Value Iteration Networks

NIPS 2016 BEST PAPER

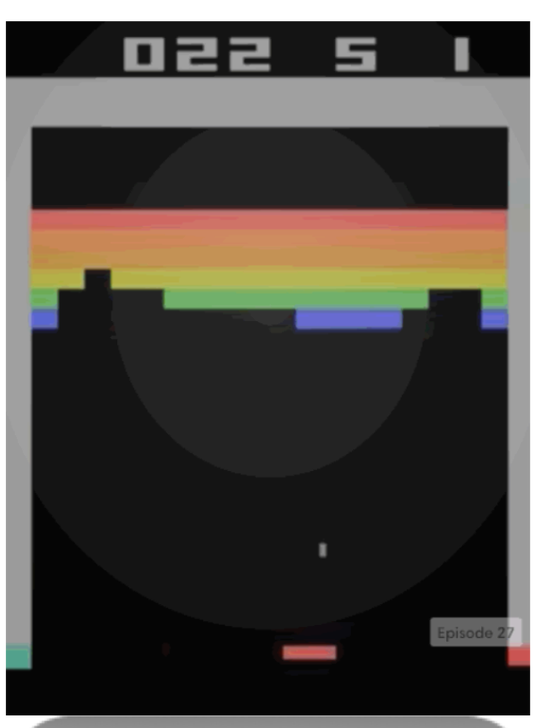
Aviv Tamar, Yi Wu, Garrett Thomas, Sergey Levine, and Pieter Abbeel

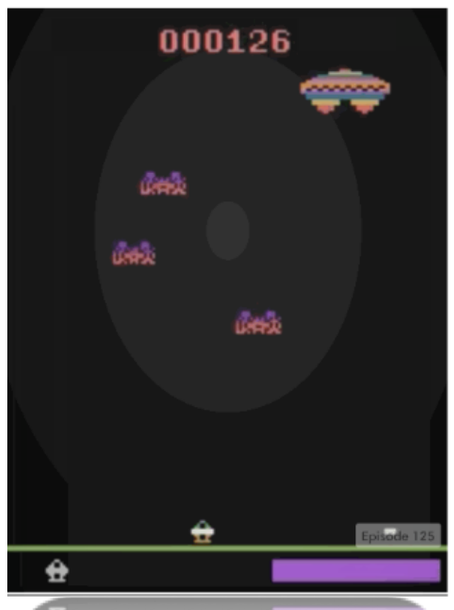
@ Berkeley Artificial Intelligence Research Lab (BAIR)

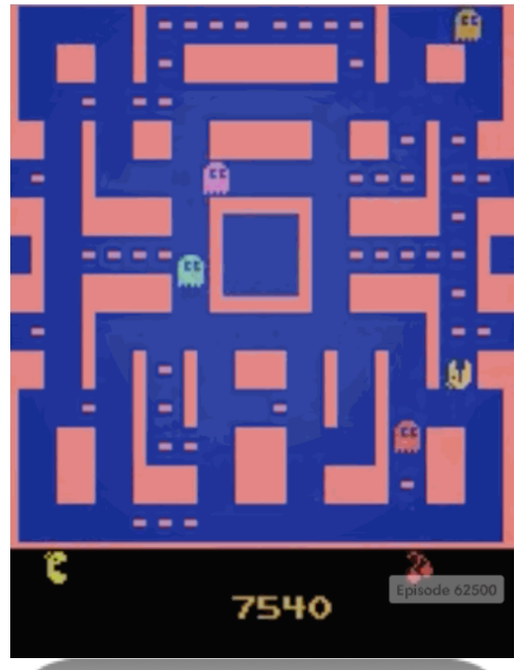
Runzhe Yang @ SJTU ACM CLASS

Introduction

- Deep RL learns policies from complicated visual input
- Learns to act, but does it **understand**?
- A simple test: generalization on grid worlds



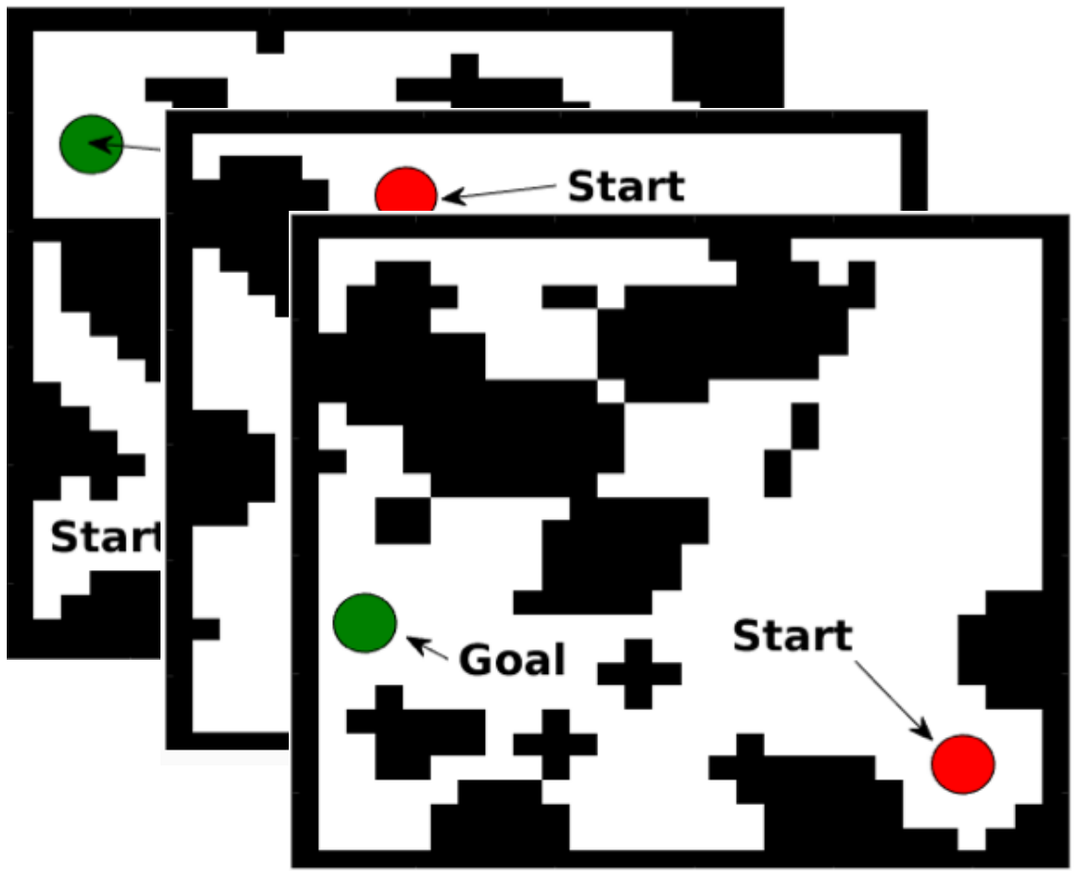




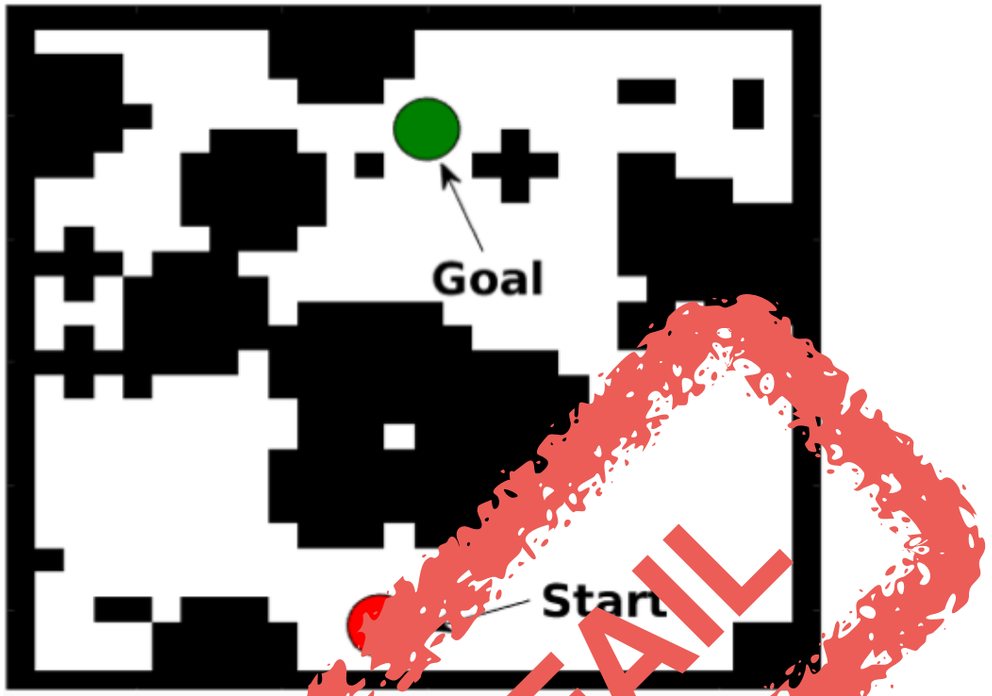
Introduction

- A simple test: generalization on grid worlds

Train



Test



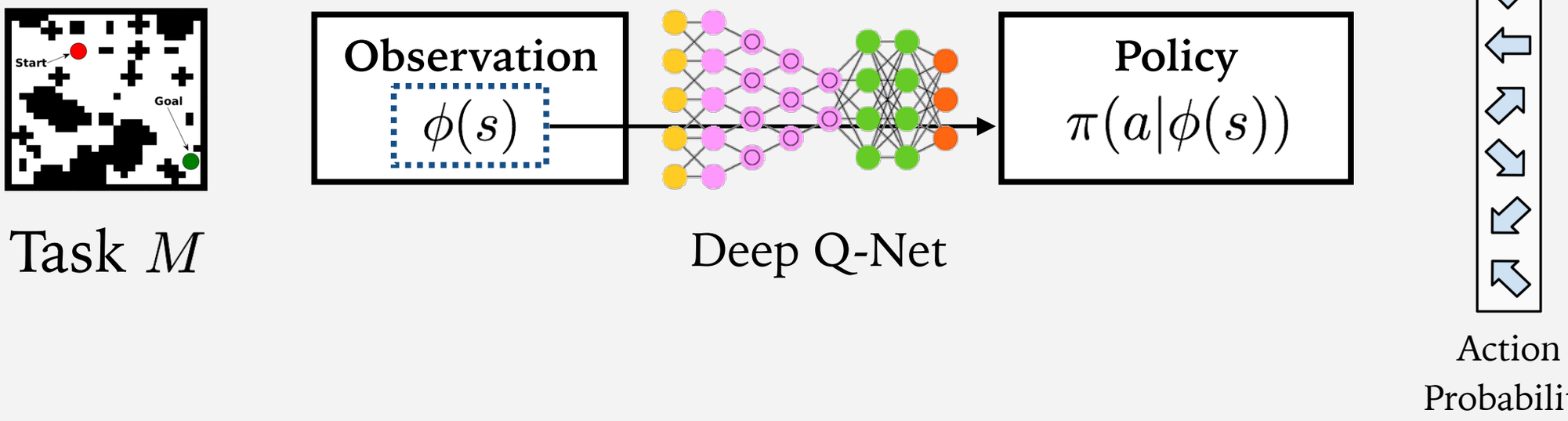
Introduction

Why doesn't it **understand**?

Runzhe Yang @ SJTU ACM CLASS

Introduction

- A neural network (NN) is trained to represent a policy

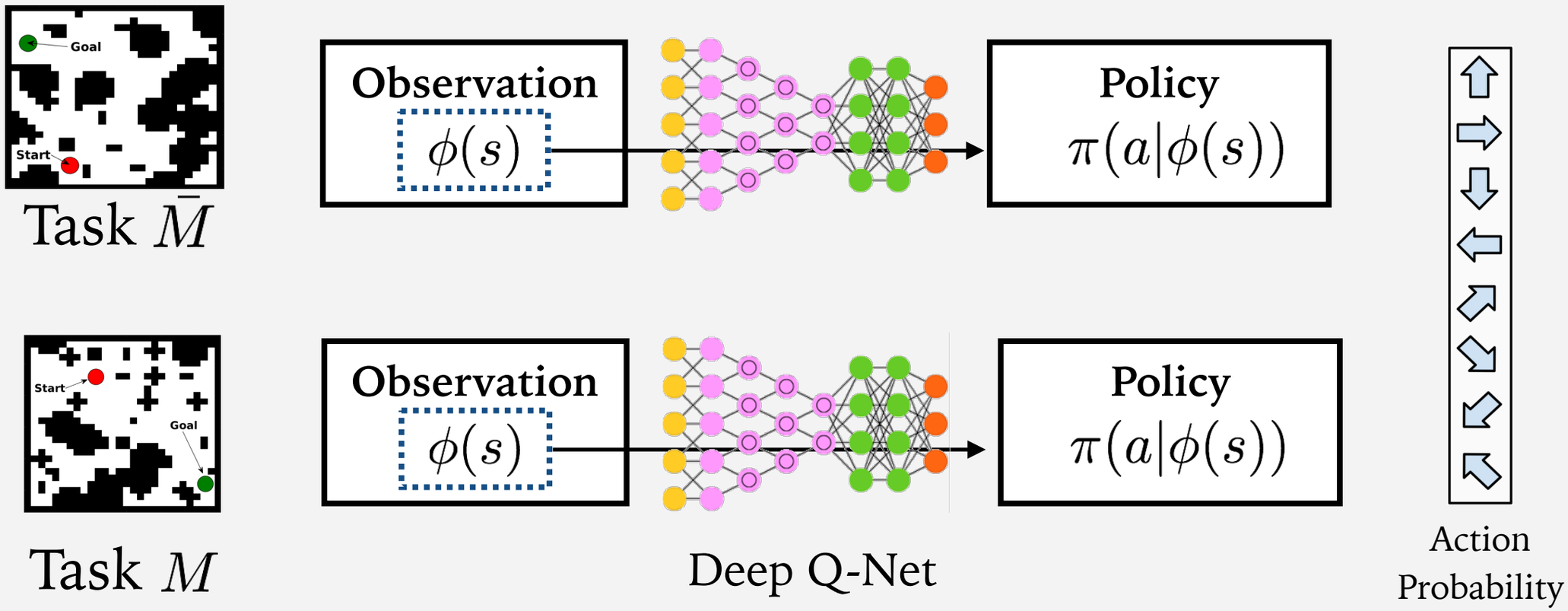


Why doesn't it **understand**?

Introduction

- A neural network (NN) is trained to represent a policy

- New task → need to re-plan

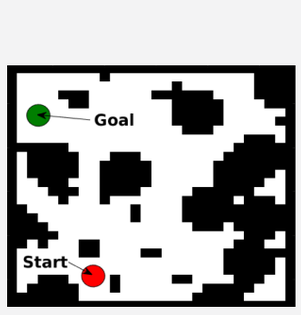


Why doesn't it **understand**?

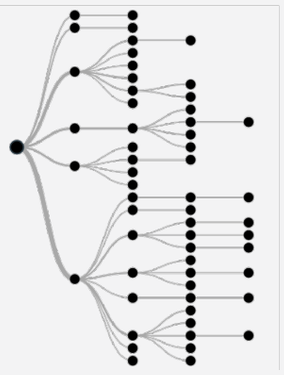
Introduction

Why doesn't it **understand**?

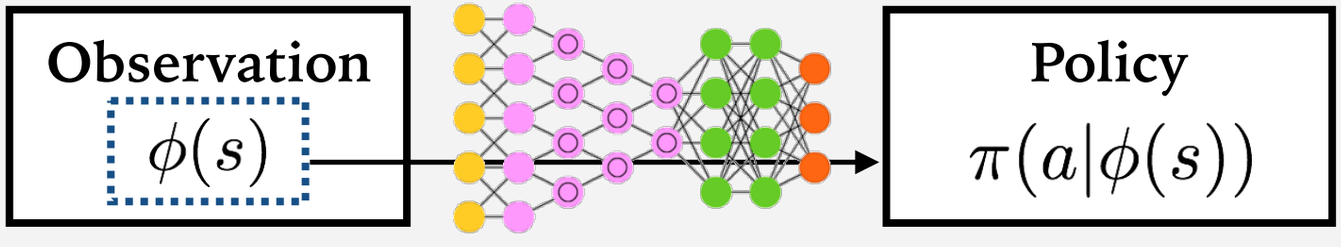
- A sequential problem requires a **planning** computation



Task $\bar{M}$



Sequential Nature

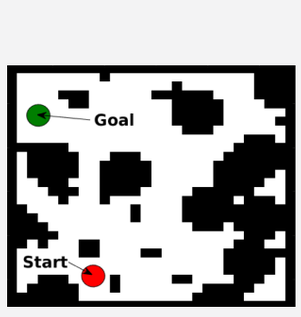


Deep Q-Net

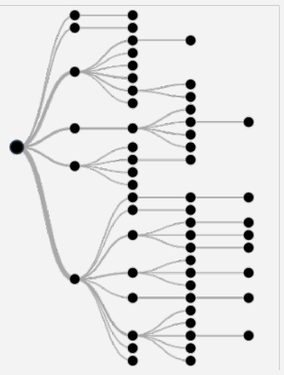
Introduction

Why doesn't it **understand**?

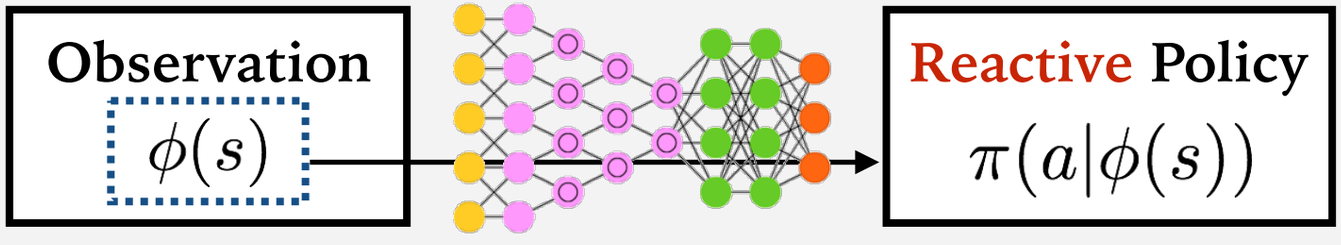
- A sequential problem requires a **planning** computation
- RL gets around that (learns a **mapping**, State → Q-value)
- **Lack of planning computation ⇒ bad understanding**



Task $\bar{M}$



Sequential Nature



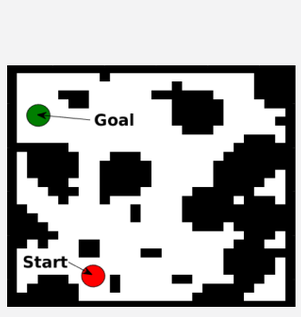
Deep Q-Net

Introduction

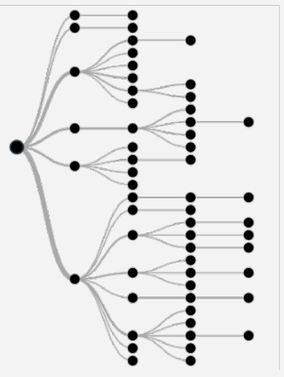
In this work:

- Learn to plan

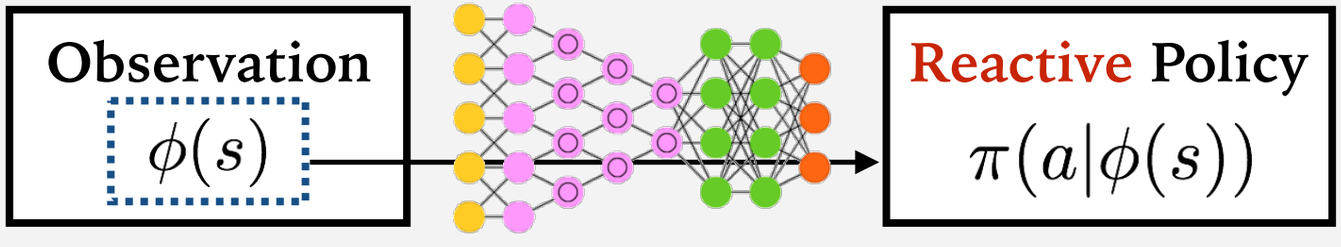
- Policies that generalize to unseen tasks



Task $\bar{M}$



Sequential Nature



Deep Q-Net

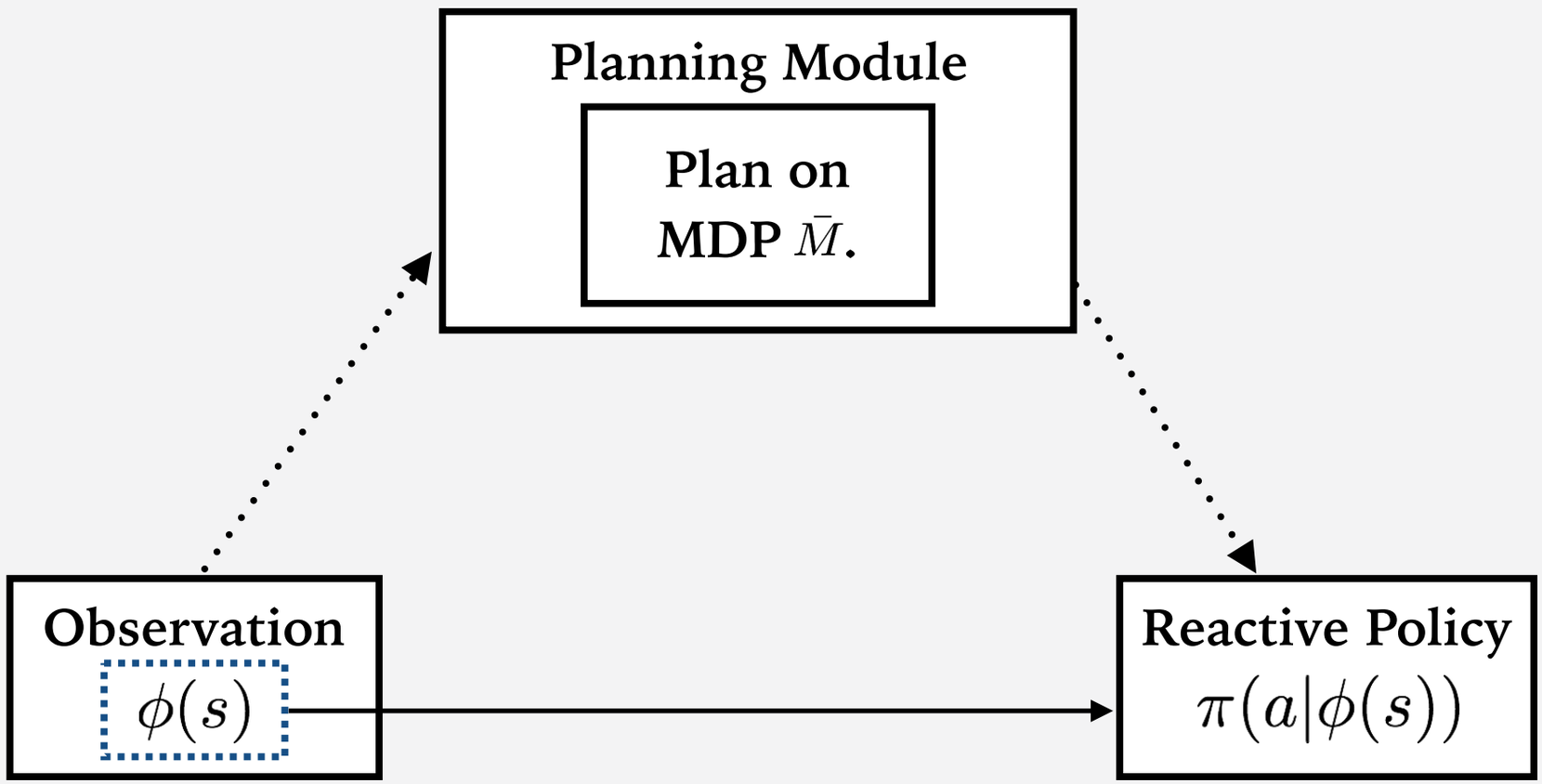
A Planning-based Policy Model

- Start from reactive policy

A Planning-based Policy Model

- Add an explicit planning computation

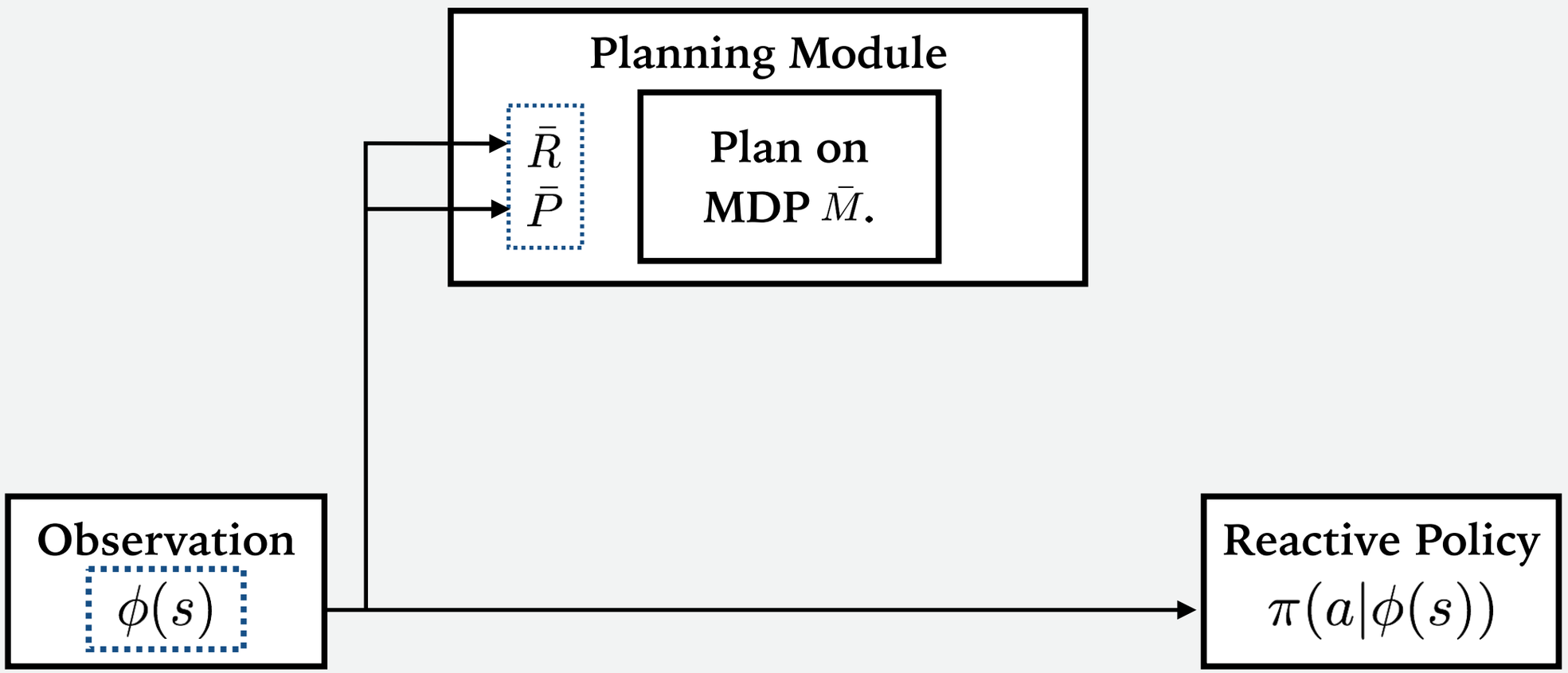
- Assumption: observation can be mapped to a useful (but **unknown**) planning computation



A Planning-based Policy Model

- NNs map observation to reward and transitions

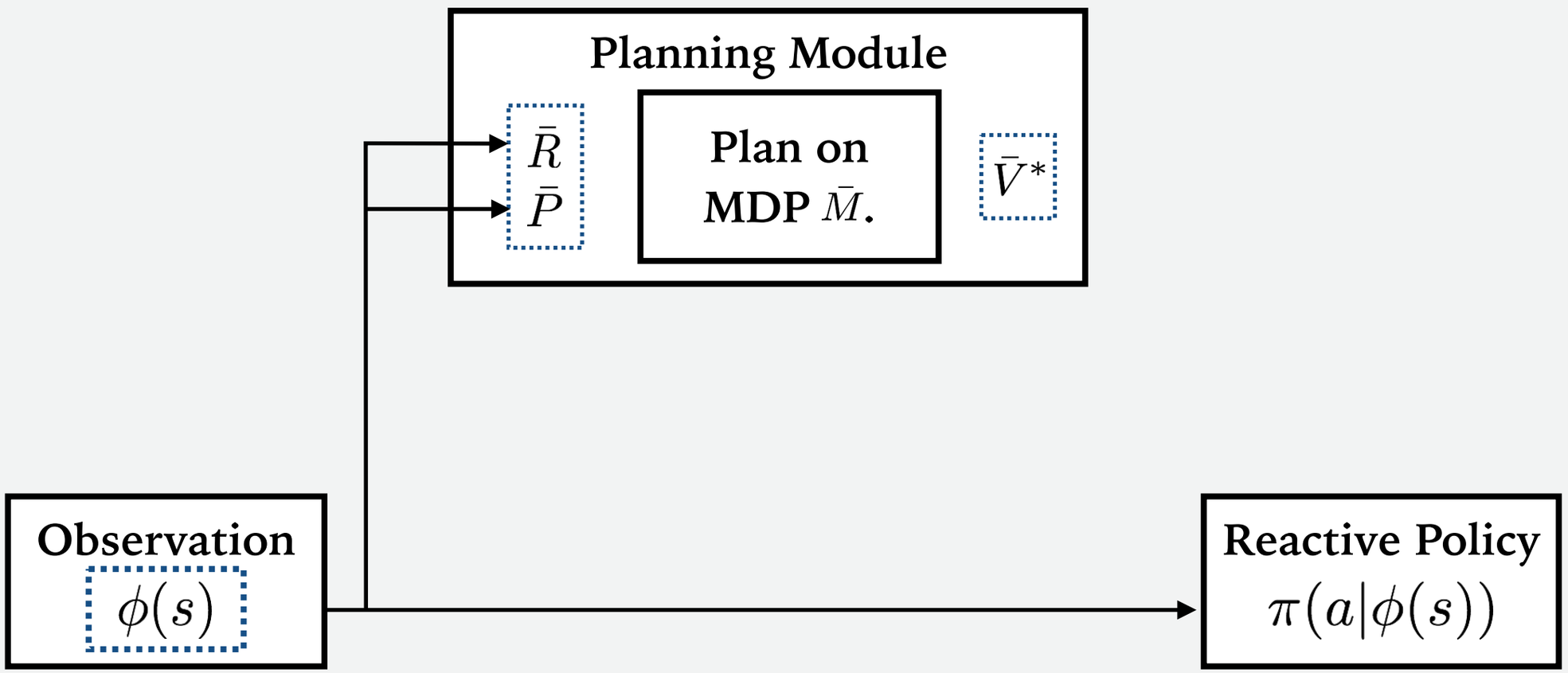
- Later, learn on new MDP



- How to use the planning computation?

A Planning-based Policy Model

- Fact 1: value function = sufficient information about plan

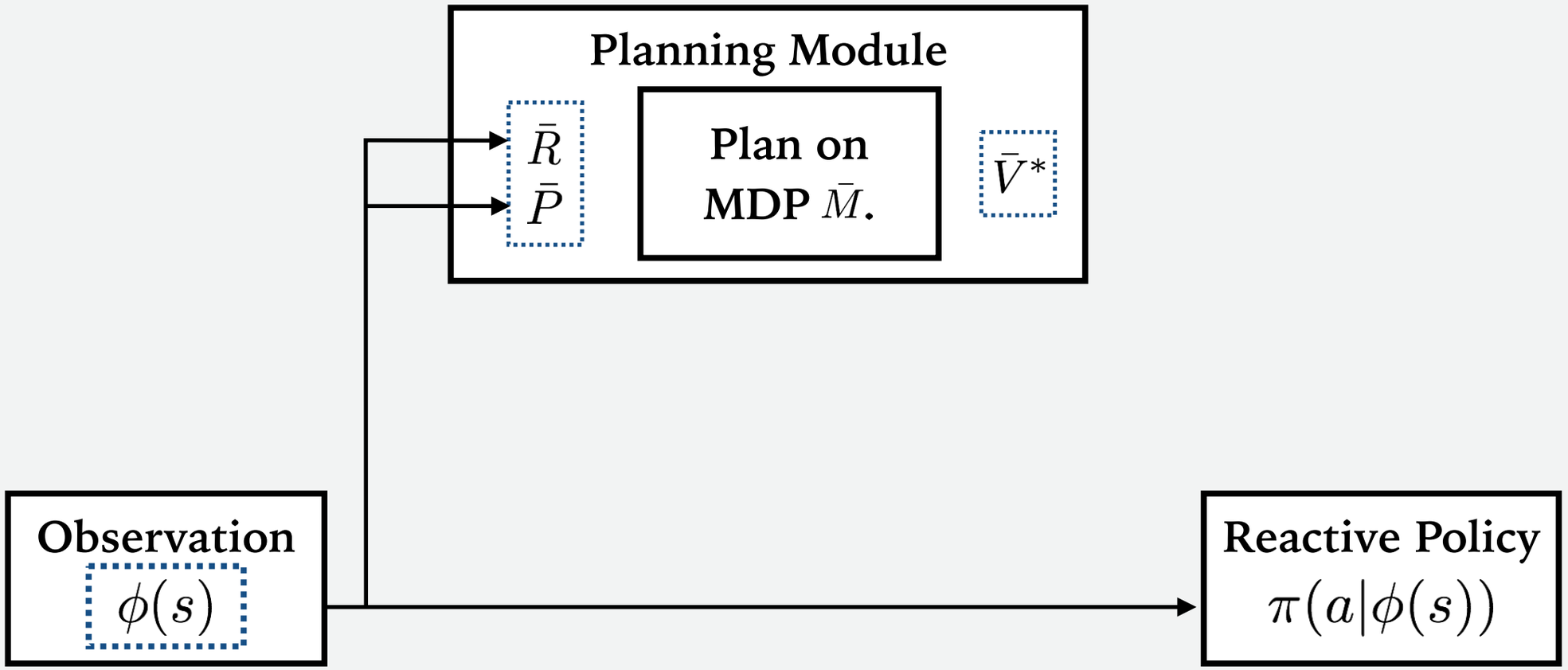


A Planning-based Policy Model

- Fact 1: value function = sufficient information about plan

- Fact 2: action prediction can require only subset of $\bar{V}^*$

$$\pi^*(a|s) = \arg\max_a R(s,a) + \gamma \sum_{s'} P(s'|s,a) V^*(s')$$

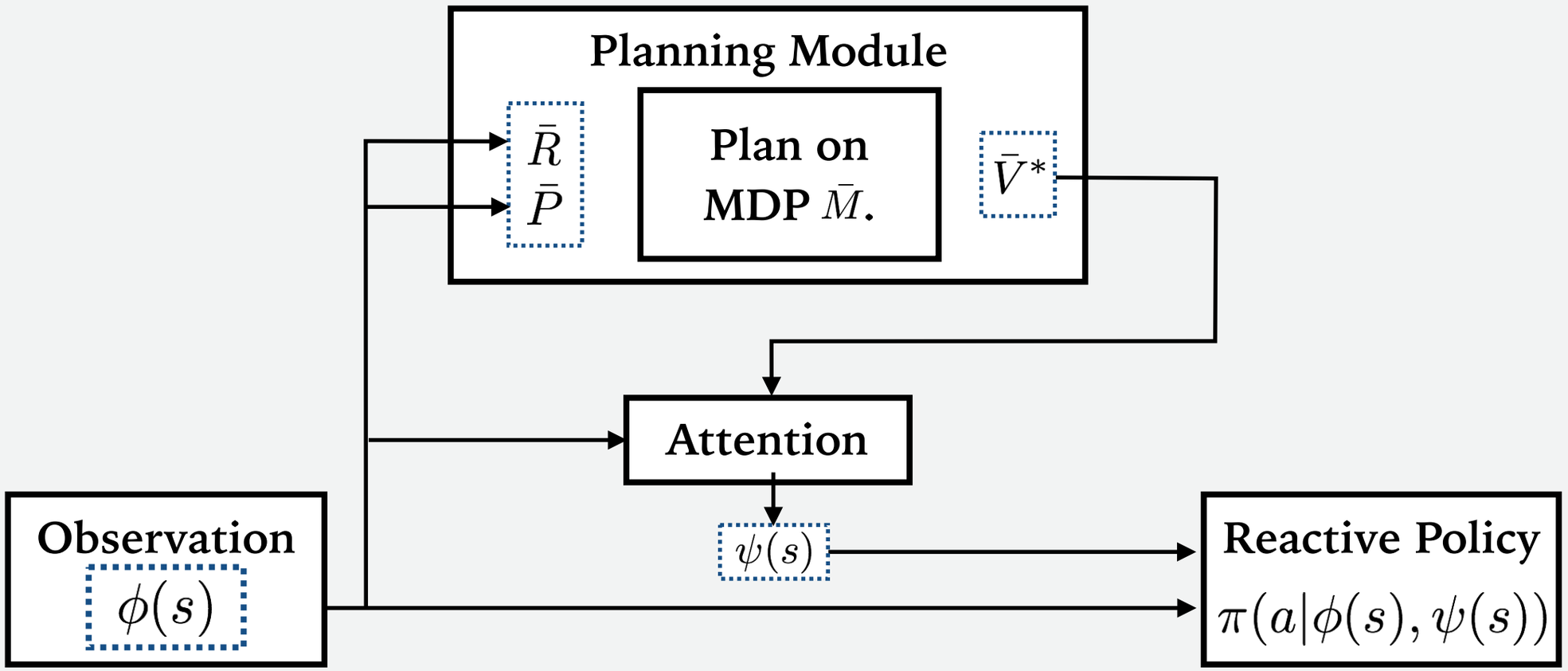


A Planning-based Policy Model

- Fact 1: value function = sufficient information about plan

- Fact 2: action prediction can require only subset of $\bar{V}^*$

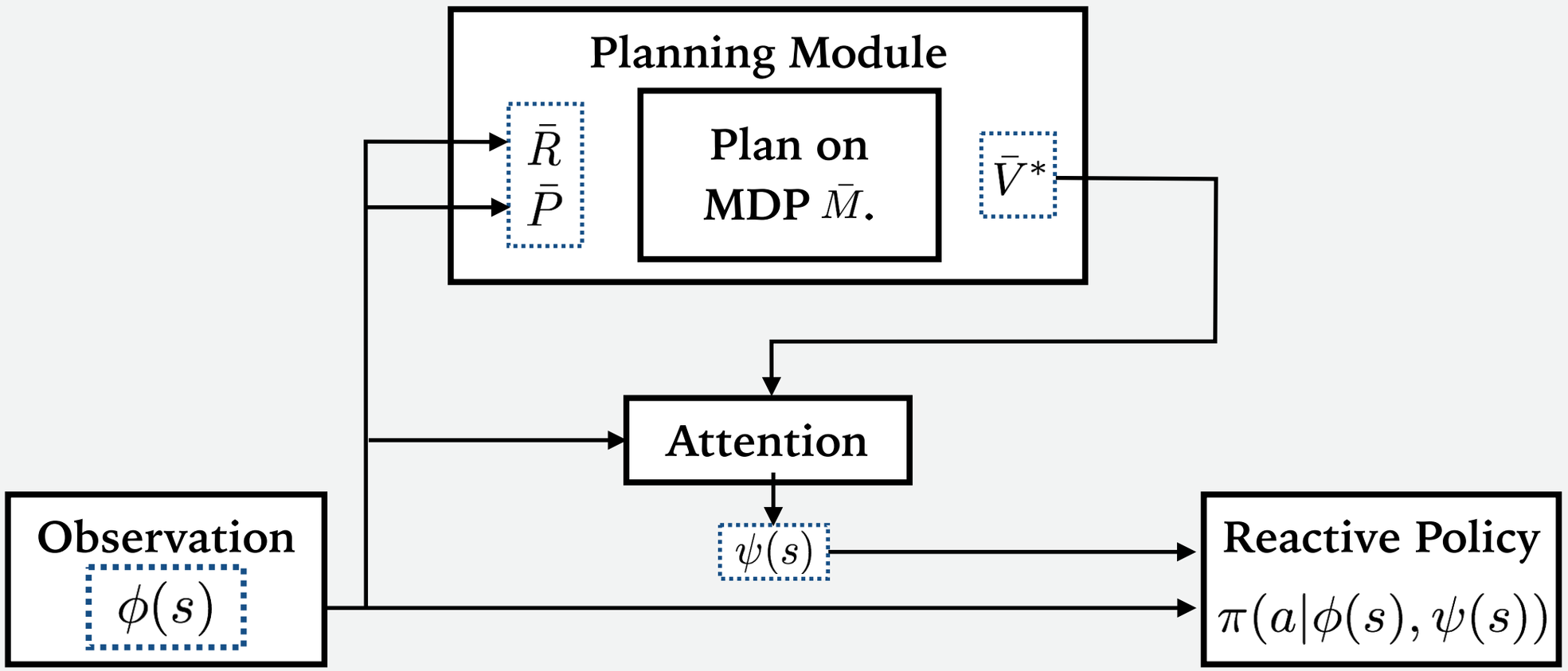
$$\pi^*(a|s) = \arg\max_a R(s,a) + \gamma \sum_{s'} P(s'|s,a)V^*(s')$$



A Planning-based Policy Model

- Policy is still a mapping $\phi(s) \longrightarrow \text{Prob}(a)$

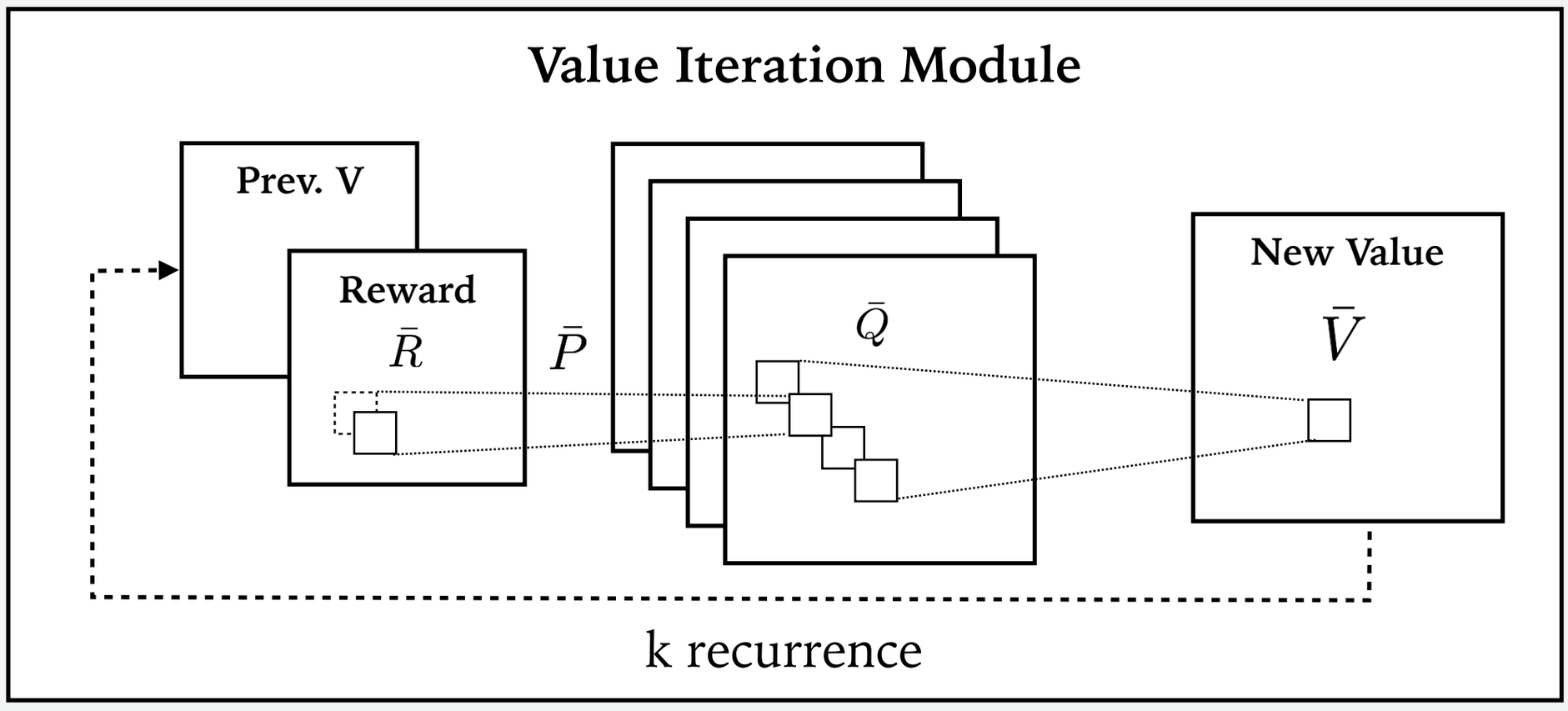
- Parameters θ for mapping $\bar{R}$, $\bar{P}$, *attention*



- How to back-prop through planning computation?

Value Iteration Network

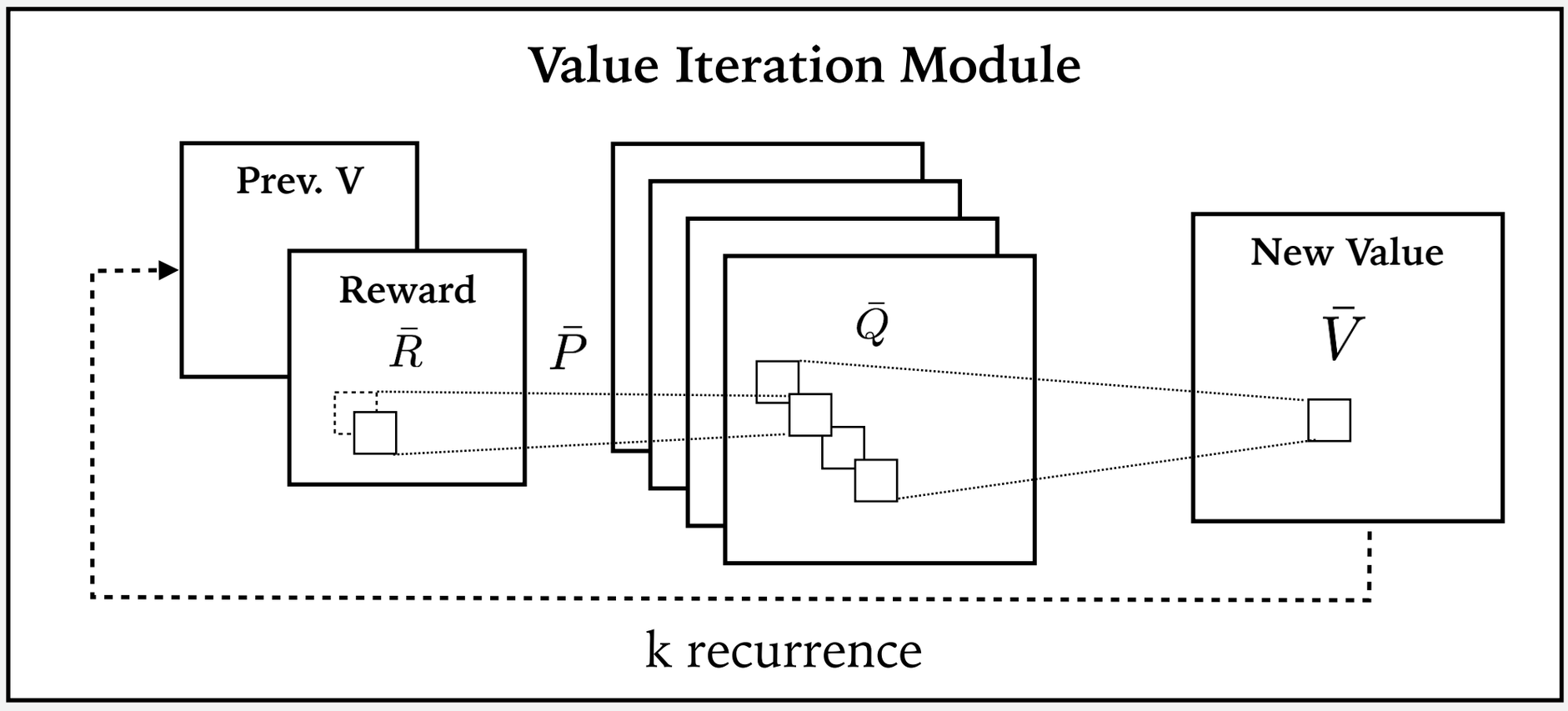
- Differential planner (Value Iteration ≈ CNN)



Conv: $\bar{Q}_{\bar{a},i'j'} = \sum_{l,i,j} W^{\bar{a}}_{l,i,j} \bar{R}_{l,i'-i,j'-j}$

Value Iteration Network

- Differential planner (Value Iteration ≈ CNN)



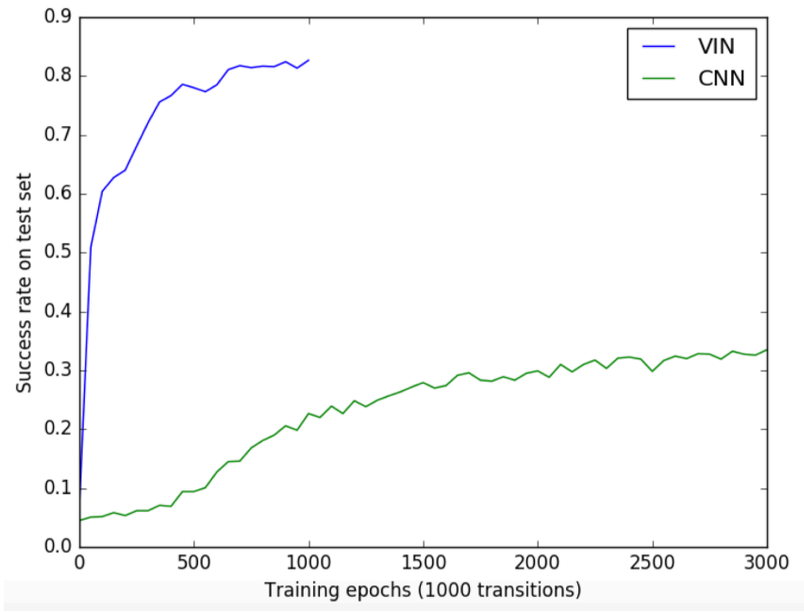
Conv: $\bar{Q}_{\bar{a},i'j'} = \sum_{l,i,j} W^{\bar{a}}_{l,i,j} \bar{R}_{l,i'-i,j'-j}$ Pool: $\bar{V}_{i,j} = \max_{\bar{a}} \bar{Q}(\bar{a}, i, j)$

Experiments

1. **Grid-World Domain**

Network	8 × 8	16 × 16
VIN	**90.9%**	**82.5%**
CNN	86.9%	33.1%

Table: RL Results – performance on **test maps.**



2. Mars Rover Navigation

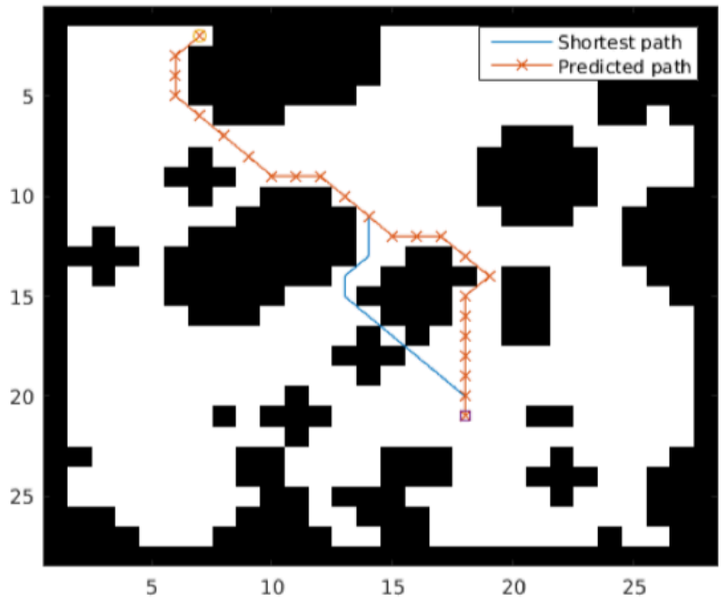
3. Continuous Control

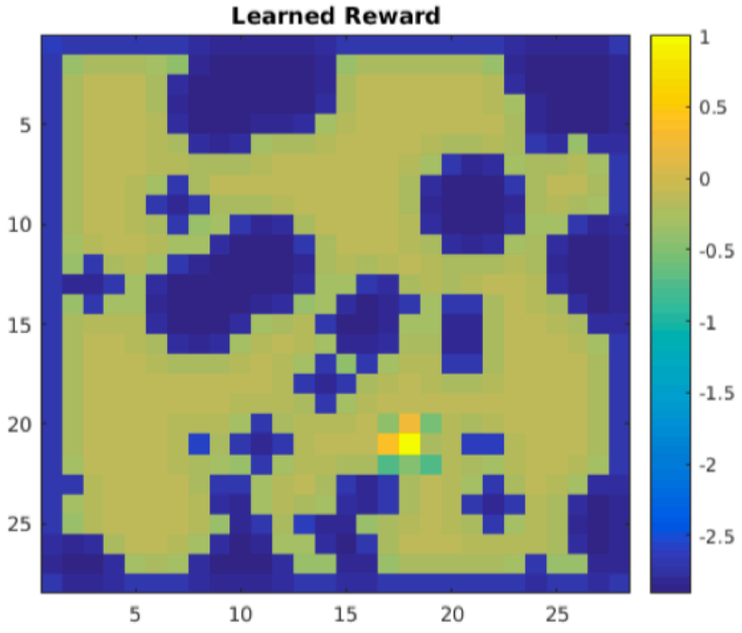
4. WebNav Challenge

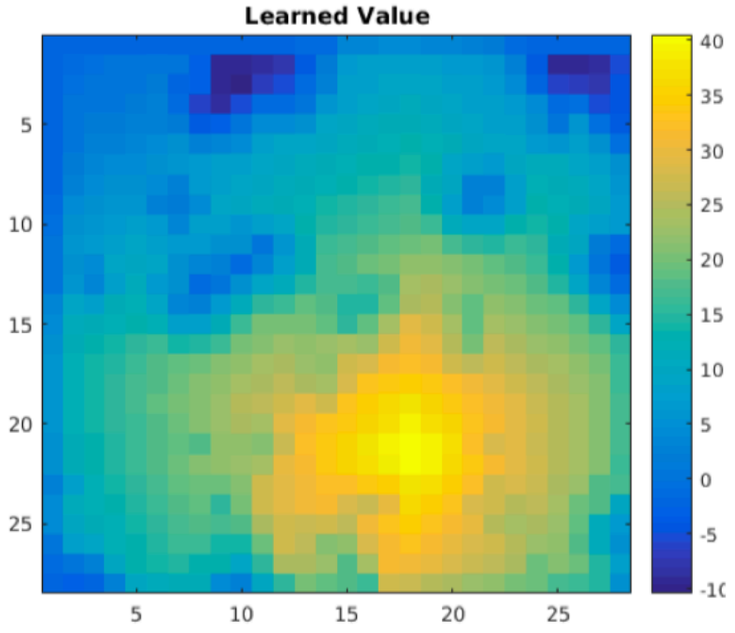
Q&A

Thank you!

Q&A


Shortest path
Predicted path
5
10
15
20
25


Learned Reward
5
10
15
20
25
1
0.5
0
-0.5
-1
-1.5
-2
-2.5


Learned Value
5
10
15
20
25
40
35
30
25
20
15
10
5
0
-5
-10