



USC Institute for
Creative Technologies

University of Southern California

Reinforcement Learning for Adaptive Theory of Mind in the Sigma Cognitive Architecture

David V. Pynadath, Paul S. Rosenbloom,
Stacy C. Marsella

The work depicted here was sponsored by the U.S. Army. Statements and opinions expressed do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.





Overall Progress on Sigma

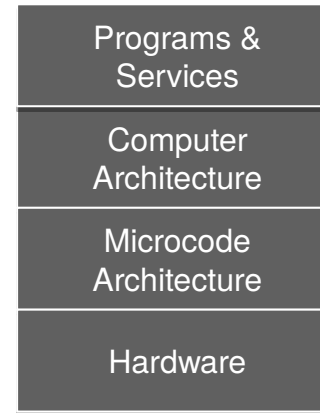
- Memory [ICCM 10]
 - Procedural (rule)
 - Declarative (semantic/episodic) [CogSci 14]
 - Constraint
 - **Distributed vectors** [AGI 14a]
- Problem solving
 - Preference based decisions [AGI 11]
 - Impasse-driven reflection [AGI 13]
 - Decision-theoretic (POMDP) [BICA 11b]
 - **Theory of Mind** [AGI 13, AGI 14b]
- Learning [ICCM 13]
 - Concept (supervised/unsupervised)
 - Episodic [CogSci 14]
 - **Reinforcement** [AGI 12a, AGI 14b]
 - Action/transition models [AGI 12a]
 - **Models of other agents** [AGI 14b]
 - Perceptual (including maps in SLAM)
- Mental imagery [BICA 11a; AGI 12b]
 - 1-3D continuous imagery buffer
 - Object transformation
 - Feature & relationship detection
- Perception
 - Object recognition (CRFs) [BICA 11b]
 - Isolated word recognition (HMMs)
 - Localization [BICA 11b]
- Natural language
 - Question answering (selection)
 - Word sense disambiguation [ICCM 13]
 - Part of speech tagging [ICCM 13]
- Graph integration [BICA 11b]
 - CRF + Localization + POMDP
- Optimization [ICCM 12]



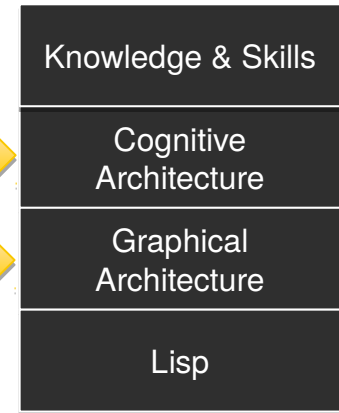
The Structure of Sigma

- Constructed in layers
 - In analogy to computer systems

Computer System



Σ Cognitive System

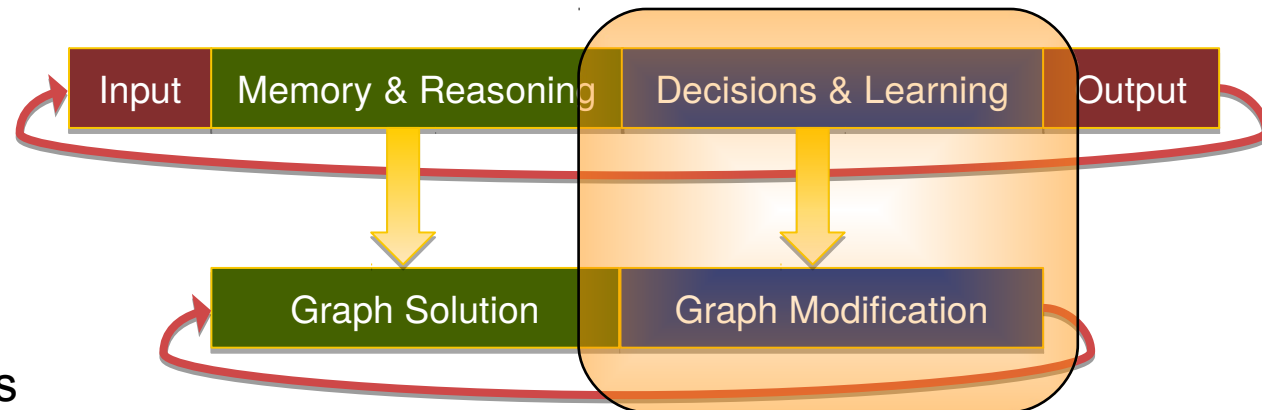


Cognitive Architecture:

- Predicates
- Conditionals
- Nested control structure

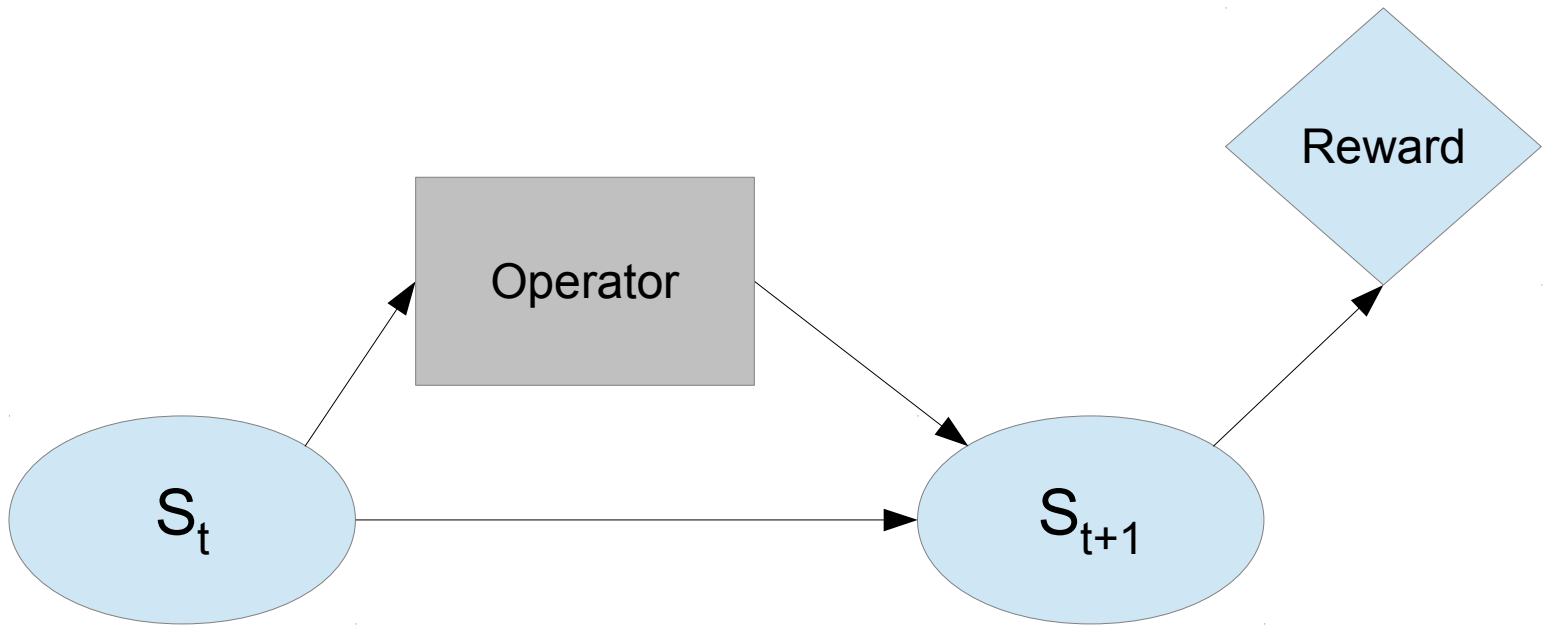
Graphical Architecture:

- Graphical models
- Piecewise-linear functions
- Gradient-descent learning





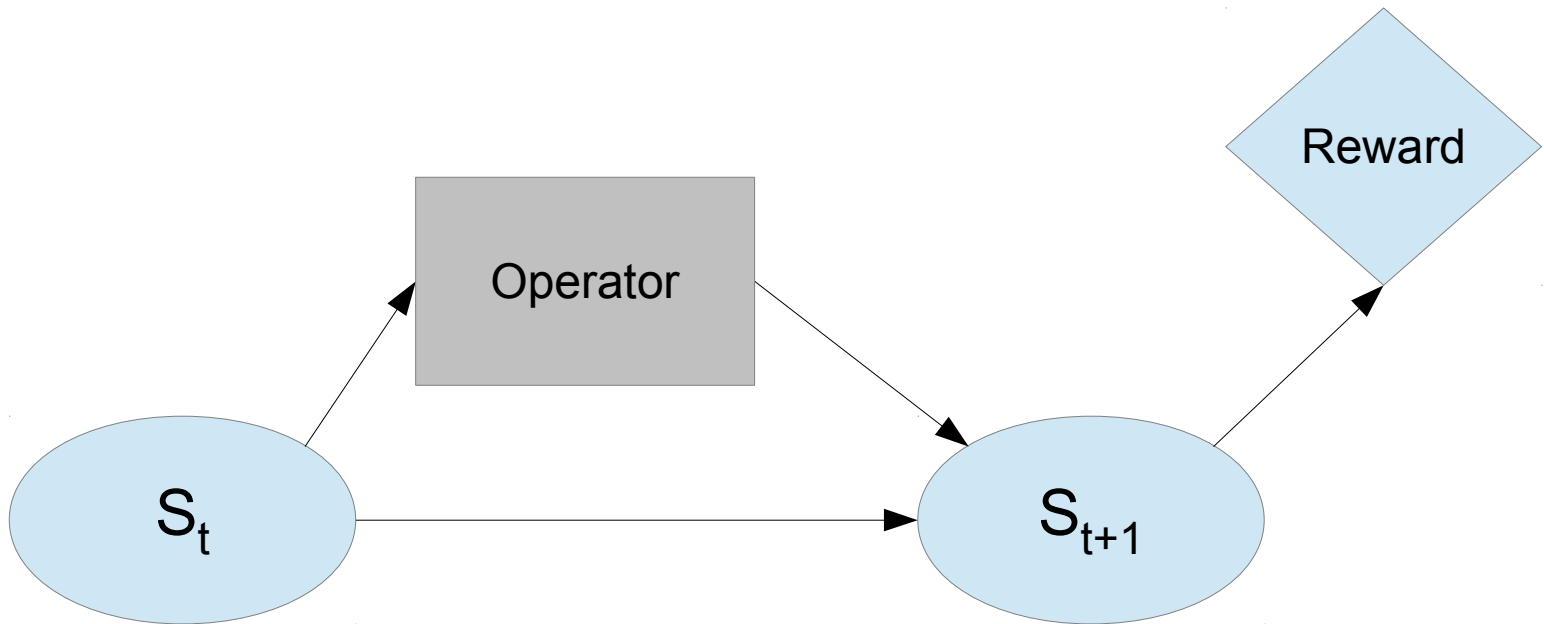
Reinforcement Learning





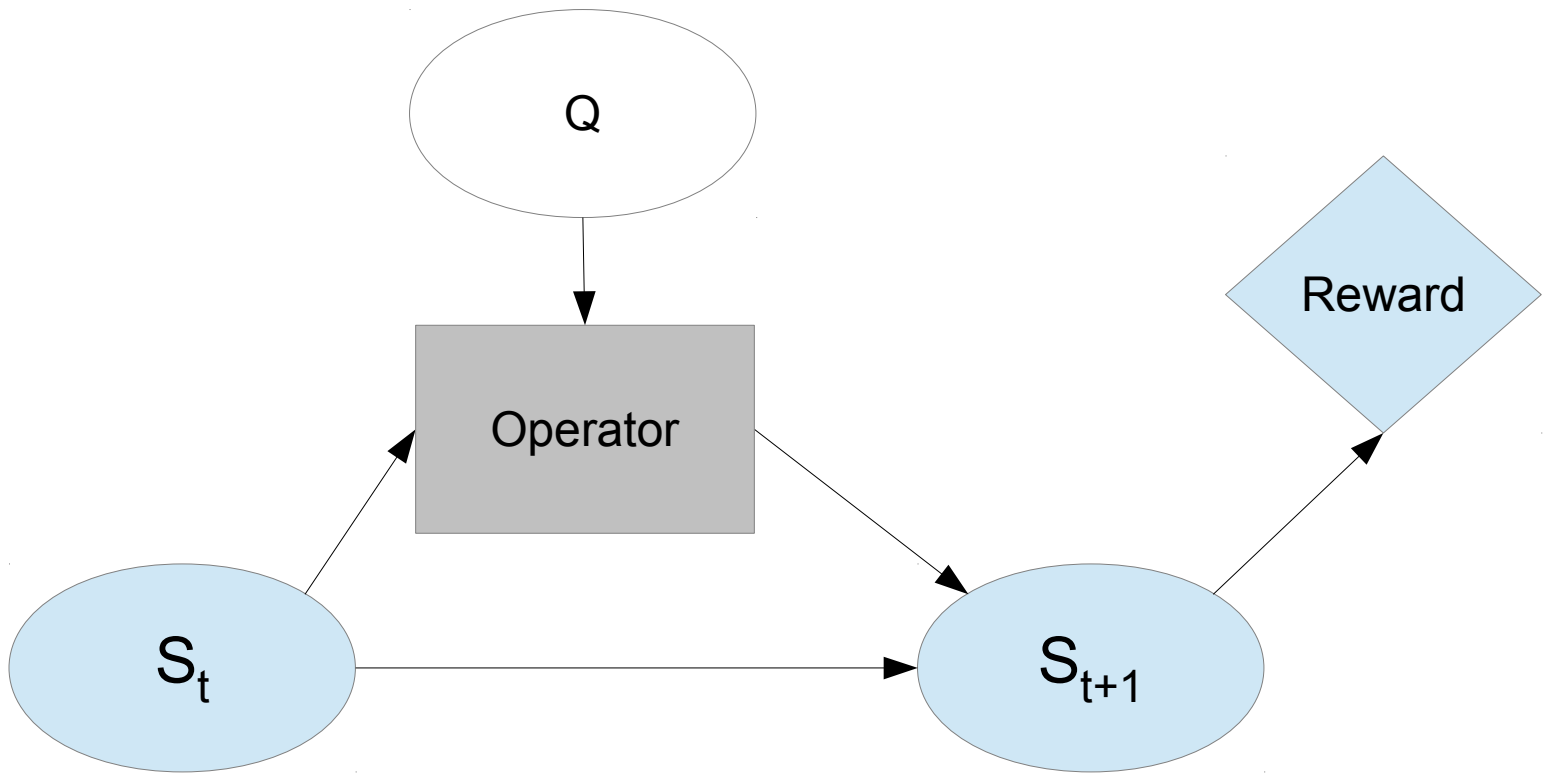
Reinforcement Learning

- We assume here:
 - Transition and reward functions are known
 - States and rewards are observable



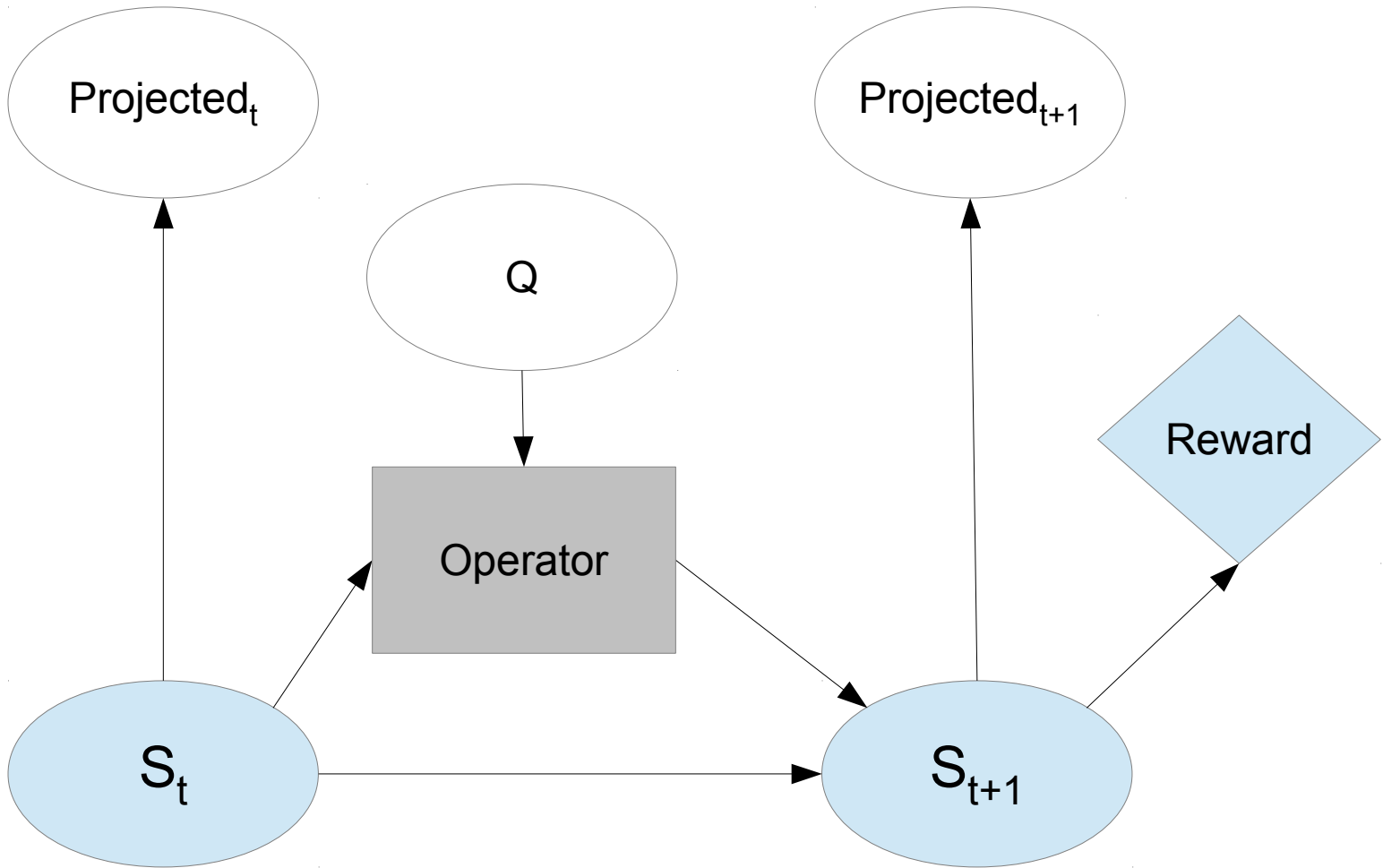


Reinforcement Learning



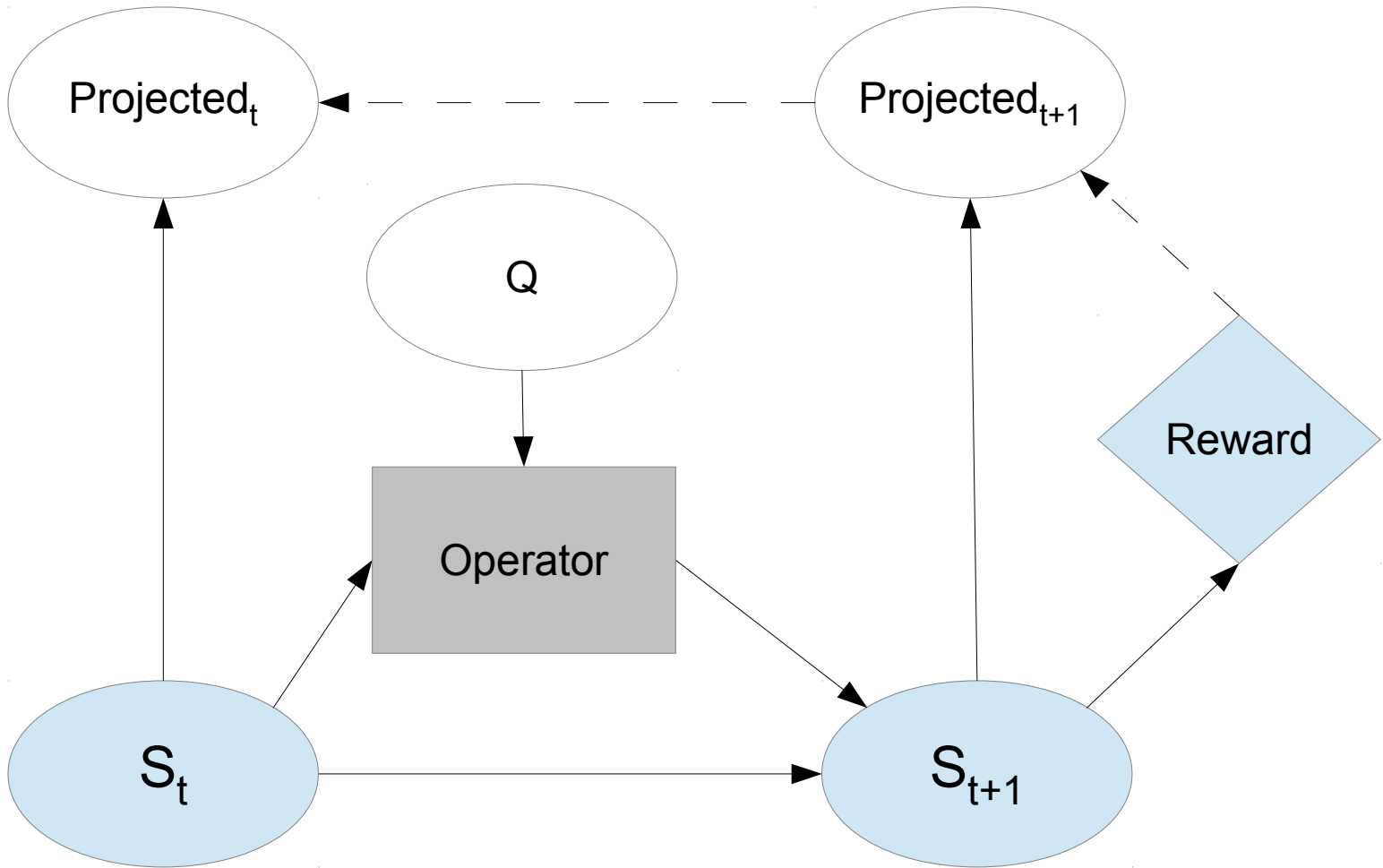


Reinforcement Learning



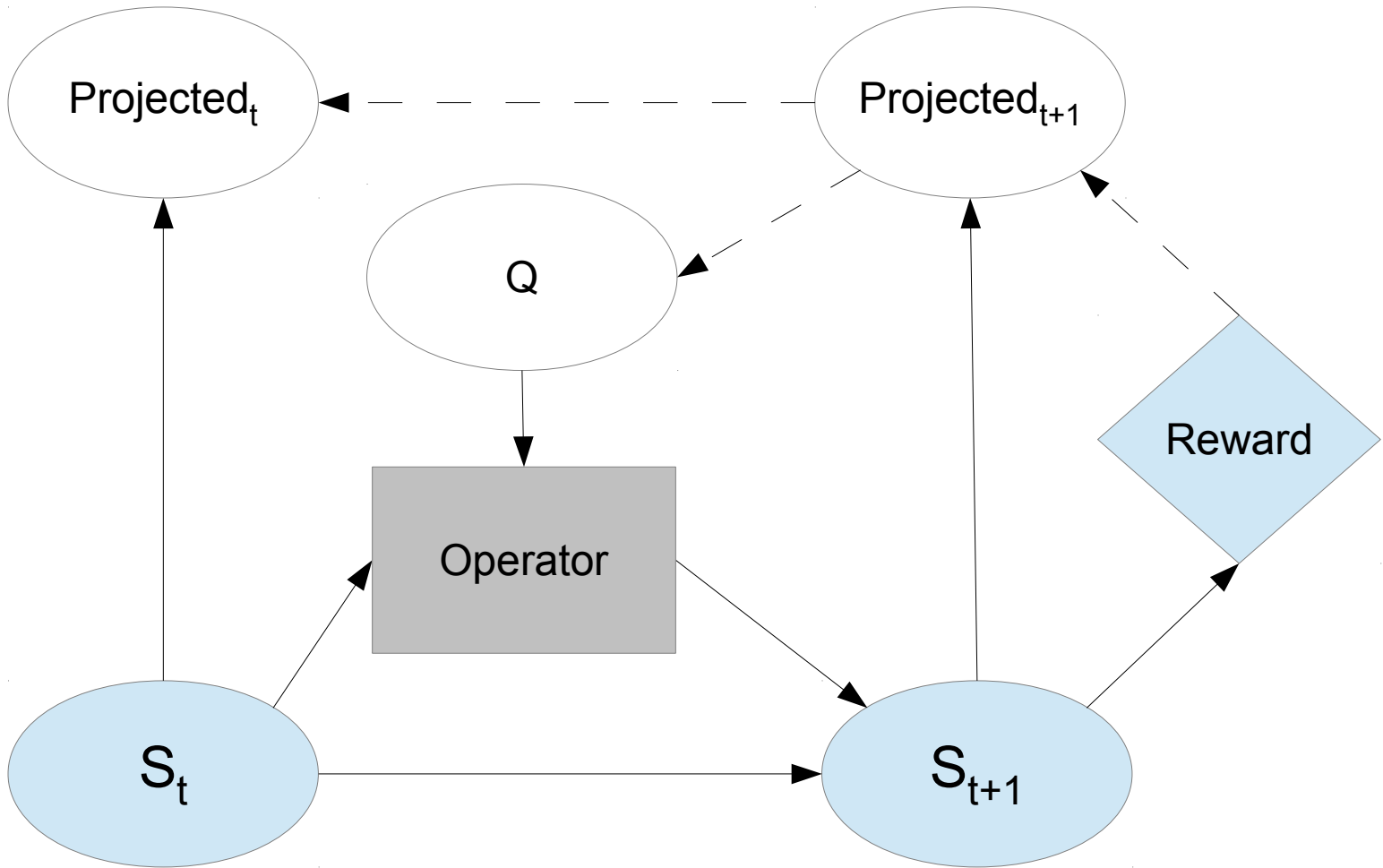


Reinforcement Learning





Reinforcement Learning



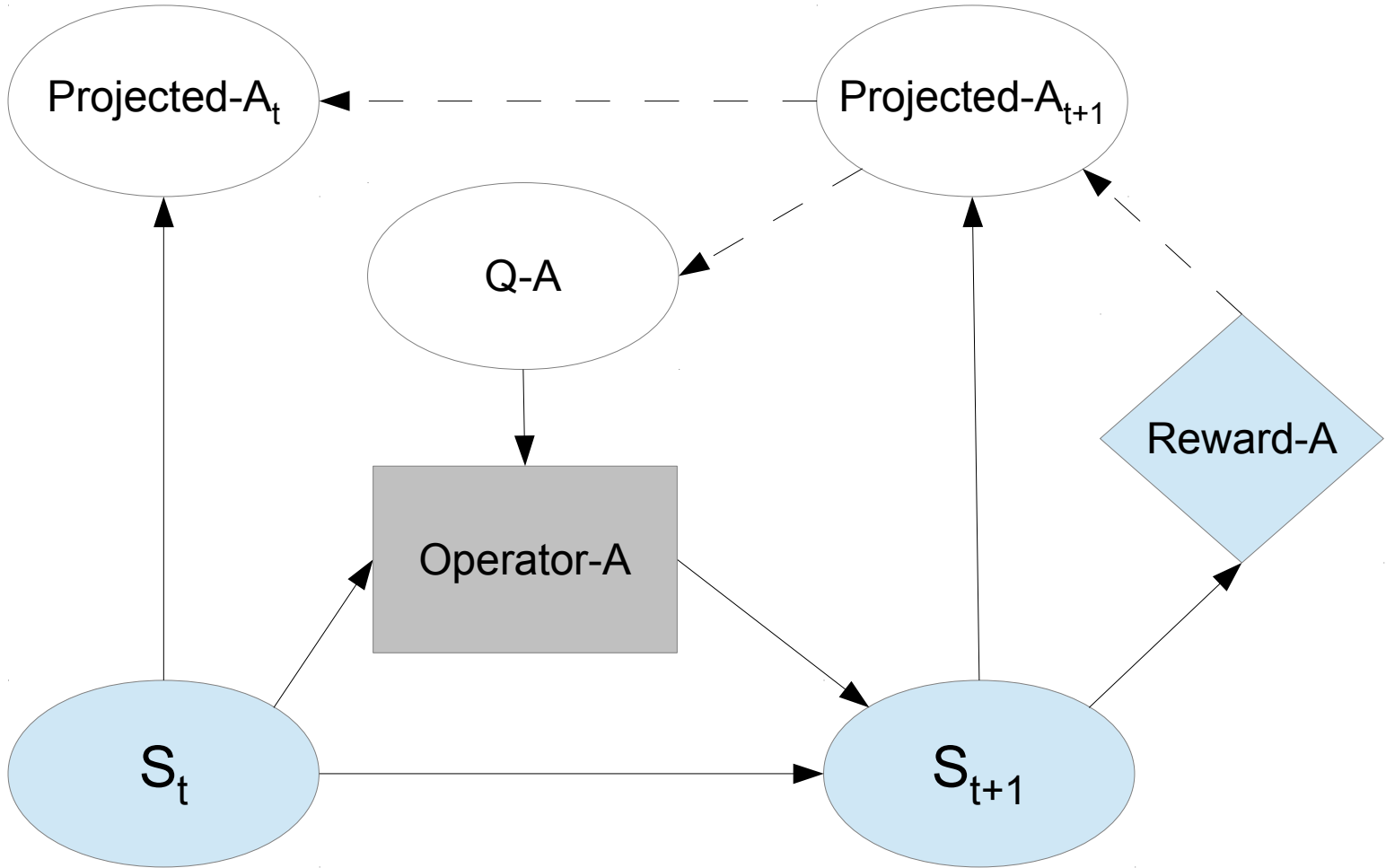


Abstract Negotiation Domain

- Two agents, A and B
 - A learns
 - B does not
- Negotiating over an allocation of fruit: apples and oranges
 - Alternate modifying the allocation on the table
 - Each can accept the current allocation on the table, ending the negotiation
 - Each has an individual reward function depending on the final allocation

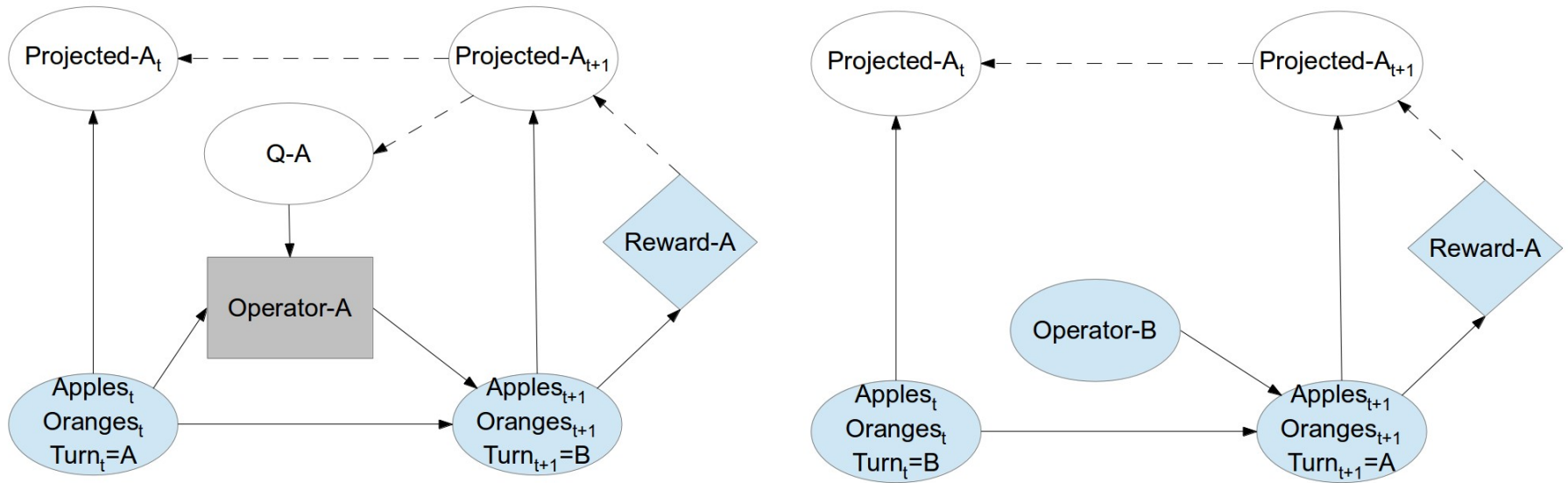


Single-Agent RL





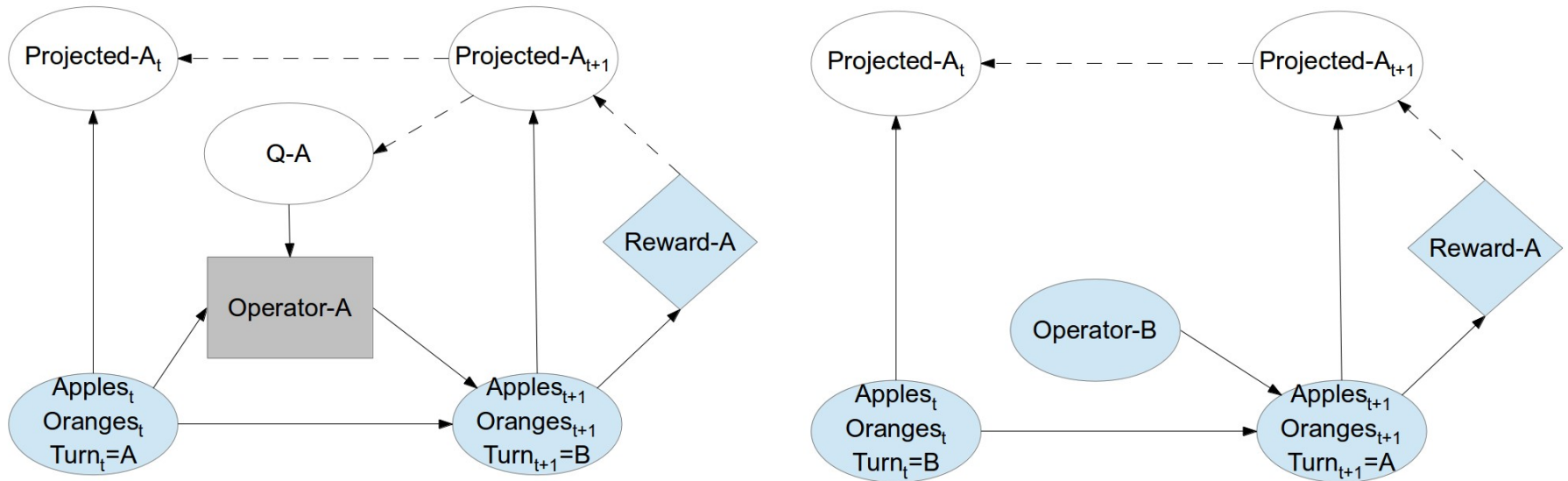
Multiagent RL



- Operator-B is not under *A*'s decision-making control
 - But it affects *A*'s expected reward
 - How should *A* model *B*'s behavior within its learning?



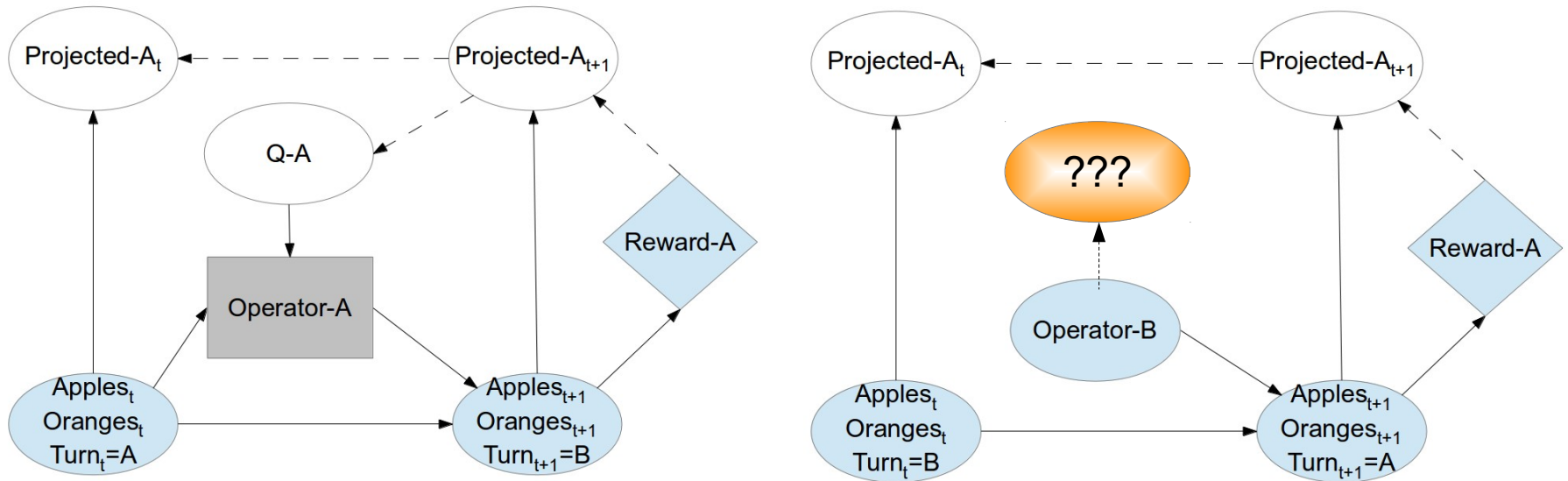
Multiagent RL: No model of B



- No model of the other agent
 - Treat agent as part of the environmental dynamics
 - e.g., Littman, 1994



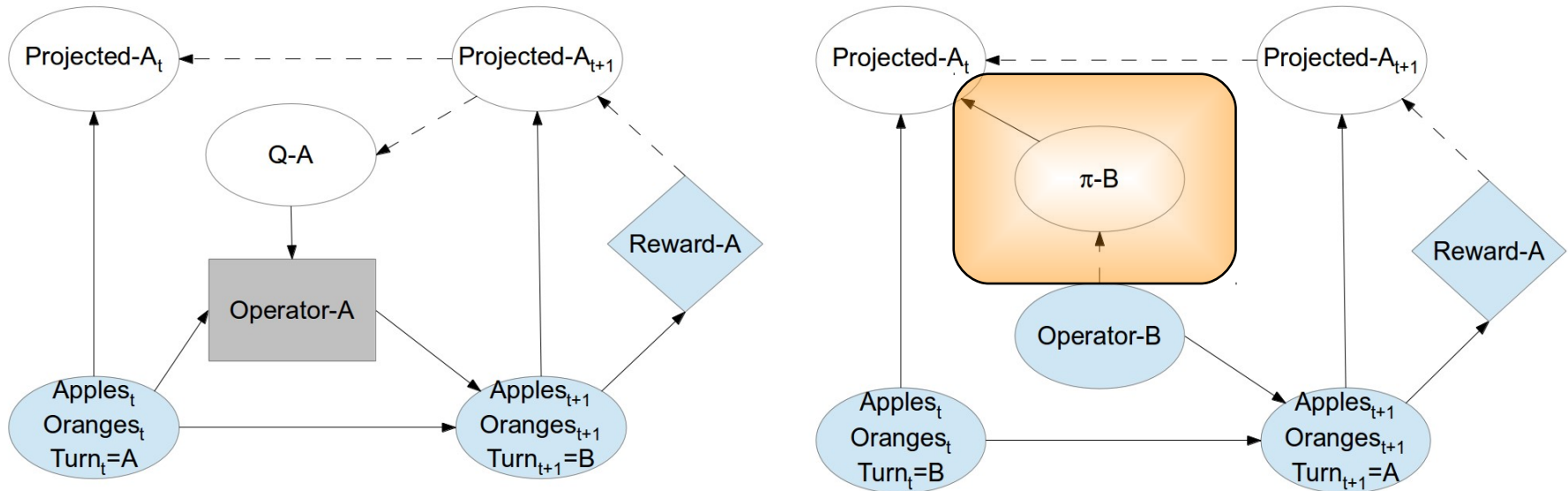
Multiagent RL: No model of B



- No model of the other agent
 - Treat agent as part of the environmental dynamics
 - e.g., Littman, 1994



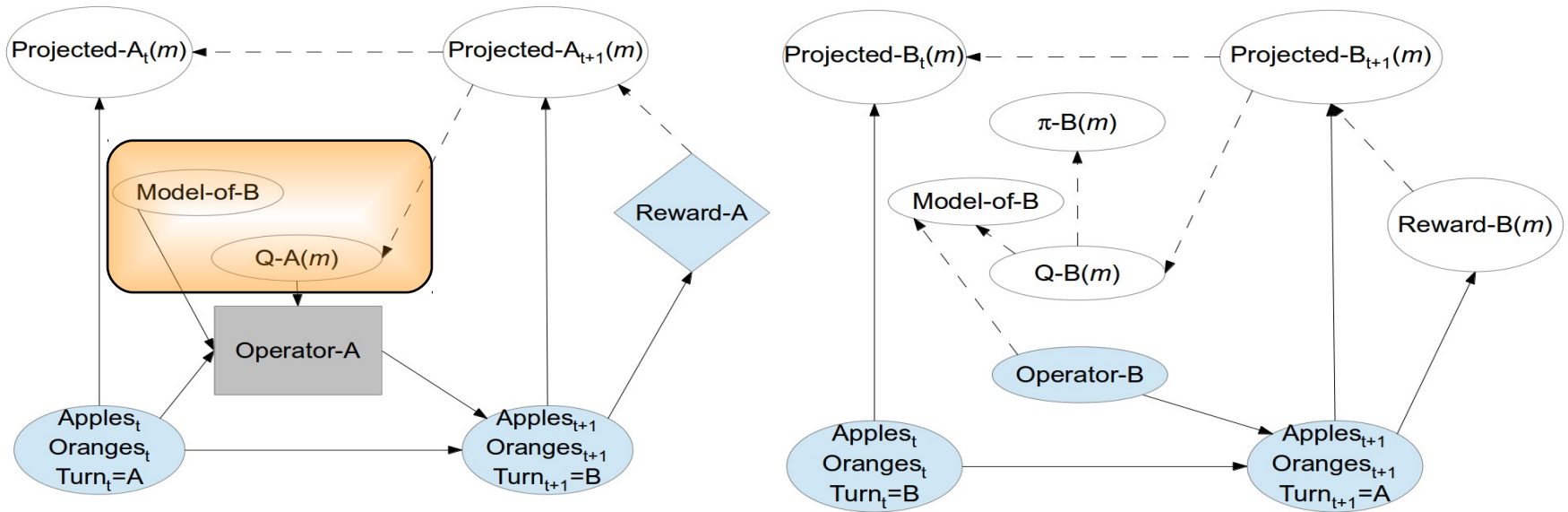
Multiagent RL: Stationary policy model of B



- Model agent as following a fixed stochastic behavior
 - Learn a stationary policy model
 - e.g., Hu & Wellman, 1998 & 2001



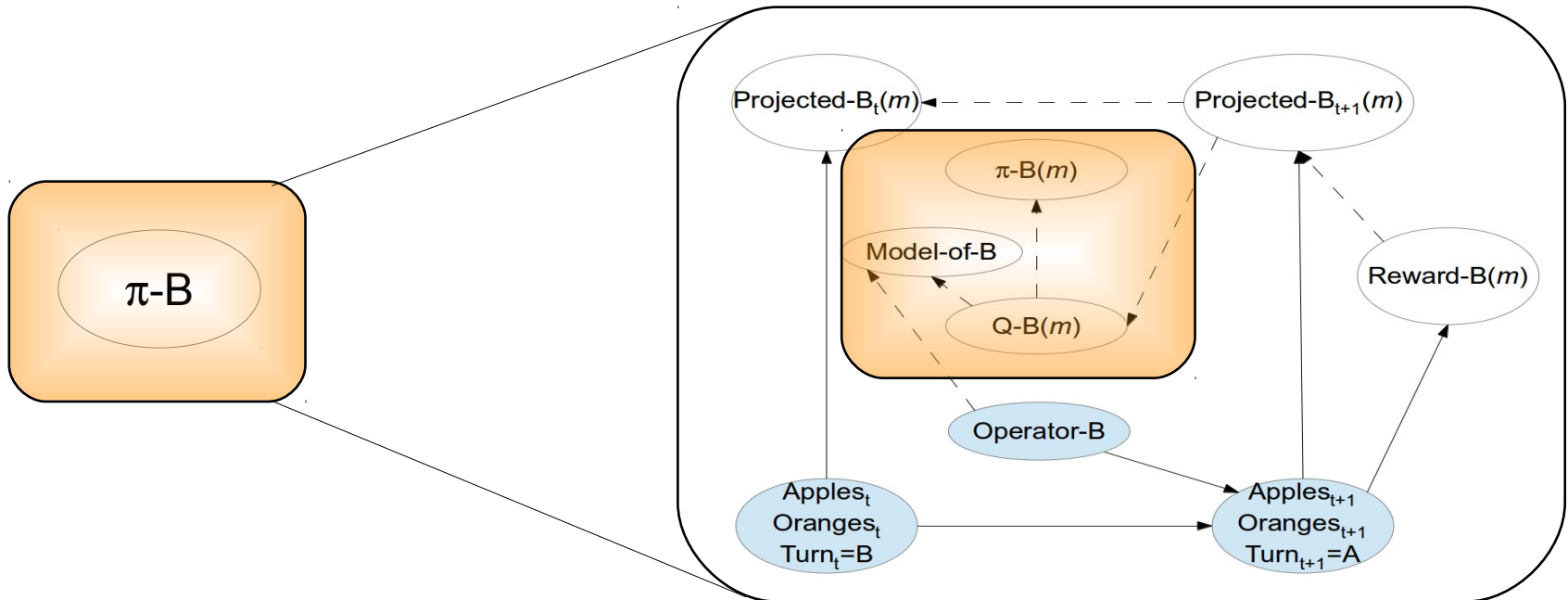
Multiagent RL: RL model of B



- Model agent as maximizing a reward function, drawn from finite subset
 - Treat agent as one of a set of candidate agent types
 - e.g., Gmytrasiewicz & Doshi, 2005; Pynadath & Marsella, 2005



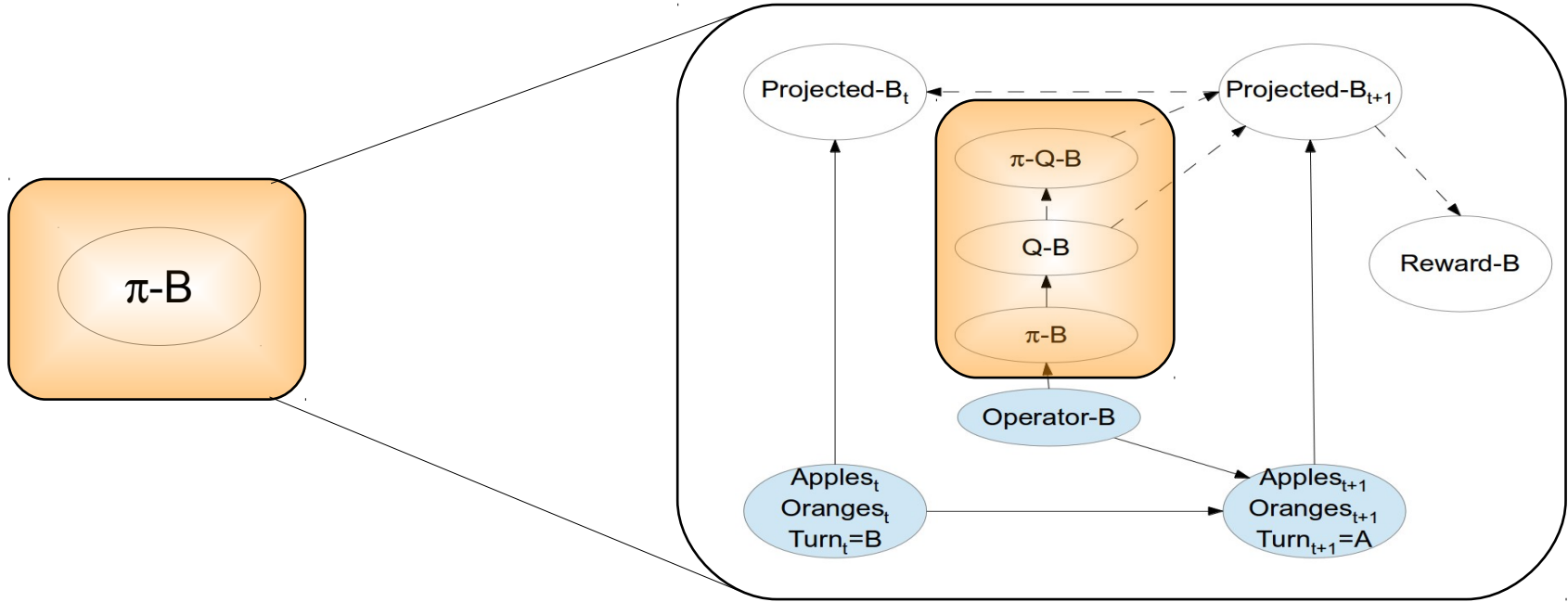
Multiagent RL: RL model of B



- Model agent as maximizing a reward function, drawn from finite subset
 - Treat agent as one of a set of candidate agent types
 - e.g., Gmytrasiewicz & Doshi, 2005; Pynadath & Marsella, 2005



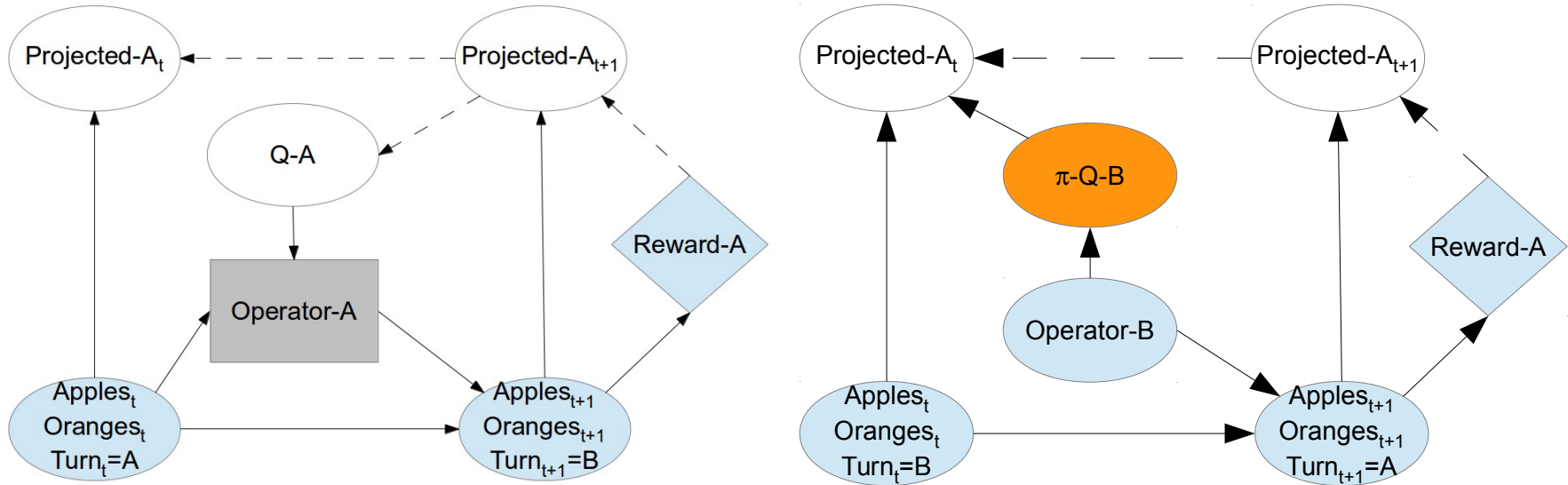
Multiagent RL: IRL of B 's Reward



- Model agent as maximizing a reward function, drawn from entire set
 - Inverse Reinforcement Learning (IRL) to infer B 's reward
 - e.g., Ng & Russell, 2000



Multiagent RL: IRL of B 's Reward



- Model agent as maximizing a reward function, drawn from entire set
 - Inverse Reinforcement Learning (IRL) to infer B 's reward
 - e.g., Ng & Russell, 2000



Results

- The four multiagent RL methods all converge to (roughly) optimal
 - All four Q functions are capable of representing the optimal policy
 - B seeks the allocation that maximizes its reward
 - It thus follows a stationary policy, with some noise

Model of B	None	Stationary Policy	Reward Subset	IRL
Msgs/decision	445	483	675	587
Msgs/cycle	306	309	1,343	560



Conclusion

- Sigma provides general support for multiagent reinforcement learning
 - Reuse the same gradient-descent mechanism
 - Change the underlying graph with different model structure of other agent
 - IRL + RL provides a novel multiagent RL
- Future work
 - Multiagent RL in both agents
 - Analyze the behaviors across all possible combinations