Advanced Topics in Reinforcement Learning

Lecture 21: Hierarchical RL II

Josiah Hanna
University of Wisconsin — Madison

Announcements

- Homework 4 due 1 minute ago. Homeworks are done! 🎉🎉
- Next week: Reproducibility and Evaluation
- No class next Thursday! 🎉🎉

Agenda

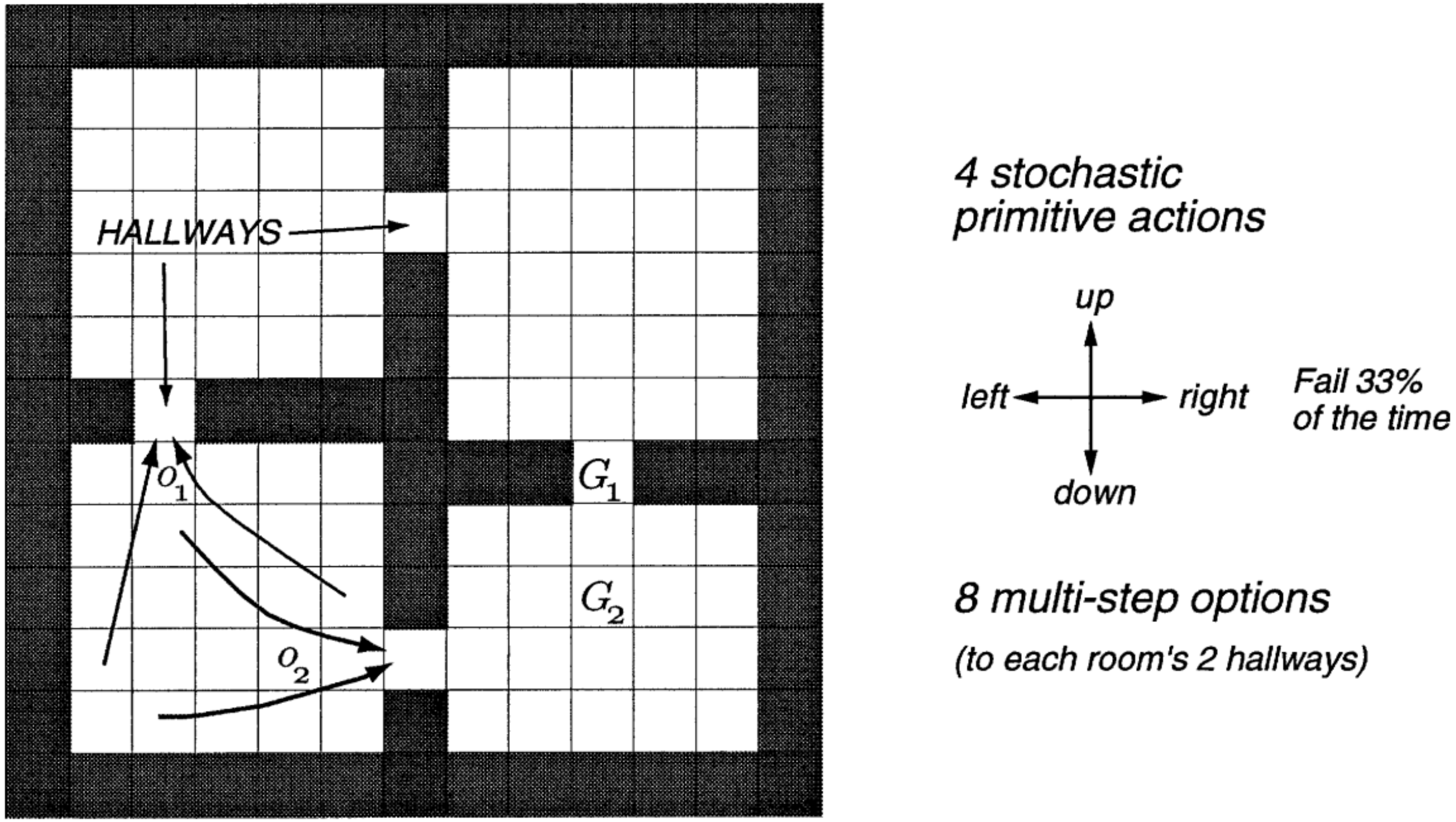
- State Abstractions.
  - What are they?
  - Types of abstractions.
  - State abstraction and deep reinforcement learning.
- An activity.

Abstraction Review

- People complete tasks by planning, learning, and acting at different level of abstraction.
  - Aids credit assignment and exploration.
- Behaviors are modular and re-used across tasks.
  - Transfer learning; subtask learning
- Different states may be functionally the same.

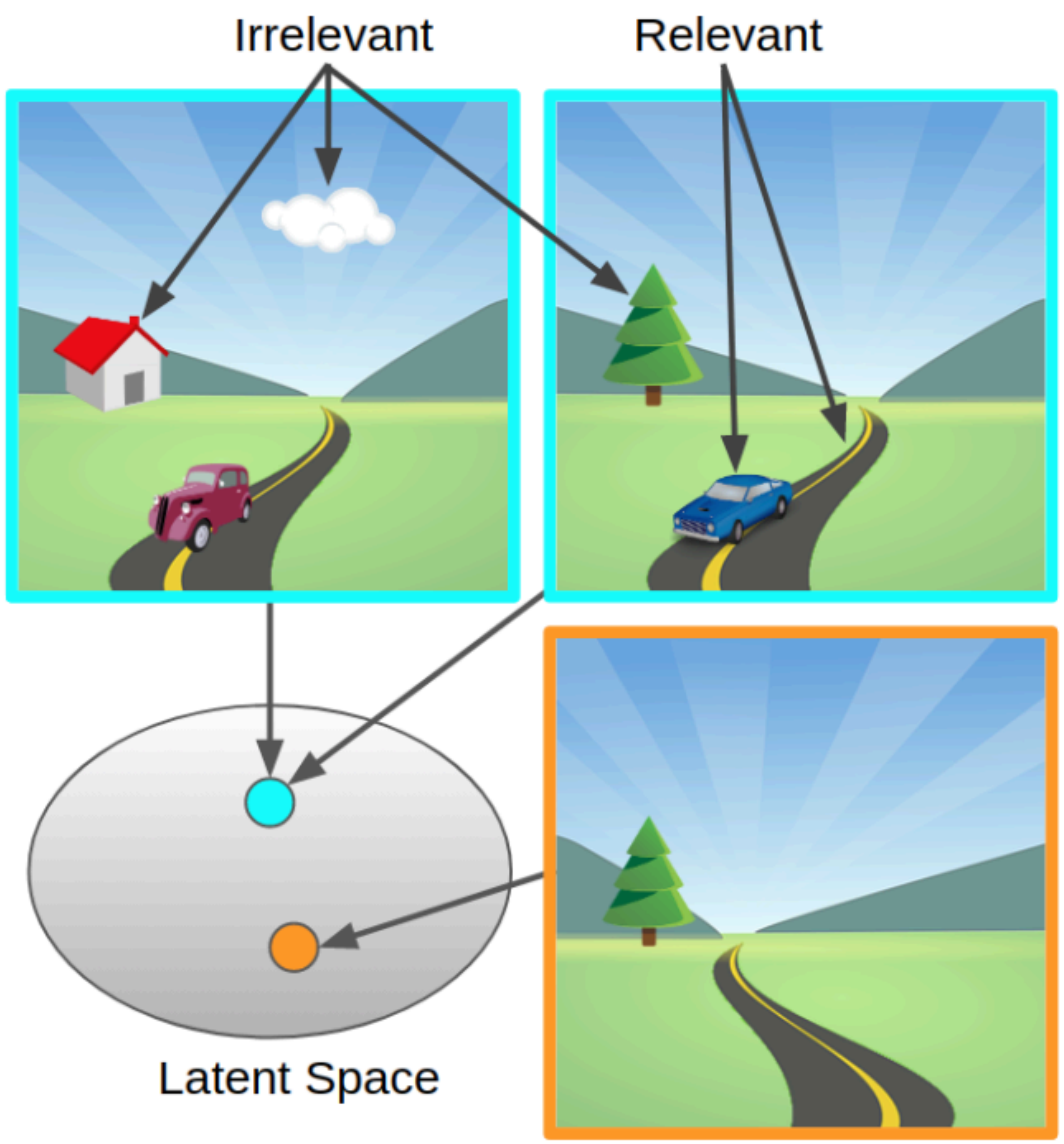
Types of Abstraction

- Temporal Abstraction
- State Abstraction



State Abstraction

- Real life problems have many states
  - But differences between states may be superficial and not affect optimal decision-making.
- A state abstraction removes irrelevant state information to promote generalization and faster learning.



Credit: Amy Zhang

Deep Bisimulation for Control. Zhang et al. 2018.

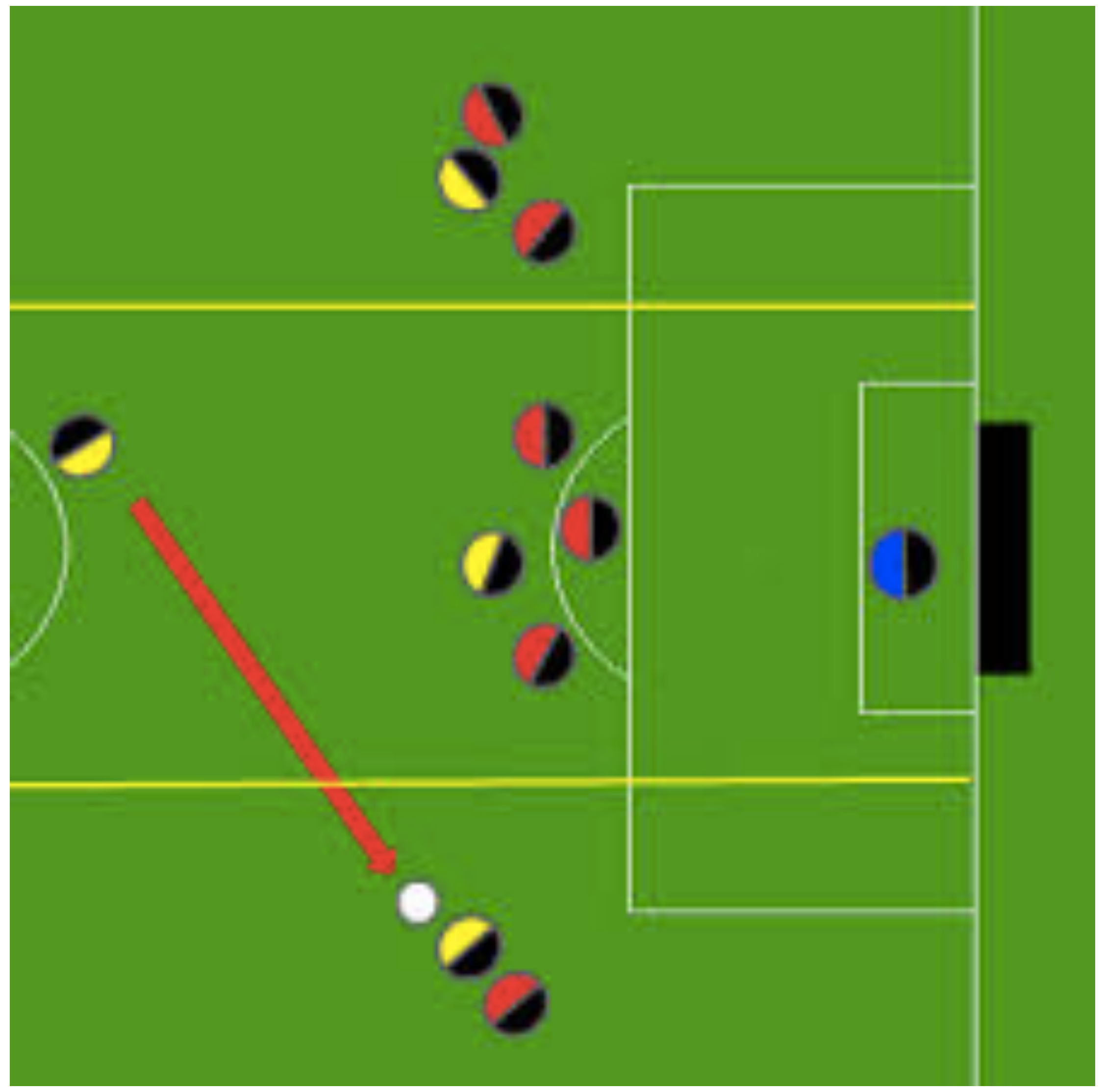
State Abstraction

- Define a state abstraction function as $\phi : \mathcal{S} \rightarrow \overline{\mathcal{S}}$.
- $\phi(s) \in \overline{\mathcal{S}}$ is an abstract state. We call $\mathcal{S}$ the ground state-space.
- The state abstraction function partitions the ground state-space into disjoint sets.
- Induces a new MDP $\langle \overline{\mathcal{S}}, \mathcal{A}, \bar{r}, \bar{p} \rangle$:
  - $\bar{r}(\bar{s}, a) = \sum_{s \in \phi^{-1}(\bar{s})} w(s) r(s, a)$

    $w(s)$ is a state weighting function.
  - $\bar{p}(\bar{s}' \mid \bar{s}, a) = \sum_{s \in \phi^{-1}(\bar{s})} \sum_{s' \in \phi^{-1}(\bar{s}')} w(s) p(s' \mid s, a)$

State Abstraction

- Weighting function is necessary to define a Markov reward and transitions.



Types of State Abstraction

- Many choices for ϕ. What properties should ϕ have?
    - Depends on what "irrelevant" means!
- Model-irrelevance: if two states, s_1 and s_2 are grouped together ($\phi(s_1) = \phi(s_2)$), then s_1 and s_2 have identical rewards and probabilities of leading to any other abstract state.
    - This property is also called bisimulation.
- $\pi^\star$-irrelevance: if two states are grouped together then the optimal action is the same in both.
- Some other choices in between: q_π- irrelevance, $q_\star$-irrelevance, $a^\star$-irrelevance.

Towards a Unified Theory of State Abstraction for MDPs. Li et al. 2006.

Deep State Abstraction

- Partitioning the state space may be difficult with high-dimensional state spaces.
- Instead, learn state abstractions with multi-layer neural networks.
- Define $\phi(s)$ as the non-linear function defined by the first k layers of a network.
- Much recent work on attempting to learn ϕ that exhibit properties.
  - "MICo: Improved representations via sampling-based state similarity for Markov decision processes." Castro et al. 2022.
  - "Learning Invariant Representations for Reinforcement Learning without Reconstruction." Zhang et al. 2021.
  - "DeepMDP: Learning Continuous Latent Space Models for Representation Learning." Gelada et al. 2019.

Activity

- Divide into groups.
- Each group is assigned an RL application.
- Goal: Set-up as an RL problem and architect an RL solution.
- Then, present your application, problem, and solution to the class.

Activity

- First, choose a presenter.
- Consider your given application and the following questions:
  - What does success look like for your application?
  - How would you formalize as an MDP or bandit?
  - Continuing or episodic?
  - What algorithm (or class of algorithms) would you use to find $\pi^\star$? Why?
  - Is off-policy learning important or necessary for the application?
  - Are their good simulators available for your domain?
  - Function approximation necessary? If so, what type?
  - Temporal abstraction necessary? If so, what options would you consider providing or learning?
  - How will you define success of learning?
- In 10 minutes, your group presenter should be prepared to present your application, RL problem formulation, and proposed solution method with the rest of the class.

Summary

- Abstraction enables more sample efficient reinforcement learning.
  - State abstractions increase generalization.
  - Options promote exploration and aid credit-assignment.
- Potentially sub-optimal convergence.
  - Maybe we have to accept this?

Action Items

- Read “Deep Reinforcement Learning that Matters.”
- Continue making progress on your final project.