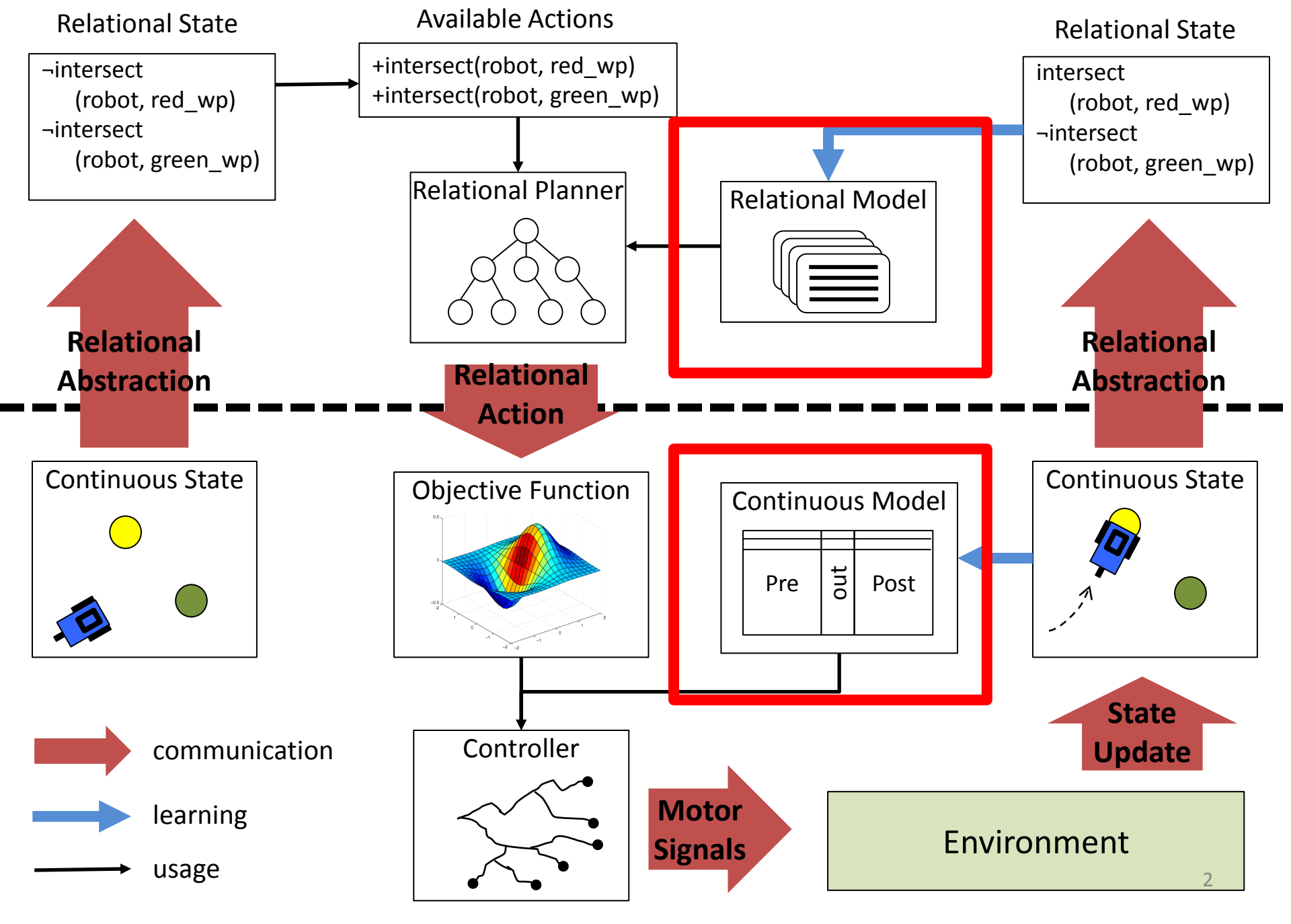
# Action Modeling in SVS

Joseph Xu

Soar Workshop 31

June 2011


Relational State
¬intersect
(robot, red_wp)
¬intersect
(robot, green_wp)
Available Actions
+intersect(robot, red_wp)
+intersect(robot, green_wp)
Relational Planner
Relational Model
Relational State
intersect
(robot, red_wp)
¬intersect
(robot, green_wp)
Relational Abstraction
Relational Action
Relational Abstraction
Continuous State
Objective Function
Continuous Model
Pre
out
Post
Continuous State
State Update
communication
learning
usage
Controller
Motor Signals
Environment
2

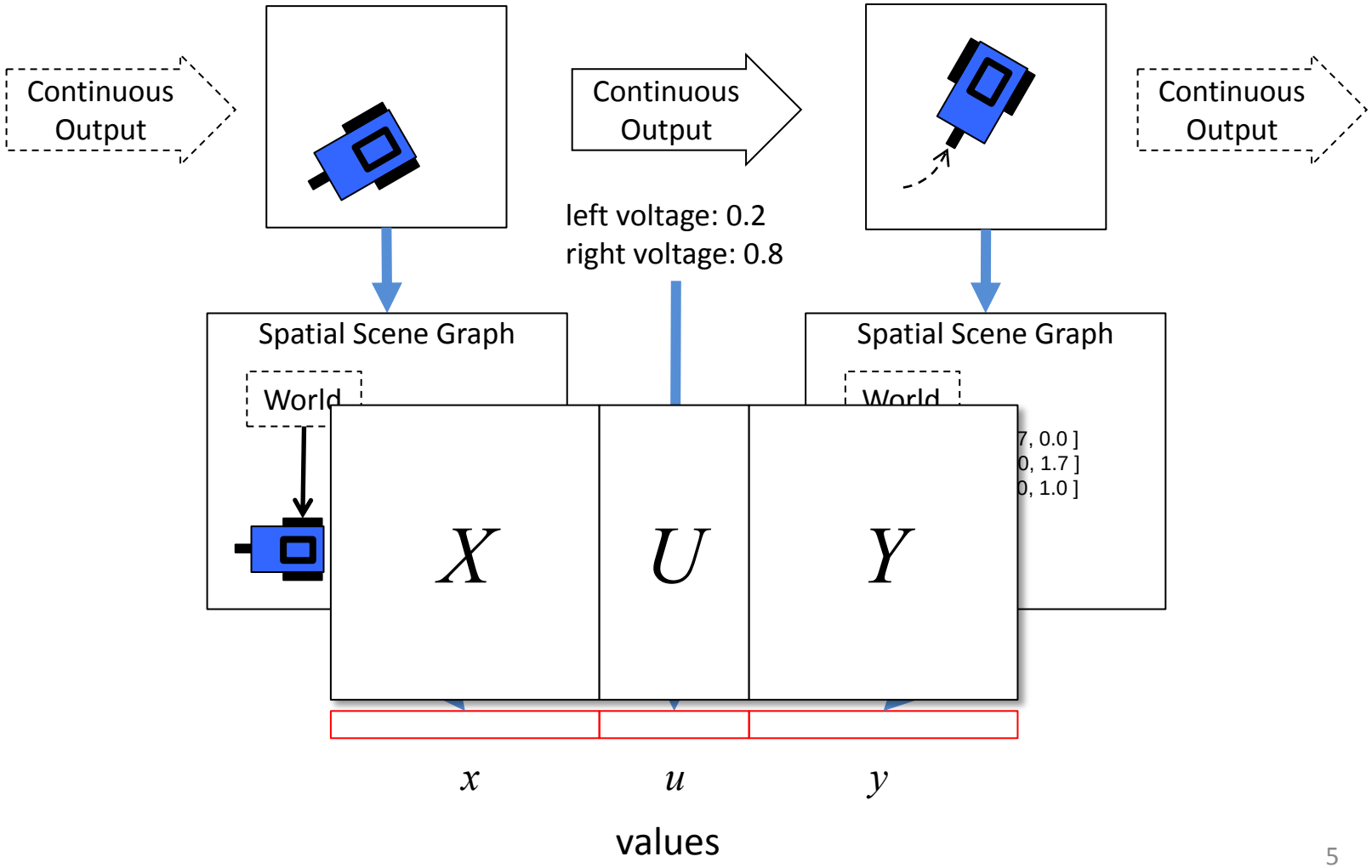
# Outline

1. Describe continuous model learning algorithm
2. Show learning results in robot domain
3. Discuss problem with algorithm
4. Briefly describe relational model learning
5. Show results on combining relational and continuous models

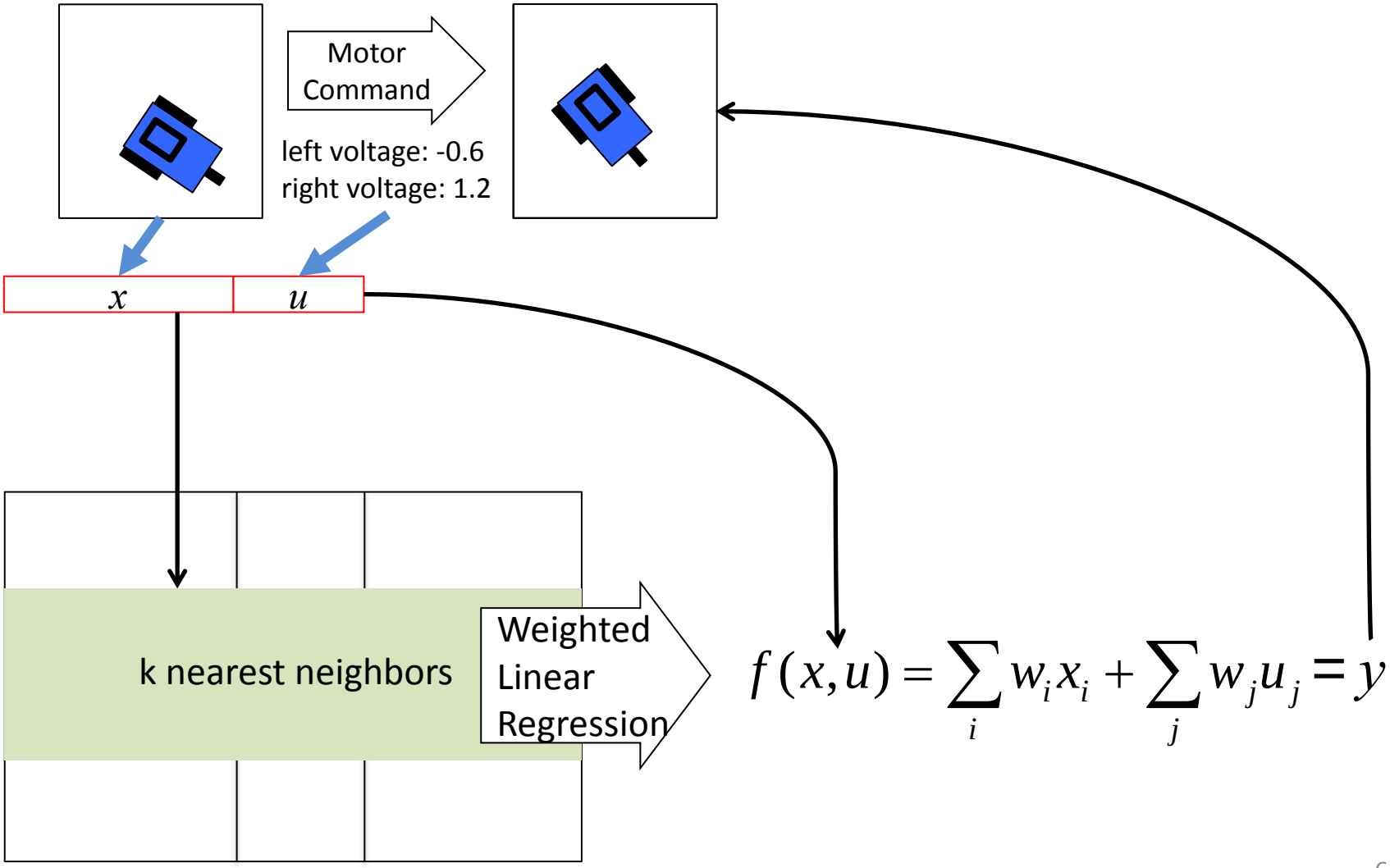
# Continuous Model Learning

- Formally, learn a function $f(x, u) \rightarrow y$
  - $x$: current continuous state
  - $u$: continuous output
  - $y$: next state
- Requirements
  - Online, incremental
  - Robust across wide variety of domains
  - Doesn't slow down with more training
- Locally Weighted Regression
  - Instance-based approach (!)
  - Almost parameter free
  - Learns arbitrarily complex functions

# Continuous Model Learning

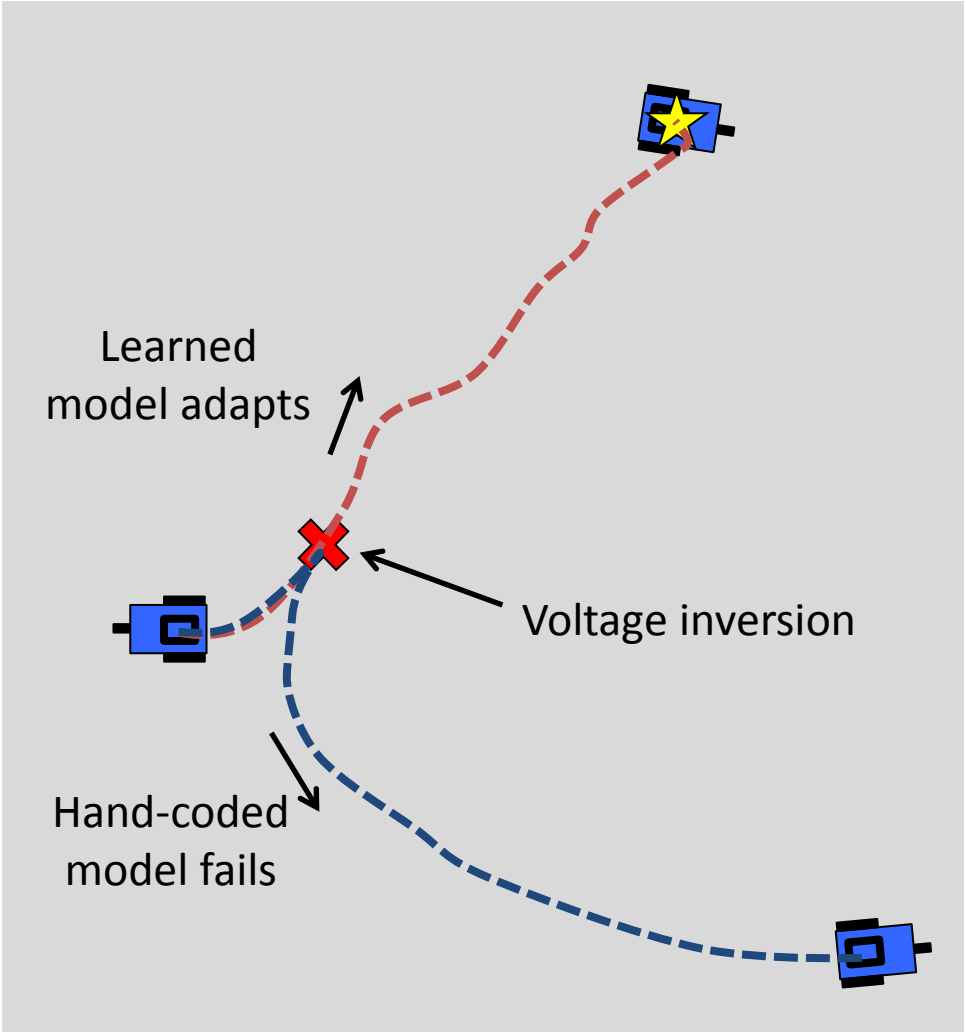


# Continuous Model Prediction

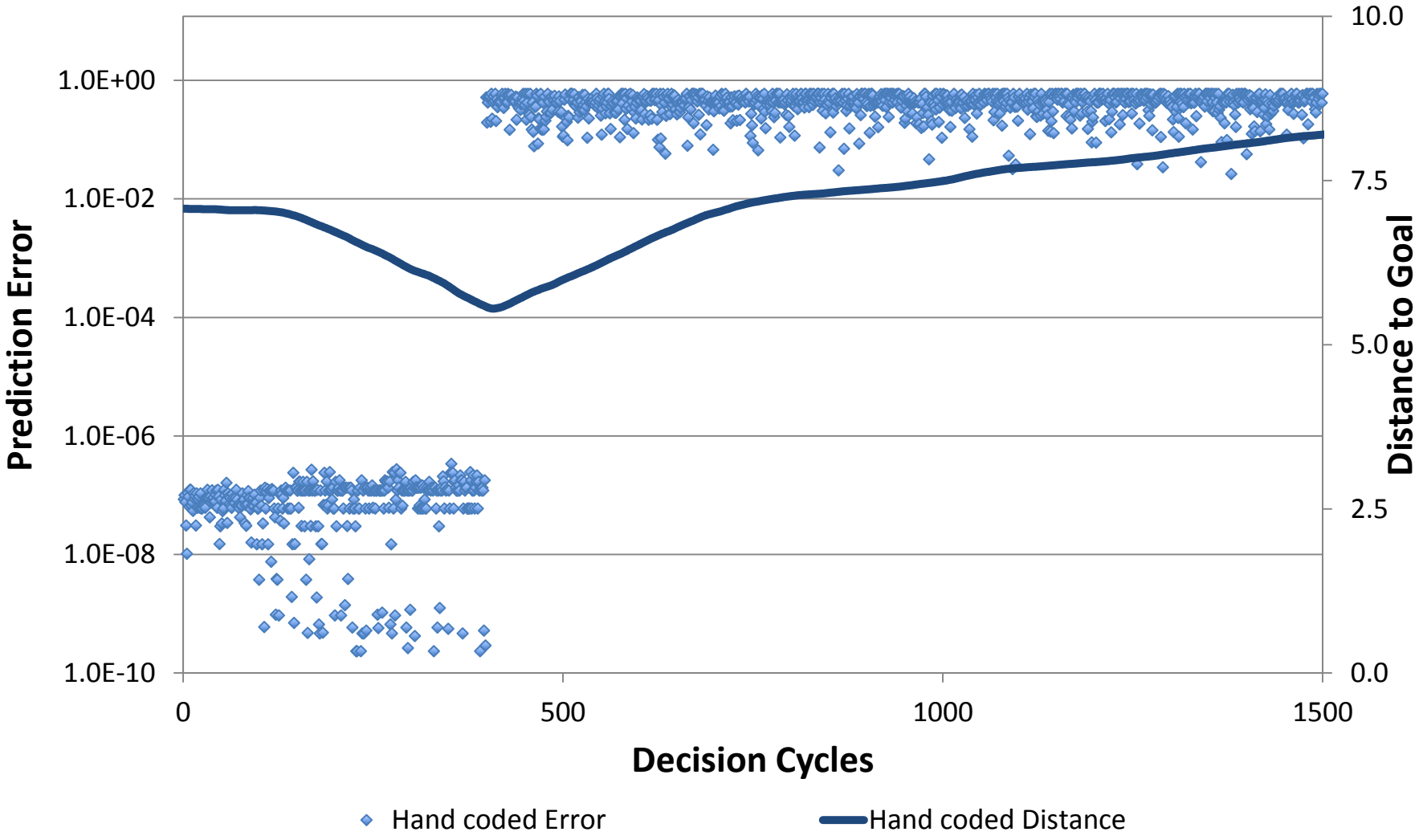


# Robot Domain Example

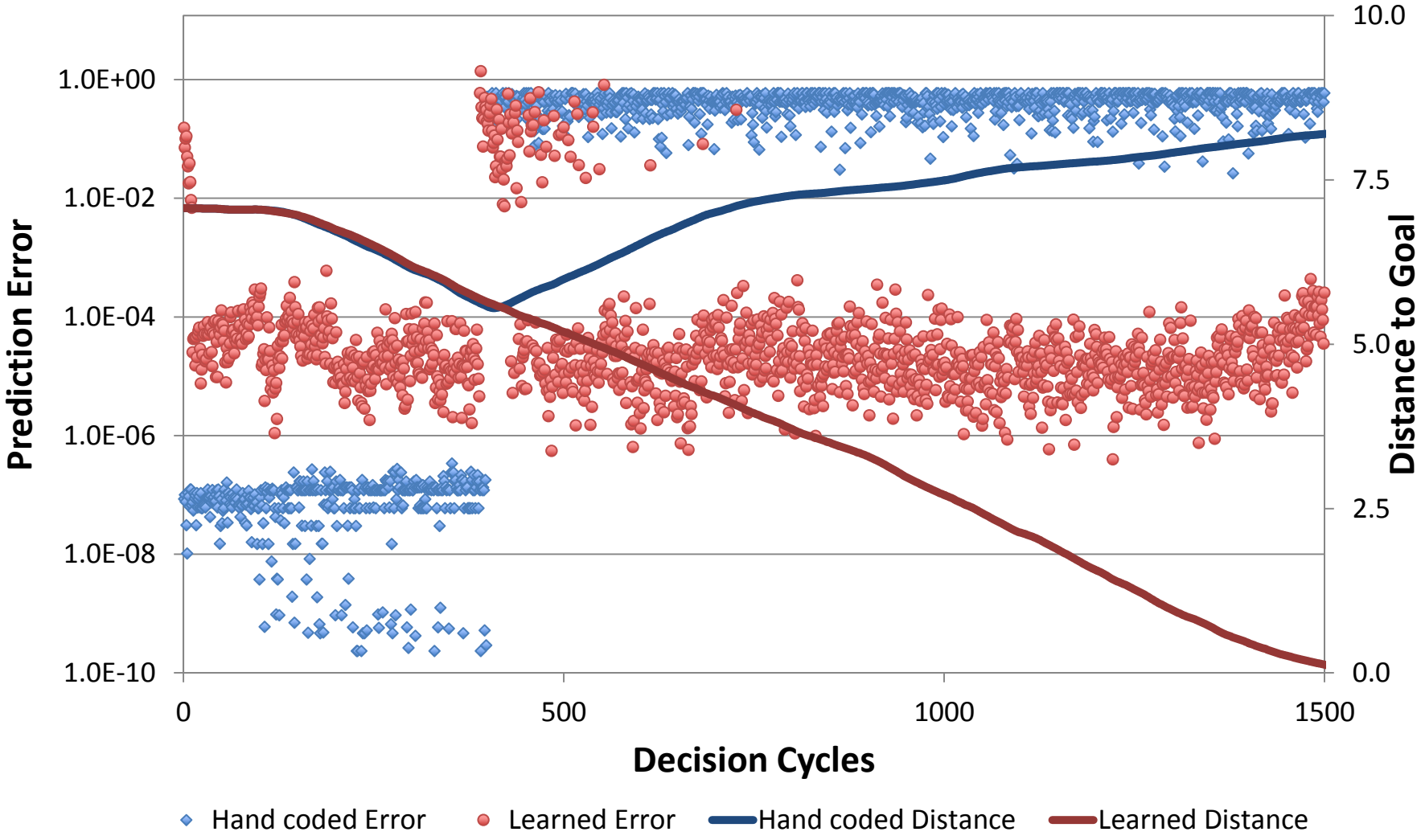
- Environment input
  - (x, y) position
  - (x, y) velocity
  - Heading angle
  - Angular velocity
  - Left wheel RPM
  - Right wheel RPM
- Agent output
  - Left and right motor voltages
- Using a stand-alone version



# Continuous Learning Performance in Robot Domain



# Continuous Learning Performance in Robot Domain



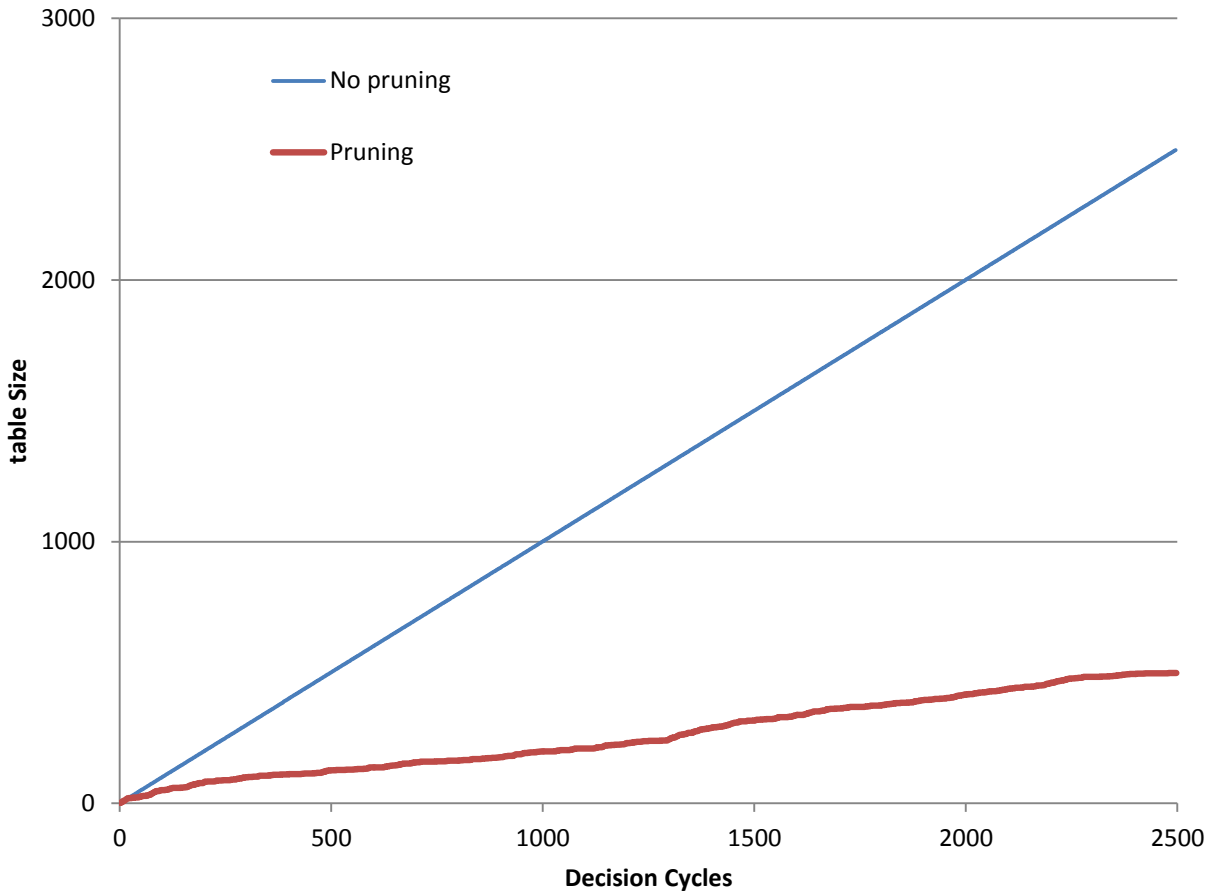
# Problem: Nearest Neighbor Search

- Slowest part of the prediction
- Using a brute force search over all training examples leads to linear time increase
- Smarter data structures such as Kd trees, ball trees, and cover trees degrade quickly as number of dimensions increase
  - Number of dimensions proportional to number of objects in scene, can grow very large
- Locality sensitive hashing?

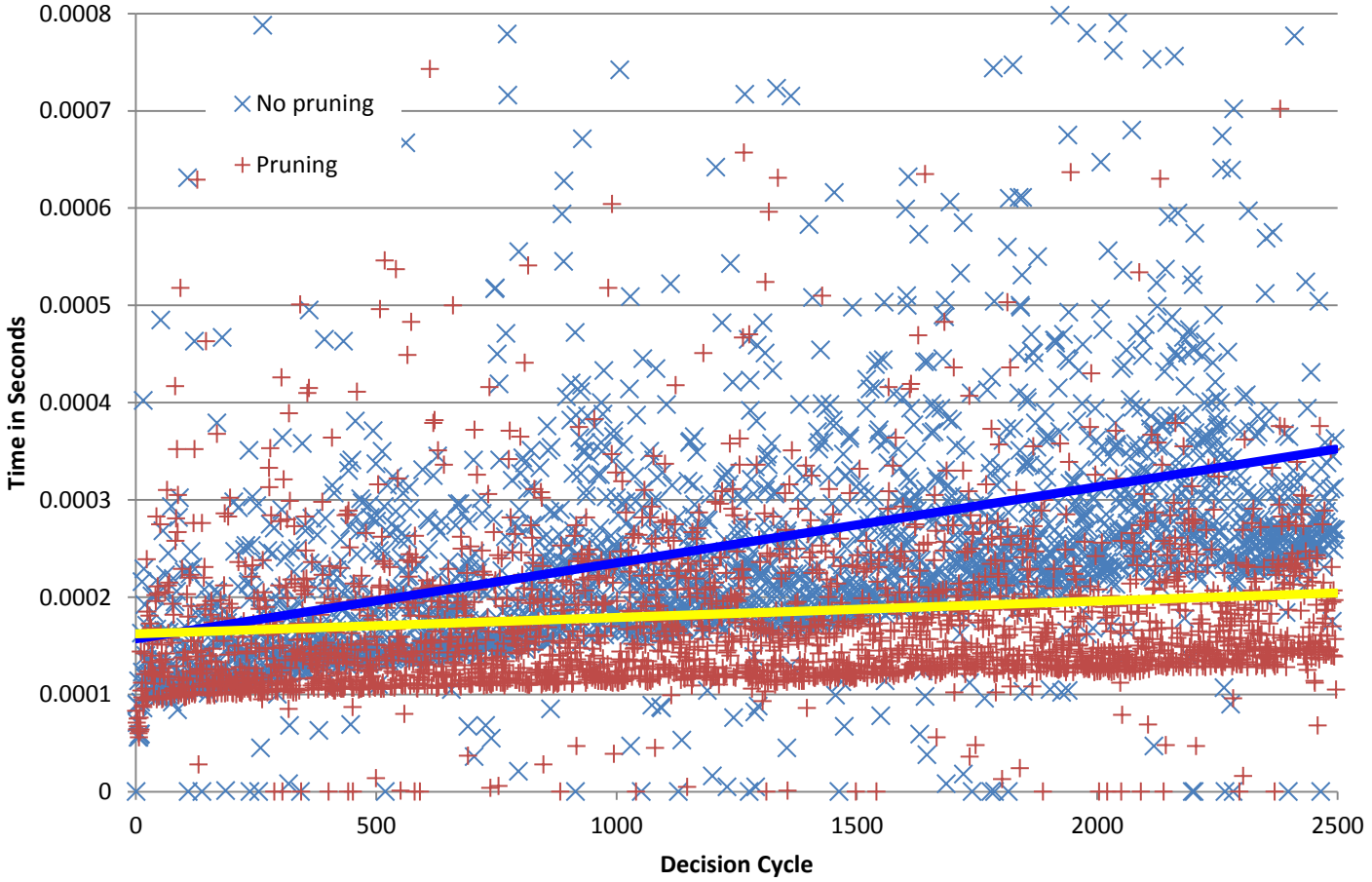
# Pruning Examples

- Selectively keep new training examples:
  - For each new example (x, u, y), make a prediction $y' = f(x, u)$ with continuous model
  - If $y' \approx y$, then assume that (x, u, y) will not improve model performance, so don't keep it

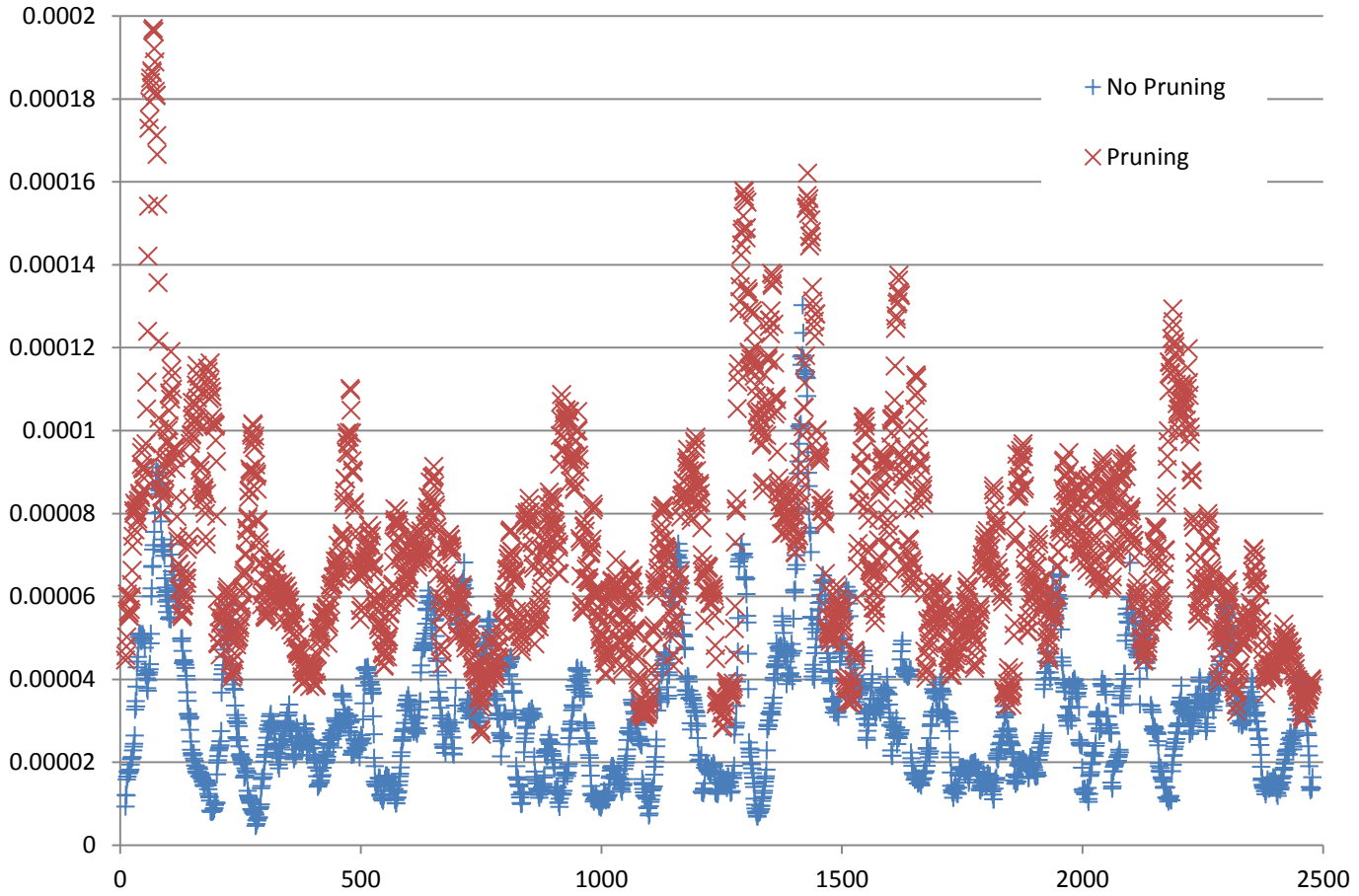
# Table Size w/wo Pruning



# Query Time w/wo Pruning

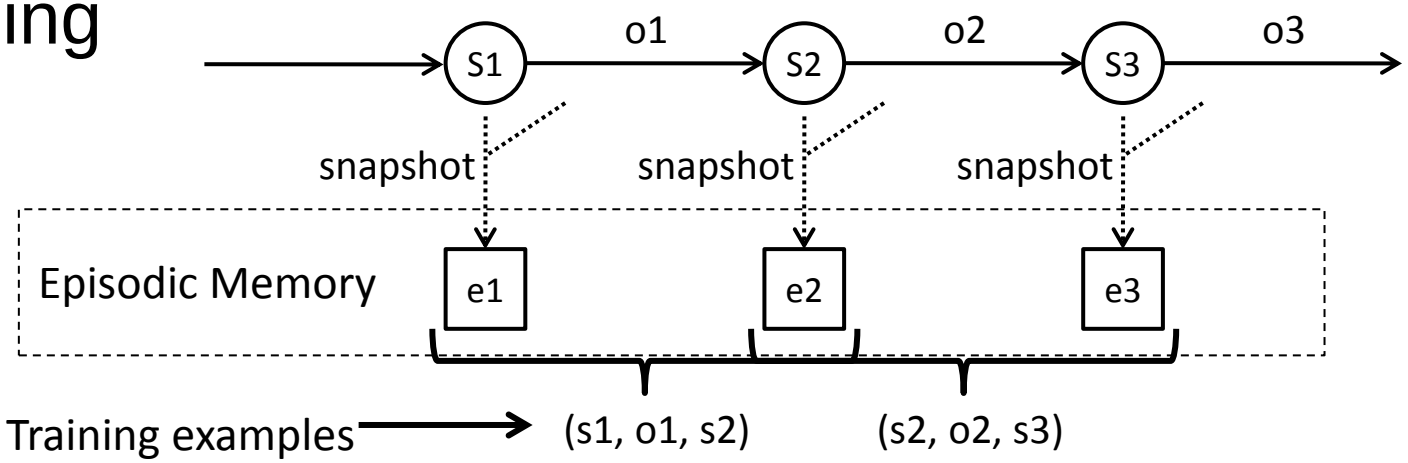


# Prediction Error w/wo Pruning

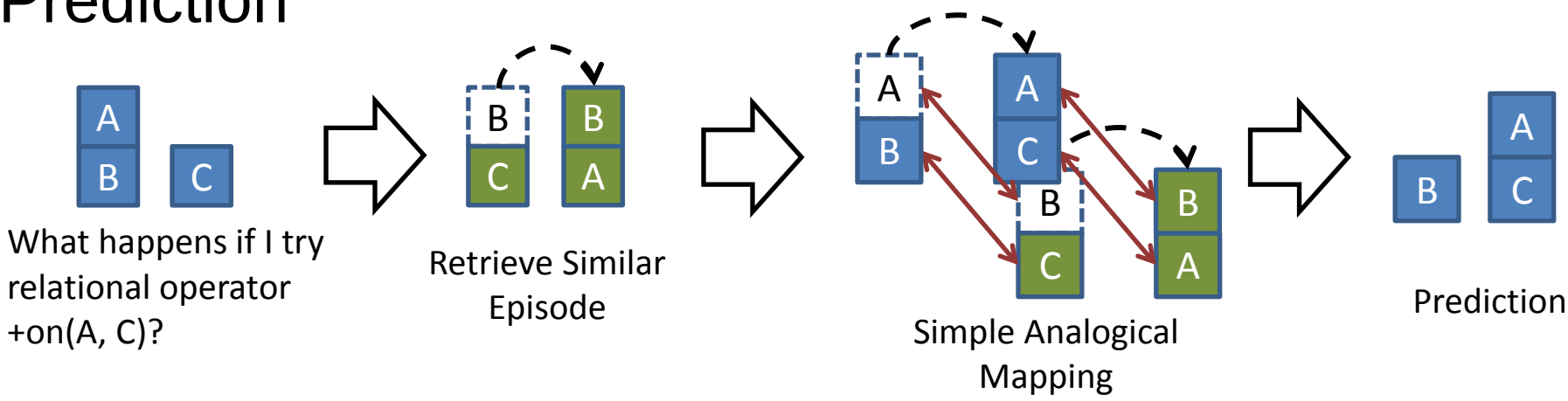


# Relational Model Learning

## Learning



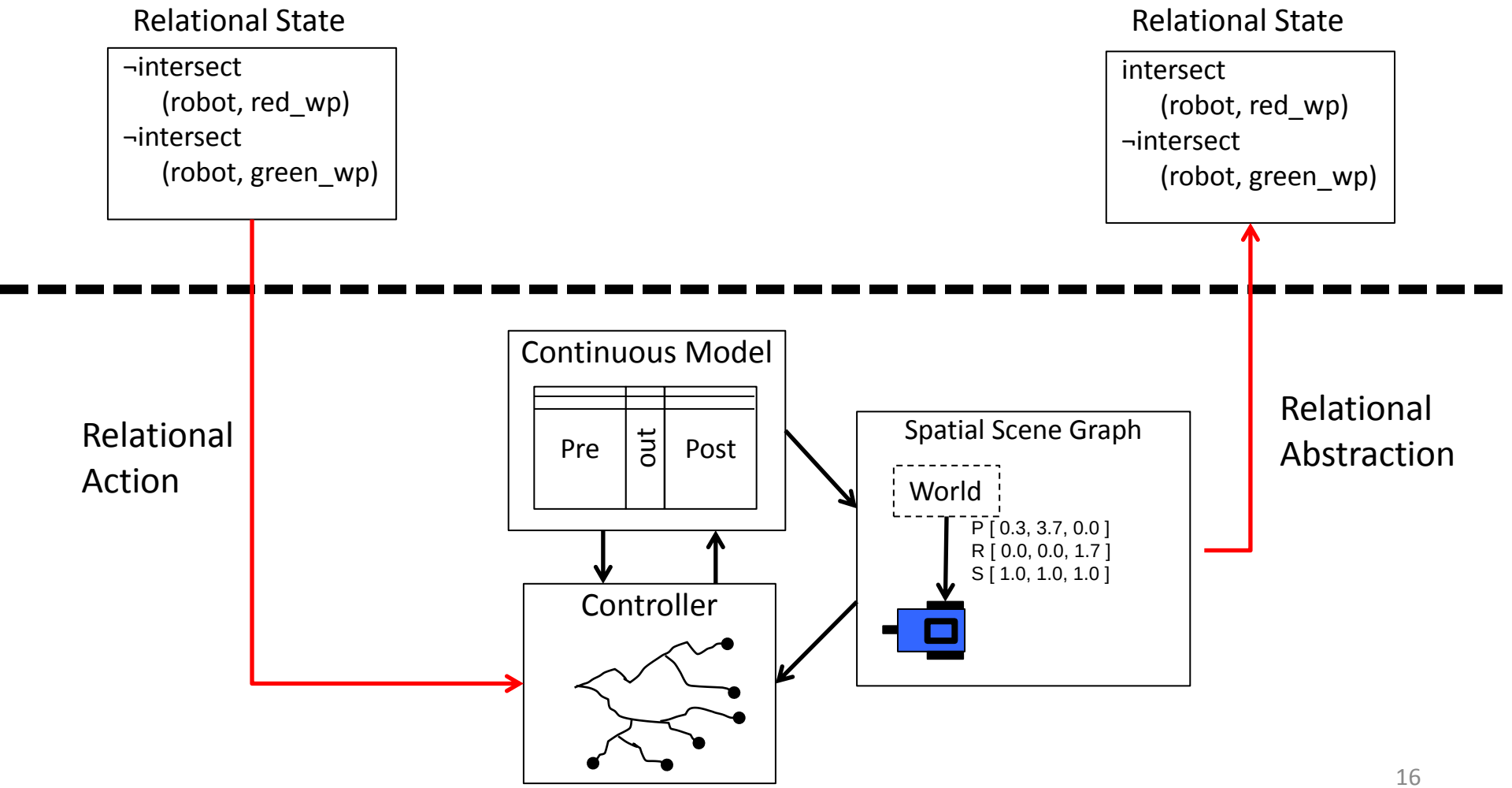

## Prediction






Xu, J. Z., Laird, J. E. (2010). *Instance-Based Online Learning of Deterministic Relational Action Models.* AAAI 2010.

# Another Way to Make Relational Predictions



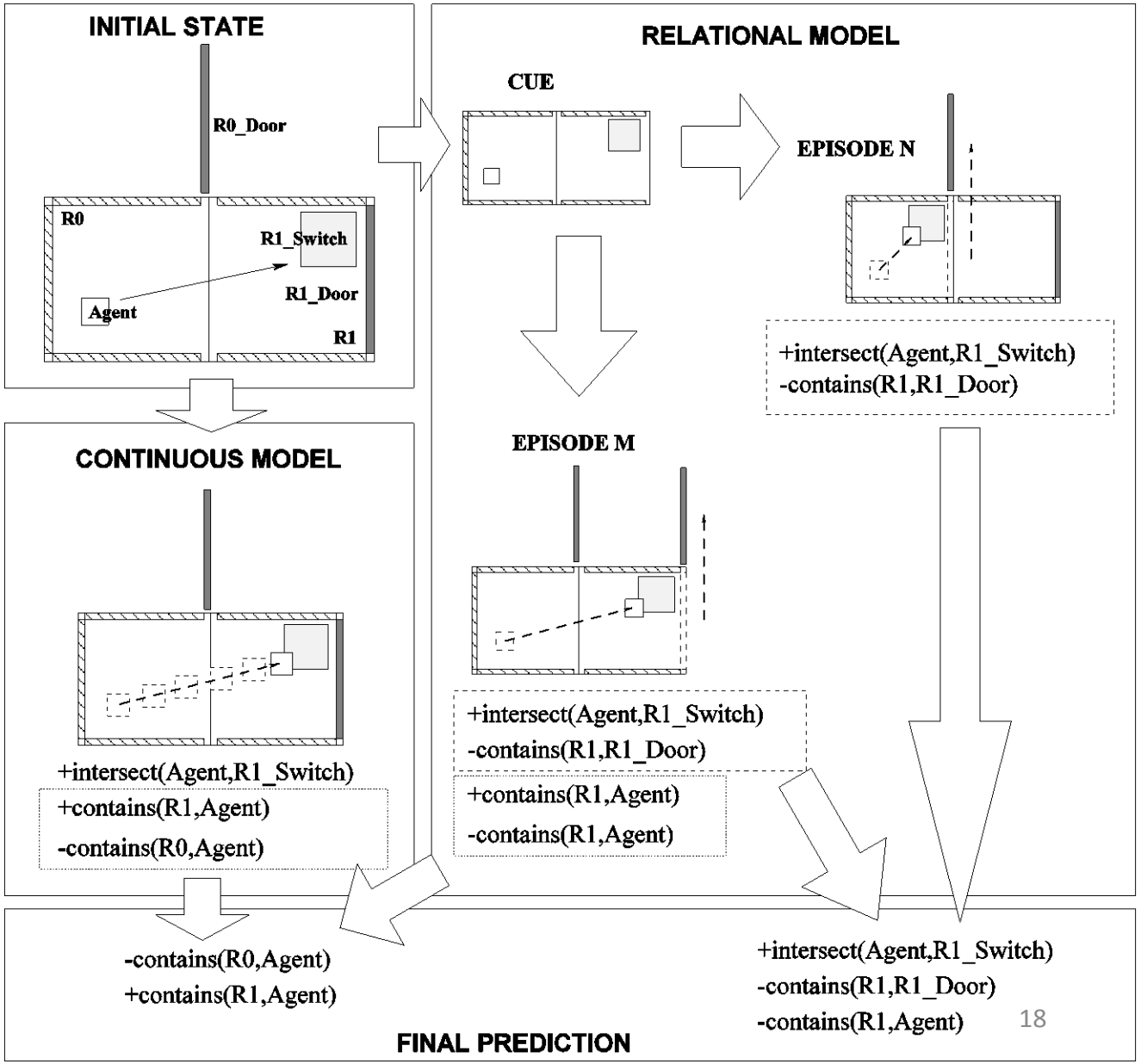
# Combining Relational and Continuous Models

- Each type of model captures different information
  - Relational model captures qualitative changes that can generalize across analogical situations, but doesn't understand space
  - Continuous model captures spatial configurations, but doesn't generalize across analogical situations
- Since we have both, we can combine them to offset each type's weakness

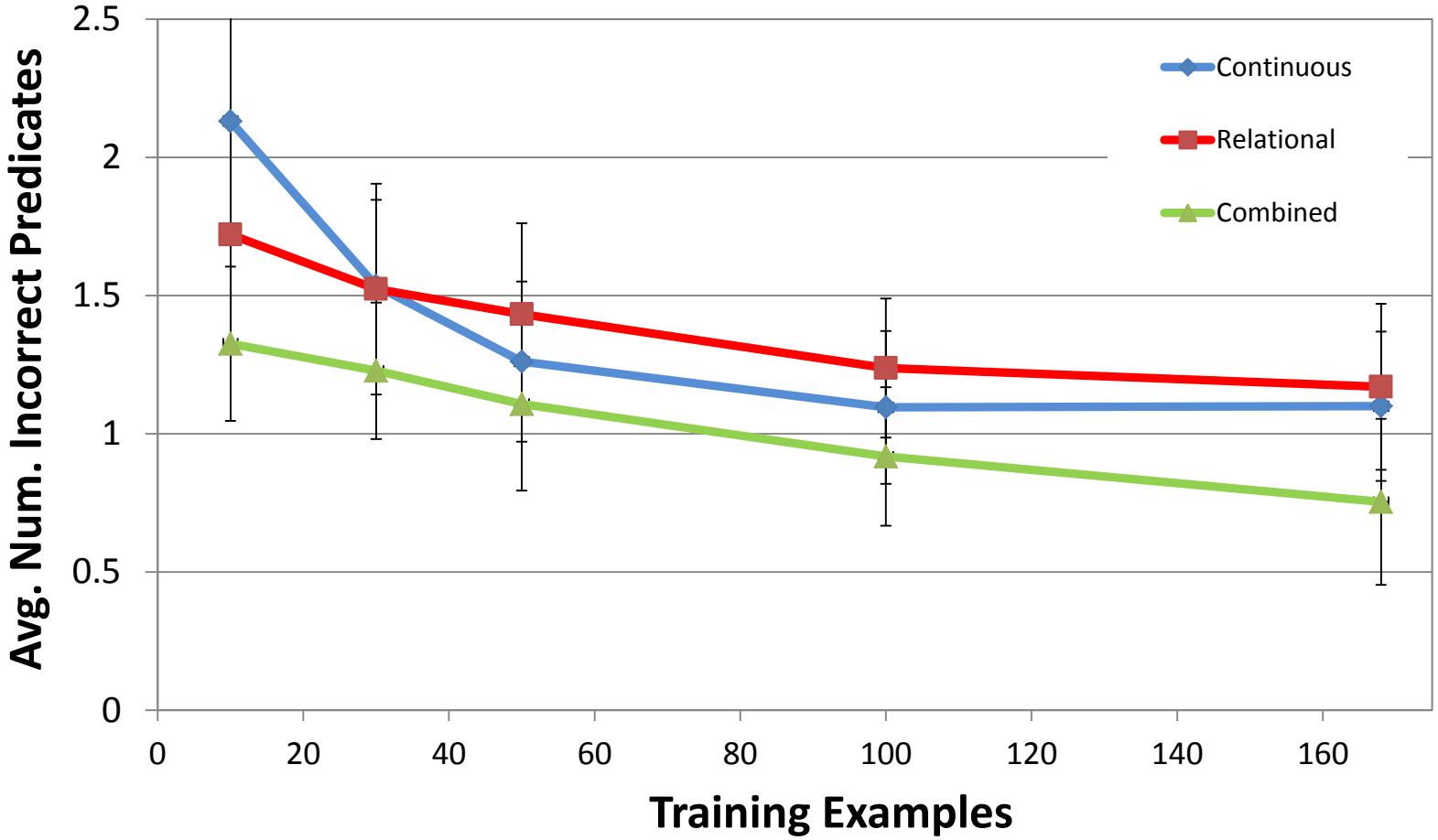
# Combining Relational and Continuous Models

Rooms Domain

- Series of connected rooms with doors and switches
- Agent outputs (x,y) velocity to move and can open doors by stepping on switches
- Relational model doesn't capture spatial configuration of rooms
- Continuous model doesn't understand the relationship between stepping on switches and opening doors



# Prediction Performance in Rooms Domain






Xu, J. Z., Laird, J. E. (2011). *Combining Learned Discrete and Continuous Action Models.* AAAI 2011.

# Conclusion

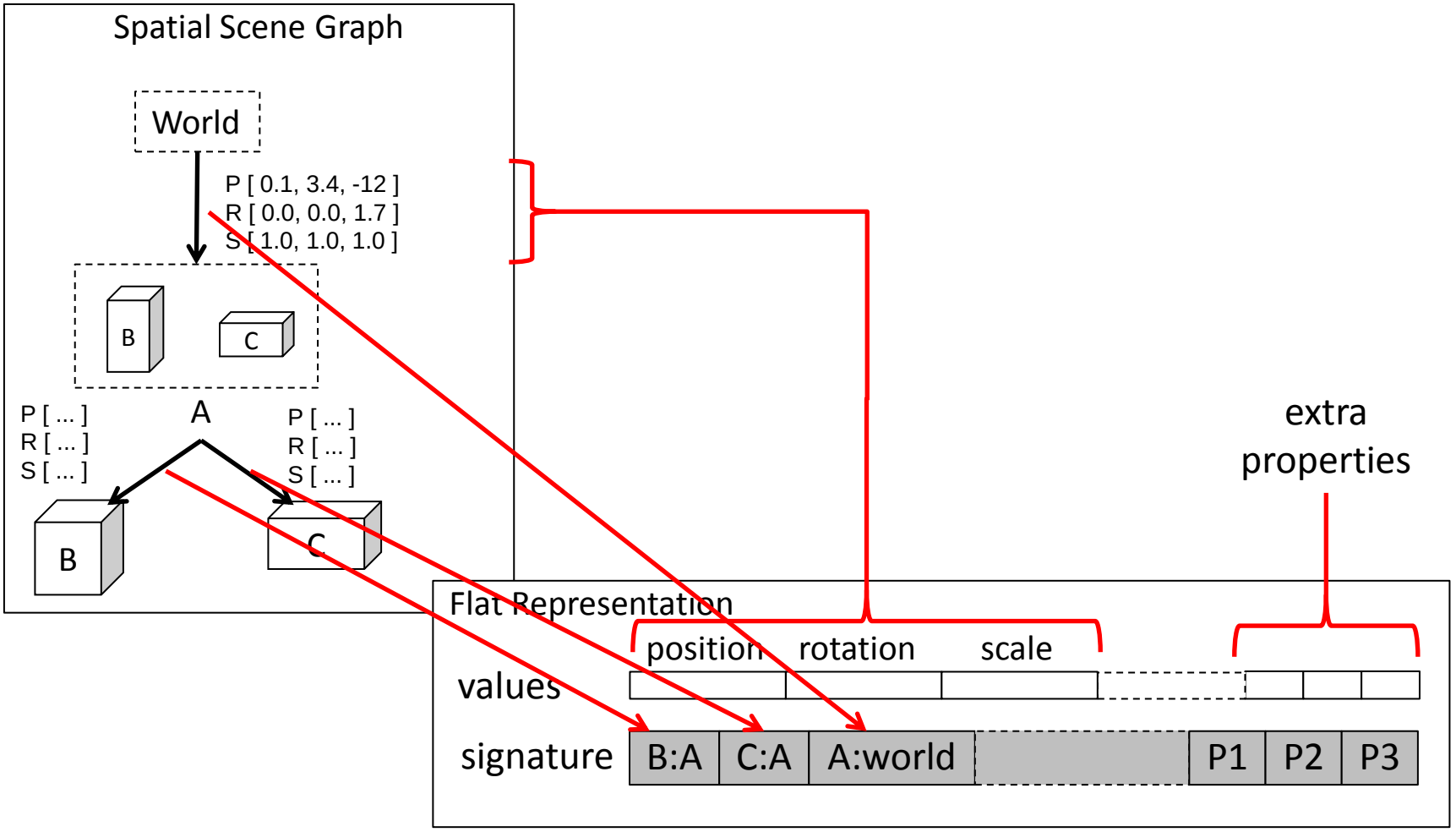
**Nuggets**

- Continuous model learns quickly in robot domain
- Continuous model adapts online to changes in dynamics
- Demonstrated a simple way to combine relational and continuous learning to improve performance

**Coal**

- Nearest neighbor search is unbounded

# Flattening Scene Graph



# Problem: Sharing Training Examples

- Current implementation of continuous model learning cannot share training examples across structurally distinct states