Towards Faster Learning of Good Decisions

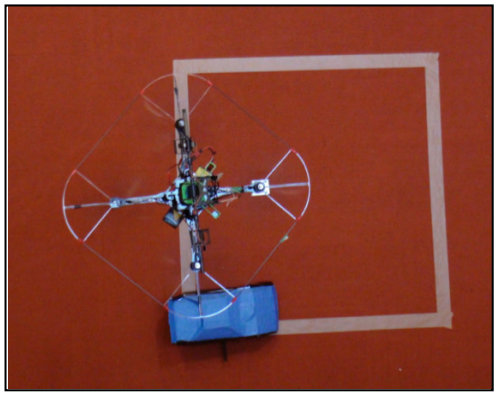
Emma Brunskill
Carnegie Mellon University

Work described down in collaboration with: Daniel Guo, Dexter Lee, Lihong Li, Yun-En Liu, Travis Mandel, Joseph Runde, Zoran Popovic

Want autonomous agents that act well (make good sequences of decisions)

Renewable resource allocation

Marketing



Machine repair

Server job scheduling

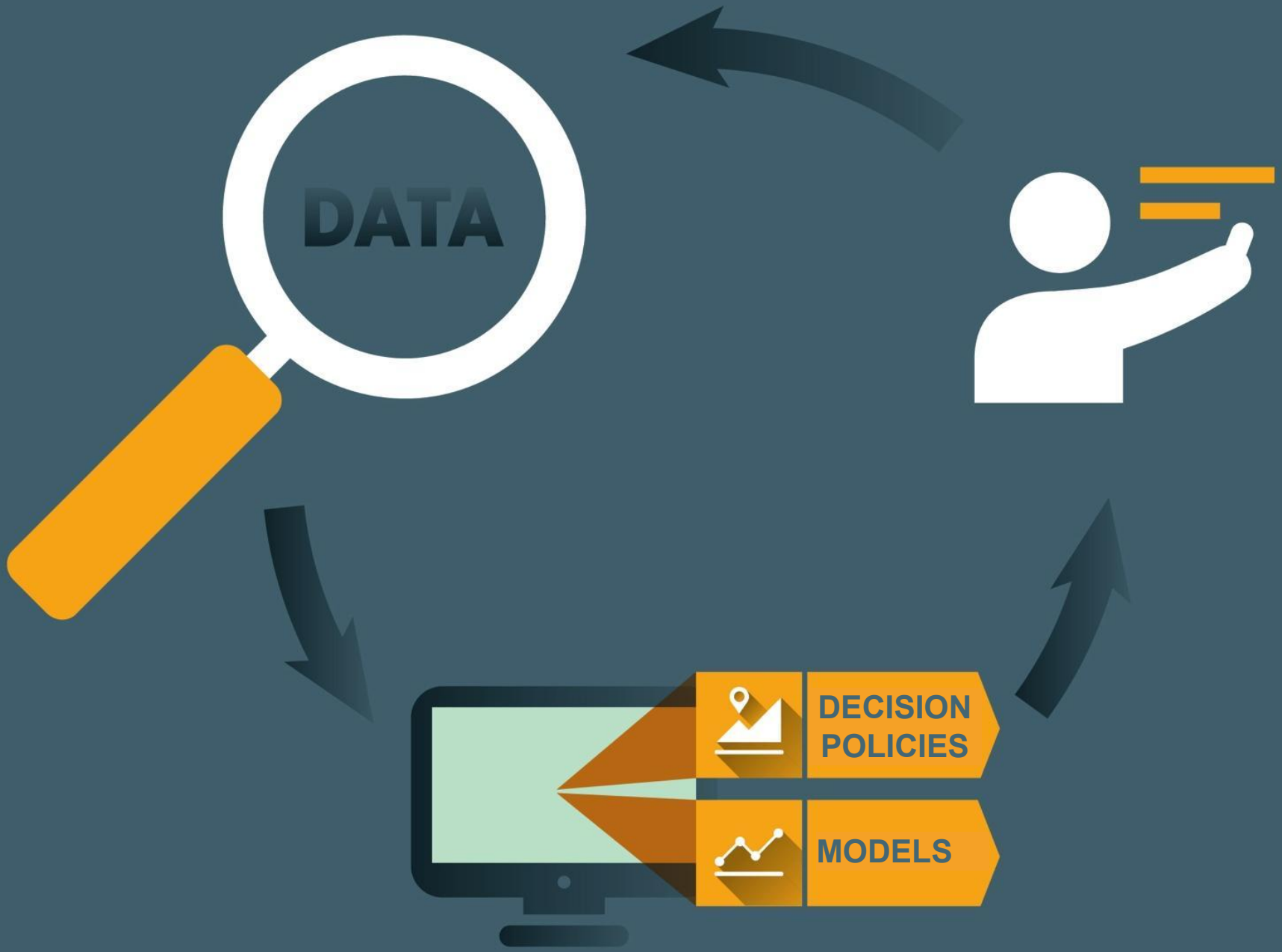
Automated customer support

Inventory ordering

Most AI algorithms developed for robots

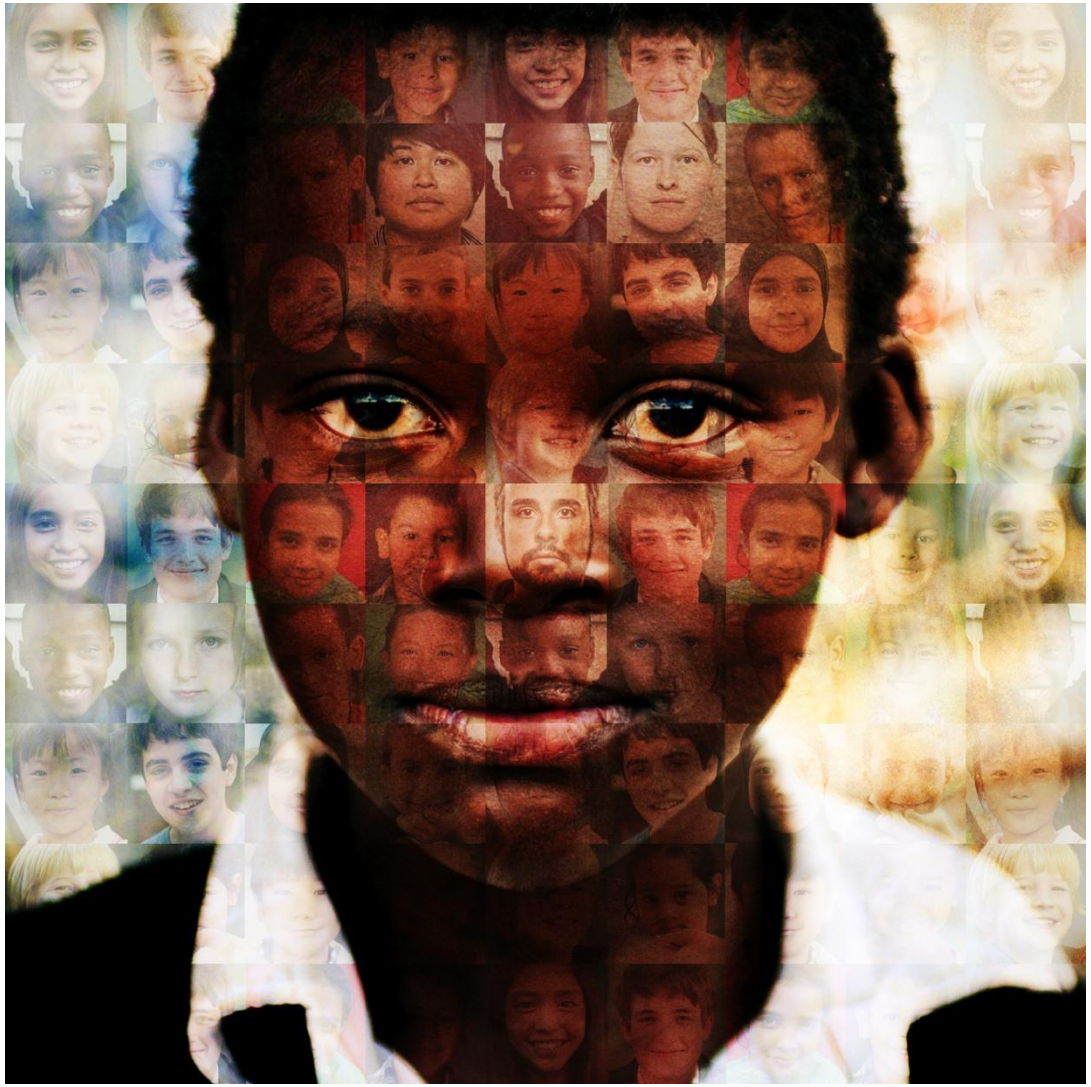
Want algorithms to enable:

Warner Brothers


DATA
DECISION
POLICIES
MODELS

Agents Making Decisions as Interact with People

Data = People



Towards Faster Learning

- Transfer learning
- Building on offline data
- Leveraging expert input

Formalizing Sample Efficiency of RL Algorithms

- **One measure: is it Probably Approximately Correct?**
  - Makes good decisions on all but the sample complexity number of steps
  - Sample complexity is polynomial function of problem parameters
  - E^3 (Kearns & Singh), R-MAX (Brafman & Tennenholtz)

Sample complexity:

number of actions may choose whose value is potentially far from optimal action's value

(informally: # of mistakes made by algorithm)

Unfortunately, this can be a lot

$$O\left(\frac{SA\log(SA/\delta)}{(1-\gamma)^3\varepsilon^2}\right)$$

E.g. (Azar et al. 2013) lower bound

Reality check:

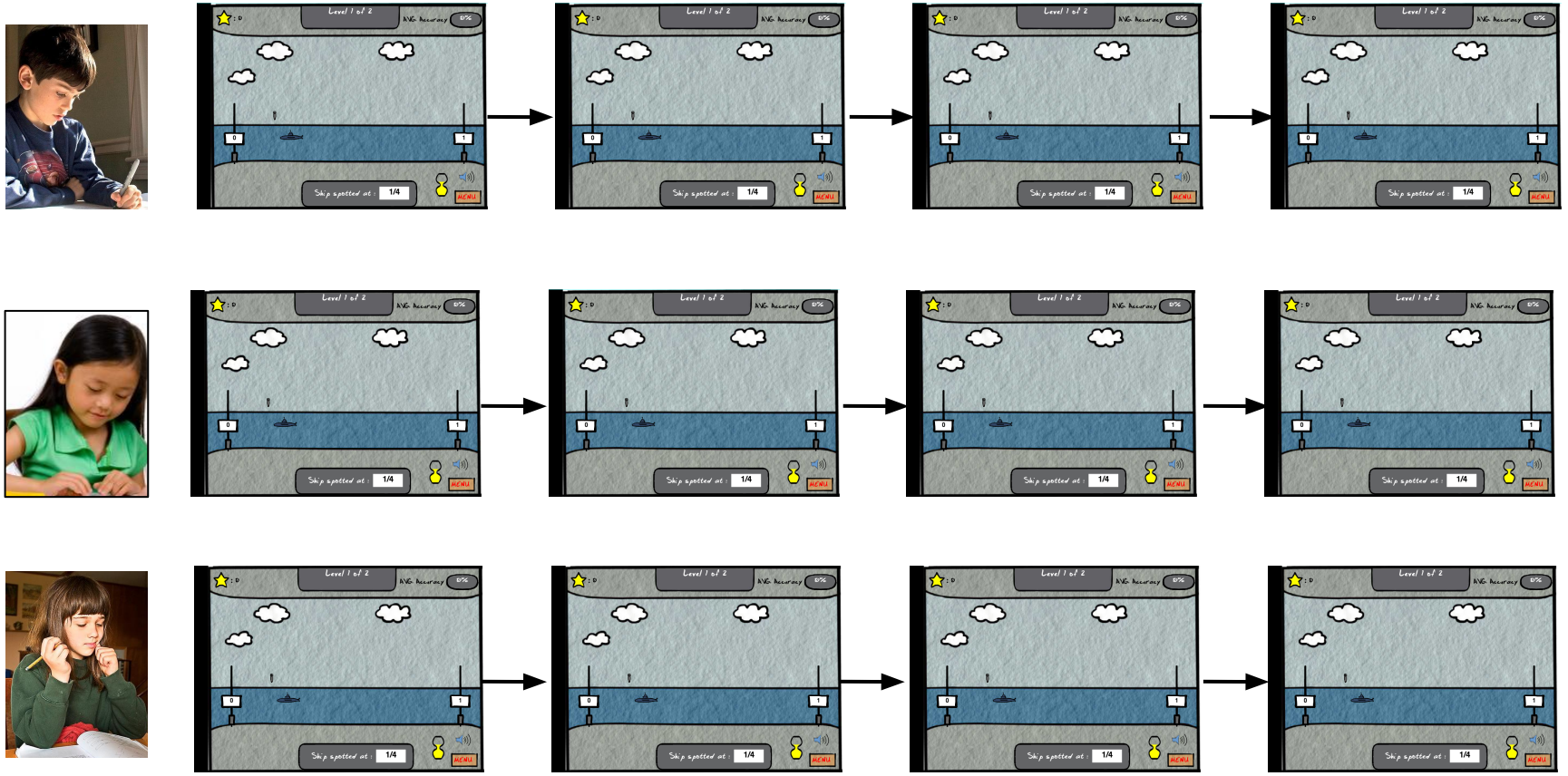
ε=0.1,γ=0.9: 10^5 samples per state

ε=0.1,γ=0.99: 10^8 samples per state

Approach: Share Knowledge

- Leverage provided information
  - Given input policy set (regret guarantees relative to best input policy, ECML 2013)
  - Given finite model sets (AAMAS 2012)
- Learn and transfer useful knowledge
  - Transfer learning (lifelong / multitask learning) (UAI 2013, NIPS 2013, ICML 2014, AAAI 2015)

Transfer / Multitask / Lifelong Learning



Each task involves sequence of decisions

Leverage related tasks to improve performance

Transfer: Fundamental Questions

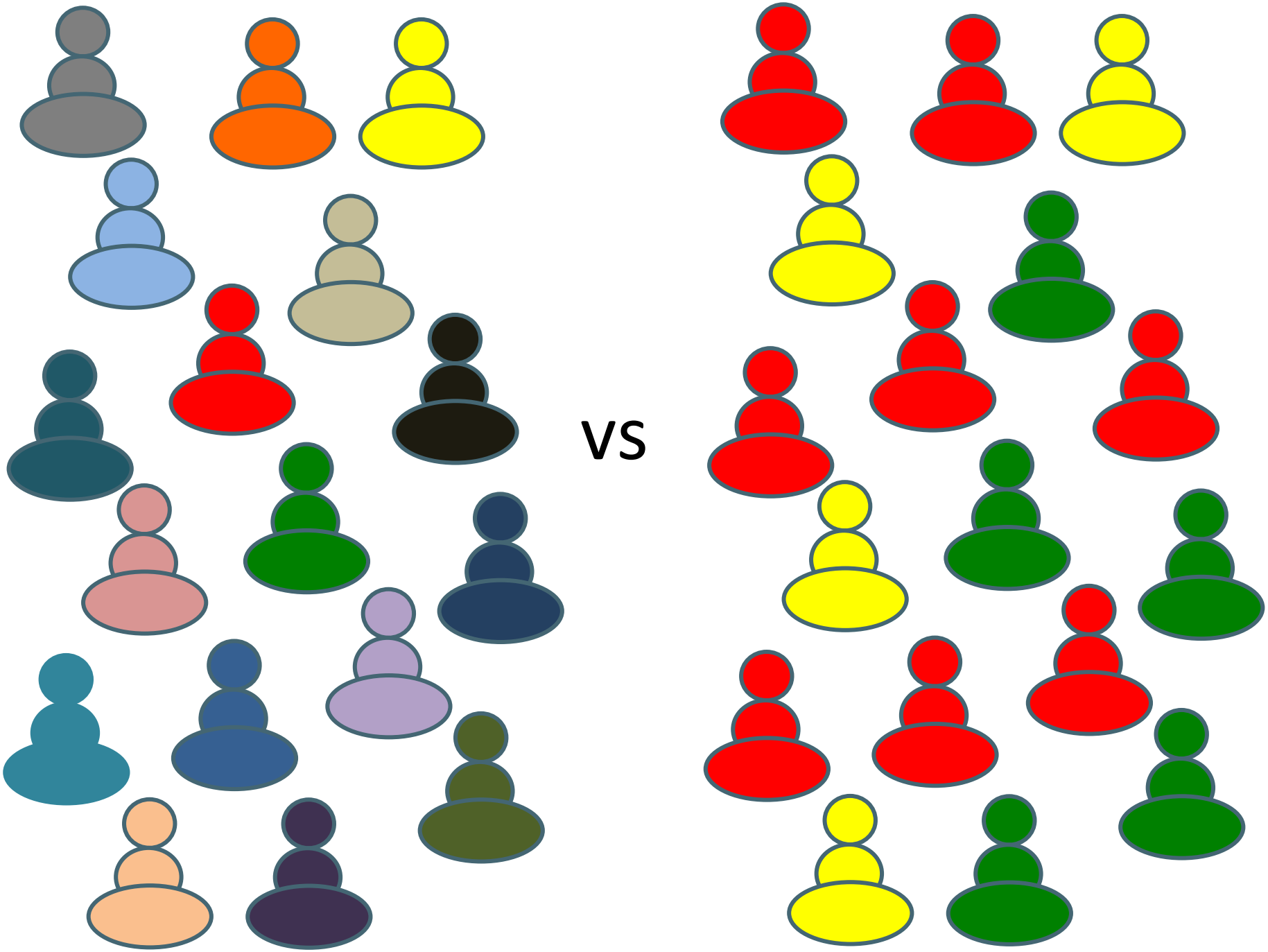
- Can learning be sped up across multiple tasks?
- Can computational costs be reduced when doing multiple tasks?
- Is different behavior optimal when an agent is maximizing performance across a set of tasks?

* task = reinforcement learning in a MDP

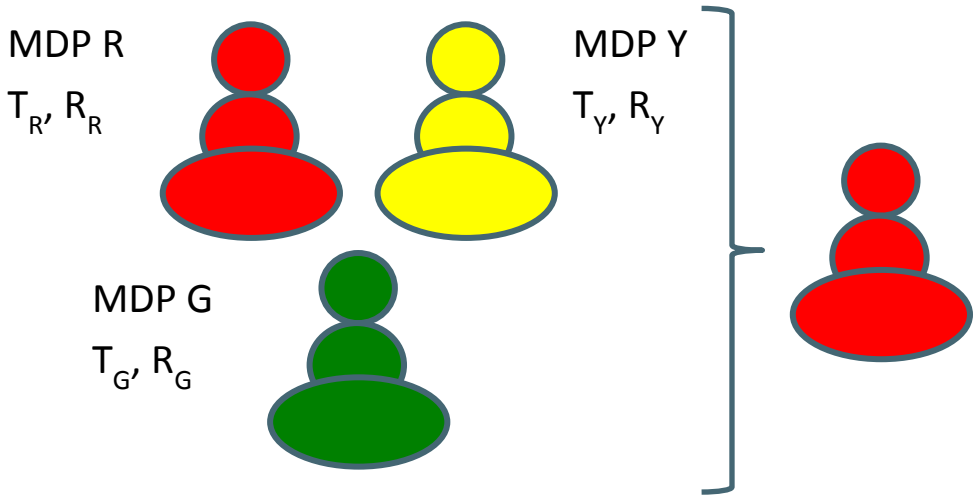
Sample complexity:

number of actions may choose whose value is potentially far from optimal action's value

Can sample complexity get smaller by leveraging prior tasks?


VS

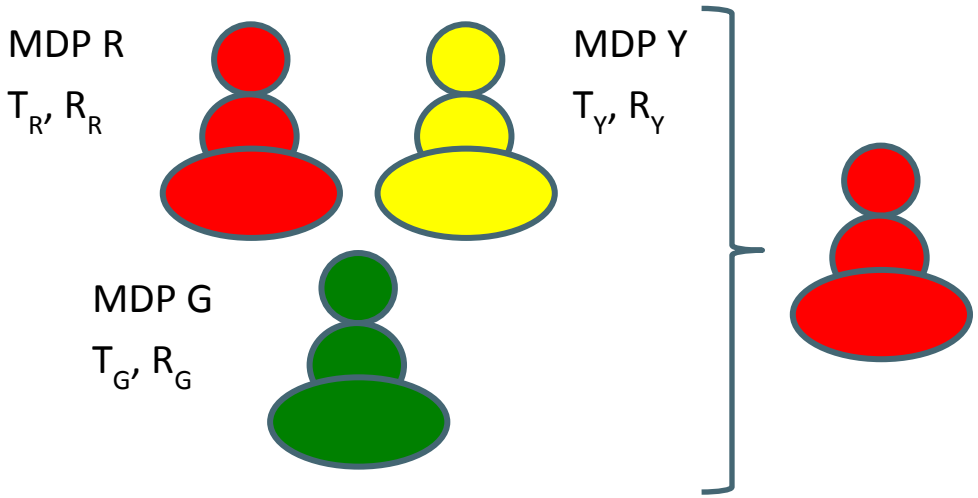
Multi-Task Reinforcement Learning with a Finite Set of Models



Sample a task from finite set of MDPs

Brunskill & Li, UAI 2013

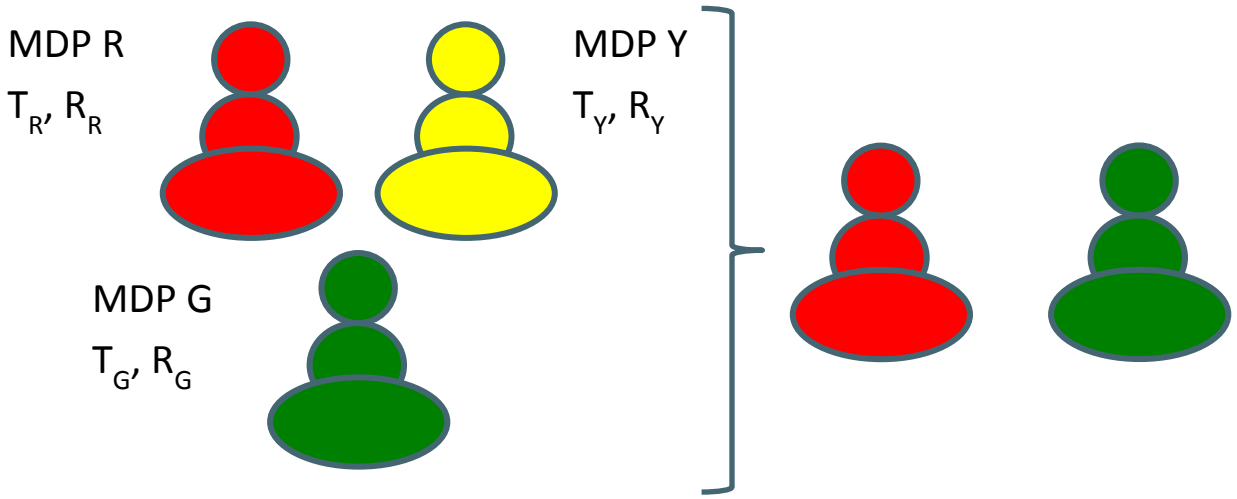
Multi-Task Reinforcement Learning with a Finite Set of Models



Act in it for H steps

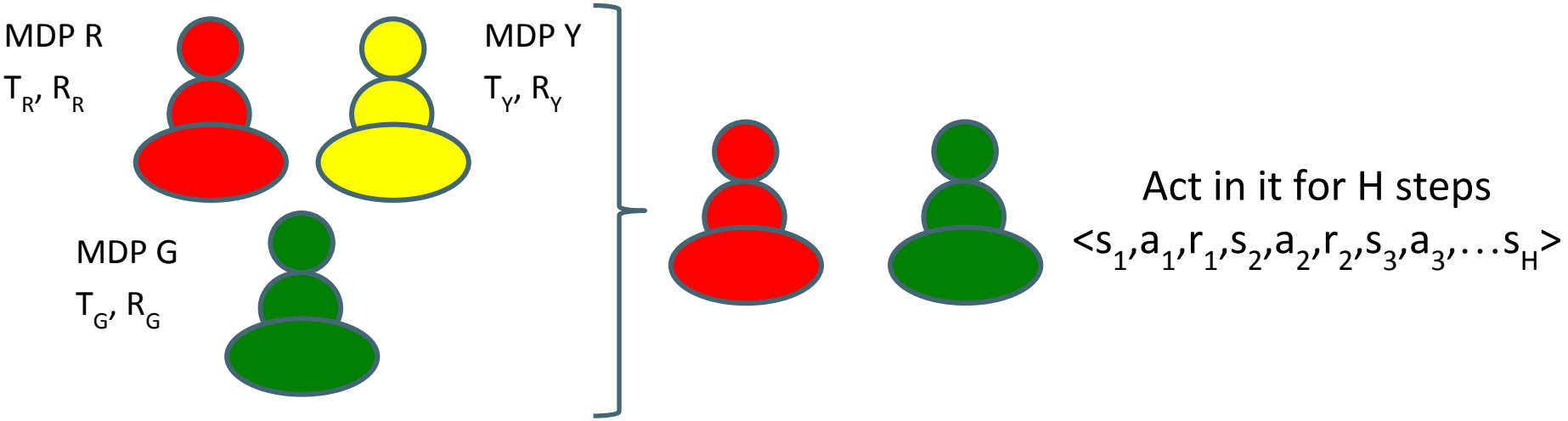
$<s_1,a_1,r_1,s_2,a_2,r_2,s_3,a_3,\ldots s_H>$

Multi-Task Reinforcement Learning with a Finite Set of Models

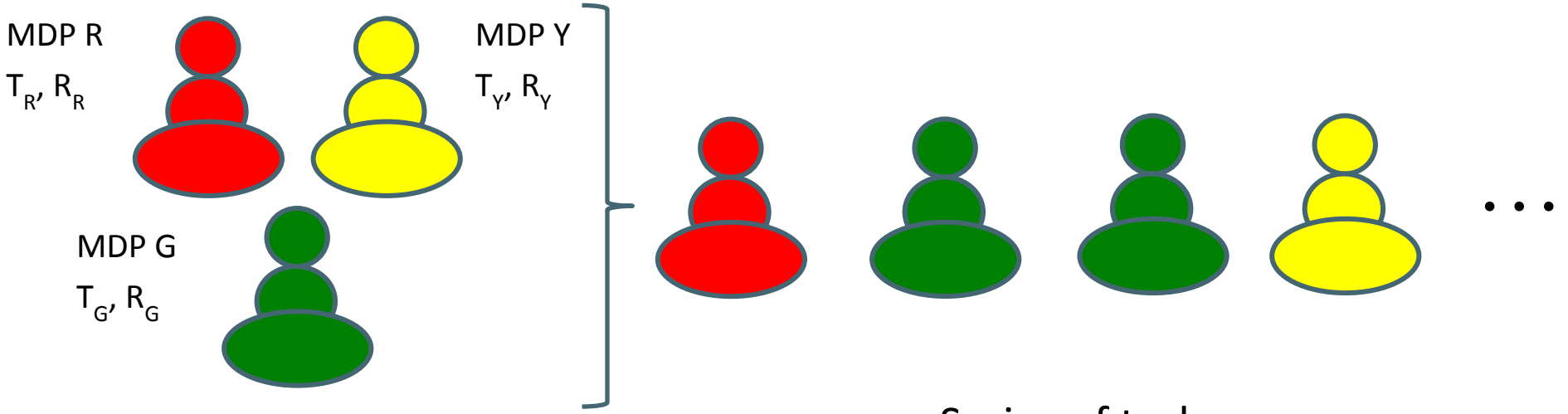


Again sample a MDP…

Multi-Task Reinforcement Learning with a Finite Set of Models

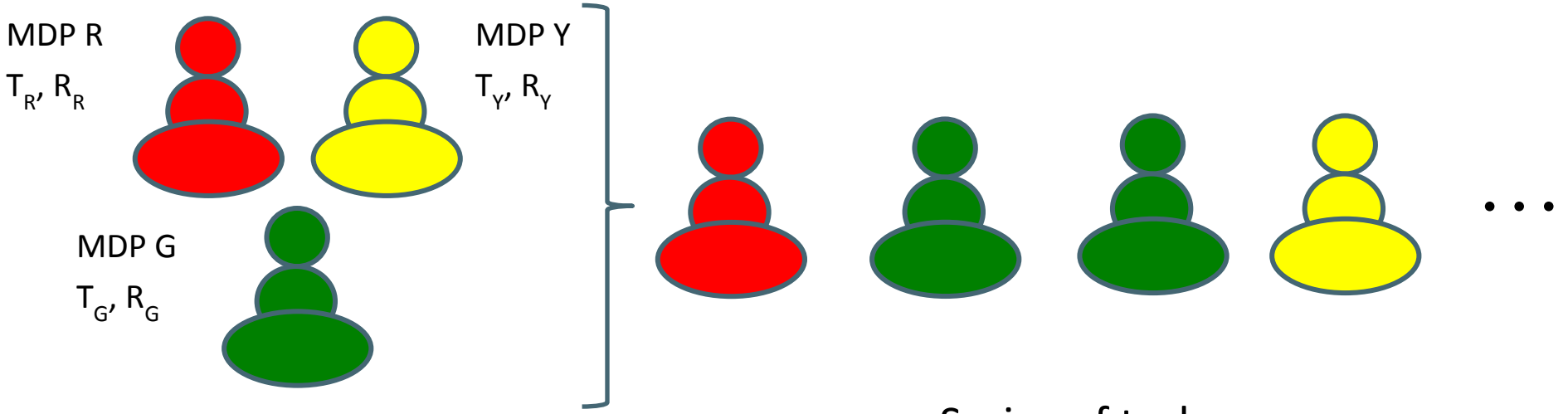


Multi-Task Reinforcement Learning with a Finite Set of Models



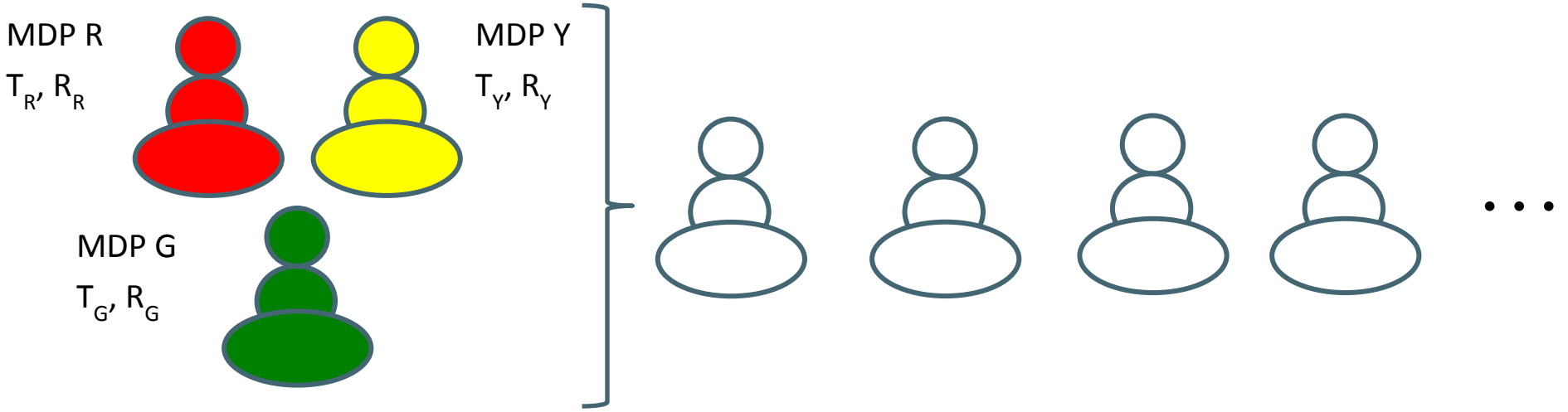
Series of tasks
Act in each task for H steps

If Knew Identity of Each Task and the MDP Parameters, No Learning Needed!

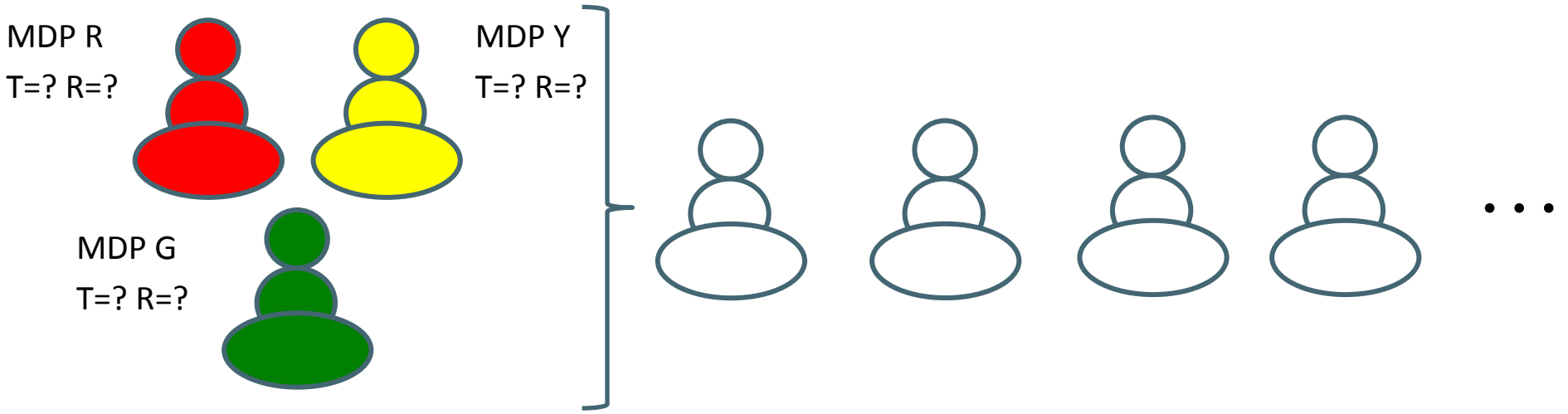


Series of tasks
Act in each task for H steps

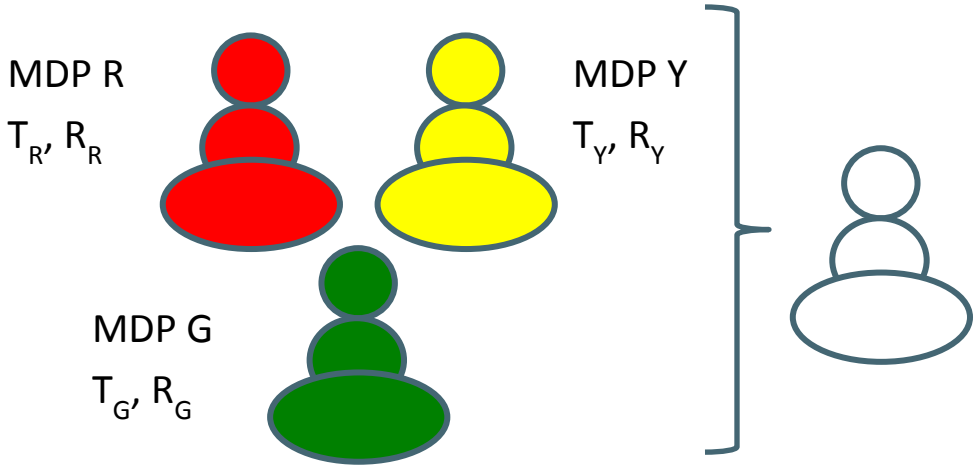
But Don’t Know the Identity of Each Task (which MDP it is)



And Don’t Know the MDP Model Parameters

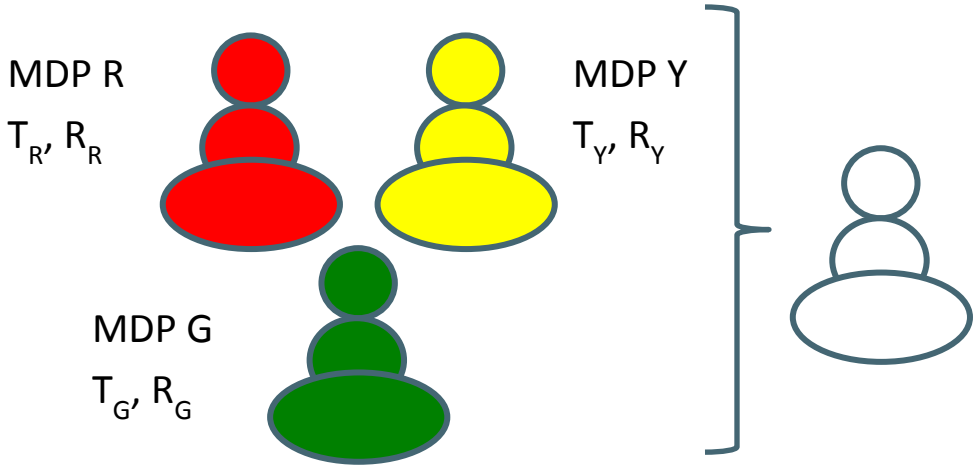


Two Key Questions

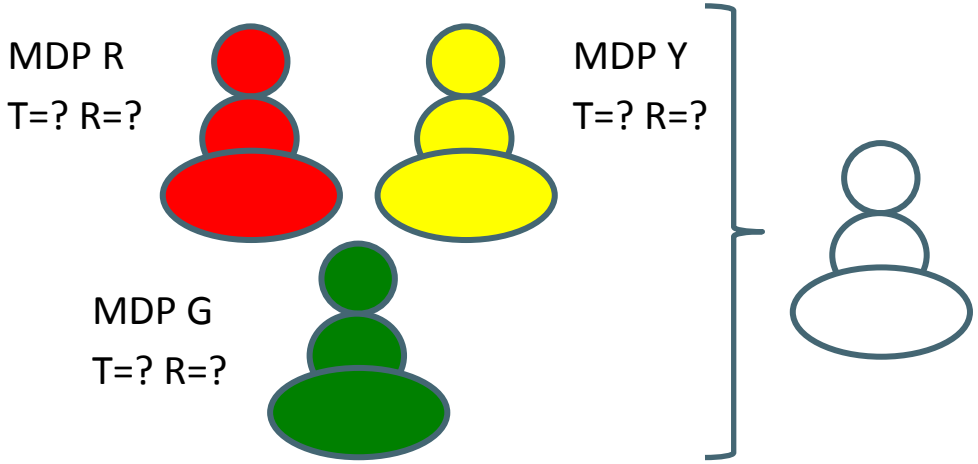


1) Is learning faster than single-task RL algorithms if know models, but not identity of current task?

Two Key Questions



1) Is learning faster than single-task RL algorithms if know models, but not identity of current task?

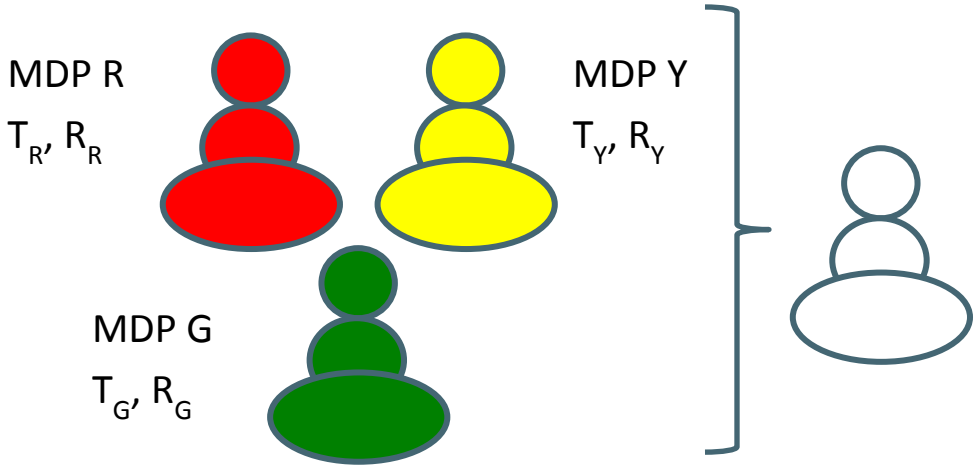


2) If yes, can we achieve similar results even if start off not knowing the MDP model parameters?

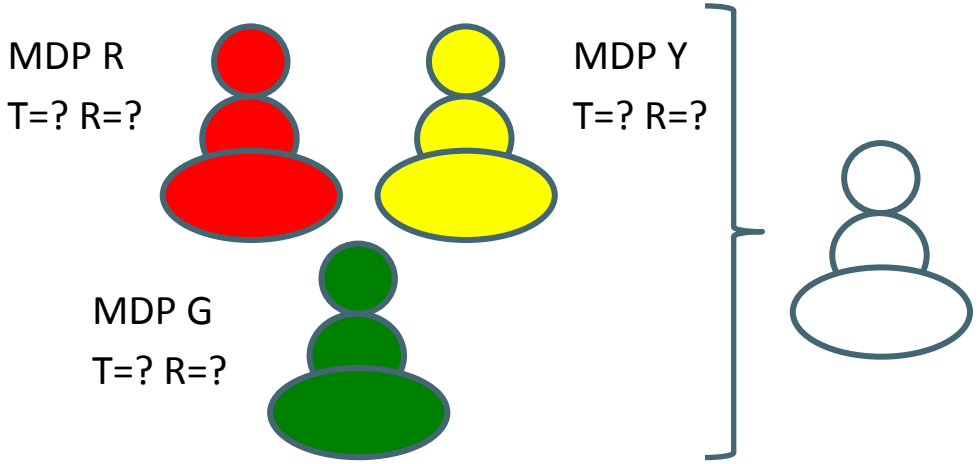
Idea: Model Identification Can be Easier than Learning Policy From Scratch

- Assume know task is 1 of *M* MDPs, where parameters of each MDP is specified
- Identify current task
  - Track which of M models are most likely given observed tuples (state, action, reward, next state)
- Proved sample complexity reduction from $|S||A| \rightarrow M$ dependence

Two Key Questions

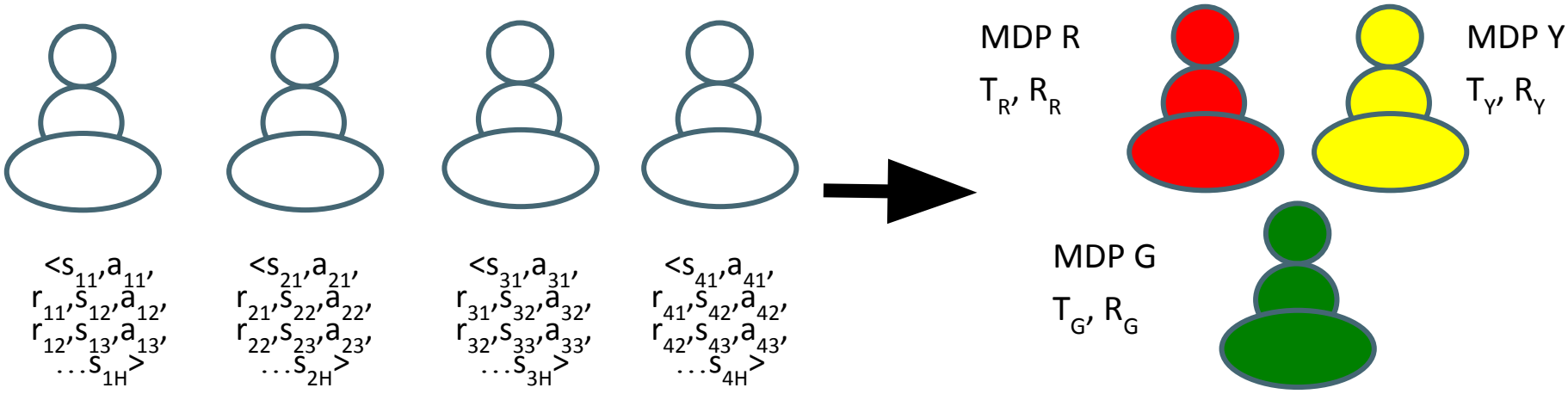


1) Is learning faster than single-task RL algorithms if know models, but not identity of current task?

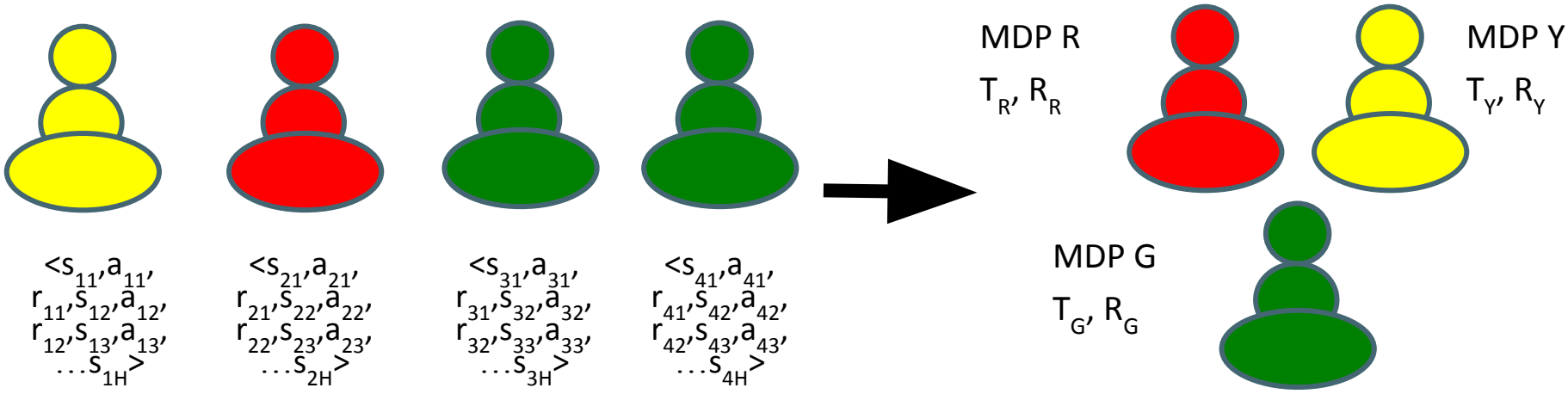


2) If yes, can we achieve similar results even if start off not knowing the MDP model parameters?

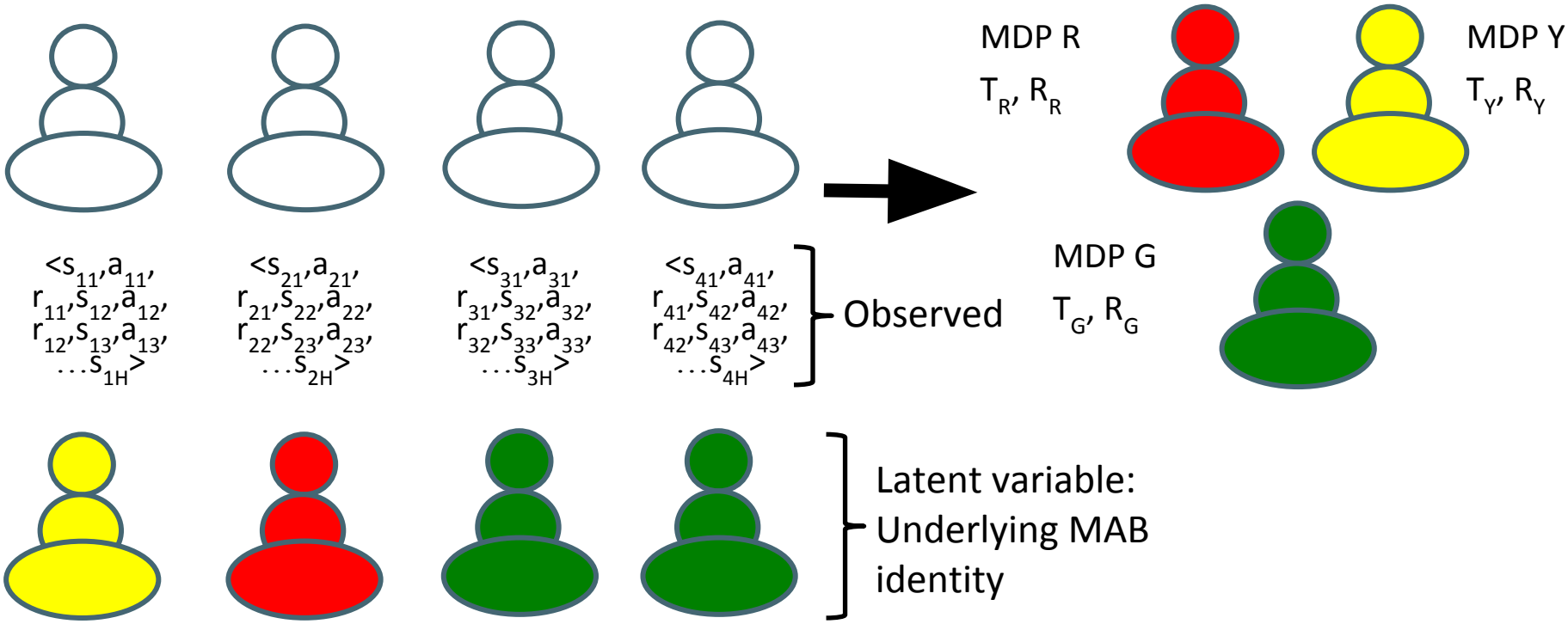
Learning Set of MDPs' Models



Learning Set of MABs' Models



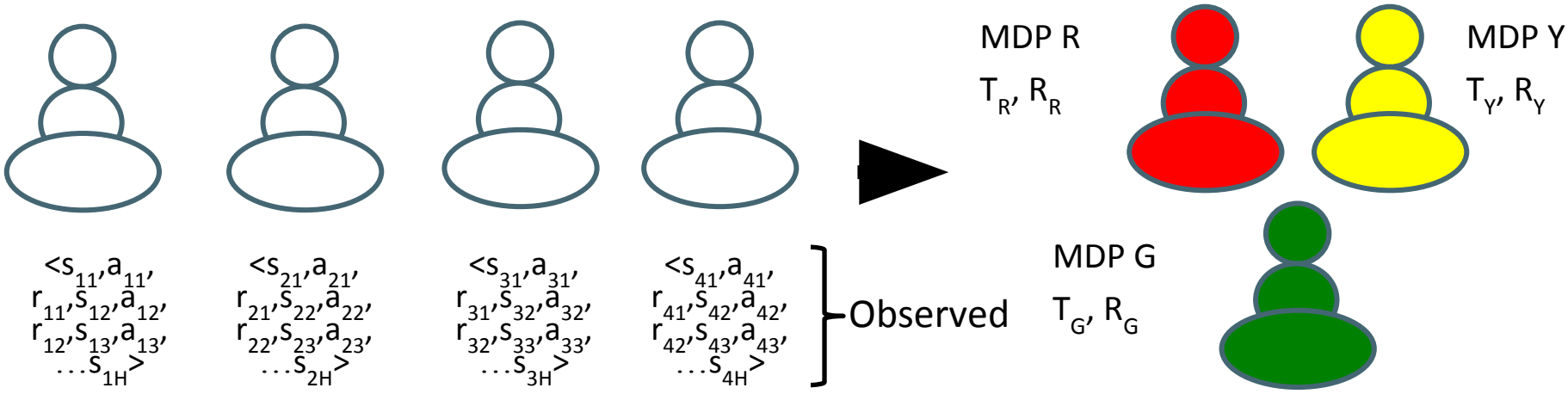
Identity of Each Task Unknown to the Learner: Latent Variable Estimation



Latent Variable Estimation

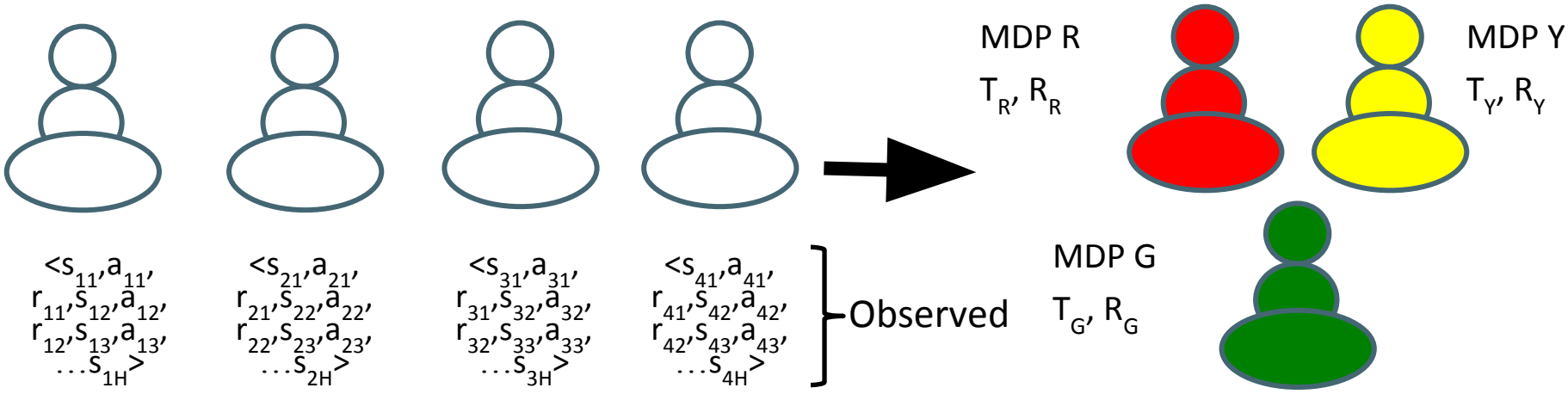
- Generally hard
- Standard techniques like expectation maximization have no finite sample guarantees & can converge to local optima

Assume Have (Sufficiently) Long Horizon Per Task



- If know all of the MDP parameters of each task perfectly, cluster tasks with identical parameters
- Know different underlying MDPs have to differ in their model parameters in at least one (s,a) pair by some difference d, so use to inform how well need to estimate parameters

Assume Have (Sufficiently) Long Horizon Per Task*



* In other work on transfer across multi-armed bandits, we relax this assumption, and use a Methods of Moments approach to derive & use finite sample bounds on latent variable estimation quality (see NIPS 2013)

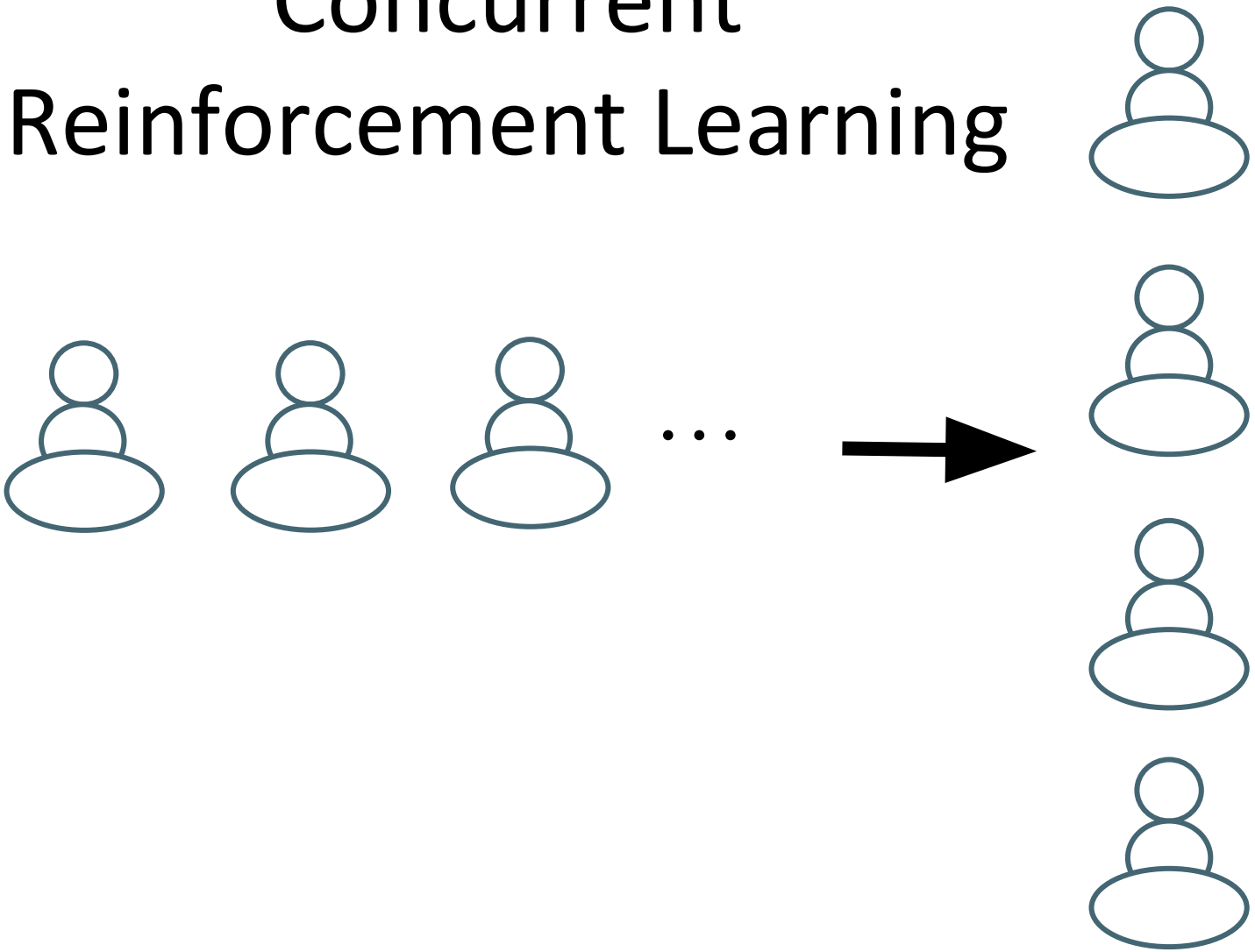
Sequential Multi-Task PAC RL in a Finite Set of Models

- For tasks i=1:T_1
  - Use single-task PAC RL algorithm E^3 in task i
- Cluster tasks into set of C MDPs
- For all further tasks i
  - Do model identification on i given C models

Key Result

- Can significantly speed learning if # MDPs << |S||A|
- **First formal result, to our knowledge, that sequential multi-task reinforcement learning can enable faster learning** (reduced sample complexity)
- **No negative transfer** in terms of sample complexity!
  - Negative transfer is when transfer can lead to worse results than if did non-transfer setting
  - Here have no such issues, at least in terms of theoretical analysis

Concurrent Reinforcement Learning



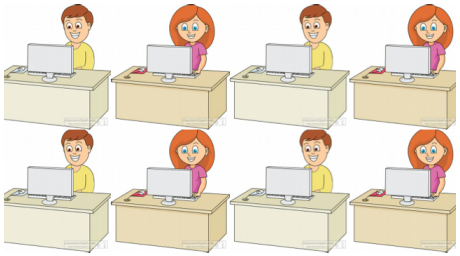
Guo and Brunskill, AAAI 2015

Class of Students

Or all customers using Amazon, or patients in a hospital, …

Concurrent but Independent

One agent doing many tasks at once

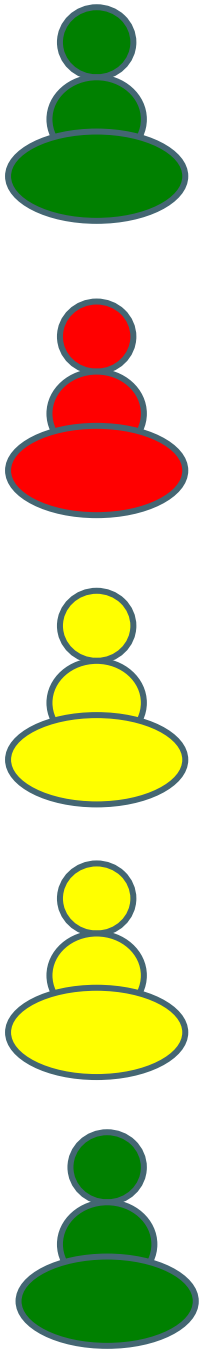


Not multiple agents doing a single task

- Very little prior work on concurrent RL, except encouraging empirical paper that might be very useful for customers (Silver et al. 2013)

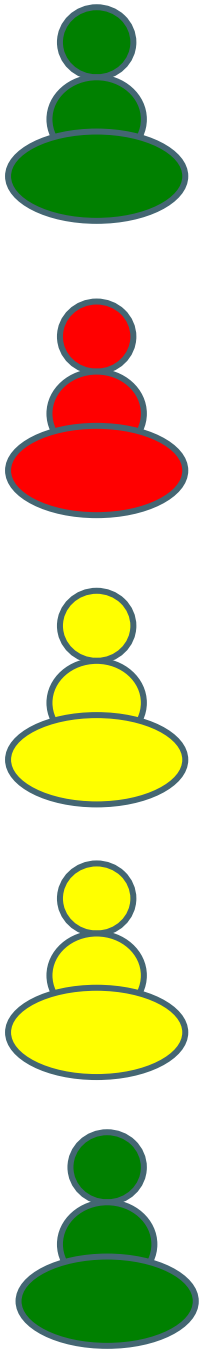
Concurrent RL

- Best could hope for: linear improvement
- Result is quite close to this!



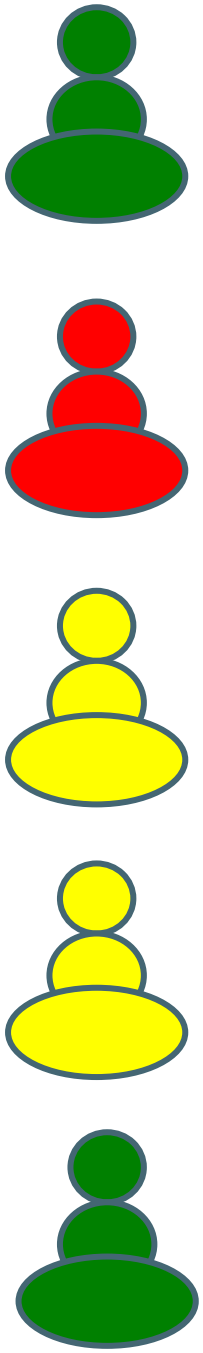
Concurrent RL in a Finite Set of MDPs: Algorithm

- For t=1:T
  - Explore state-action space in each MDP
- Cluster tasks
- Run concurrent MBIE
- Similar to sequential task, but now doing clustering and sharing while acting as act in a single task



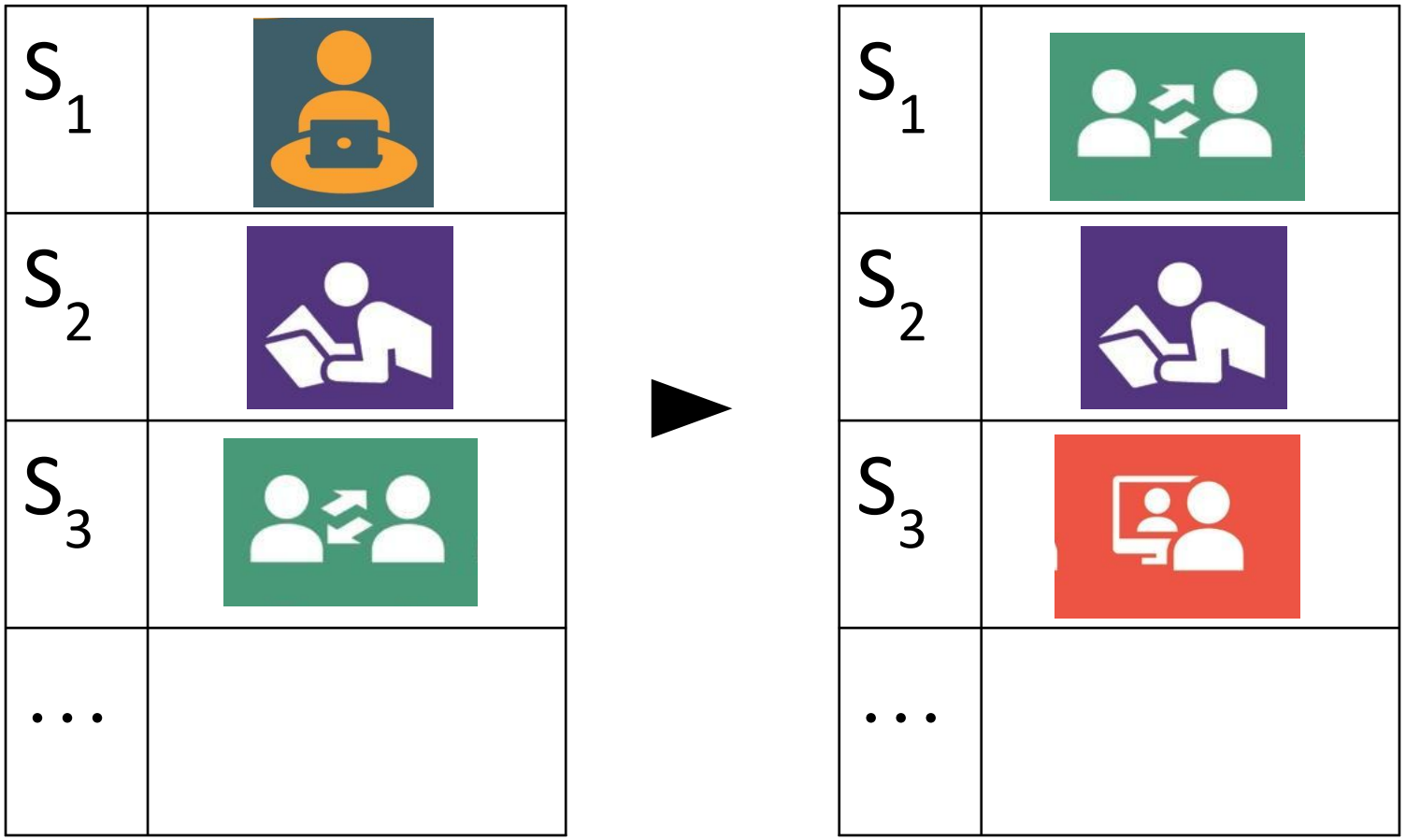
Concurrent RL in a Finite Set of MDPs: Intuition

- If time to cluster is small relative to experience needed to learn a good policy
- Then get approximately linear speedup (in terms of sample complexity) over not sharing information



Building on Offline Data

Off Policy Reinforcement Learning



Data gathered using previous policy (could be stochastic policy or multiple policies)

Want to output an optimal or good policy for future use

INTRODUCTION TO

Artificial Intelligence

Course | Discussion | Progress | Information | Prof

The instructors

Sebastian Thrun

Sebastian Th
a Research
Professor of
Computer Sc
at Stanford
University, a Google Fellow, a memb
the National Academy of Engineerin
the German Academy of Sciences. T
is best known for his research in rob
and machine learning.

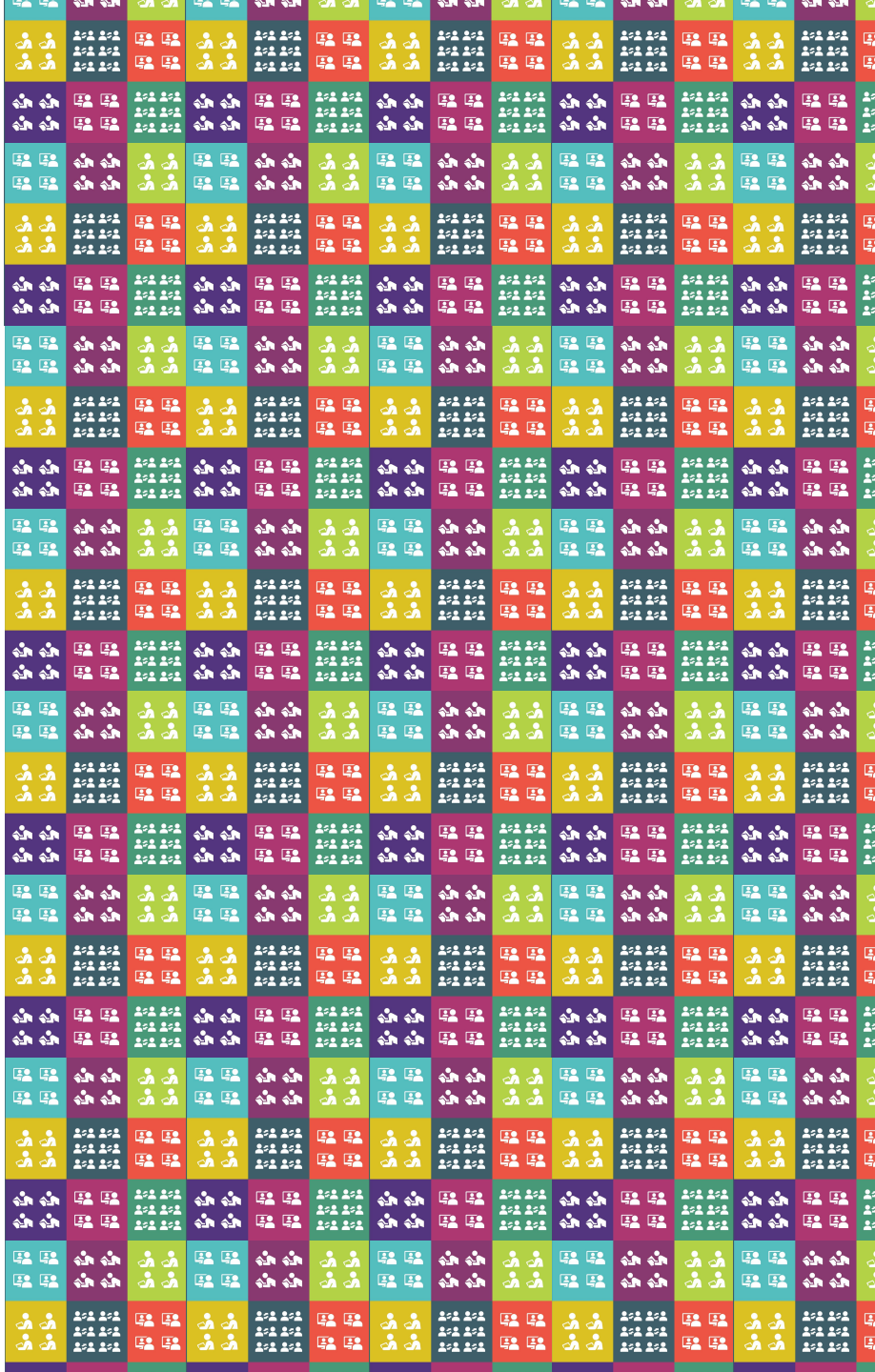
Peter Nor

Peter Norvig
Director of
Research at
Google Inc. H
also a Fellow
American

Increasing Amount of Decision Data, Increasing Opportunity for Better New Policies

- Electronic medical record systems
- Massive open online classes, tutoring systems
- Consumer marketing
- Home energy monitoring

Want Good Estimates of Generalization Performance




Mandel, Liu, Brunskill & Popović, AAMAS 2014
Level 1:8
Fork
Challenge:
Use old data to figure out good policies to deploy
MENU
OPTIONS

State Representation

Asked for a hint after 20s
Got correct after 40s

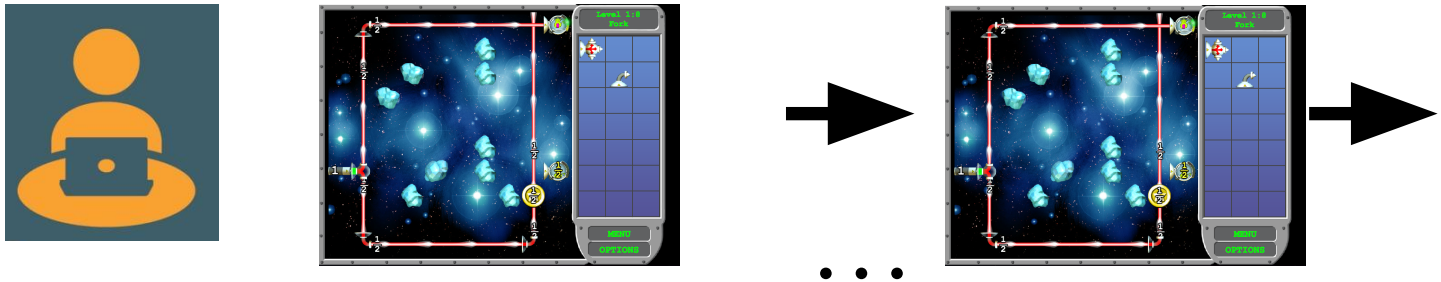
• • •

Got wrong after 15s
Got correct after 16s

- Vector of feature values
  - <NumberHintsRequested=1, ProblemsSinceHintRequested=1, TotalElapsedTime=56s, NumberTimesGotCorrectWithoutHint=1, TotalNumberOfProblemsDone=2, …>

Mandel, Liu, Brunskill & Popović, AAMAS 2014

State Representation



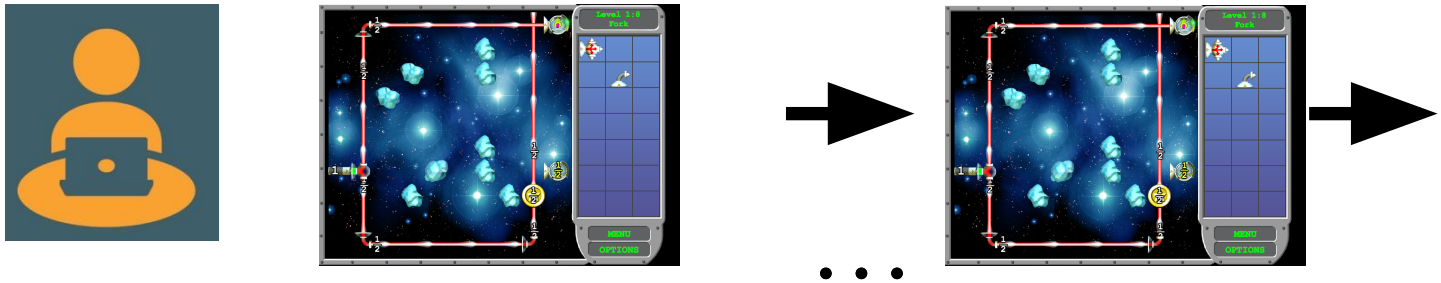
Asked for a hint after 20s
Got correct after 40s

Got wrong after 15s
Got correct after 16s

- Vector of feature values
  - <NumberHintsRequested=1, ProblemsSinceHintRequested=1, TotalElapsedTime=56s, NumberTimesGotCorrectWithoutHint=1, TotalNumberOfProblemsDone=2, …>
- Probability distribution over latent state
  - Prob(GraphicalComparison & SymbolicComparison) = 0.27, Prob(GraphicalComparison & Not SymbolicComparison) = .63, Prob(Not GraphicalComparison & SymbolicComparison) = 0.03, Prob(not GraphicalComparison & Not SymbolicComparison) = .07

Mandel, Liu, Brunskill & Popović, AAMAS 2014

State Representation



Asked for a hint after 20s
Got correct after 40s

. . .

Got wrong after 15s
Got correct after 16s

- Vector of feature values
  - <NumberHintsRequested=1, ProblemsSinceHintRequested=1, TotalElapsedTime=56s, NumberTimesGotCorrectWithoutHint=1, TotalNumberOfProblemsDone=2, …>
- Probability distribution over latent state
  - Prob(GraphicalComparison & SymbolicComparison) = 0.27, Prob(GraphicalComparison & Not SymbolicComparison) = .63, Prob(Not GraphicalComparison & SymbolicComparison) = 0.03, Prob(not GraphicalComparison & Not SymbolicComparison) = .07
- Prediction over responses to future activities
  - Prob get next graphical activity right =0.9, Prob get next improper fraction activity right =0.4
  - …

Mandel, Liu, Brunskill & Popović, AAMAS 2014

General Formulation: Unbiased Offline Policy Evaluation Across Representations for Short Horizons

Representation & Algorithm to Compute Policy

$activity_{11}$, $observation_{12}$, $reward_{12}$, $activity_{12}$, $observation_{13}$, ...
$activity_{21}$, $observation_{22}$, $reward_{22}$, $activity_{22}$, $observation_{23}$,...
...

Unbiased Estimate of Policy Performance

Mandel, Liu, Brunskill & Popović, AAMAS 2014

Guarantees

- Unbiased estimate of expected future performance of input representation—policies
  - Importance sampling to compare policies generated from variety of representations
  - Cross validation to predict generalization

Mandel, Liu, Brunskill & Popović, AAMAS 2014

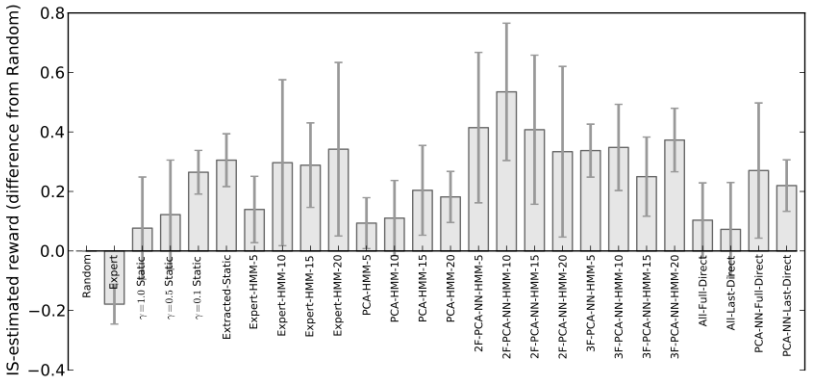
Deployment: Refraction Game

- Find adaptive policy for concept to give to a player to maximize total number of concepts complete before quit

Deployment: Refraction Game Offline Data

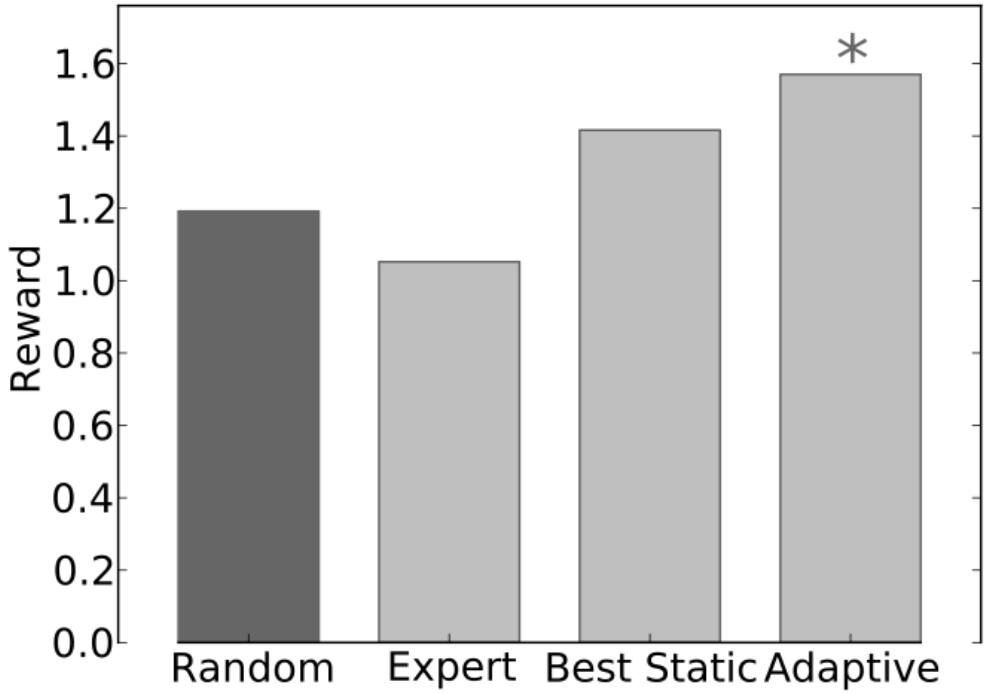
- 180 features of game per level
- Collected 11,000 players' data using random level ordering

Used Offline Evaluation Method to Compare Many Representation-Policies



- Best policy in offline evaluation is adaptive policy using PCA+neural network representation
- Previously used expert ordering is estimated as being worse than random!
- Highlights non-trivial nature of designing good policies

… and Best Scoring Offline Policy Improved Concept Completion by 32%

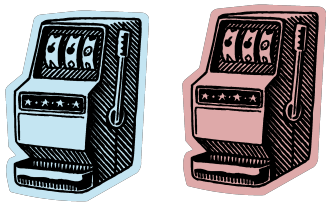


- Tried 4 policies with 2000 new learners
- Compared to random & expert

Towards Faster Learning

- Transfer learning
- **Building on offline data**
- **Leveraging expert input**

Multi-Armed Bandits

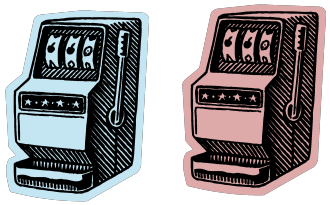


Evaluate Algorithms

Utilize Prior Data

Mandel, Liu, Brunskill & Popović, AAAI 2015

Multi-Armed Bandits



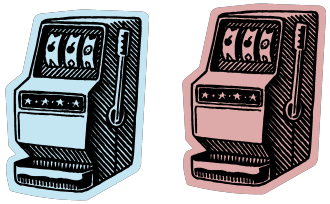
Evaluate Algorithms

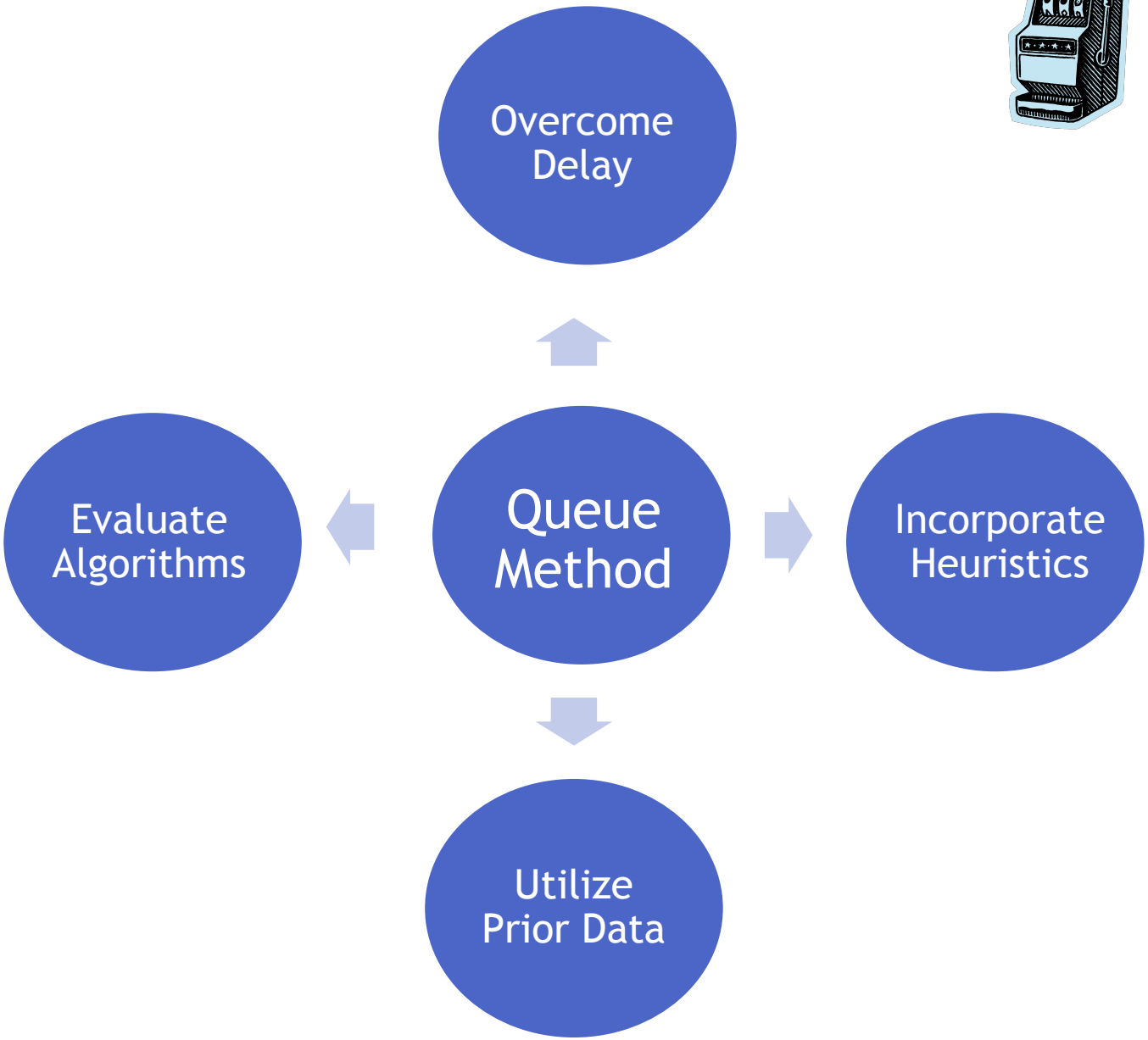
Incorporate Heuristics

Utilize Prior Data

Mandel, Liu, Brunskill & Popović, AAAI 2015

Multi-Armed Bandits





Mandel, Liu, Brunskill & Popović, AAAI 2015

Queue Method

- Store data (outcomes of pulling arms) in queues, one per arm
- Update a base algorithm (often which has formal performance guarantees) only using queues
  - Formal bounds
- Sampling distribution

Leveraging Heuristics While Maintaining Guarantees

- Given
  - Base algorithm (w/formal performance guarantees)
  - Heuristic algorithm
- Store data (outcomes of pulling arms) in queues, one per arm
- Update base algorithm only using queue data
- Sampling distribution: mixes heuristic and base but bounds lengths of queues

Evaluate Algorithms Using Prior Data

- Store prior data (outcomes of pulling arms) in queues, one per arm
- Draw outcome from queue according to action requested by algorithm to evaluate
- Update algorithm given queue outcome
- Halt when reach an empty queue

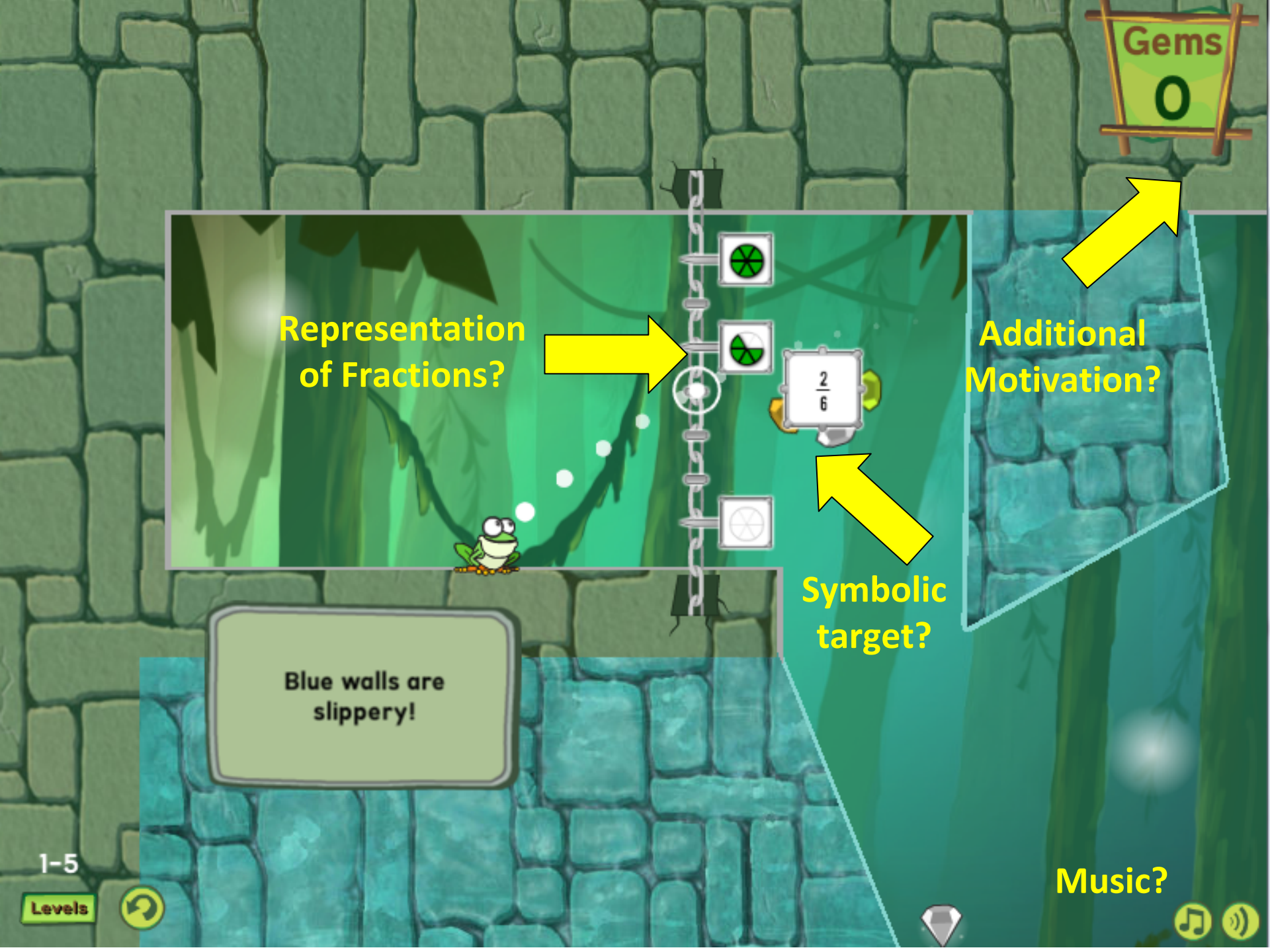
Evaluate Algorithms Using Prior Data

- Store prior data (outcomes of pulling arms) in queues, one per arm
- Draw outcome from queue according to action requested by algorithm to evaluate
- Update algorithm given queue outcome
- Halt when reach an empty queue
- More efficient than rejection sampling
- Conditionally unbiased*

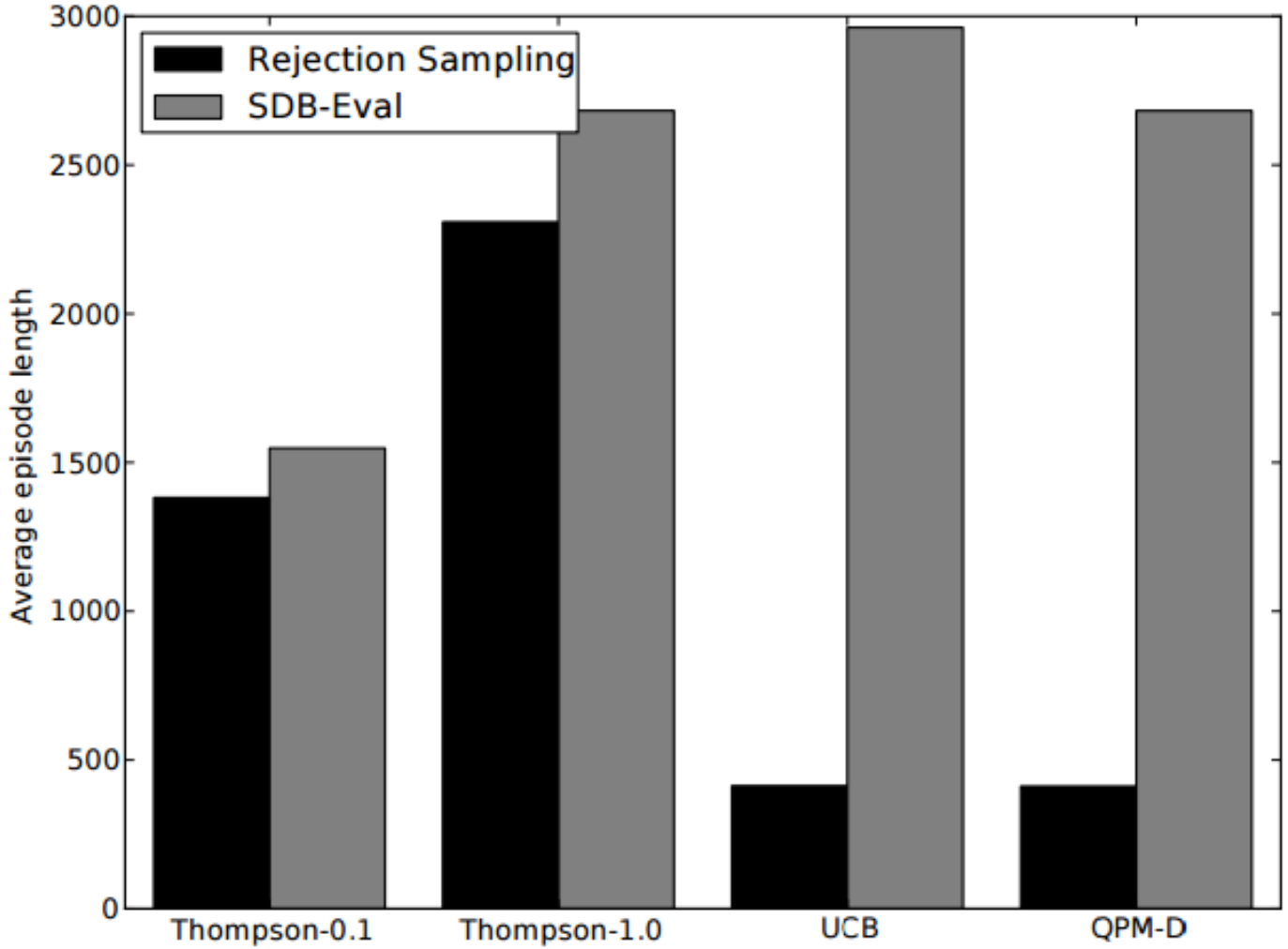
Regret Bounds

Theorem 1. *For algorithm 1 with any choice of procedure* GETSAMPLINGDIST *and any online bandit algorithm* BASE, $\mathbb{E}[R_T] \leq \mathbb{E}\left[R_T^{\text{BASE}}\right] + \sum_{i=1}^{N} \Delta_i \, \mathbb{E}[S_{i,T}]$ *where* $S_{i,T}$ *is the number of elements pulled for arm* i *by time* T*, but not yet shown to* BASE.

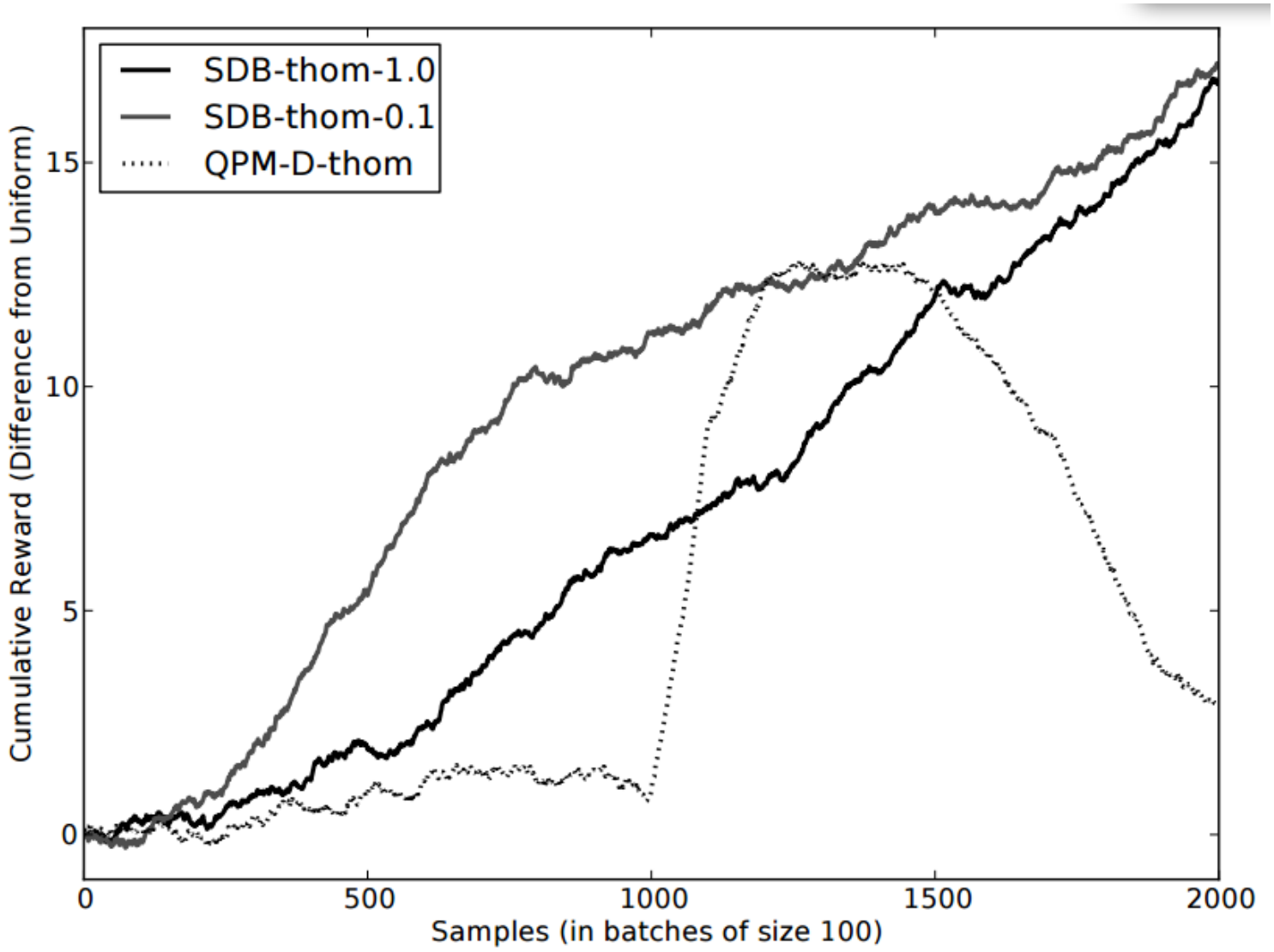

Gems
0
2
6
Blue walls are slippery!
1-5
Levels


Gems
0
Representation of Fractions?
2/6
Additional Motivation?
Symbolic target?
Blue walls are slippery!
1-5
Levels
Music?

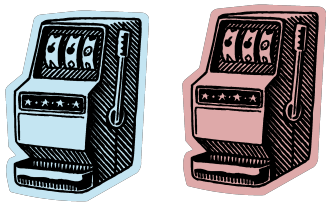
Offline Data Efficiency

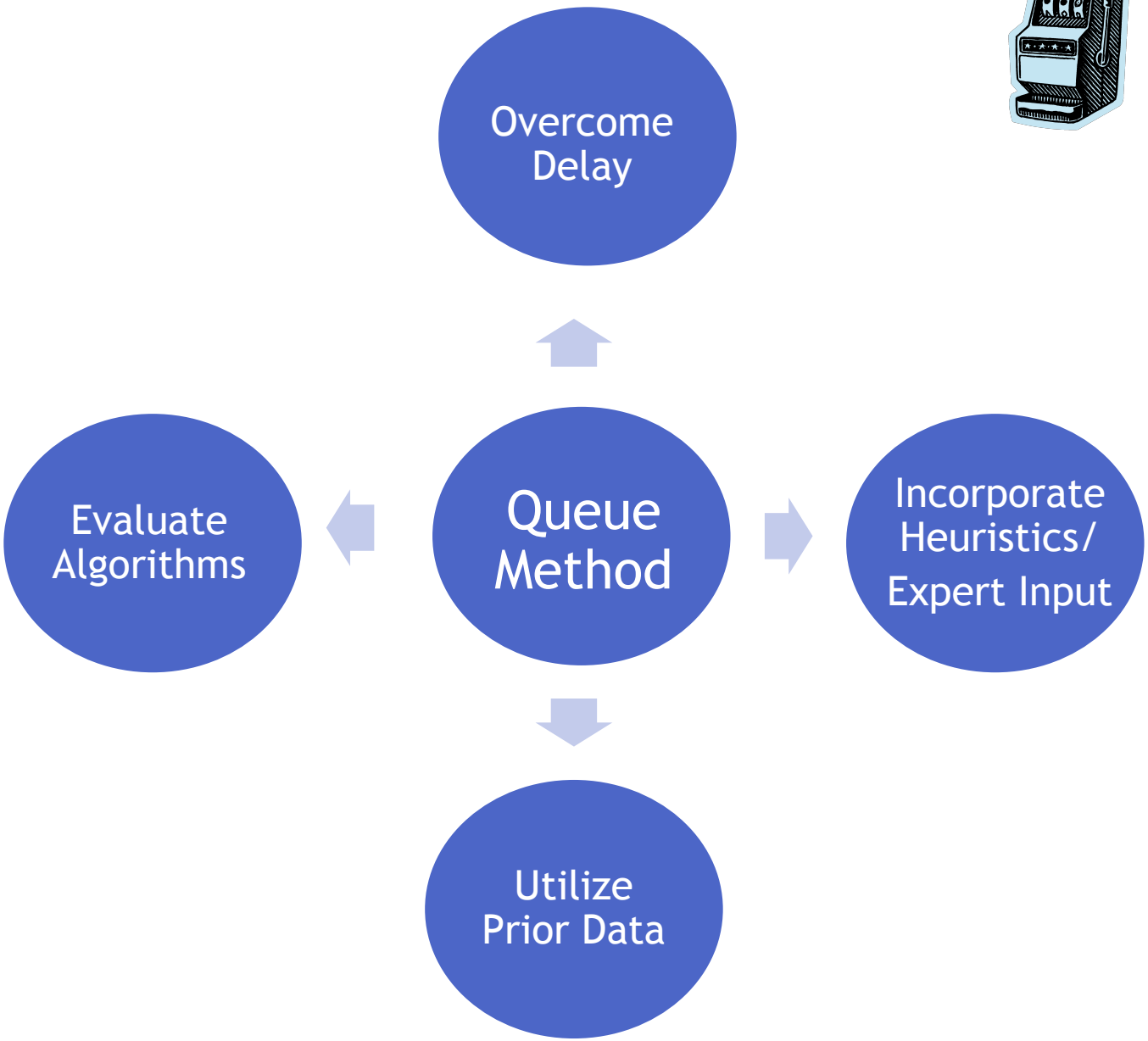


Algorithm Performance Evaluated Using Offline Data



Multi-Armed Bandits





Mandel, Liu, Brunskill & Popović, AAAI 2015

Towards Faster Learning

- Transfer learning
- Building on offline data
- **Leveraging expert input**

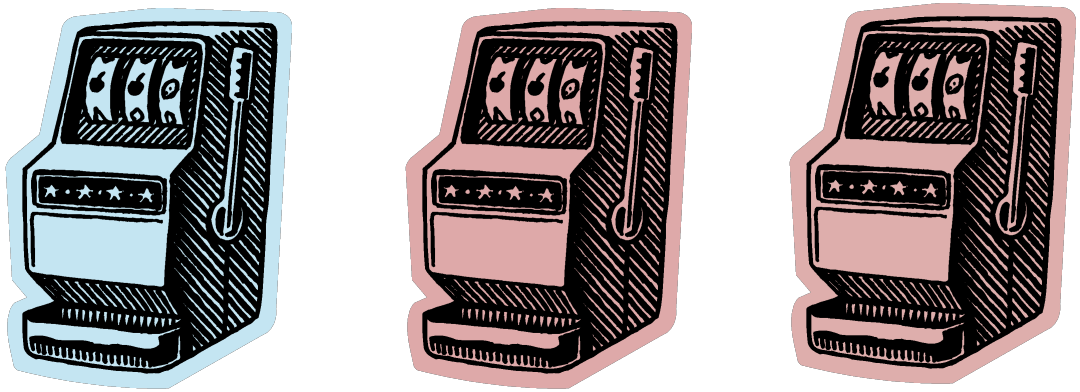
Faster Learning With Policy Advice

- Online reinforcement learning
- Computational and speed of learning (sample complexity, regret) tend to scale with size of state/action space
- Policy advice: given finite set of policies
- Objective: perform as well as best policy

Azar, Lazaric, Brunskill, ECML 2013

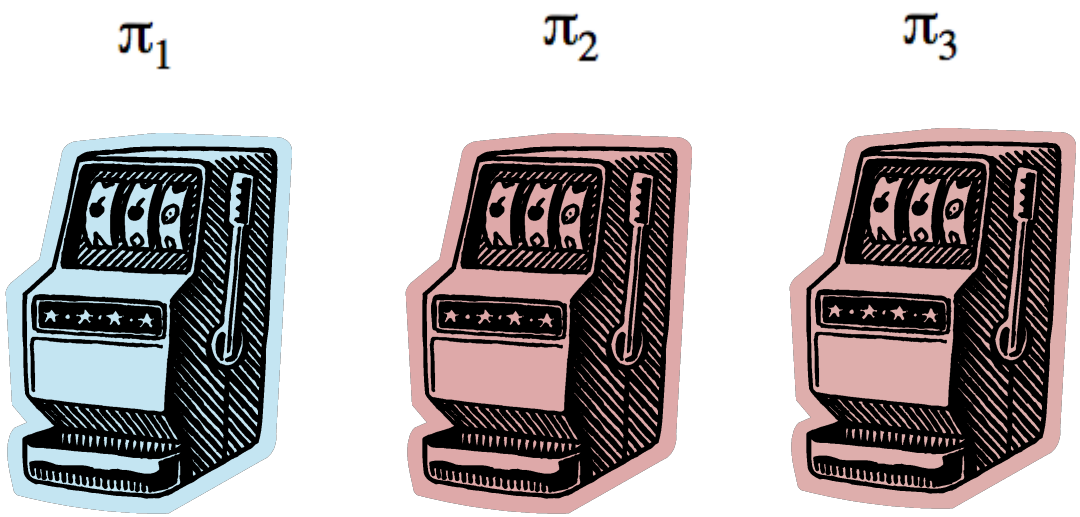
RL with Policy Advice (RLPA)

π_1 π_2 π_3



Azar, Lazaric, Brunskill, ECML 2013

RL with Policy Advice (RLPA)



- Keep upper bound on avg. reward per policy
- Use to optimistically select policy

Azar, Lazaric, Brunskill, ECML 2013

RLPA Regret Bounds

For any $T \geq T^+ = f^{-1}(H^+)$ *the regret of RLPA is bounded as*

$$\Delta(s) \leq f(T)\sqrt{Tm(\log(T/\delta))} + f(T)m\big(\log_2(T^+) + 2\log_2(T)\big)$$

$$\tilde{O}(\sqrt{mT}) \text{ (with } f(T) = \log(T),\ T^+ = \exp(H^+))$$

- No dependence on size of state-action space
- Sqrt dependence on m, # of input policies
- Dependent only on span H+ of best policy

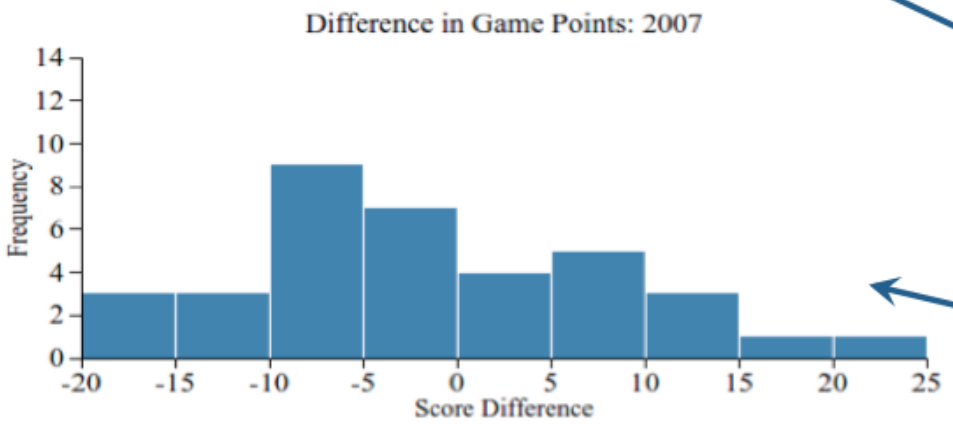
Azar, Lazaric, Brunskill, ECML 2013

Closing the Graveyard of Ambitions

Closing the Graveyard of Ambitions

	Celtics	Nets	Knicks	76ers	Raptors	Bulls	Cavaliers	Pistons
Rockets	9	5	8	9	14	2	0	-3
Magic	17	11	-6	-1	-5	7	-11	-17
Lakers	-17	-12	-1	8	-12	22	8	2
Blazers	-4	-3	-18	-1	8	13	9	1

[-20, -16]	3
[-15, -11]	3
[-10, -6]	9
[-5, -1]	7
[0, 4]	4
[5, 9]	5
[10, 14]	3
[15, 19]	1
[20, 24]	1



Randomly generated tables

Dynamic histograms

Adaptive Hints

Hint: Check your answer for the range [-10, -6], and [5, 9].

* Remember that we are counting the number of times that the score difference in a game falls within these ranges.

Sample Efficient Online RL

- State space size influences how quickly can learn a good policy
- What representation should we use?
  - One that enables us to represent good policy

Abstract from Demonstration

- Cobo et al. (2011, 2014)
- Leverage expert input
- Without being bounded by expert performance
- Identify which features used by experts
- Do RL on that feature space

Learning the Student Features Used to Decide How To Teach

done_lo3
pre_lo3
previous_right_lo3
done_lo4
pre_lo4
previous_right_lo4
done_lo5
pre_lo5
previous_right_lo5
done_lo6
pre_lo6
previous_right_lo6
done_my_lo
pre_my_lo
previous_right_my_lo
previous_time_before_last_problem_lo
previous_lo
previous_time_spent_last_problem_lo
time_before_last_problem_lo
attempt_last_problem_lo
time_spent_last_problem_lo
previous_attempt_last_problem_lo
pretest_time_spent
right_lo2
previous_left_lo2
previous_correctness
right_lo3
previous_left_lo3
previous_attempts
right_lo4
previous_left_lo4
previous_time_spent
right_lo5

current_lo
left_lo1
previous_done_lo1
correctness
left_lo2
previous_done_lo2
attempts
left_lo3
previous_done_lo3
time_spent
left_lo4
previous_done_lo4
time_spent_all
left_lo5
previous_done_lo5
right_first_try
left_lo6
previous_done_lo6
wrong_first_try
left_my_lo
previous_done_my_lo
pretest_score
right_lo1
previous_left_lo1
previous_left_lo5
previous_right_first_try
right_lo6
previous_left_lo6
previous_wrong_first_try
right_my_lo
previous_left_my_lo
done_lo1
pre_lo1

6 features of student learning process better than 70 at predicting teacher's decisions!

Lee, Runde, Jibril, Wang Brunskill, LAS 2015

Towards Faster RL

Warner Brothers