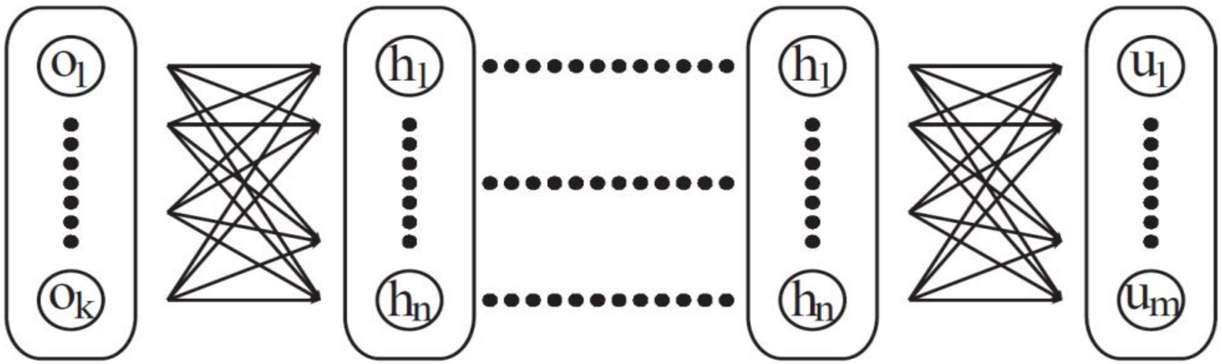
Review: Single Task Policy Gradient



$$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$$

$$\mathcal{L}_{\mathcal{T}_i}(f_\phi) = -\mathbb{E}_{\mathbf{x}_t, \mathbf{a}_t \sim f_\phi, q_{\mathcal{T}_i}} \left[\sum_{t=1}^{H} R_i(\mathbf{x}_t, \mathbf{a}_t) \right]$$

How to Choose Initial Parameters to Speed Learning?



$$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$$

$$\mathcal{L}_{\mathcal{T}_i}(f_\phi) = -\mathbb{E}_{\mathbf{x}_t, \mathbf{a}_t \sim f_\phi, q_{\mathcal{T}_i}} \left[\sum_{t=1}^{H} R_i(\mathbf{x}_t, \mathbf{a}_t) \right]$$

Parameters for Faster Future RL

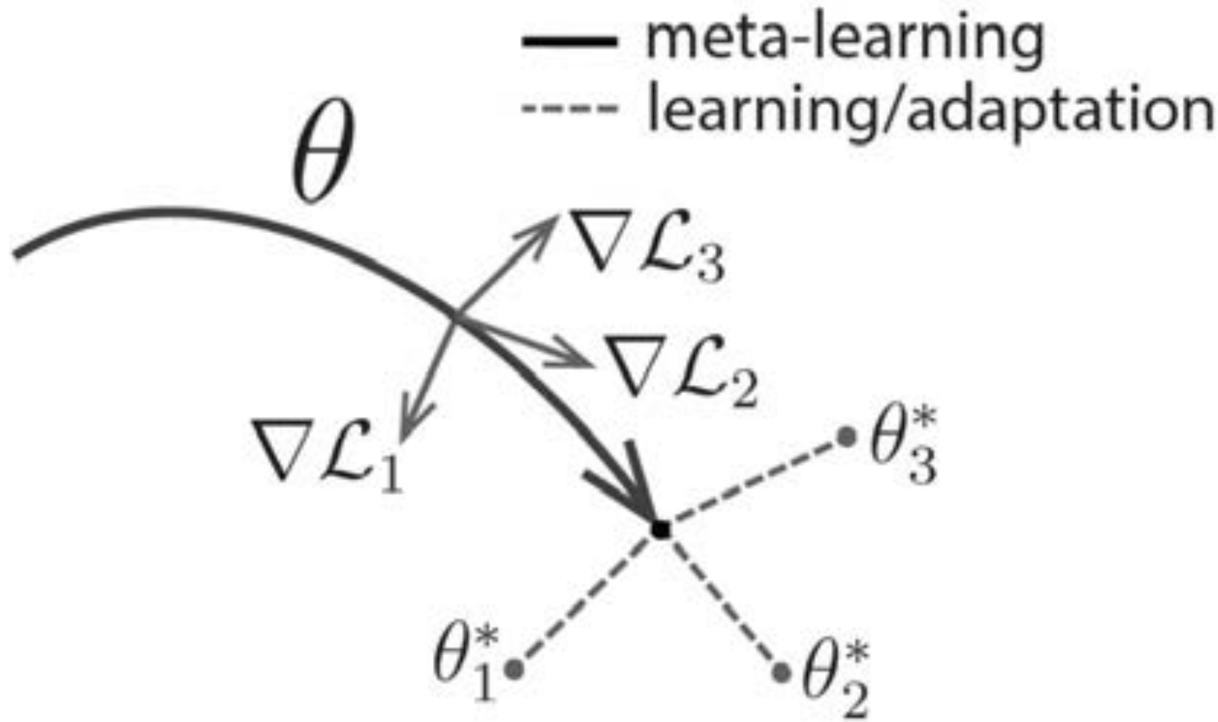
Finn et al., "Model-Agnostic Meta-Learning" ICML 2017

$$\min_{\theta} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta'_i}) = \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}\left(f_{\theta - \alpha \nabla_{\theta} \mathcal{L}_{\mathcal{T}_i}(f_{\theta})}\right)$$

set of tasks

Model Agnostic Meta-Learning

Finn et al., "Model-Agnostic Meta-Learning" ICML 2017



$\longrightarrow$ Learn θ so that it is "close" to good θ for many tasks:

One gradient step from θ on task yields high reward

Parameters for Faster Future RL

Finn et al., “Model-Agnostic Meta-Learning” ICML 2017

$$\min_{\theta} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta'_i}) = \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)})$$

set of tasks

Update meta-parameters θ by SGD

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta'_i})$$

MAML for RL

Finn et al., “Model-Agnostic Meta-Learning” ICML 2017

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters

randomly initialize θ
while not done **do**
  Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
  for all $\mathcal{T}_i$ **do**
    Sample K trajectories $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{a}_1, ...\mathbf{x}_H)\}$ using f_θ in $\mathcal{T}_i$
    Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ using $\mathcal{D}$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation 4
    Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
    Sample trajectories $\mathcal{D}_i' = \{(\mathbf{x}_1, \mathbf{a}_1, ...\mathbf{x}_H)\}$ using $f_{\theta_i'}$ in $\mathcal{T}_i$
  end for
  Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$ using each $\mathcal{D}_i'$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation 4
end while

Meta-Learning Parameters

Finn et al., "Model-Agnostic Meta-Learning" ICML 2017

Slide from Sergey Levine

supervised learning: $f(x) \to y$

supervised meta-learning: $f(\mathcal{D}_{\text{train}}, x) \to y$

model-agnostic meta-learning: $f_{\text{MAML}}(\mathcal{D}_{\text{train}}, x) \to y$

$$f_{\text{MAML}}(\mathcal{D}_{\text{train}}, x) = f_{\theta'}(x)$$

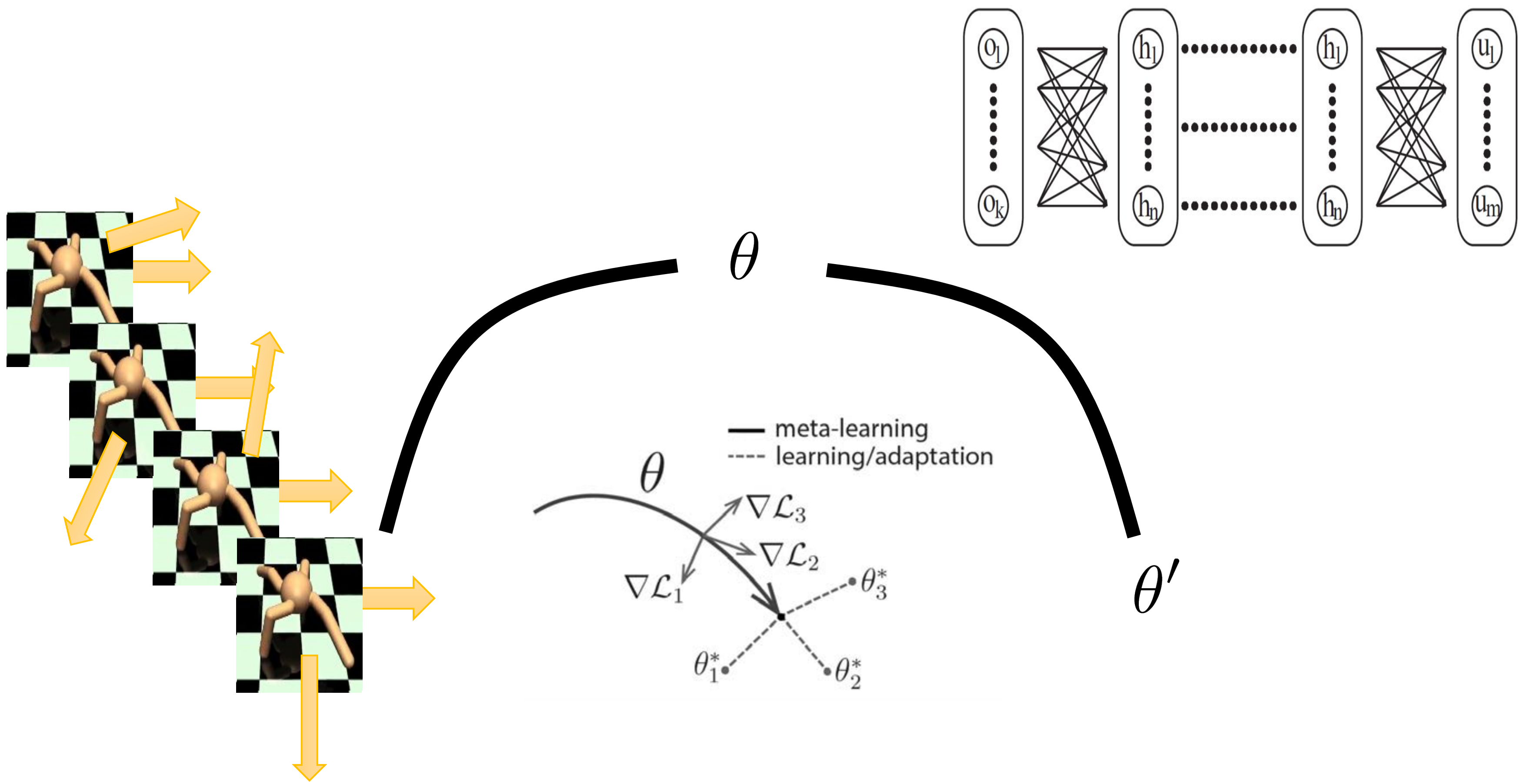
$$\theta' = \theta - \alpha \sum_{(x,y) \in \mathcal{D}_{\text{train}}} \nabla_\theta \mathcal{L}(f_\theta(x), y)$$

Just another computation graph…

Can implement with any autodiff package (e.g., TensorFlow)

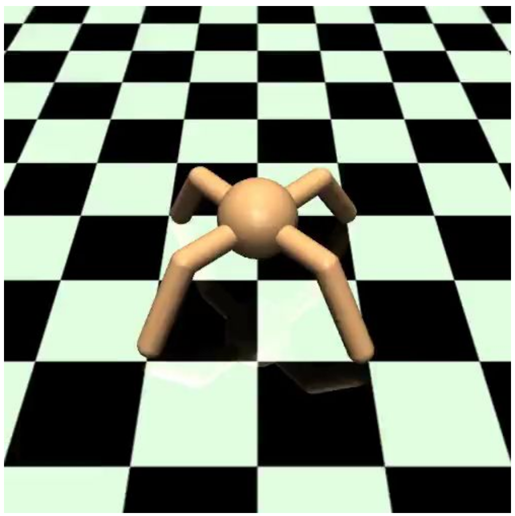
But has favorable inductive bias…

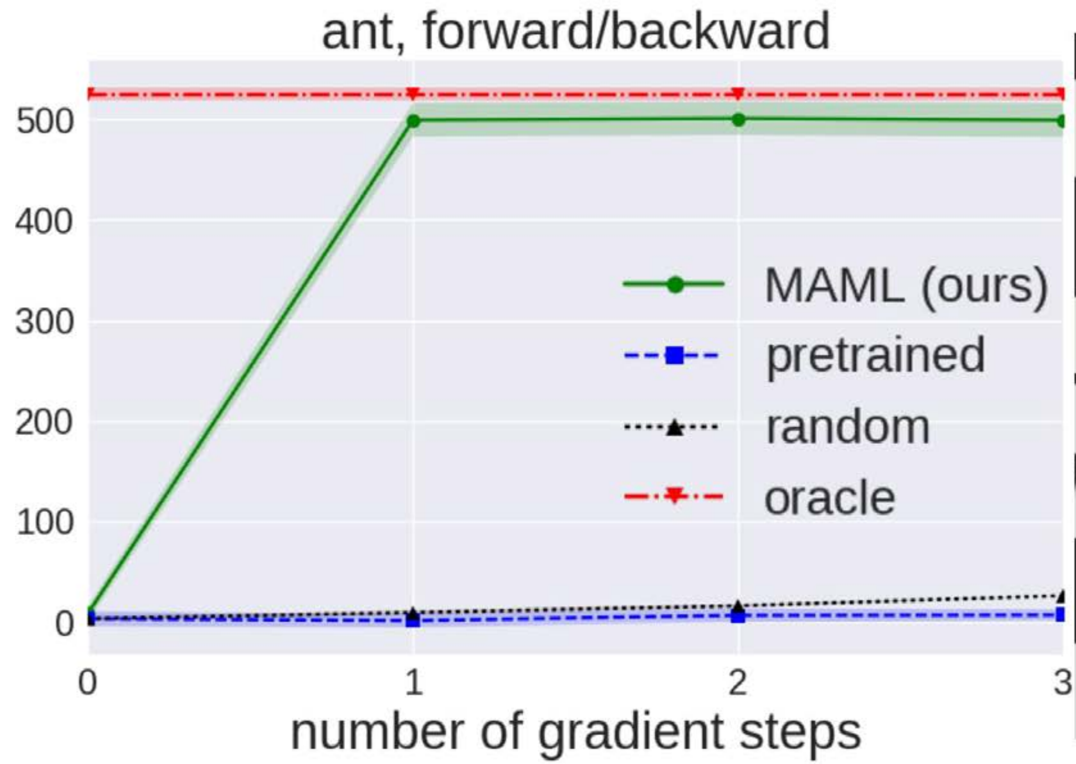
Train Meta-Parameters Across Set of Tasks



Model-agnostic meta-learning: accelerating PG

Finn et al., “Model-Agnostic Meta-Learning” ICML 2017





Many nice extensions (including model based)
Very helpful for 1-shot learning in related tasks

Two Core Parts of Multi-Task / Meta RL

- Summarize experience across tasks
  - As a finite set of tasks (clustering)
  - As a low dimensional subspace
  - As a set of parameters near to desired set
- Use summary to improve learning in new task
  - As initialization to standard RL algorithm
  - To new RL algorithm to direct exploration

Open Questions & Directions

- Detecting and recovering from negative transfer
- Changing how to behave in current tasks to improve future performance on later tasks
- Curriculum design and meta-learning

Multi-Task / Meta RL

- Summarize experience across tasks
    - As a finite set of tasks (clustering)
    - As a low dimensional subspace
    - As a set of parameters near to desired set
- Use summary to improve learning in new task
    - As initialization to standard RL algorithm
    - To new RL algorithm to direct exploration