# Soar-RL Discussion: Future Directions, Open Questions, and Why You Should Use It

31st Soar Workshop

# Soar-RL Today

- Robust framework for integrated online RL
- Big features:
  - Hierarchical RL
  - Internal actions
  - Architectural constraints
    - Generalization, time
- Active research pushing in those directions

# Soar-RL Tomorrow

- Future research directions
  - Additional architectural constraints
  - Fundamental advances in RL
- Increased adoption
  - Usage cases
  - Understanding barriers to use

# Why You Should Use Soar-RL

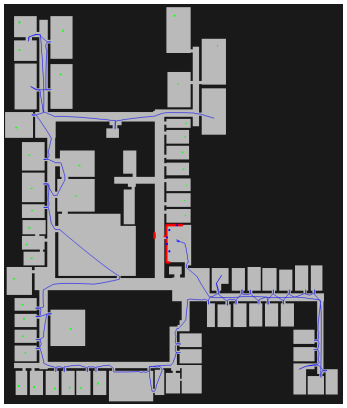
- Changing environment
- Optimized policies over environment actions
- Balanced exploration and exploitation

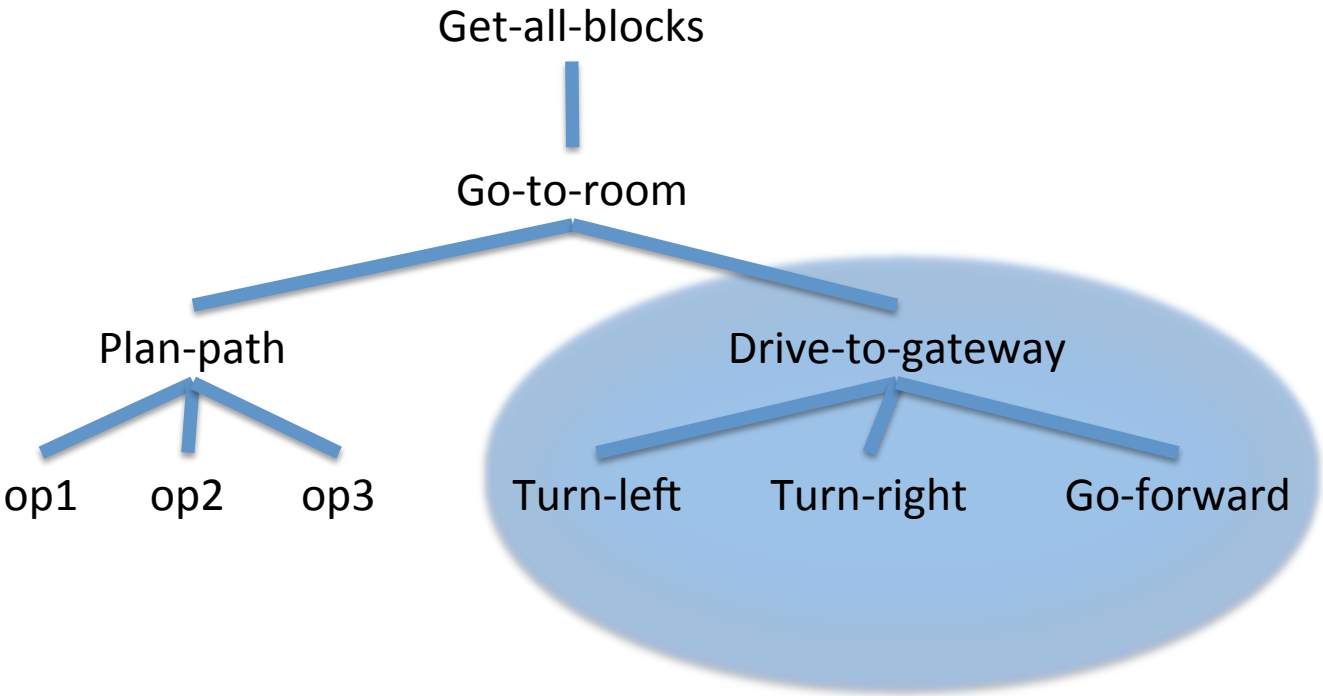
- Why *I* would use Soar-RL if *I* were *you*

# Non-stationary Environments

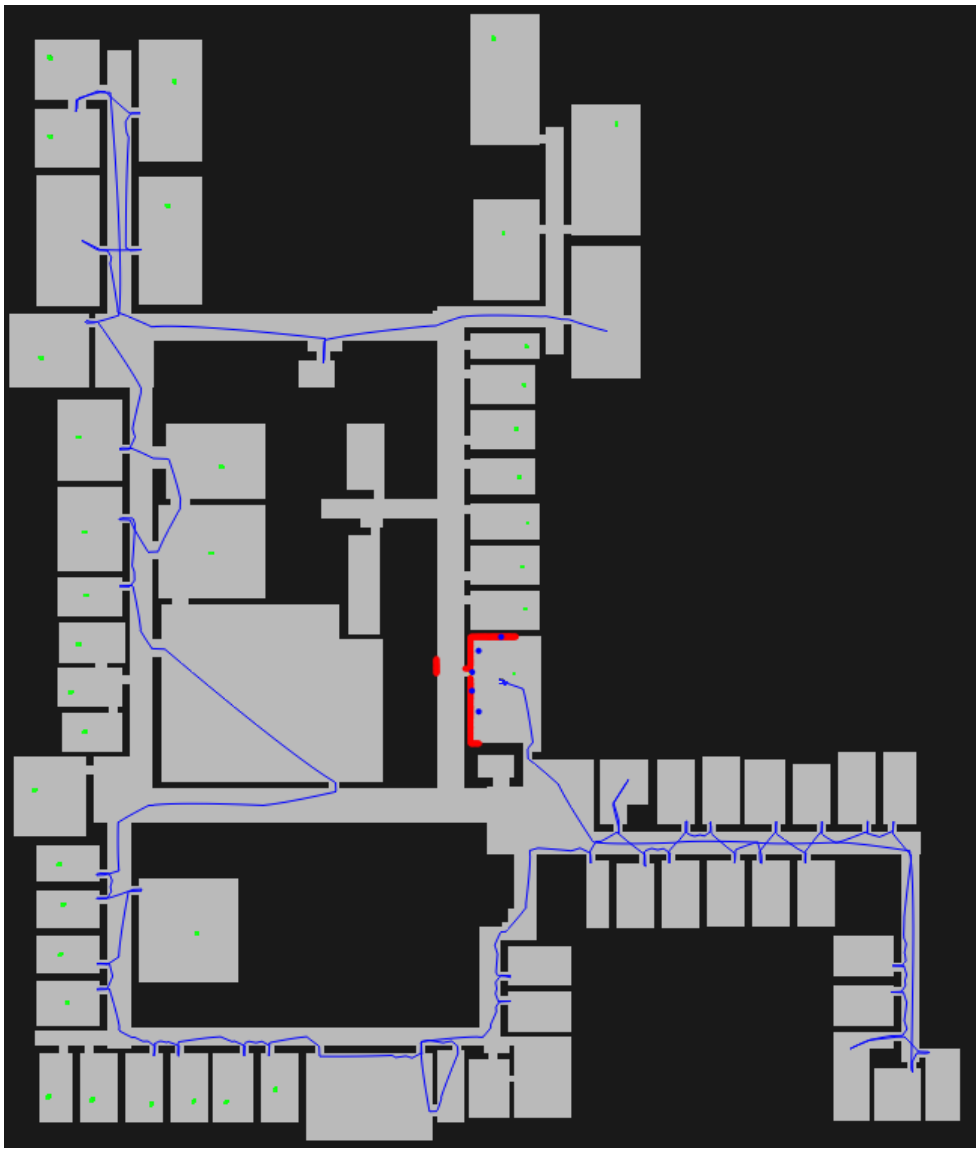
- Dynamic environment regularities
  - Adversarial opponent in a game
  - Weather or seasons in a simulated world
  - Variations between simulated and embodied tasks
  - Limited transfer learning, or tracking
- Agents that persist for long periods of time

# Pragmatic Soar-RL

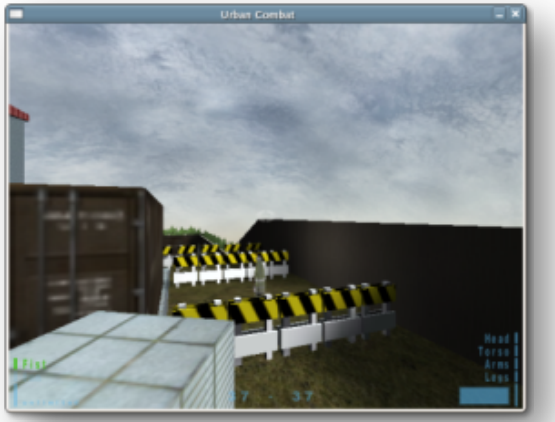


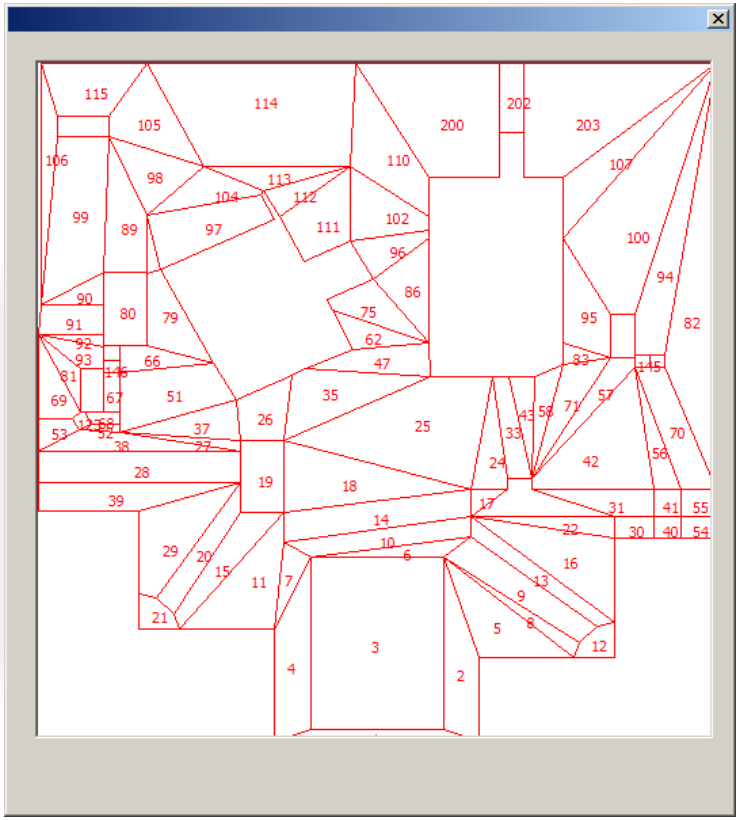


# Optimizing External Actions



# Balanced Exploration & Exploitation



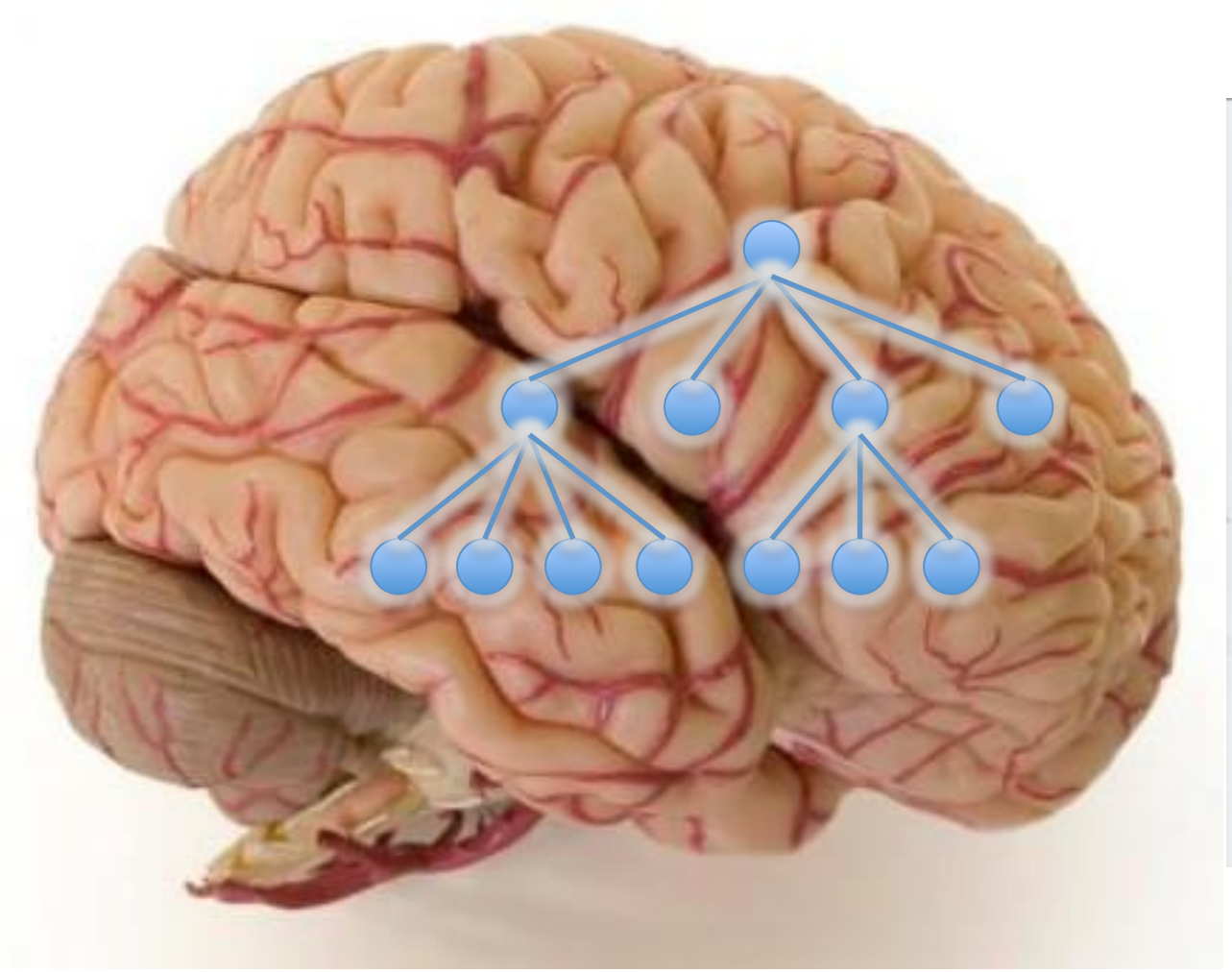



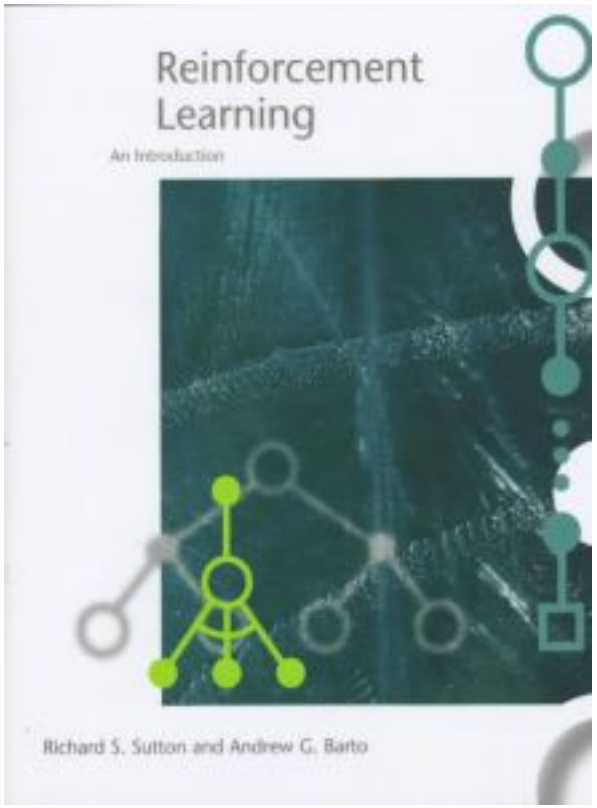

- Stationary, but stochastic actions

# Practical Reasons

- Programmer time is expensive
- Learn in simulation, near-transfer to embodied agent
- If hand-coded behavior can guarantee doctrine, then so can RL behavior
  - Mix symbolic and adaptive behaviors FIX

# Biggest Barrier to Using Soar-RL





# Other Barriers to Adoption?

# I Know What You're Thinking

# Future Directions for Soar-RL

- Parameter-free framework
- Issues of function approximation
- Scaling to general intelligence
- MDP characterization

# Parameter-free framework

- Want fewer free parameters
  - Less developer time finding best settings
  - Stronger architectural commitments
- Parameters are set initially, and can evolve over time

**Policies**

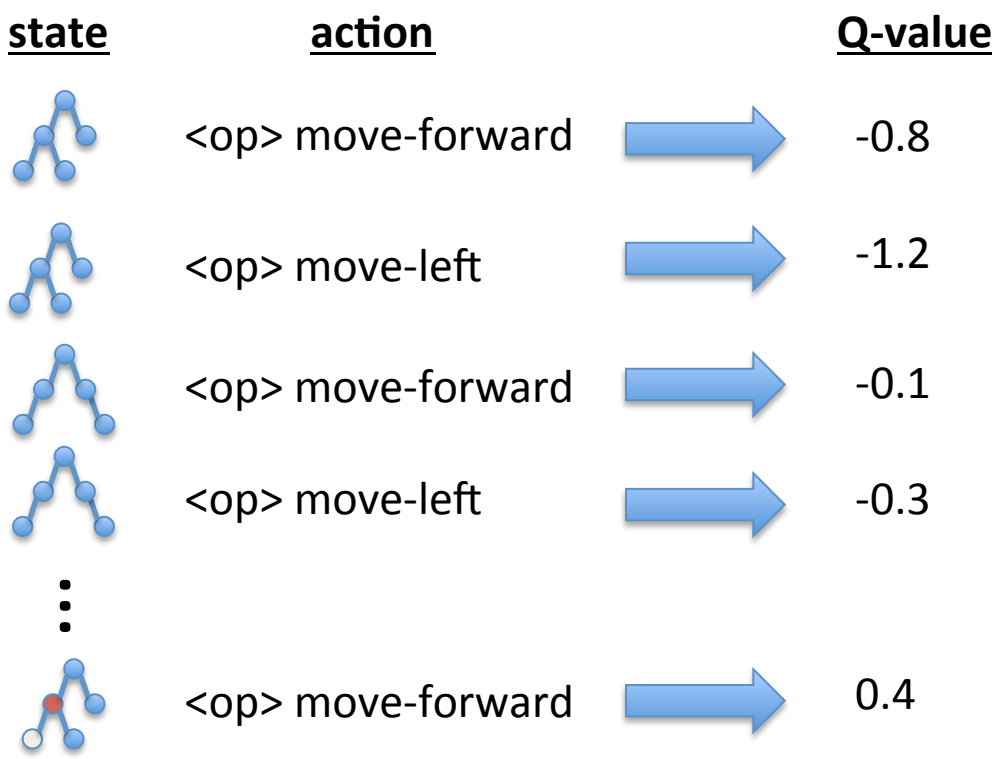
Exploration
Gap handling
Learning algo.

**Knobs**

Learning rate
Exploration rate
Initial values
Eligibility decay

# Soar-RL Value

- Q-value: expected future reward after taking action *a* in state *s*



# Soar-RL Value Function Approximation

- Typically agent WM has lots of knowledge
  - Some knowledge irrelevant to RL state
  - Generalizing over relevant knowledge can improve learning performance
- Soar-RL factorization is non-trivial
  - Independent rules, but dependent features
  - Linear combination of rules & values
  - Rules that fire for more than one operator proposal
- Soar-RL is a good framework for exploration

# Scaling to General Intelligence

- Actively extending Soar to long-term agents
  - Scaling long-term memories
  - Long runs with chunking
- Can RL contribute to long-term agents?
  - Intrinsic motivation
  - Origin of RL rules
  - Correct factorization
    - Rules, feature space, time
    - What is learned with Soar-RL and what is hand coded
  - Learning at the evolutionary time scale

# MDP Characterization

- MDP: Markov Decision Process
- Interesting problems are non-Markovian
  - Markovian problems are solvable
- Goal: use Soar to tackle interesting problems
- Are SARSA/Q-learning the right algorithms?

(The catch: basal ganglia performs temporal-difference updates)


Questions?