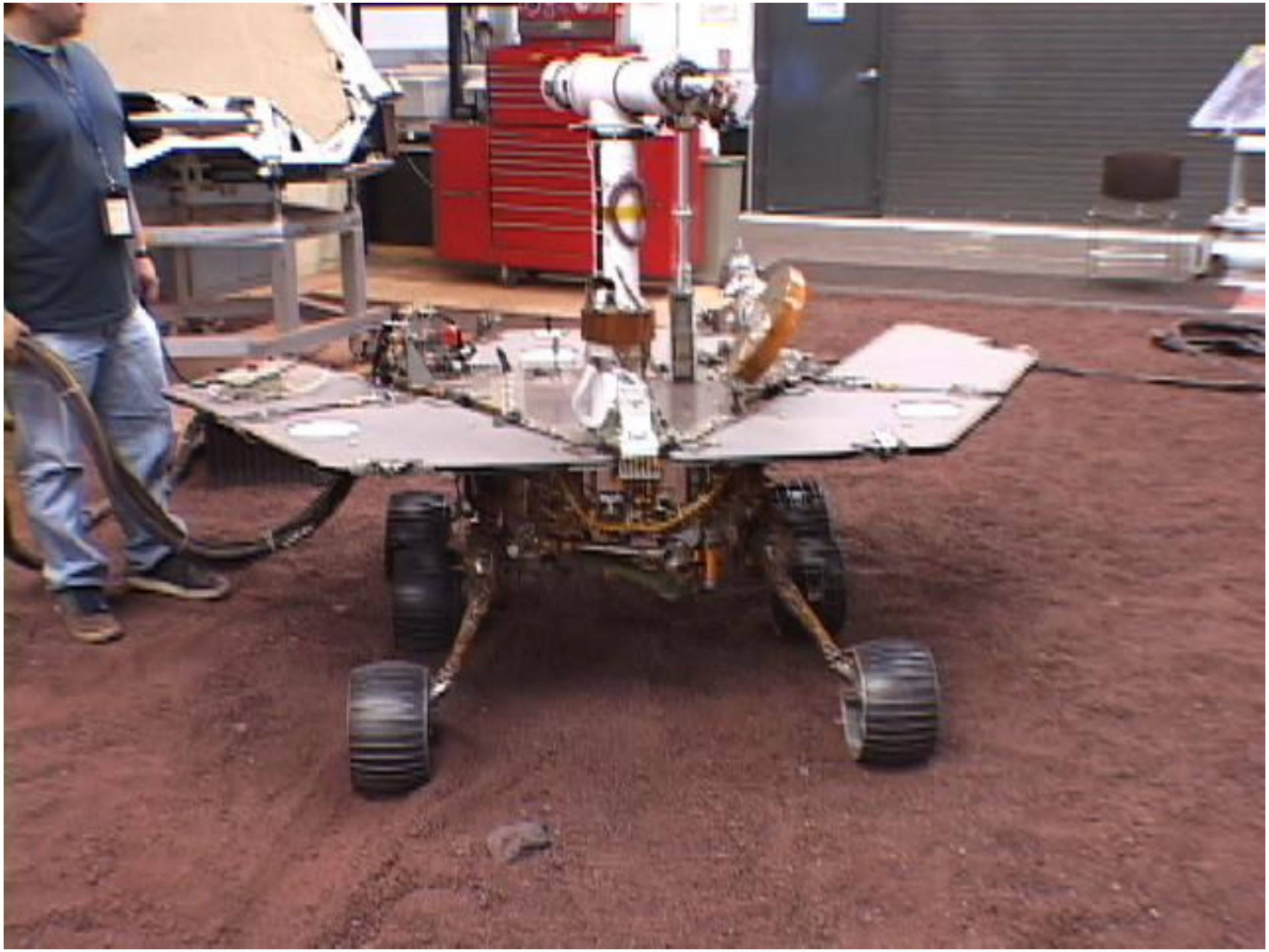
Efficient Learning of Dynamics Models Using Terrain Classification

Bethany Leffler **Chris Mansley** Michael Littman

Robotic Motivation of Navigation Task



Robotic Motivation of Navigation Task

Robotic Motivation of Navigation Task

Navigation

Traditional

- Dynamics of the agent are known or learned
- Planning is done with respect to the model

Model-Based RL

- Dynamics of the agent are learned
- Planning is done with respect to the model

Navigation

Traditional

- Dynamics of the agent are known or learned
- Planning is done with respect to the model
- Assumes a single dynamic or model for all states

Model-Based RL

- Dynamics of the agent are learned
- Planning is done with respect to the model

Navigation

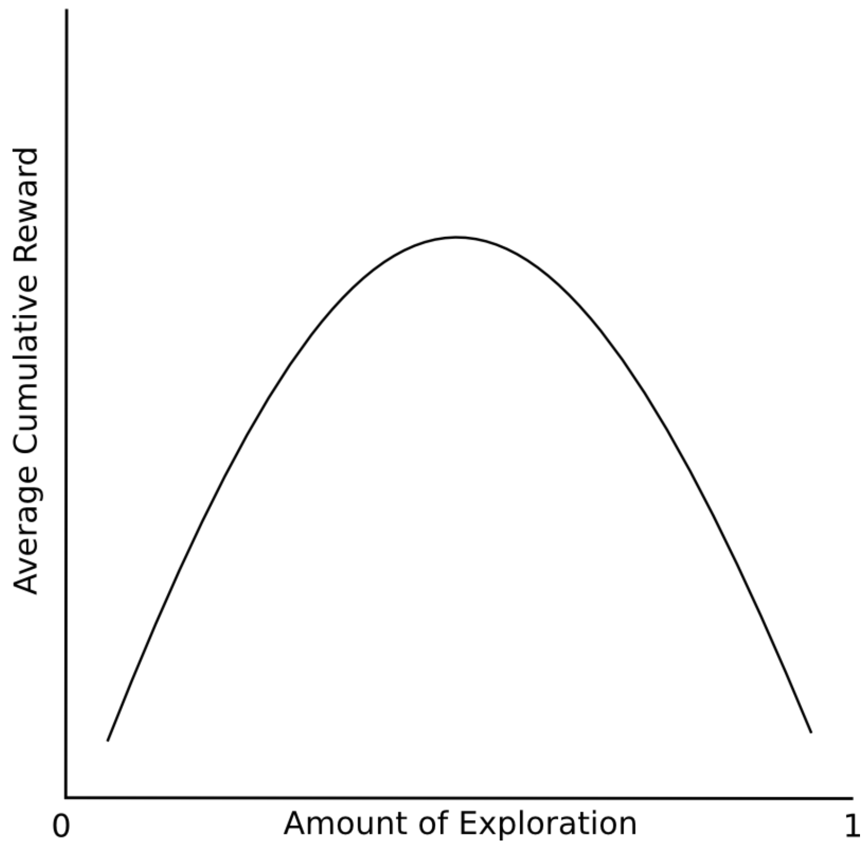
Traditional

- Dynamics of the agent are known or learned
- Planning is done with respect to the model
- Assumes a single dynamic or model for all states

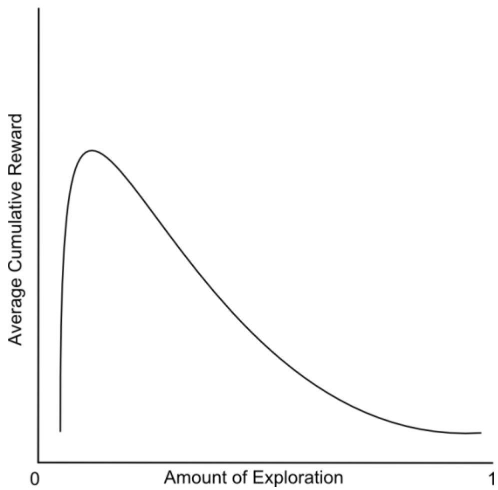
Model-Based RL

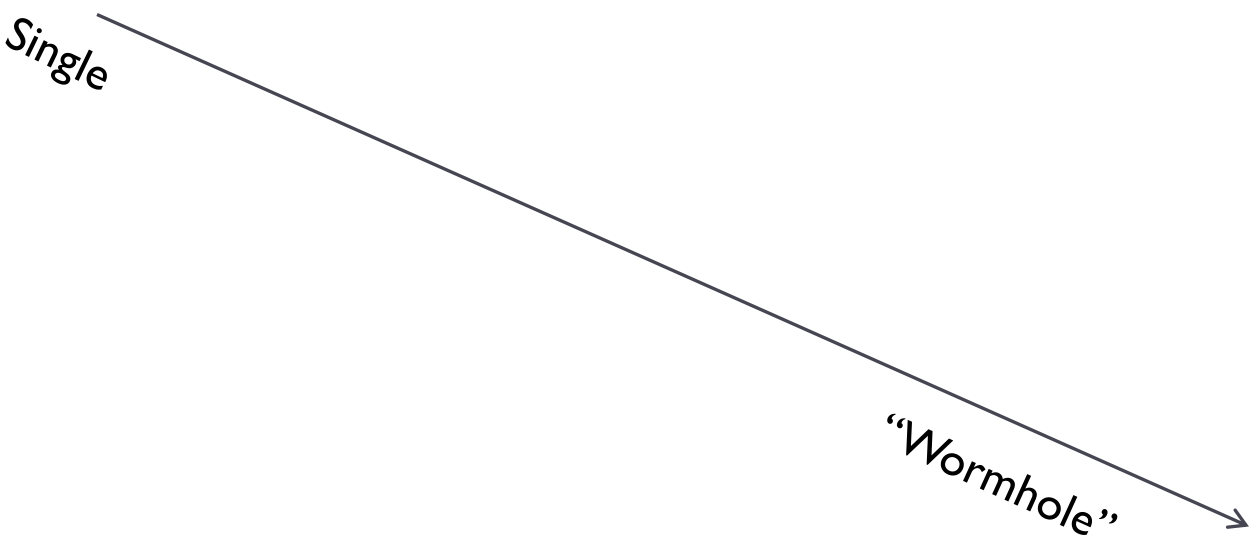
- Dynamics of the agent are learned
- Planning is done with respect to the model
- Assumes each state may have a different dynamics model

Exploration vs. Exploitation

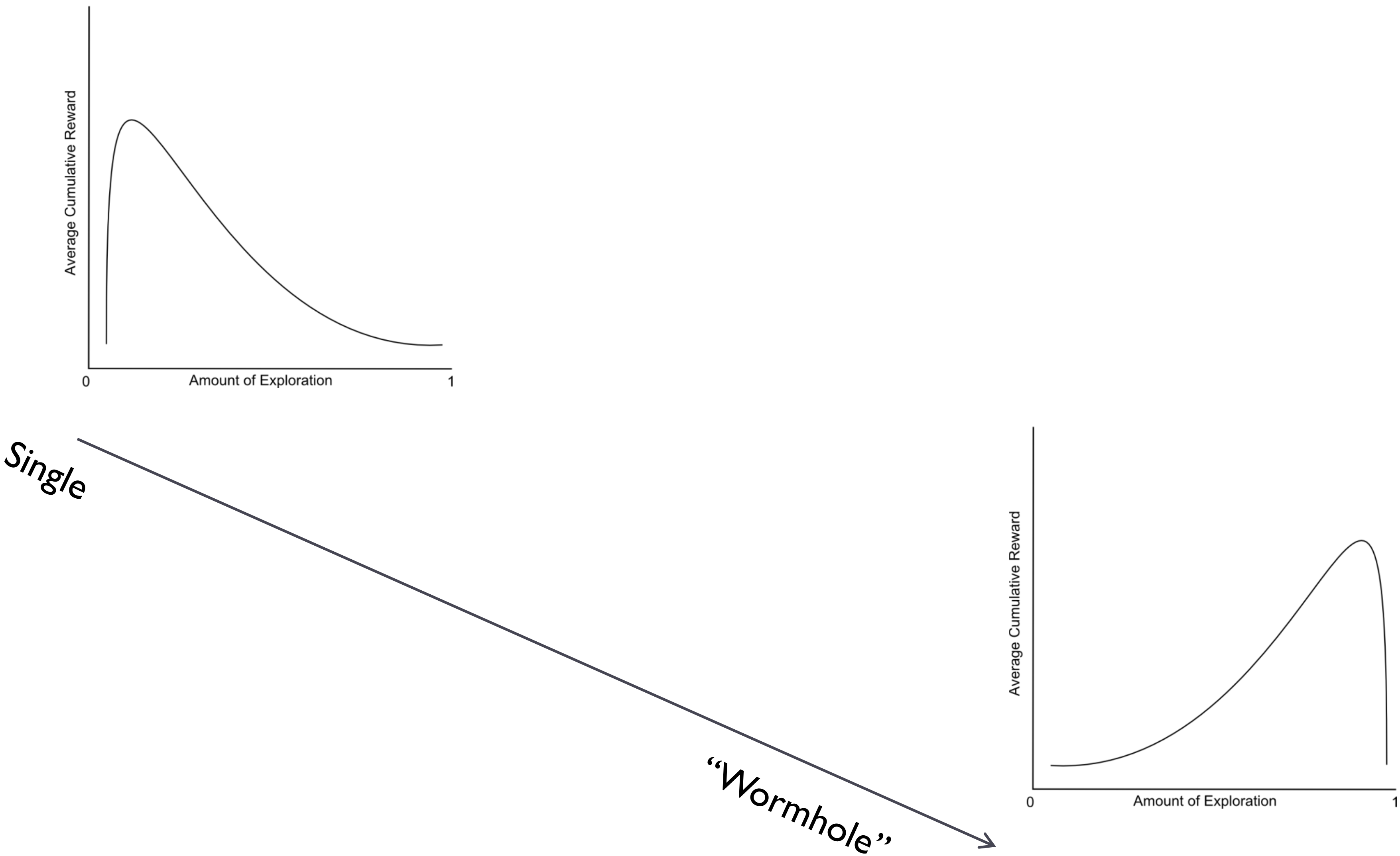


Environmental Model Matching

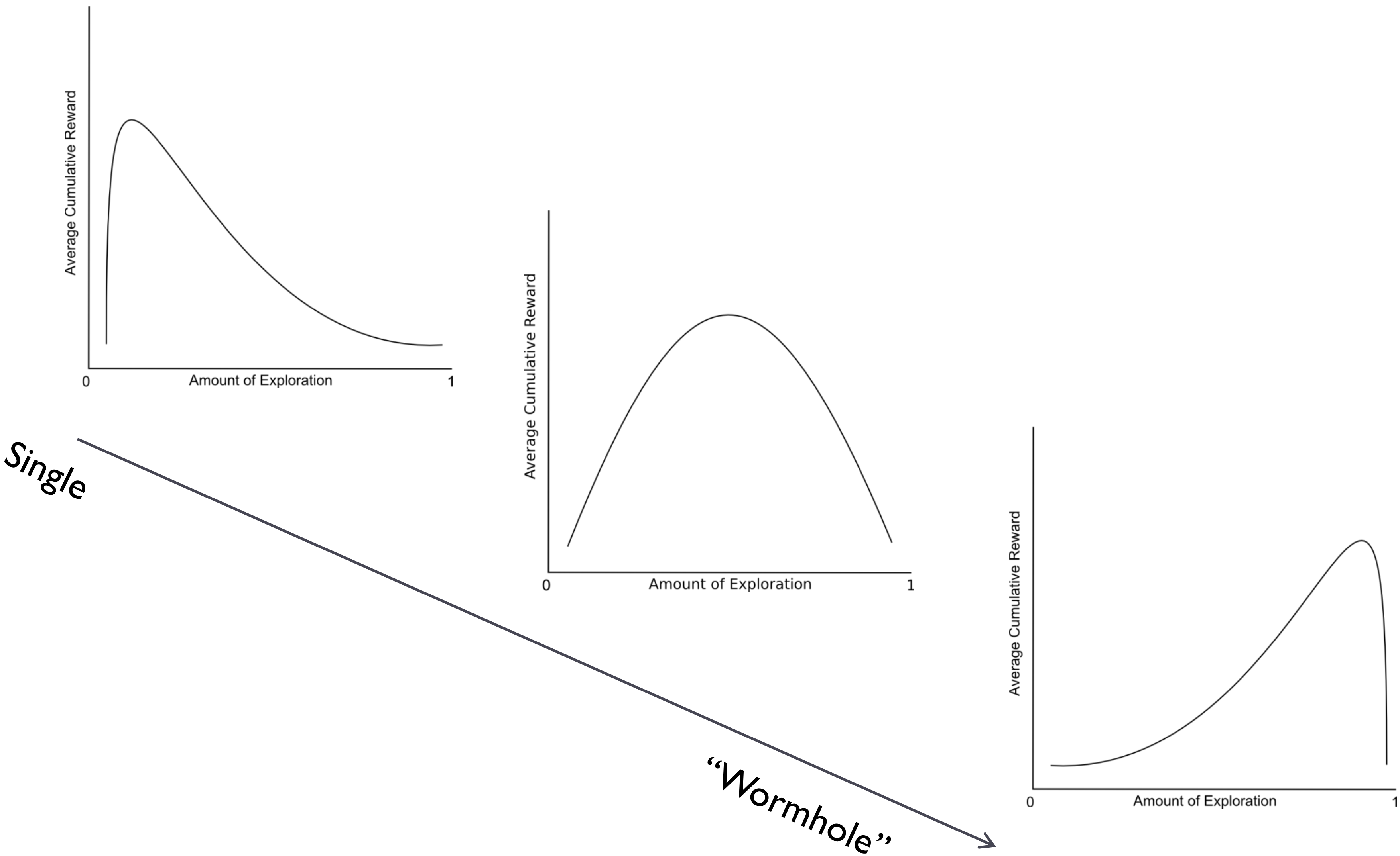




Environmental Model Matching



Environmental Model Matching



Our Algorithm

- In Leffler et al 2007, we defined such an algorithm
- This work extends that paper by
  - Empirically demonstrating the significance of adding a single extra model in this framework
  - Fully integrating autonomy into the system, removing the need for hand tuning
  - Comparing against other algorithms for generalization in RL
  - Enabling further extensions

Additional Assumptions

- Dynamics Indicator
  - There exists a function that indicates what area of the state space has similar dynamics
  - This function is often simply a single feature

Relocatable Action Model (RAM) – MDP

[Sherstov and Stone, 2005]

MDP

S – State

A – Action

$R: S \to \mathfrak{R}$ – Reward

$T: S \times A \to \Pr(S)$

– Transition Function

RAM-MDP

S – State

A – Action

$R: S \to \mathfrak{R}$ – Reward

$\kappa: S \to C$ – Cluster Function

$t: C \times A \to \Pr(O)$ – RAM

$\eta: S \times O \to S$ – Next-State Function

C – Cluster / Type

O – Outcome

Relocatable Action Model (RAM) – MDP

[Sherstov and Stone, 2005]

MDP

S – State

A – Action

$R: S \rightarrow \mathfrak{R}$ – Reward

$T: S \times A \rightarrow \Pr(S)$

– Transition Function

RAM-MDP

S – State

A – Action

$R: S \rightarrow \mathfrak{R}$ – Reward

$\kappa: S \rightarrow C$ – Cluster Function

$t: C \times A \rightarrow \Pr(O)$ – RAM

$\eta: S \times O \rightarrow S$ – Next-State Function

C – Cluster / Type

O – Outcome

Relocatable Action Model (RAM) – MDP
[Sherstov and Stone, 2005]

MDP

S – State

A – Action

$R: S \rightarrow \Re$ – Reward

$T: S \times A \rightarrow \Pr(S)$

– Transition Function

RAM-MDP

S – State

A – Action

$R: S \rightarrow \Re$ – Reward

$\kappa: S \rightarrow C$ – Cluster Function

$t: C \times A \rightarrow \Pr(O)$ – RAM

$\eta: S \times O \rightarrow S$ – Next-State Function

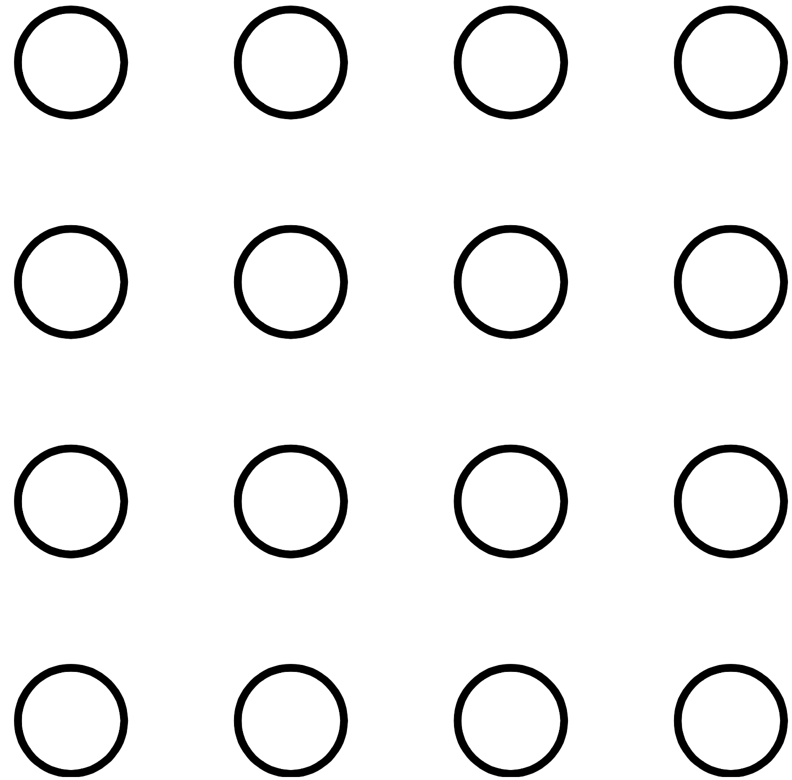
C – Cluster / Type

O – Outcome

RAM-Rmax

[Leffer et al., 2007]

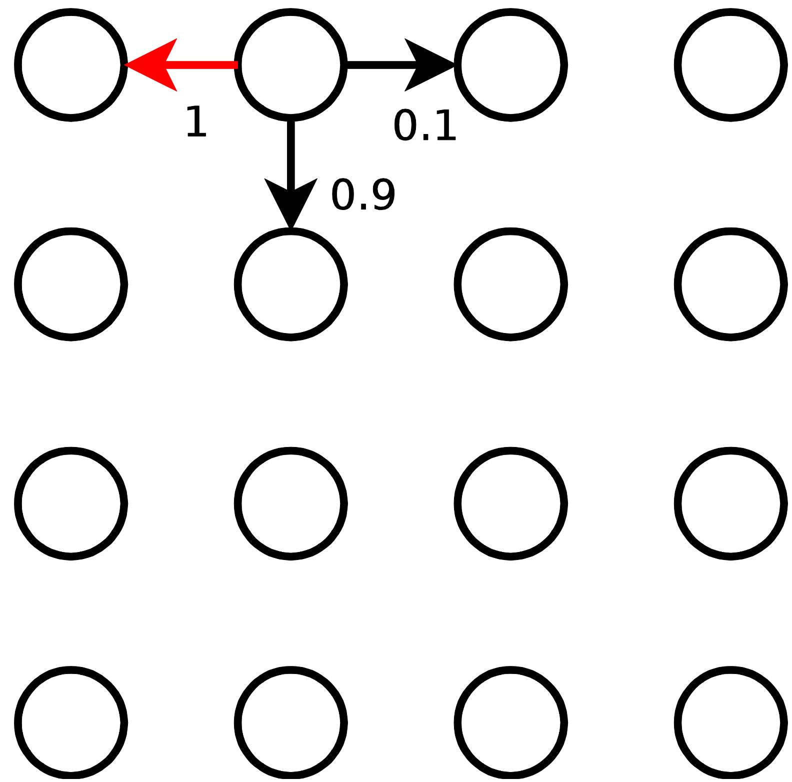
- State Space



RAM-Rmax

[Leffer et al., 2007]

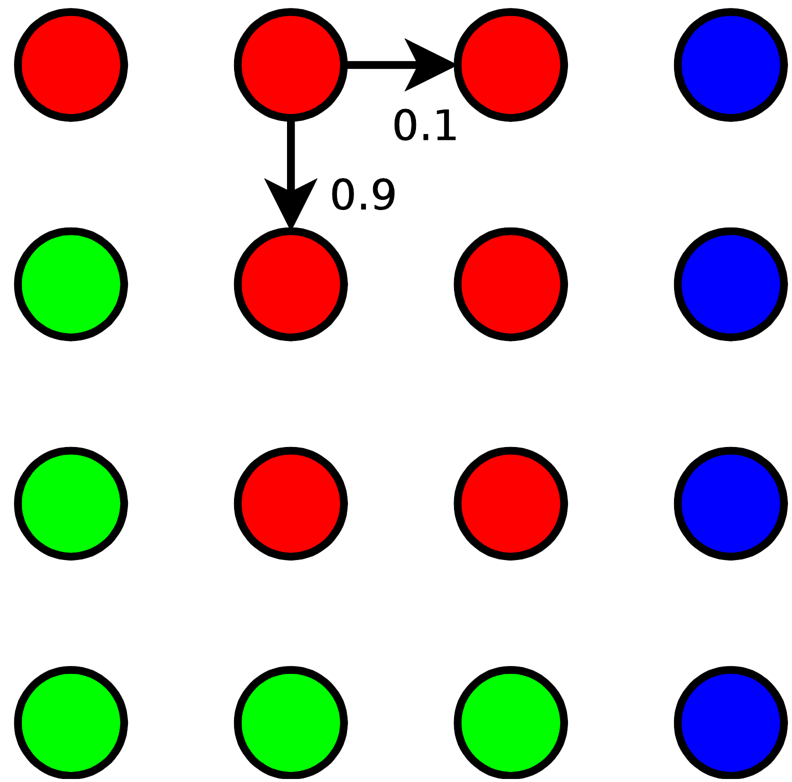
- State Space
- Observe Transitions



RAM-Rmax

[Leffer et al., 2007]

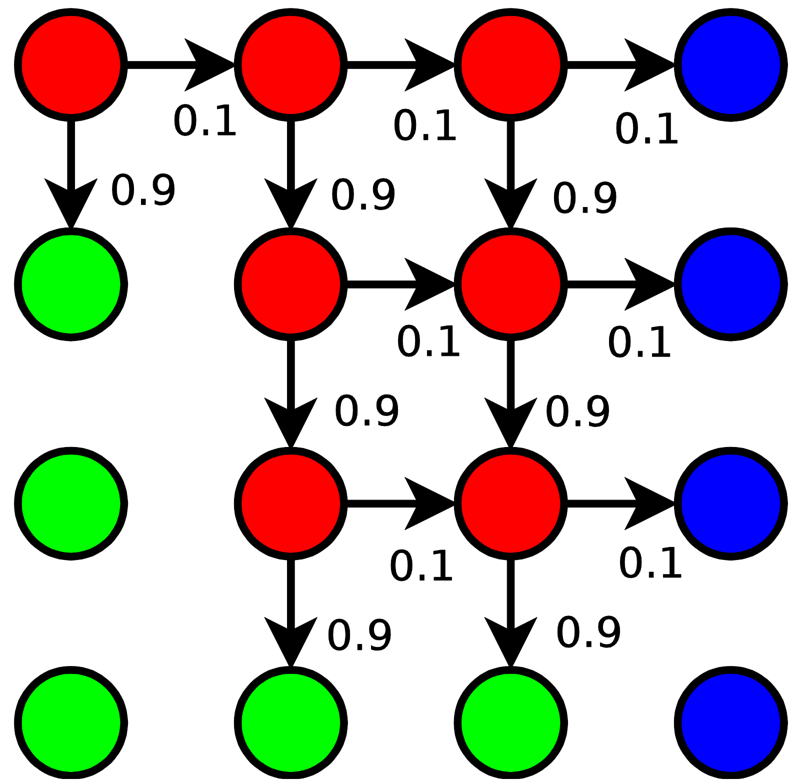
- State Space
- Observe Transitions
- Assign transition statistics to the clusters



RAM-Rmax

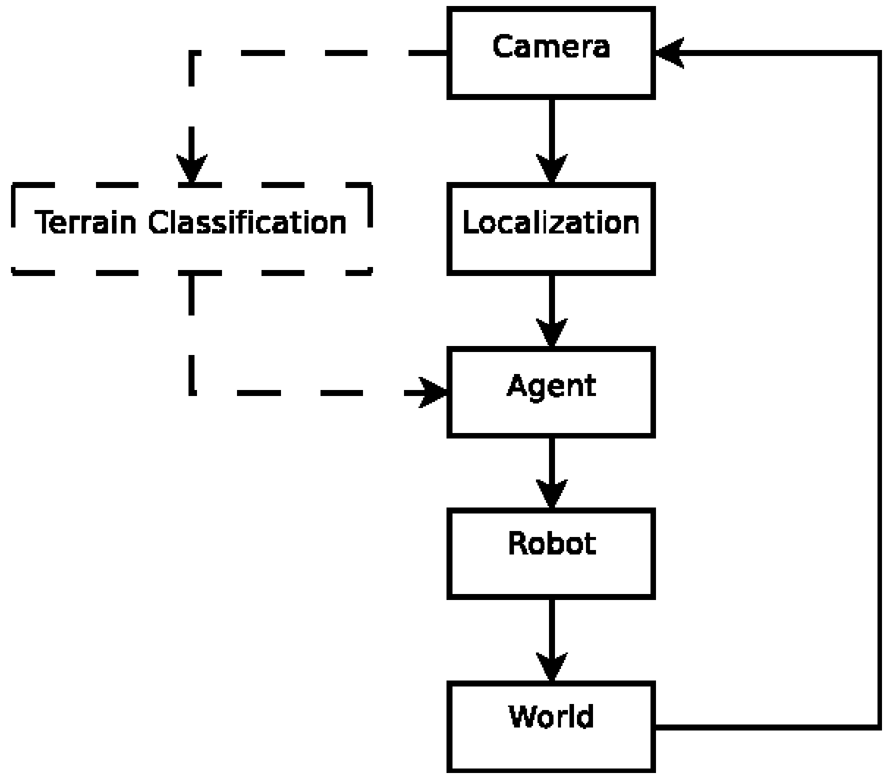
[Leffer et al., 2007]

- State Space
- Observe Transitions
- Assign transition statistics to the clusters
- Use these statistics to plan



System Architecture

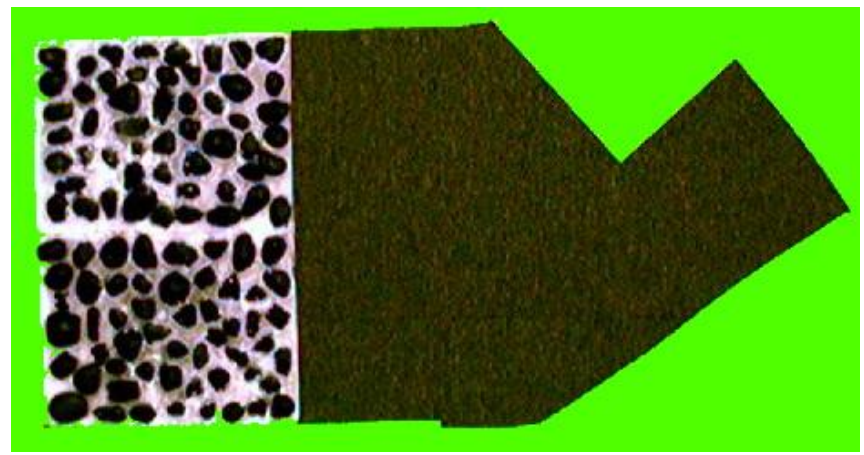
- Camera
- Terrain Classification
- Localization
- RAM-Rmax
- Action



Terrain Classification

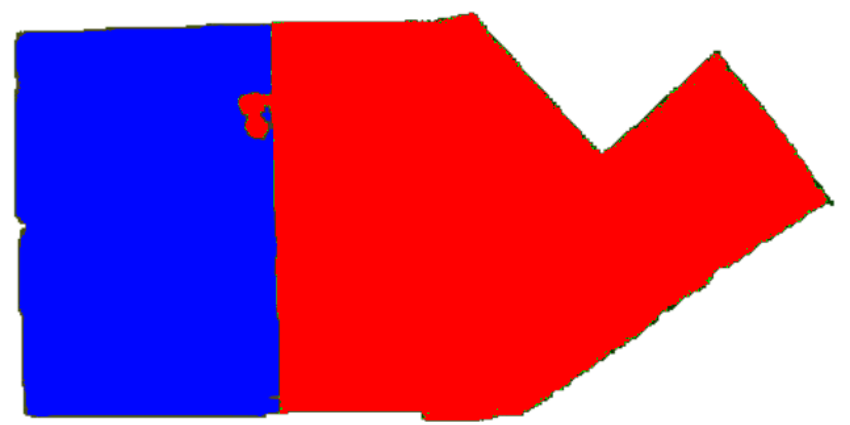
[Comanicu and Meer, 2002]

- "Off the shelf" segmentation of terrain into two areas
- The only parameter given to the segmentation algorithm was to limit the size of the smallest area found



Terrain Classification
[Comanicu and Meer, 2002]

- "Off the shelf" segmentation of terrain into two areas
- The only parameter given to the segmentation algorithm was to limit the size of the smallest area found

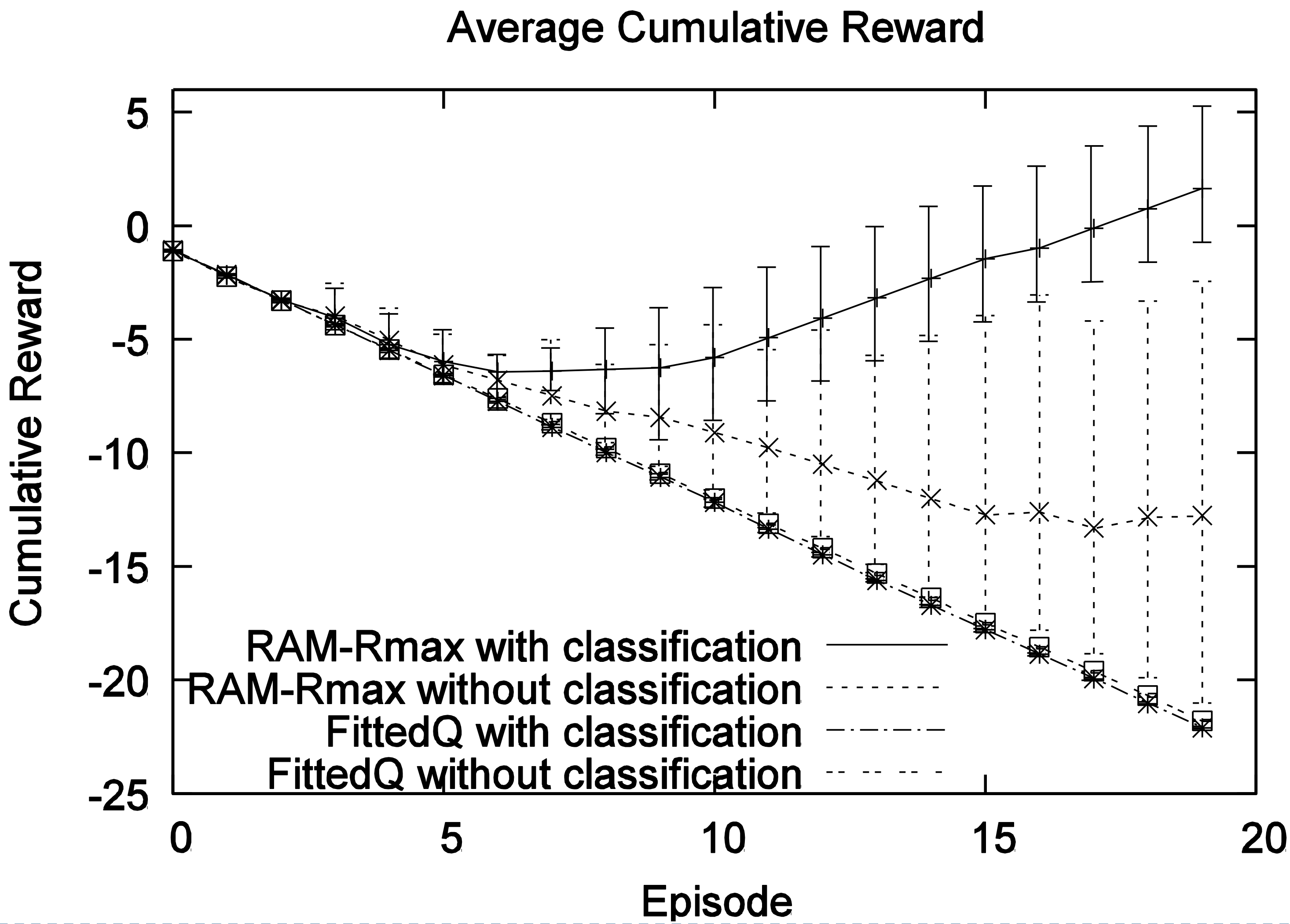


Task Description

- Navigate to Goal
- Reaching the goal or falling out ends the episode
- If you assume one dynamics model, the variance will be large enough that positioning the robot at the goal is close to impossible

States	12000
Actions	3
Step Cost	-0.1
Out of Bounds	-1

Cumulative Reward



Success Rates

- In the last ten episodes, RAM-Rmax with the cluster information succeeded reaching the goal 96% of the time. With one cluster, it only reach the goal 34% of the time.
- Fitted Q Iteration was unable to reach the goal with or without cluster information in 20 episodes.

Conclusions

- Used a framework that allows us to add prior information in a principled way
- Showed that this framework reduces exploration in natural environments
- Empirically demonstrated that the addition of a single extra cluster can radically improve performance
- More powerful than the simple addition of an extra feature to function approximation methods
- Further reduced the dependency on hand tuning from the previous work resulting in a more automated system

Continuous Domains

[Brunskill et al., 2008]

- Instead of representing the model as a set of discrete statistics, learn a Gaussian
- Use the continuous offset (RAM model) with Fitted Value Iteration to solve

Feature Selection

[Li et al., 2008]

- Which features are good dynamics indicators?
- We can learn this
- This enables us to incorporate additional sensors, either alone or in combination

Thank You

Citations

- Brunskill, E., Leffer, B. R., Li, L., Littman, M. L., and Roy, N. (2008). CORL: A continuous-state offset-dynamics reinforcement learner. In Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence (UAI-08).
- Leffer, B. R., Littman, M. L., and Edmunds, T. (2007). Efficient reinforcement learning with relocatable action models. In Proceedings of the 22nd Conference on Artificial Intelligence (AAAI-07),
- Li, L., Littman, M. L., and Walsh, T. J. (2008). Knows what it knows: A framework for self-aware learning. In Proceedings of the Twenty-Fifth International Conference on Machine Learning (ICML-08).
- Sherstov, A. A. and Stone, P. (2005). Improving action election in MDP's via knowledge transfer. In Proceedings of the 20th Conference on Artificial Intelligence (AAAI-05)
- Comanicu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Machine Intell. (TPAMI-02)