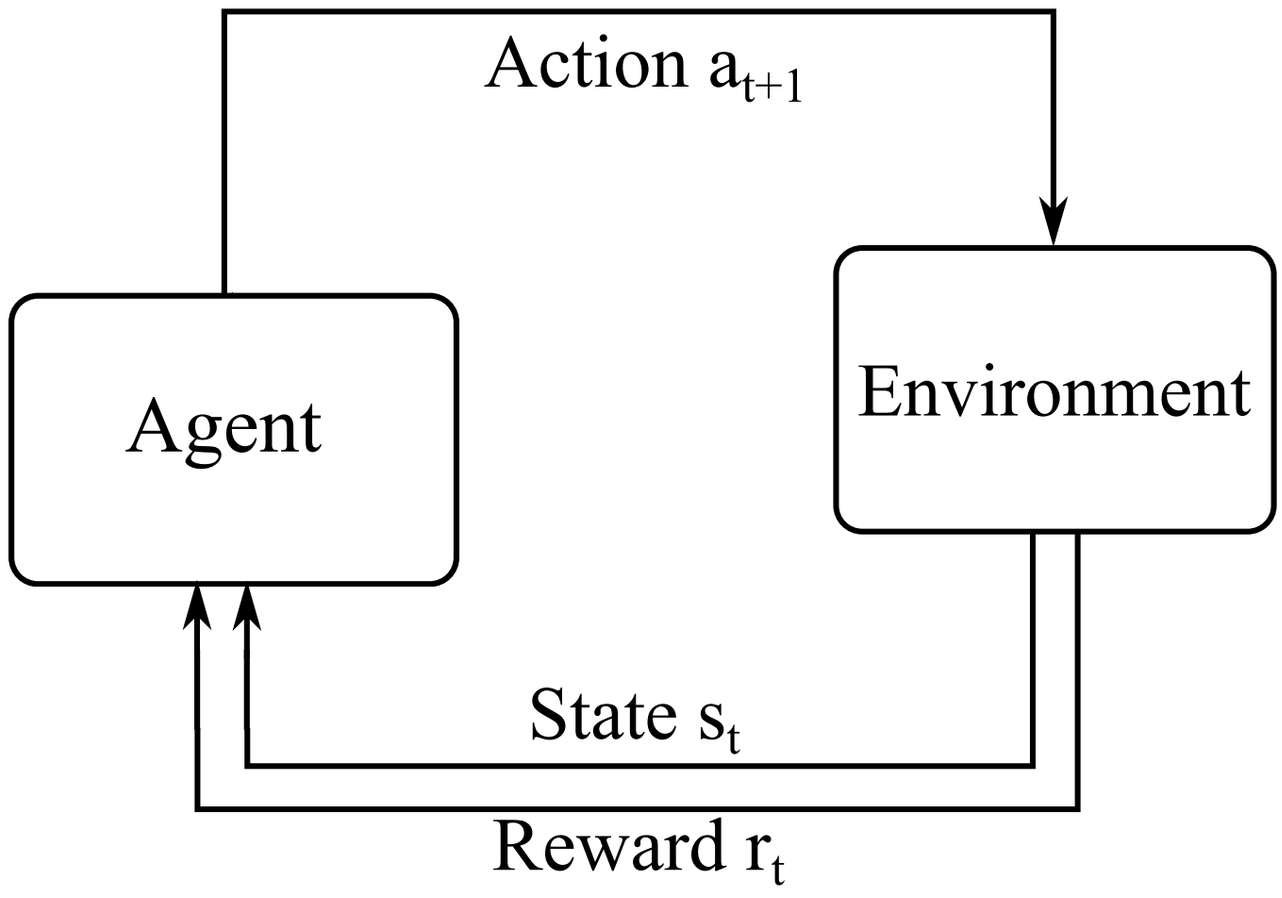
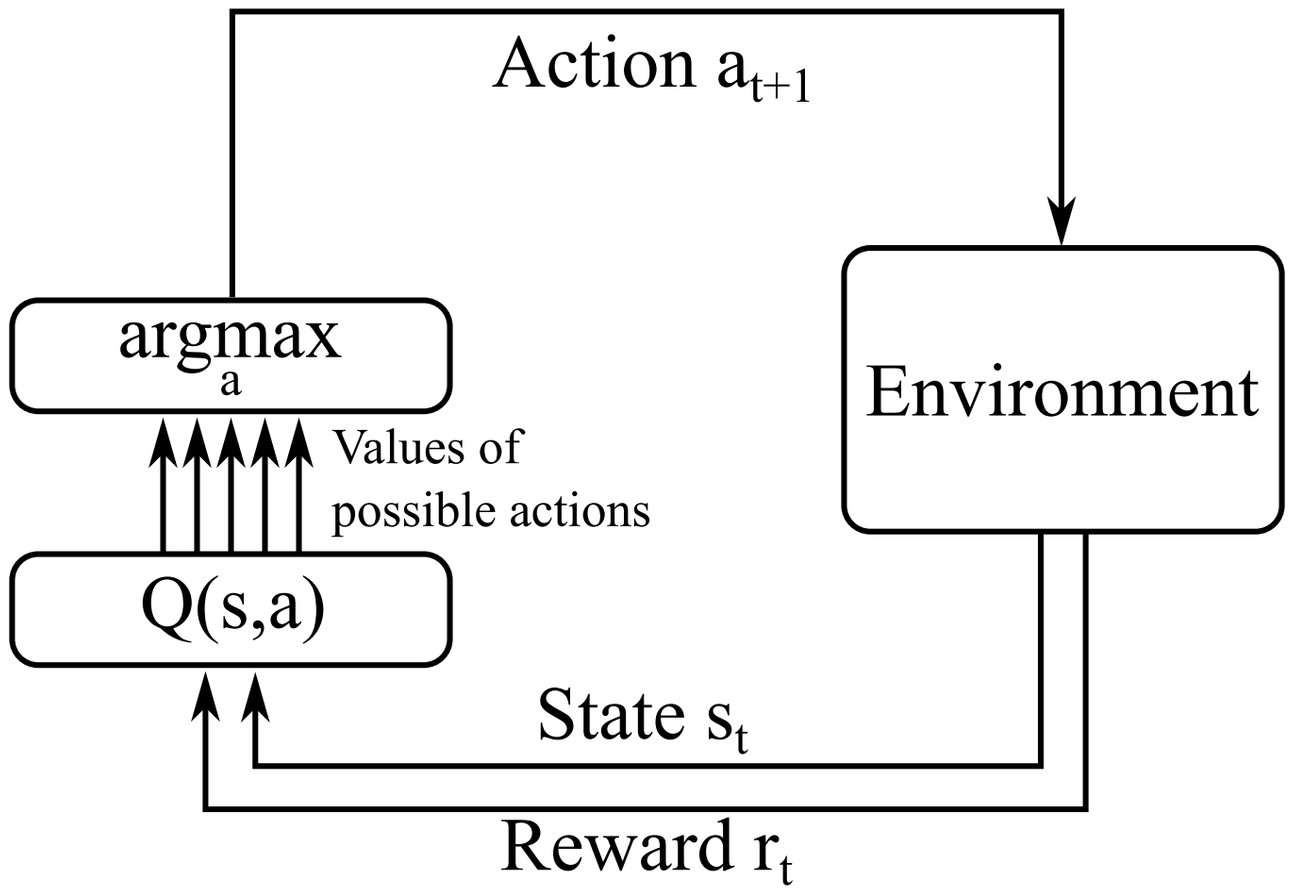
Estimating Q(s,s') with Deep Deterministic Dynamics Gradients

Ashley D. Edwards, Himanshu Sahni, Rosanne Liu, Jane Hung, Ankit Jain, Rui Wang, Adrien Ecoffet, Thomas Miconi, Charles Isbell, Jason Yosinski

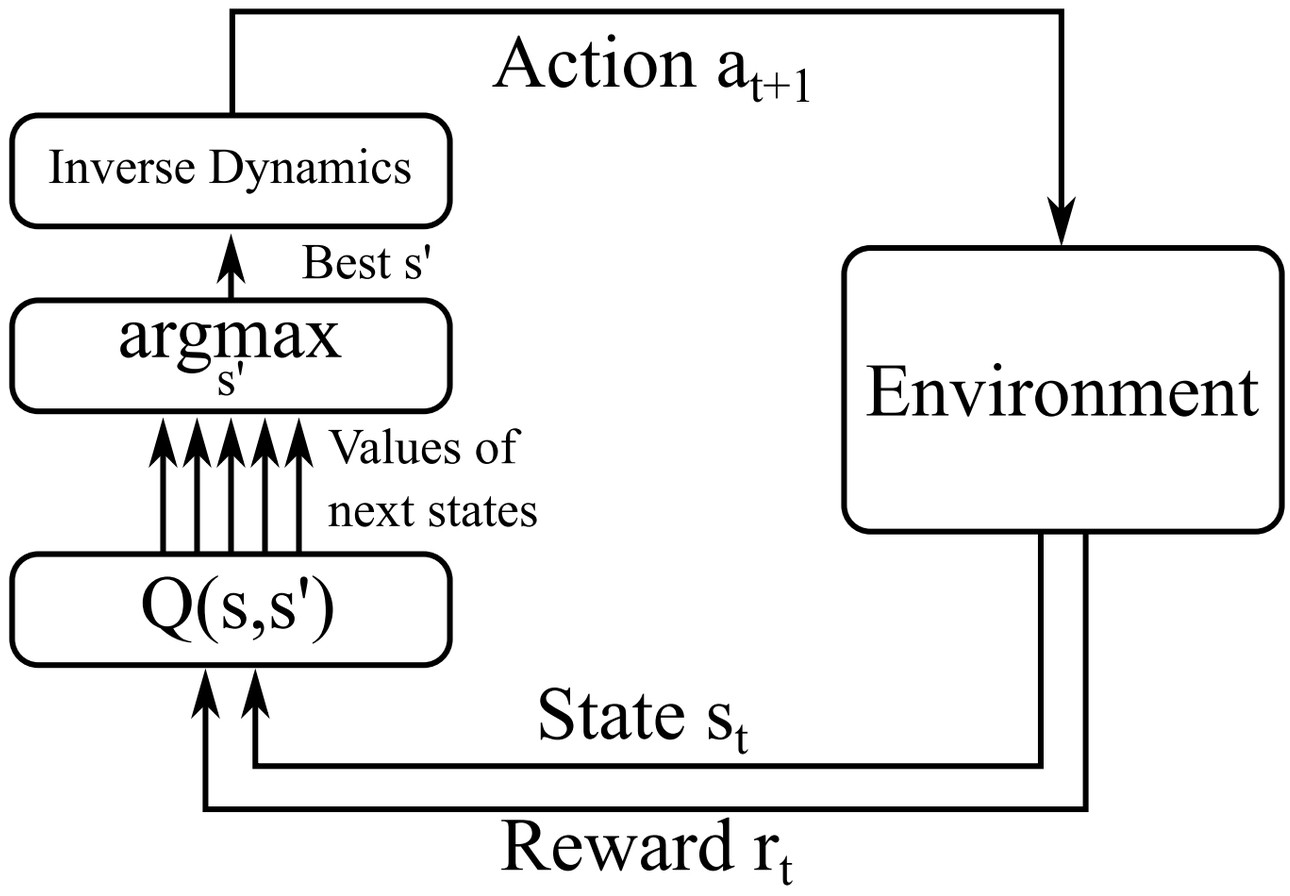
Motivation



Motivation


Action a_{t+1}
argmax
a
Values of
possible actions
Q(s,a)
Environment
State s_t
Reward r_t

Motivation


Action a_{t+1}
Inverse Dynamics
Best s'
argmax
s'
Values of
next states
Q(s,s')
Environment
State s_t
Reward r_t

Motivation

- Naturally learns well in redundant action spaces
- Transfers across permuted action spaces
- Enables learning from observations obtained from completely random policies

The QSS formulation for RL

$$Q(s, s') \doteq r + \gamma \max_{s'' \in N(s')} Q(s', s'')$$

The QSS formulation for RL

$$Q(s, s') \doteq r + \gamma \max_{s'' \in N(s')} Q(s', s'')$$

$$Q(s, s') = Q(s, s') + \alpha[r + \gamma \max_{s'' \in N(s)} Q(s', s'') - Q(s, s')]$$

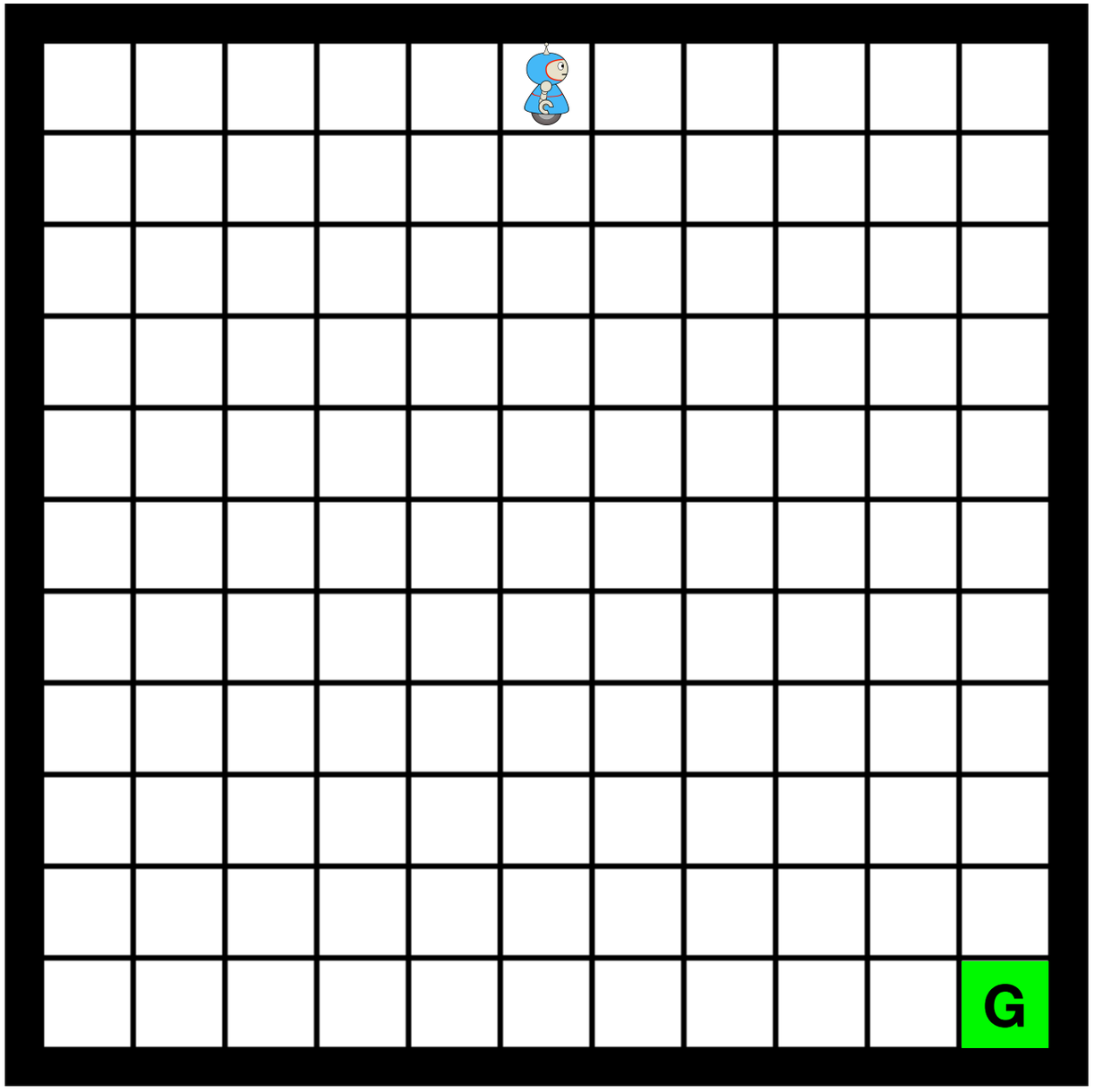
The QSS formulation for RL

$$Q(s, s') \doteq r + \gamma \max_{s'' \in N(s')} Q(s', s'')$$

$$Q(s, s') = Q(s, s') + \alpha[r + \gamma \max_{s'' \in N(s)} Q(s', s'') - Q(s, s')]$$

$$\tau(s) = \arg\max_{s' \in N(s)} Q(s, s')$$

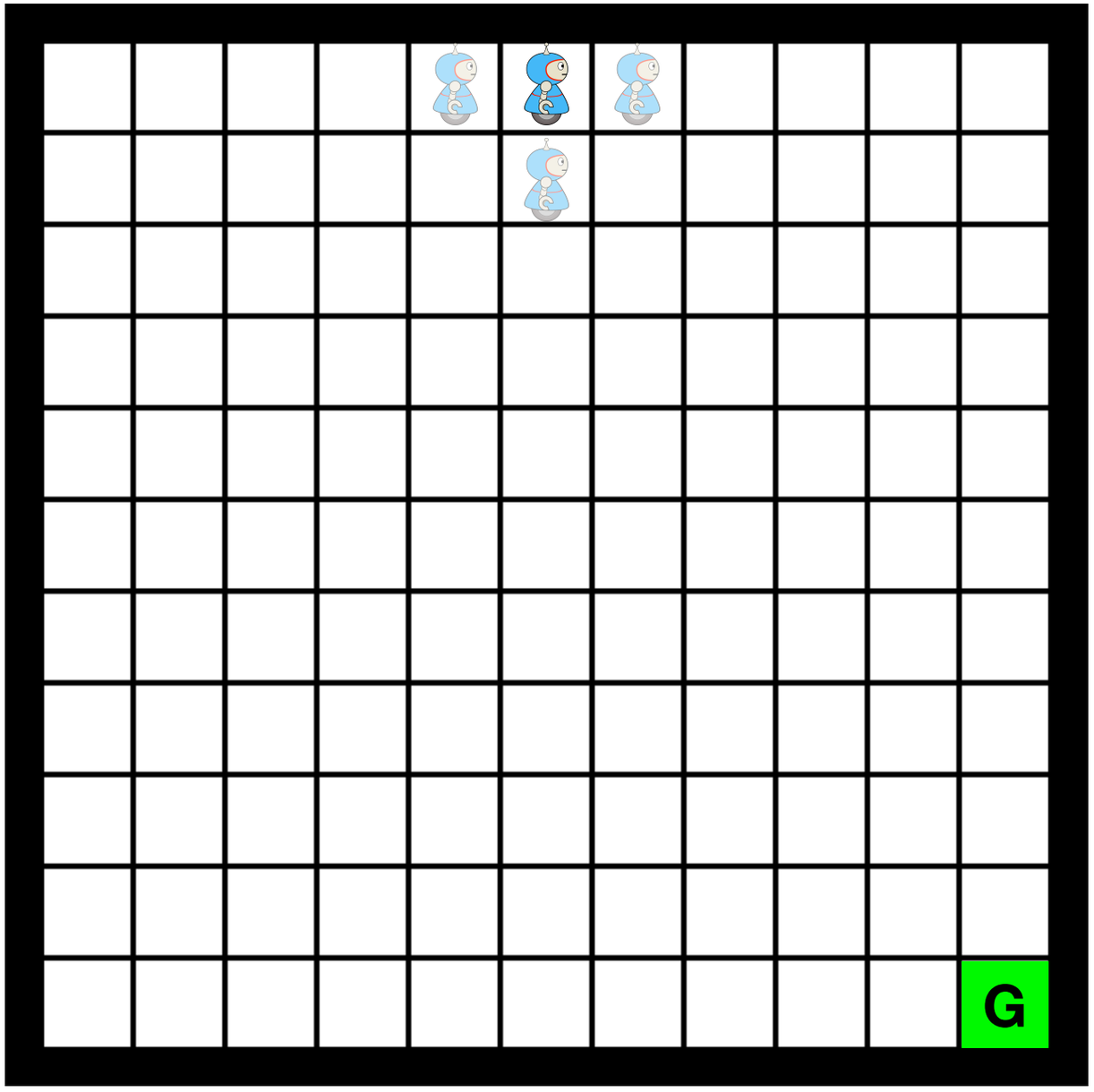
The QSS formulation for RL



Control

1) Find neighboring states N(s)
2) Select state with largest value
3) Determine action

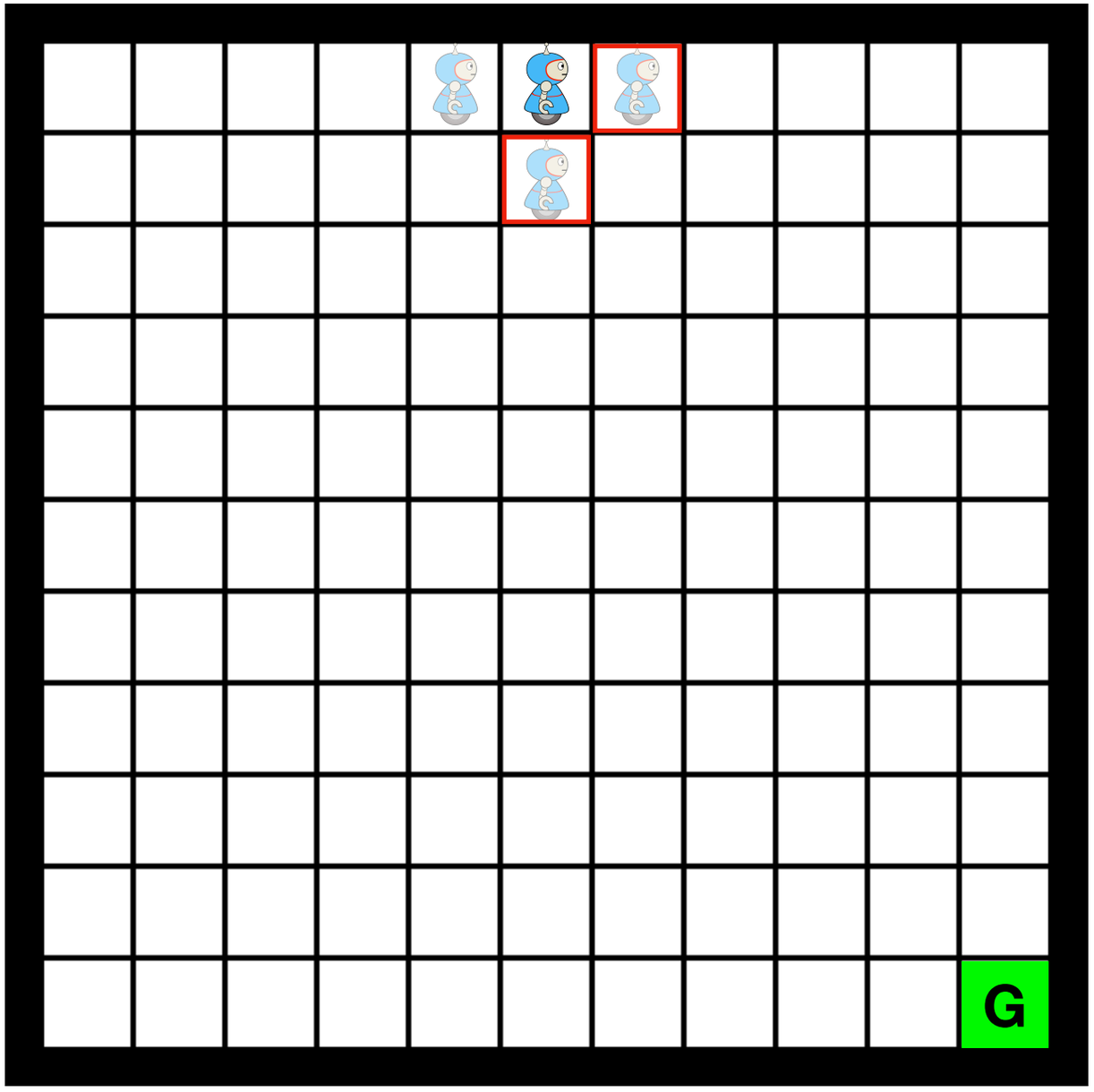
The QSS formulation for RL



Control

1) Find neighboring states N(s)
2) Select state with largest value
3) Determine action

The QSS formulation for RL

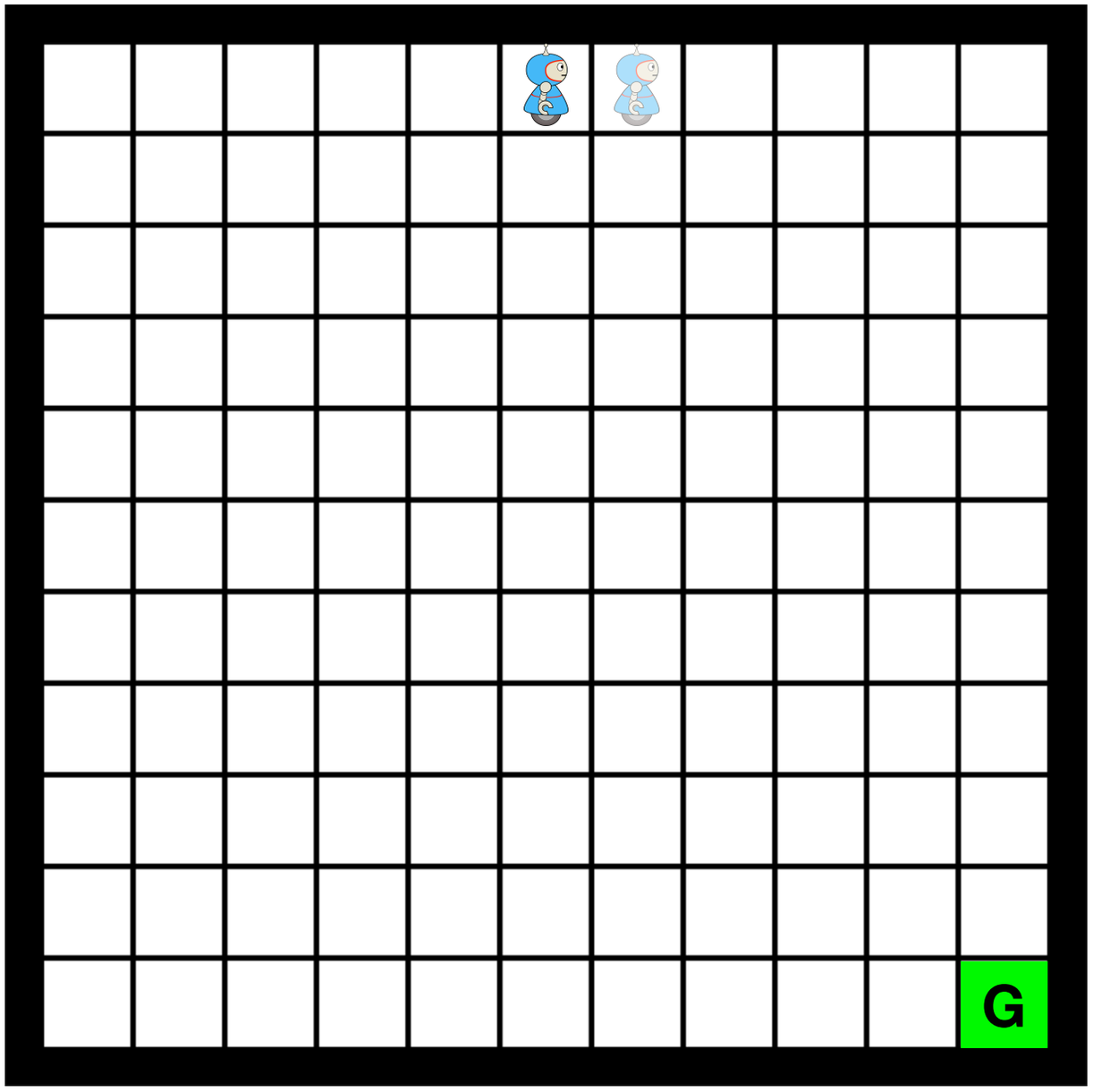


Control

1) Find neighboring states N(s)
2) Select state with largest value
3) Determine action

$$\tau(s) = \underset{s' \in N(s)}{\arg\max}\, Q(s, s')$$

The QSS formulation for RL



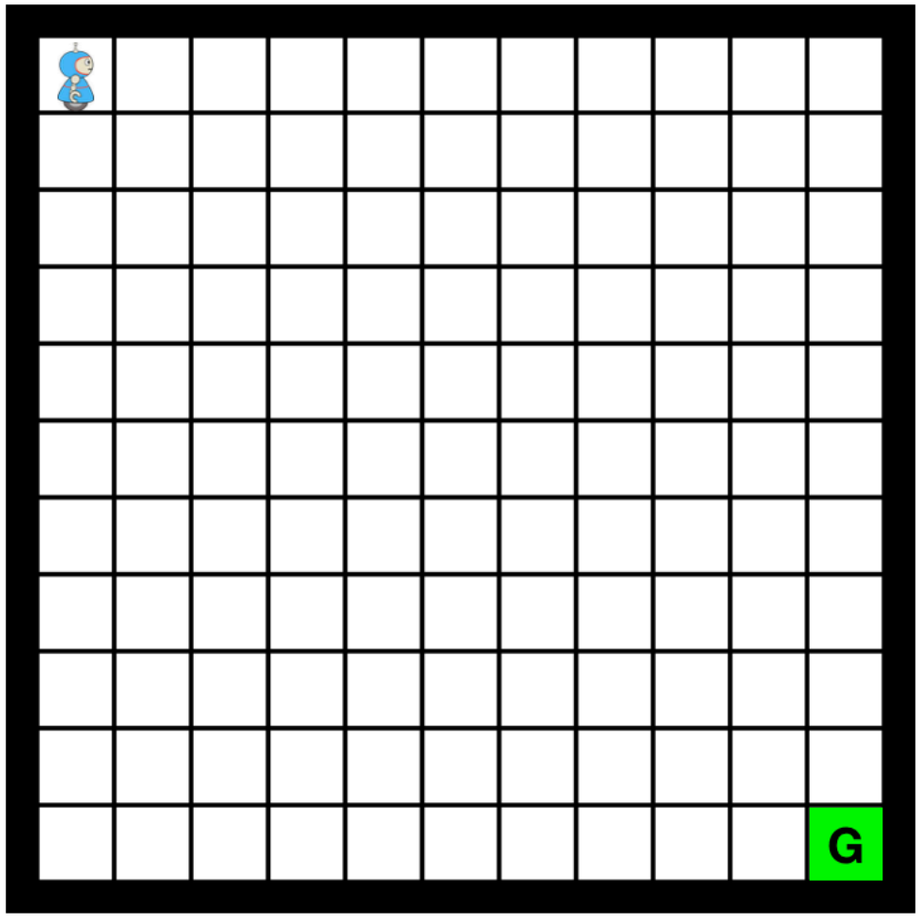
Control

1) Find neighboring states N(s)
2) Select state with largest value
3) Determine action

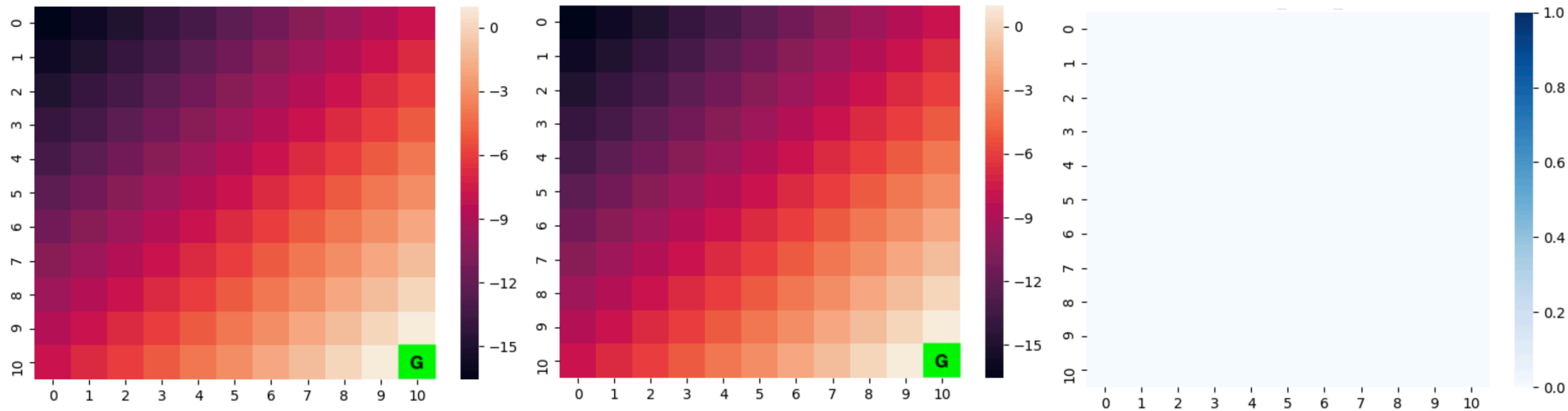
$$\pi(s) = I(s, \tau(s))$$

Tabular experiments

- Q(s,s') stored in table
- Given N(s) and I(s,s')
- 11x11 gridworld
- 4 actions



Tabular experiments

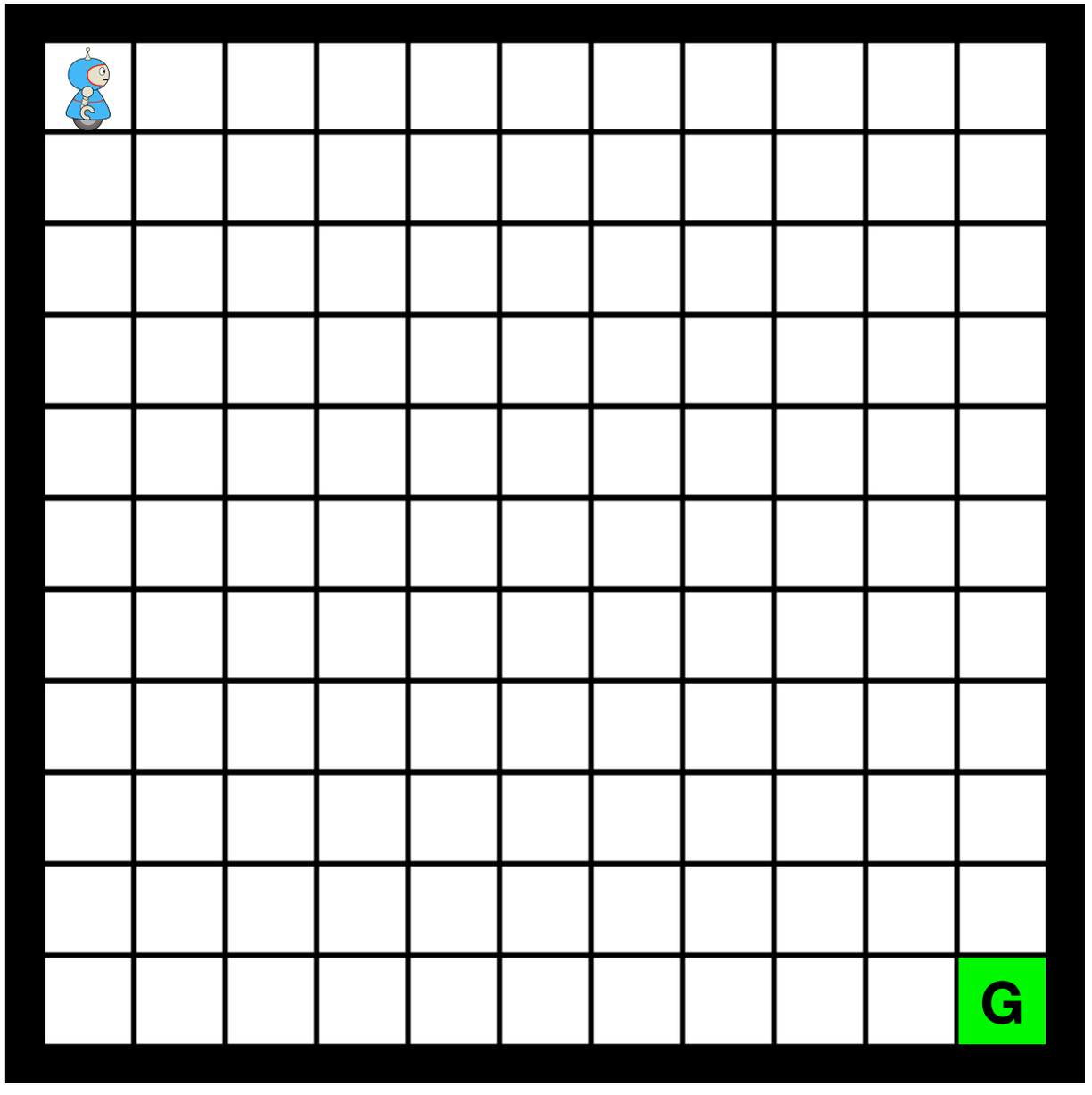


(a) $\max_a Q(s, a)$ (b) $\max_{s'} Q(s, s')$ (c) $\frac{QSS - QSA}{|QSS|}$

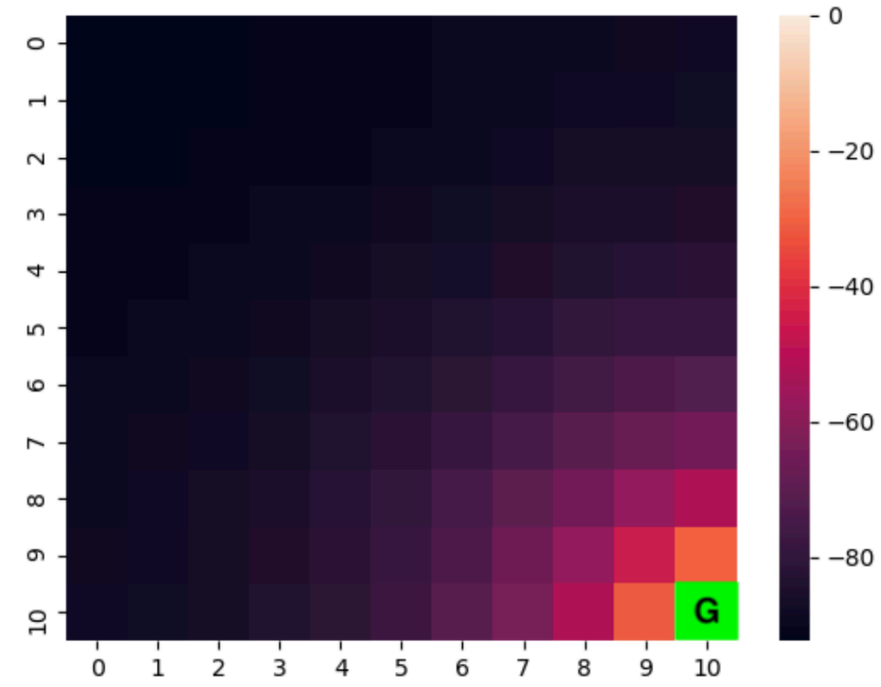
Example of equivalence of QSA and QSS

Tabular experiments

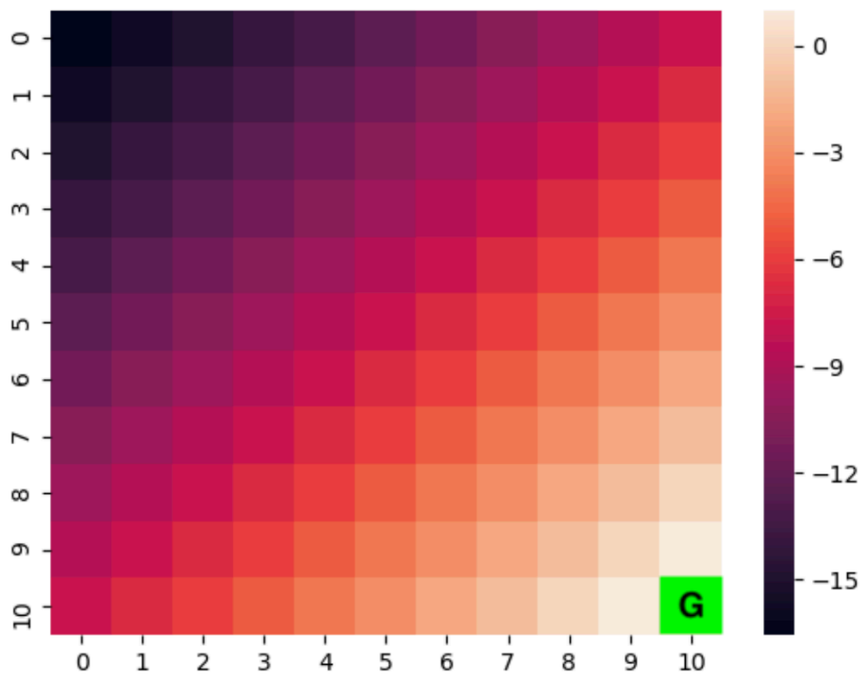
- 4 actions with probability of slipping
- Given N(s) and I(s,s')



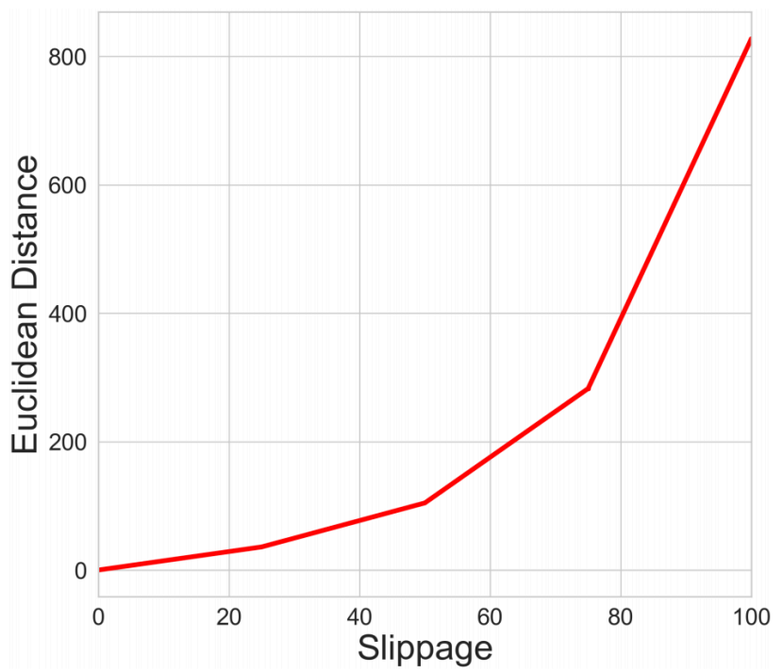
Tabular experiments



(a) $\max_a Q(s, a)$



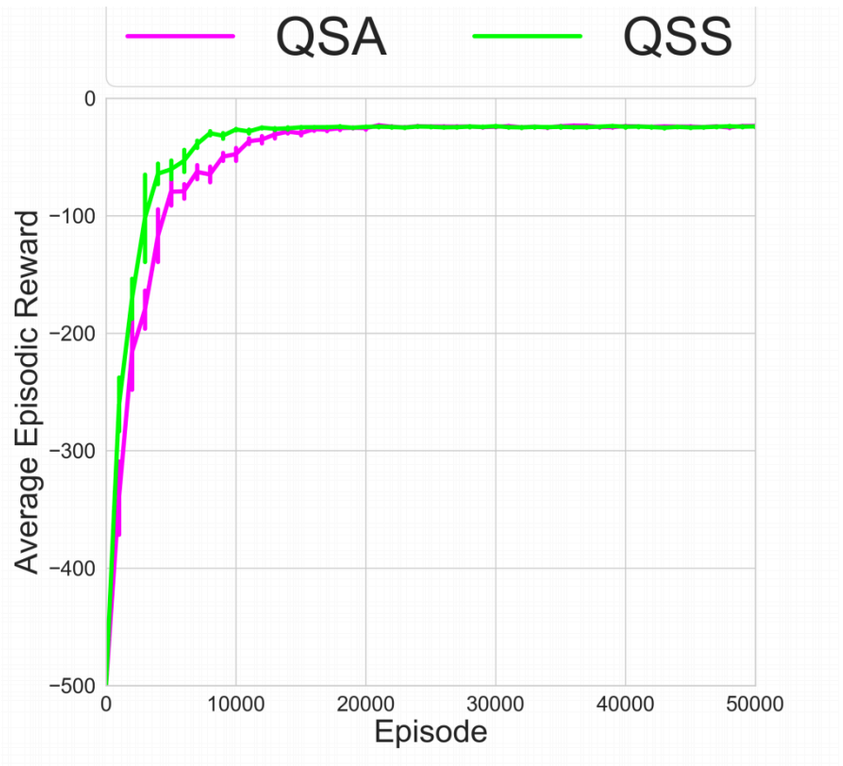
(b) $\max_{s'} Q(s, s')$



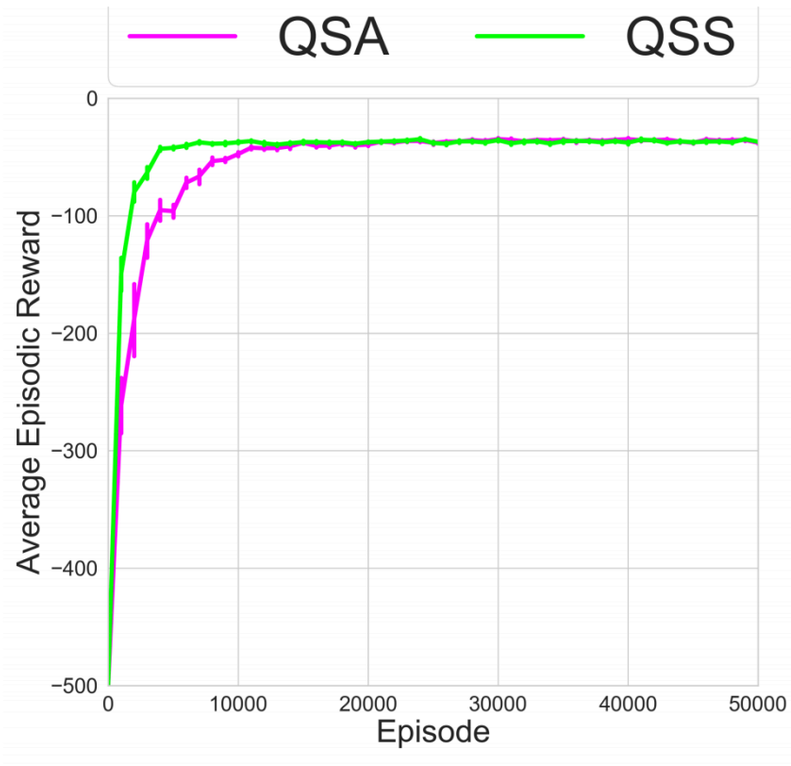
(c) Value distance

Example of QSS with stochastic actions

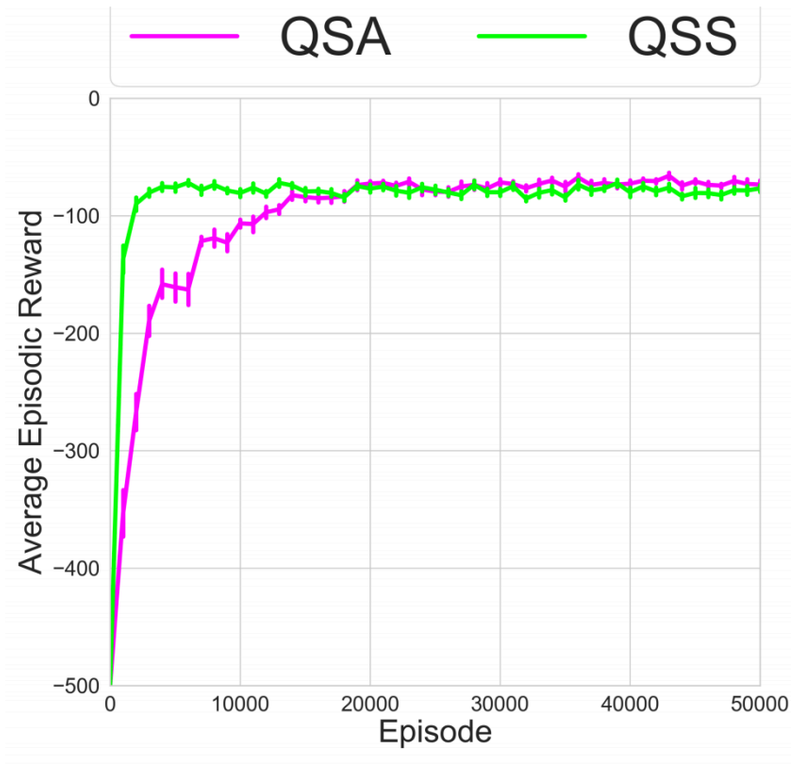
Tabular experiments



(a) 25%



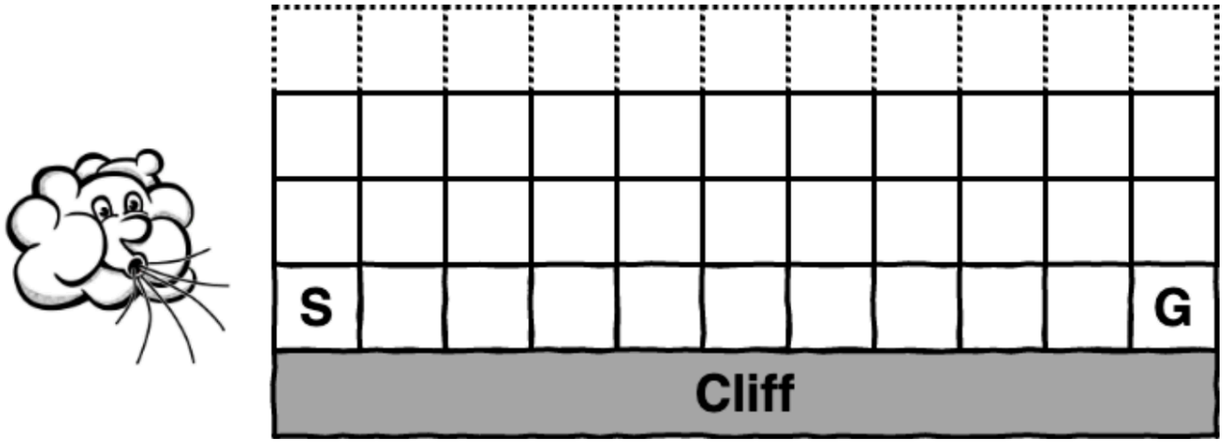
(b) 50%



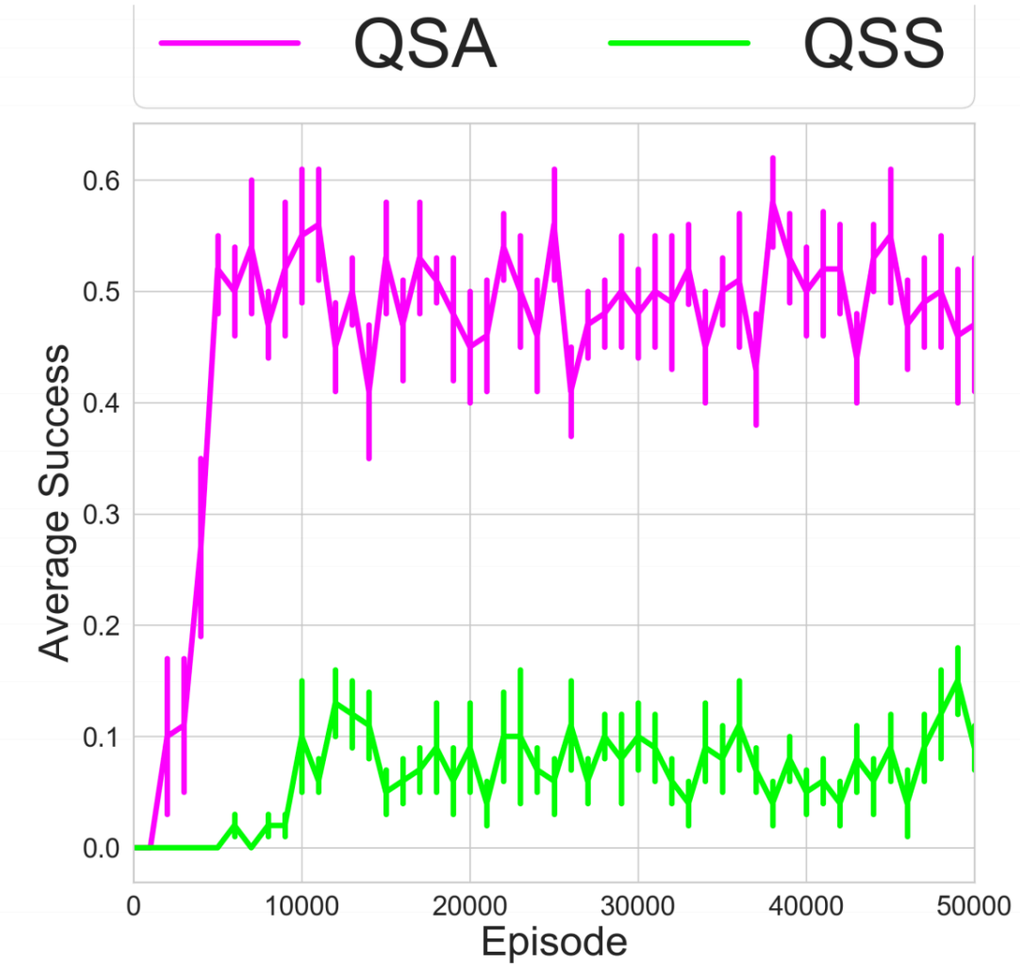
(c) 75%

Example of QSS with stochastic actions

Tabular experiments



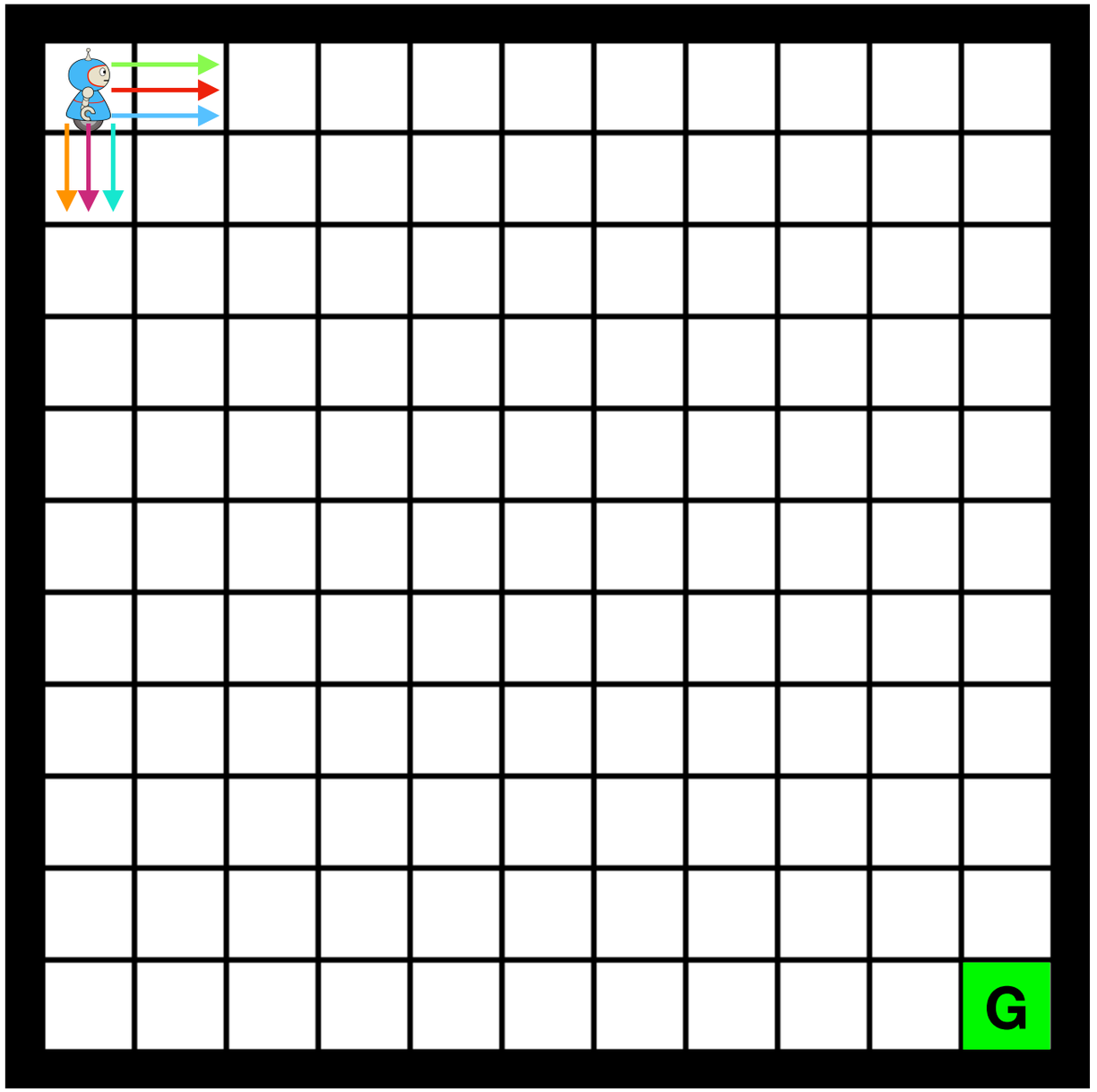
50% slippage along edge



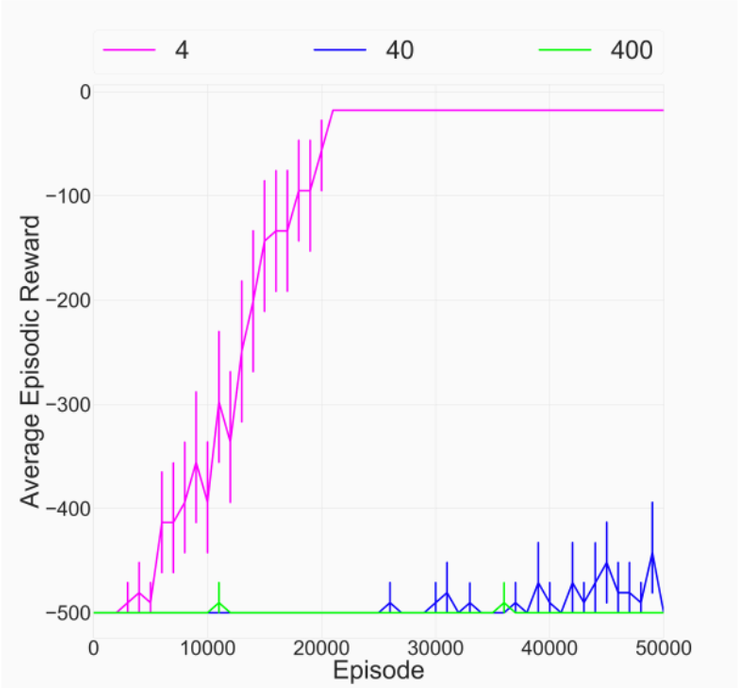
Example of QSS with stochastic actions

Tabular experiments

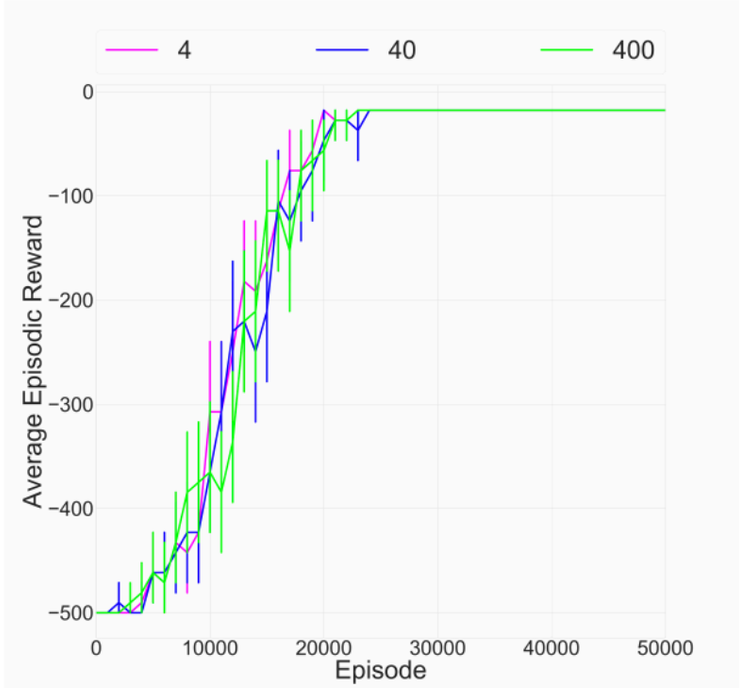
- 4 base actions with redundancy
- Learn I(s,s’)



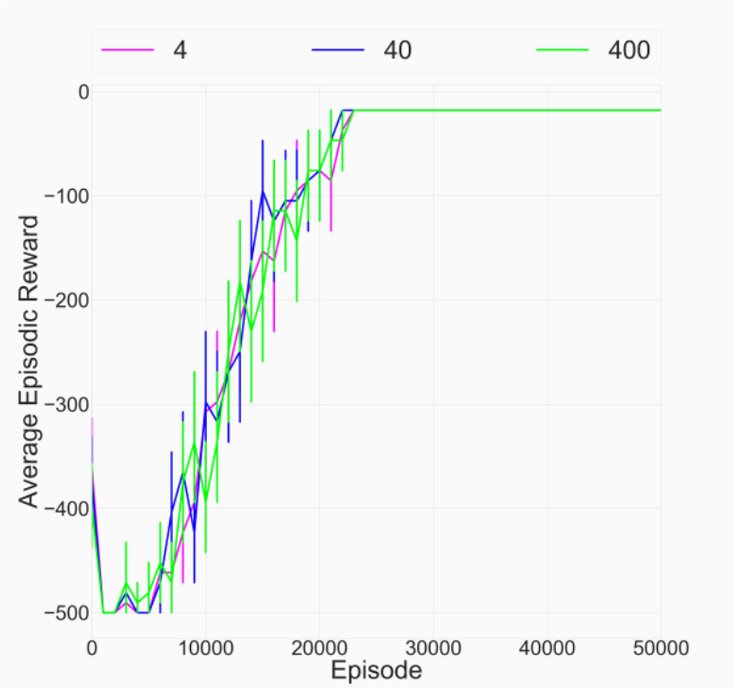
Tabular experiments



(a) QSA



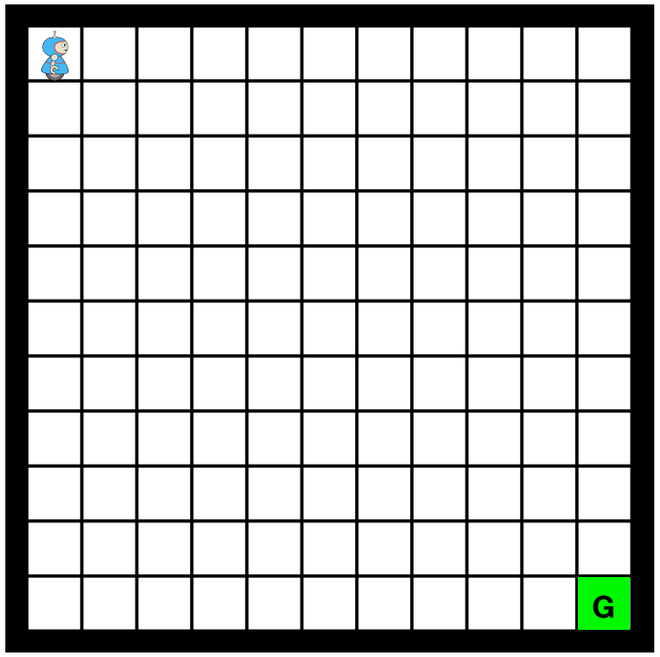
(b) QSS



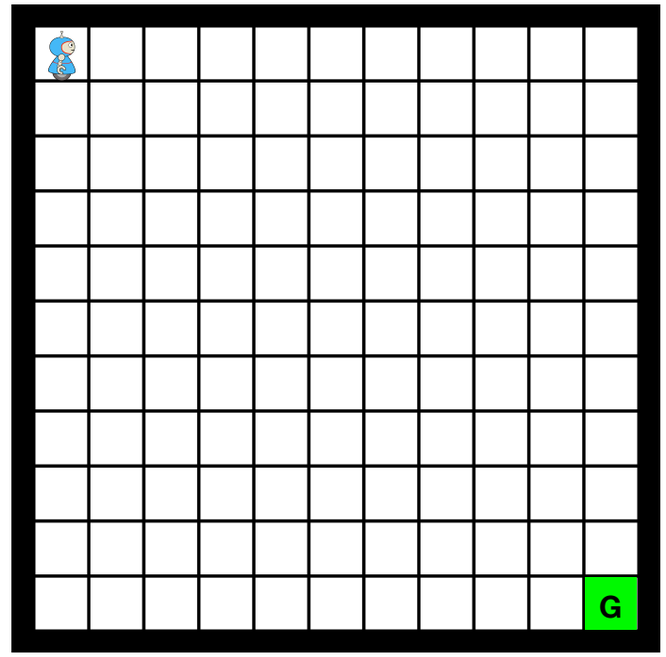
(c) QSS + inverse dynamics

QSS with redundant actions

Tabular experiments

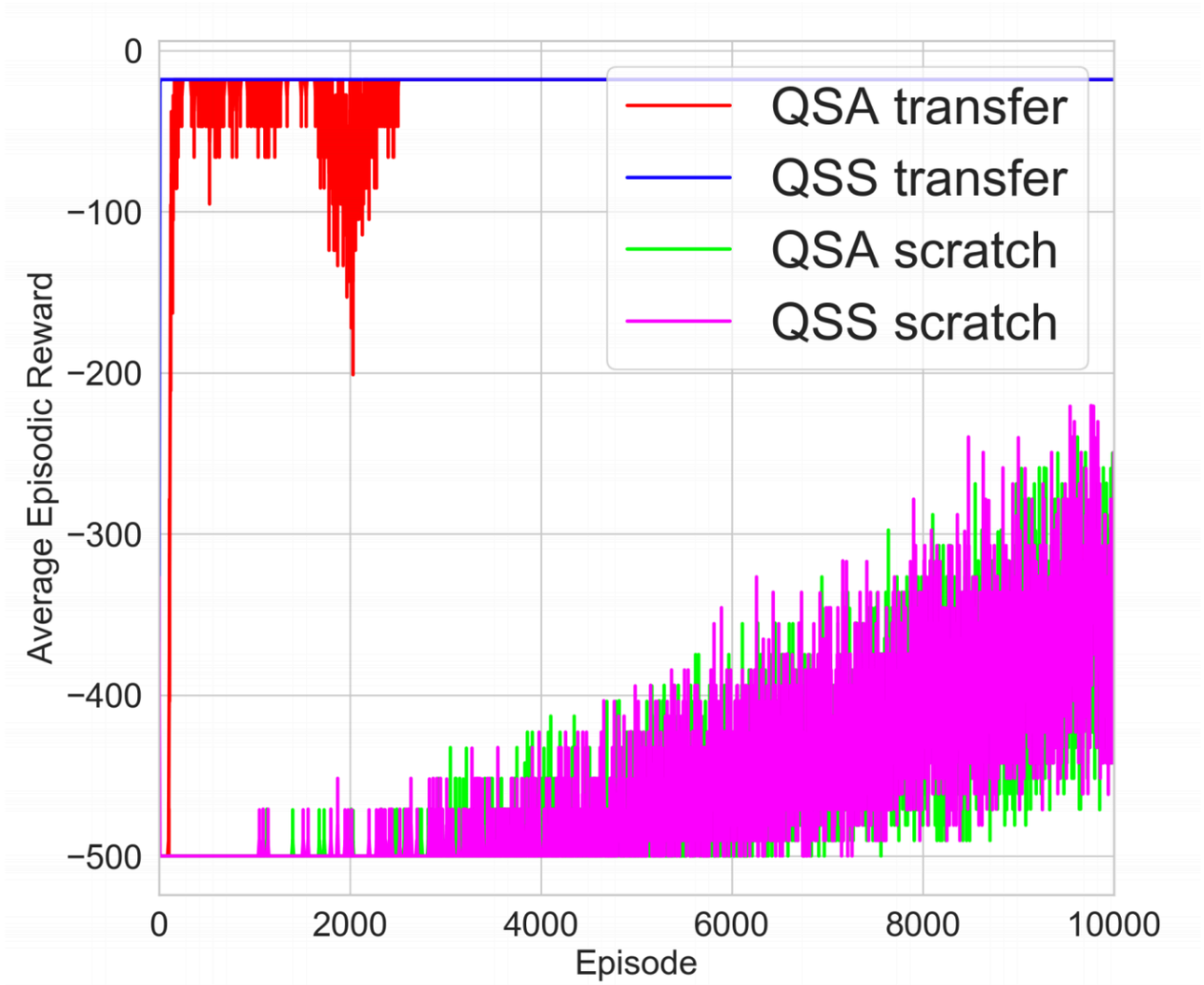


- 11x11 gridworld
- 4 actions 0, 1, 2, 3
- Given N(s)
- Learn I(s,s’)



- 11x11 gridworld
- 4 actions 3, 2, 1, 0
- Given N(s)
- Learn I(s,s’)

Tabular experiments



QSS with permuted actions

Tabular experiments

- Experiments demonstrated properties of QSS
- Most problems cannot be solved in tabular setting
- How to learn in settings with large/continuous state spaces?

The QSS formulation for RL

$$Q(s, s') \doteq r + \gamma \max_{s'' \in N(s')} Q(s', s'')$$

$$Q(s, s') = Q(s, s') + \alpha[r + \gamma \max_{s'' \in N(s)} Q(s', s'') - Q(s, s')]$$

$$\pi(s) = I(s, \tau(s))$$

$$\tau(s) = \arg\max_{s' \in N(s)} Q(s, s')$$

The QSS formulation for RL

$$Q(s, s') \doteq r + \gamma \max_{s'' \in N(s')} Q(s', s'')$$

$$Q(s, s') = Q(s, s') + \alpha[r + \gamma \max_{s'' \in N(s)} Q(s', s'') - Q(s, s')]$$

$$\pi(s) = I(s, \tau(s))$$

$$\tau(s) = \arg\max_{s' \in N(s)} Q(s, s')$$

The QSS formulation for RL

$$Q(s, s') \doteq r + \gamma \max_{s'' \in N(s')} Q(s', s'')$$

$$Q(s, s') = Q(s, s') + \alpha[r + \gamma \max_{s'' \in N(s)} Q(s', s'') - Q(s, s')]$$

$$\pi(s) = I(s, \tau(s))$$

$$\tau(s) = \arg\max_{s' \in N(s)} Q(s, s')$$

The QSS formulation for RL

$$Q(s, s') \doteq r + \gamma Q(s, \tau(s))$$

$$Q(s, s') = Q(s, s') + \alpha[r + \gamma \max_{s'' \in N(s)} Q(s', s'') - Q(s, s')]$$

$$\pi(s) = I(s, \tau(s))$$

$$\tau(s) = \arg\max_{s' \in N(s)} Q(s, s')$$

The QSS formulation for RL

$$Q(s, s') \doteq r + \gamma Q(s, \tau(s))$$

$$Q(s, s') = Q(s, s') + \alpha[r + \gamma \max_{s'' \in N(s)} Q(s', s'') - Q(s, s')]$$

$$\pi(s) = I(s, \tau(s))$$

$$\tau(s) = \arg\max_{s' \in N(s)} Q(s, s')$$

The QSS formulation for RL

$$Q(s, s') \doteq r + \gamma Q(s, \tau(s))$$

$$Q(s, s') = Q(s, s') + \alpha[r + \gamma Q(s', \tau(s')) - Q(s, s')]$$

$$\pi(s) = I(s, \tau(s))$$

$$\tau(s) = \underset{s' \in N(s)}{\arg\max}\, Q(s, s')$$

The QSS formulation for RL

$$Q(s, s') \doteq r + \gamma Q(s, \tau(s))$$

$$Q(s, s') = Q(s, s') + \alpha[r + \gamma Q(s', \tau(s')) - Q(s, s')]$$

$$\pi(s) = I(s, \tau(s))$$

$$\tau(s) = \underset{s' \in N(s)}{\arg\max}\, Q(s, s')$$

The QSS formulation for RL

$$Q(s, s') \doteq r + \gamma Q(s, \tau(s))$$

$$\mathcal{L}_\theta = \sum_i \| y - Q_{\theta_i}(s, s') \|$$

$$y = r + \gamma \min_{i=1,2} Q_{\theta'_i}(s', \tau_{\psi'}(s'))]$$

$$\pi(s) = I(s, \tau(s))$$

$$\tau(s) = \arg\max_{s' \in N(s)} Q(s, s')$$

The QSS formulation for RL

$$Q(s, s') \doteq r + \gamma Q(s, \tau(s))$$

$$\mathcal{L}_\theta = \sum_i \| y - Q_{\theta_i}(s, s') \|$$

$$y = r + \gamma \min_{i=1,2} Q_{\theta'_i}(s', \tau_{\psi'}(s'))]$$

$$\pi(s) = I(s, \tau(s))$$

$$\tau(s) = \arg\max_{s' \in N(s)} Q(s, s')$$

The QSS formulation for RL

$$Q(s, s') \doteq r + \gamma Q(s, \tau(s))$$

$$\mathcal{L}_\theta = \sum_i \| y - Q_{\theta_i}(s, s') \|$$

$$y = r + \gamma \min_{i=1,2} Q_{\theta'_i}(s', \tau_{\psi'}(s'))]$$

$$\mathcal{L}_\omega = \| I_\omega(s, s') - a \|$$

$$\tau(s) = \arg\max_{s' \in N(s)} Q(s, s')$$

The QSS formulation for RL

$$Q(s, s') \doteq r + \gamma Q(s, \tau(s))$$

$$\mathcal{L}_\theta = \sum_i \| y - Q_{\theta_i}(s, s') \|$$

$$y = r + \gamma \min_{i=1,2} Q_{\theta'_i}(s', \tau_{\psi'}(s'))]$$

$$\mathcal{L}_\omega = \| I_\omega(s, s') - a \|$$

$$\tau(s) = \arg\max_{s' \in N(s)} Q(s, s')$$

The QSS formulation for RL

$$Q(s, s') \doteq r + \gamma Q(s, \tau(s))$$

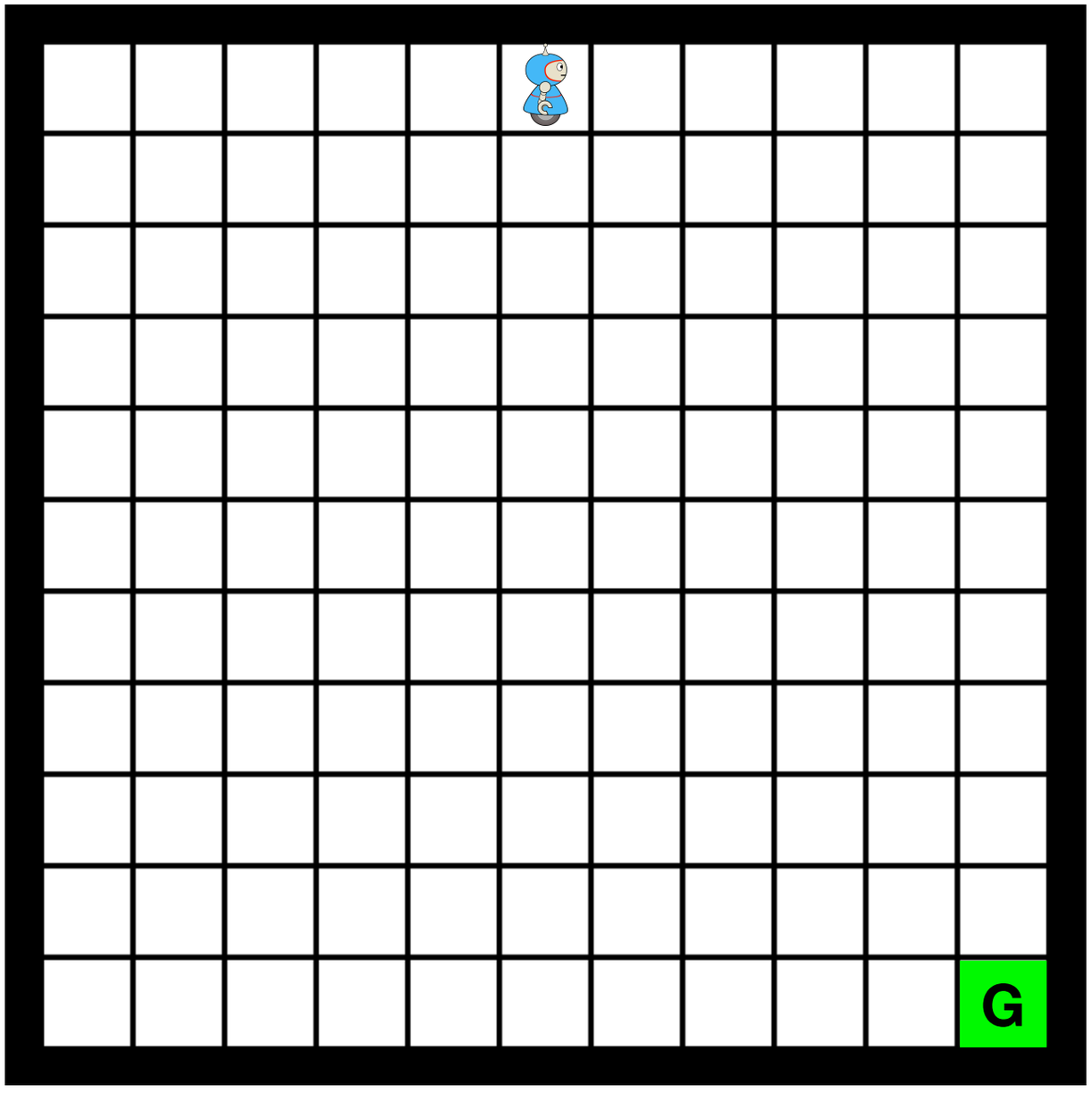
$$\mathcal{L}_\theta = \sum_i \| y - Q_{\theta_i}(s, s') \|$$

$$y = r + \gamma \min_{i=1,2} Q_{\theta'_i}(s', \tau_{\psi'}(s'))]$$

$$\mathcal{L}_\omega = \| I_\omega(s, s') - a \|$$

$$\mathcal{L}_\psi = -Q_\theta(s, \tau_\psi(s))$$

Continuous QSS via D3G

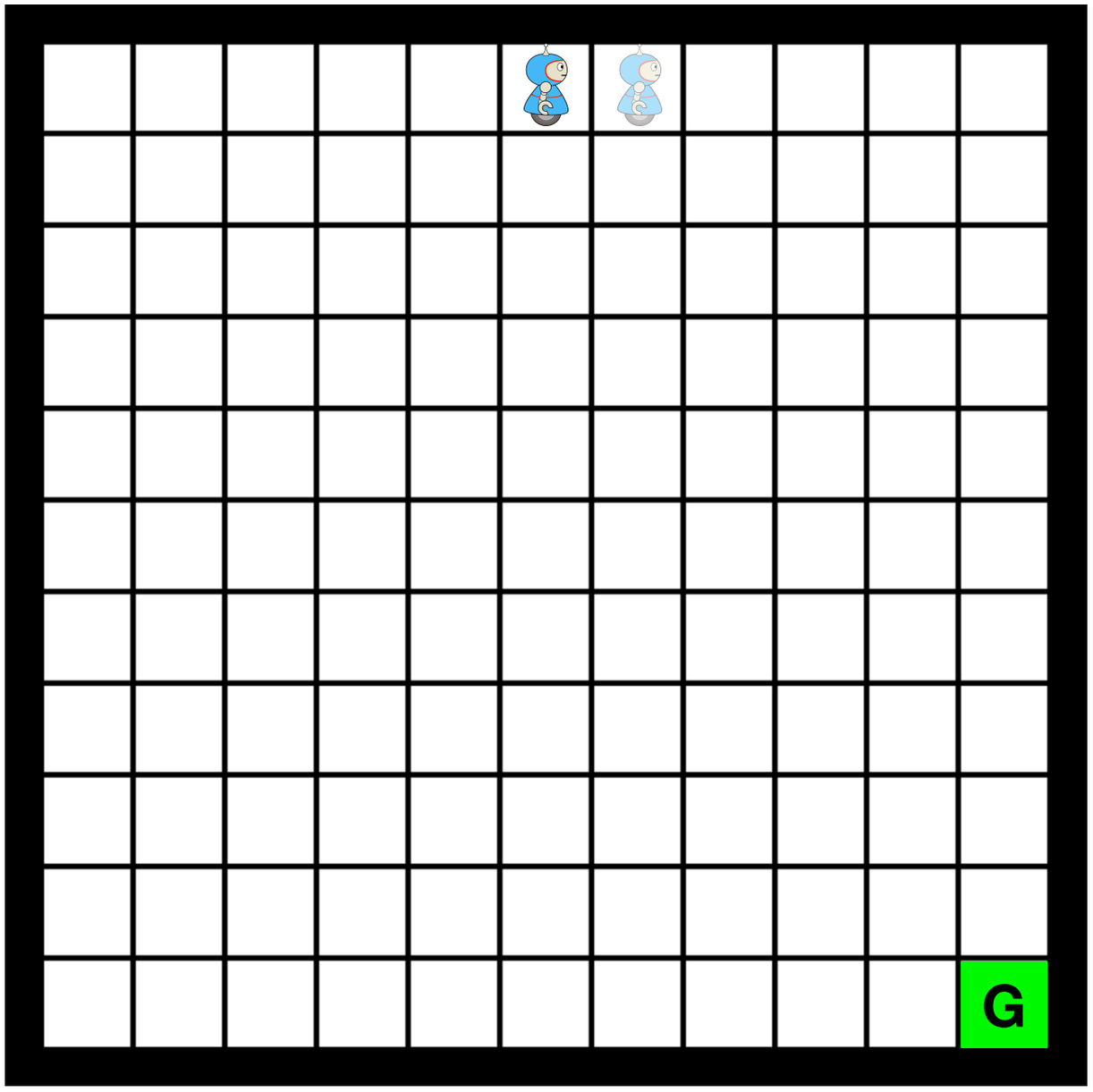


Control

1) ~~Find neighboring states N(s)~~
2) Select state with largest value
3) Determine action

$$s'_\tau \leftarrow \tau(s)$$

Continuous QSS via D3G

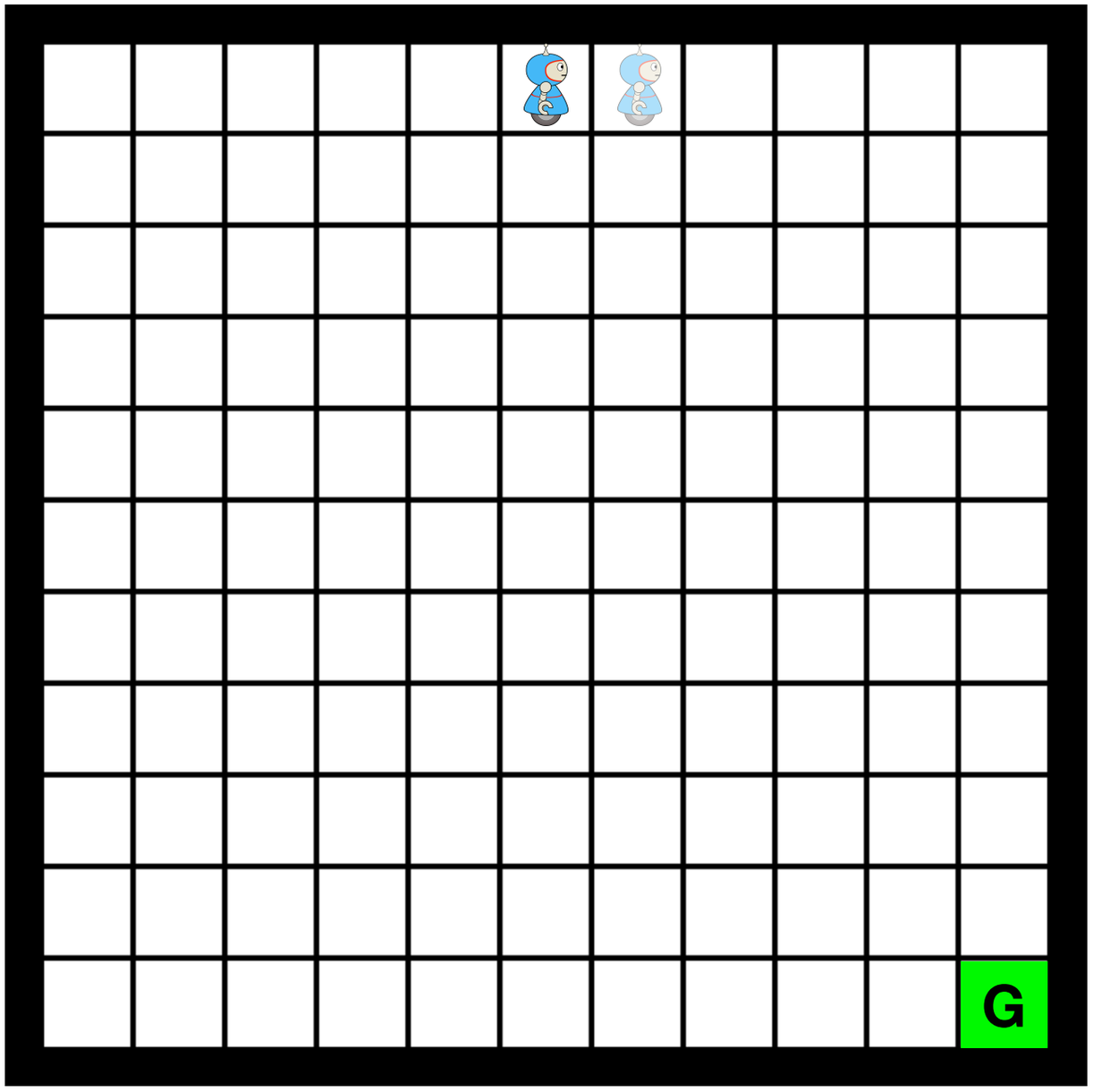


Control

1) ~~Find neighboring states N(s)~~
2) Select state with largest value
3) Determine action

$$s'_\tau \leftarrow \tau(s)$$

Continuous QSS via D3G



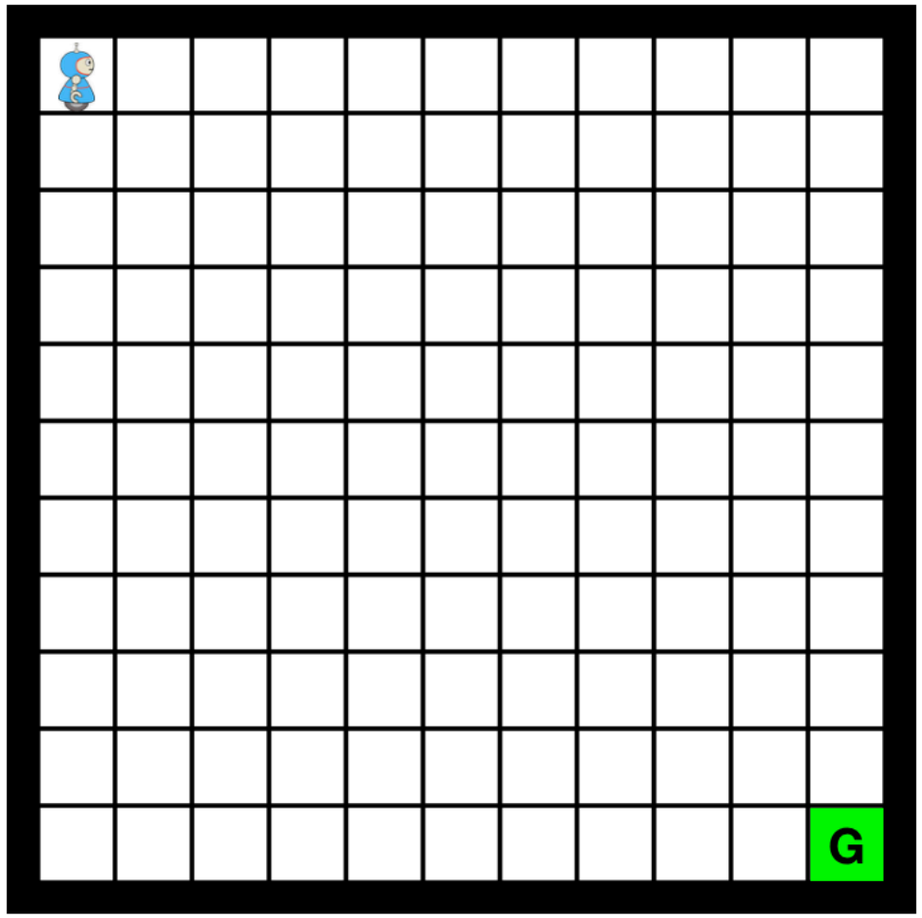
Control

1) ~~Find neighboring states N(s)~~
2) Select state with largest value
3) Determine action

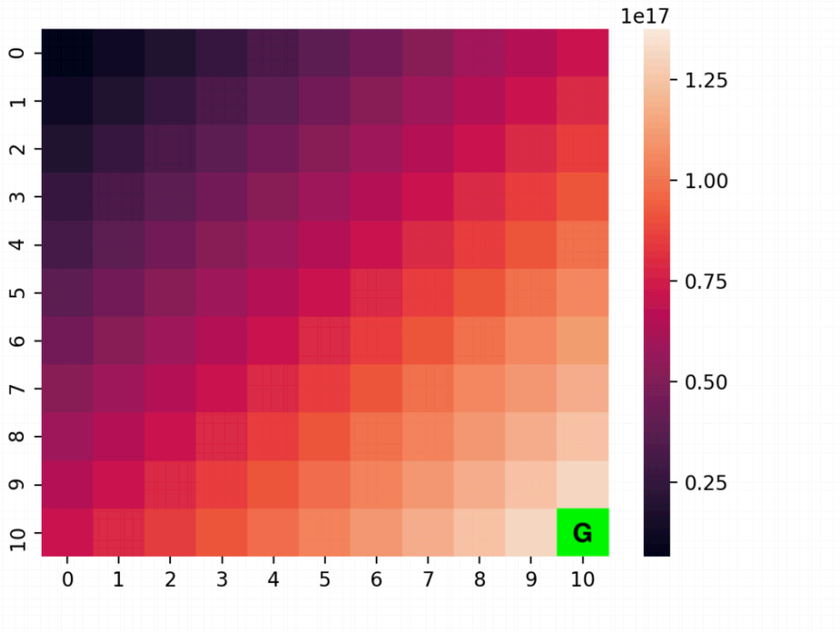
$$\pi(s) \leftarrow I(s, s'_\tau)$$

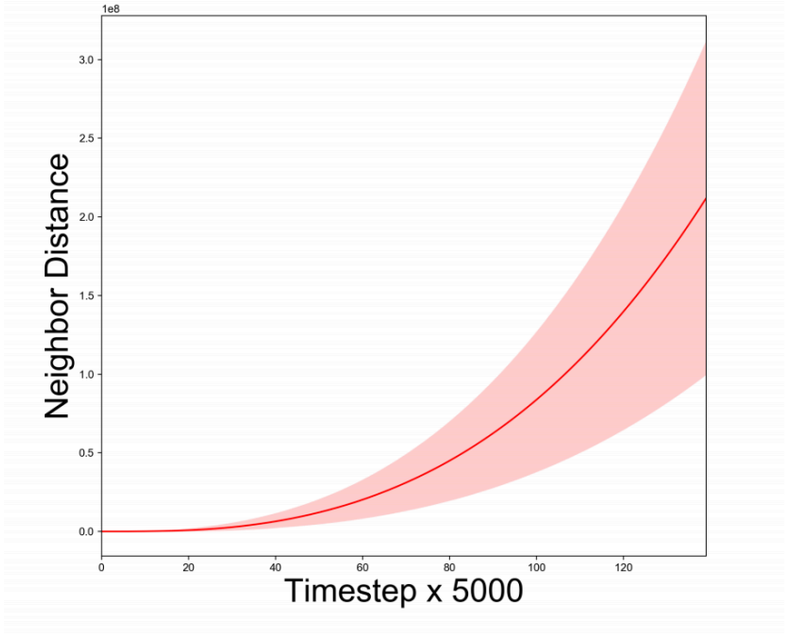
D3G Control Experiments

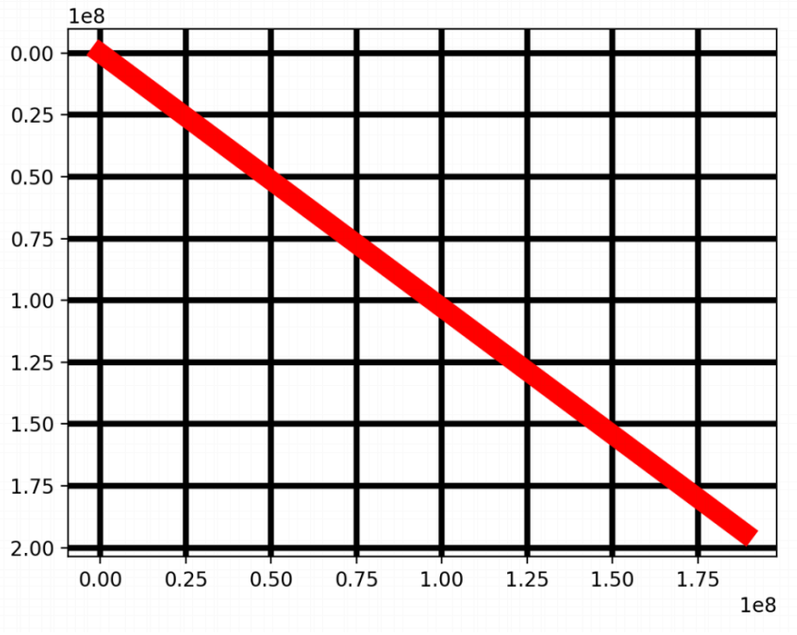
- 11x11 gridworld
- Mujoco tasks
- I(s,s') learned



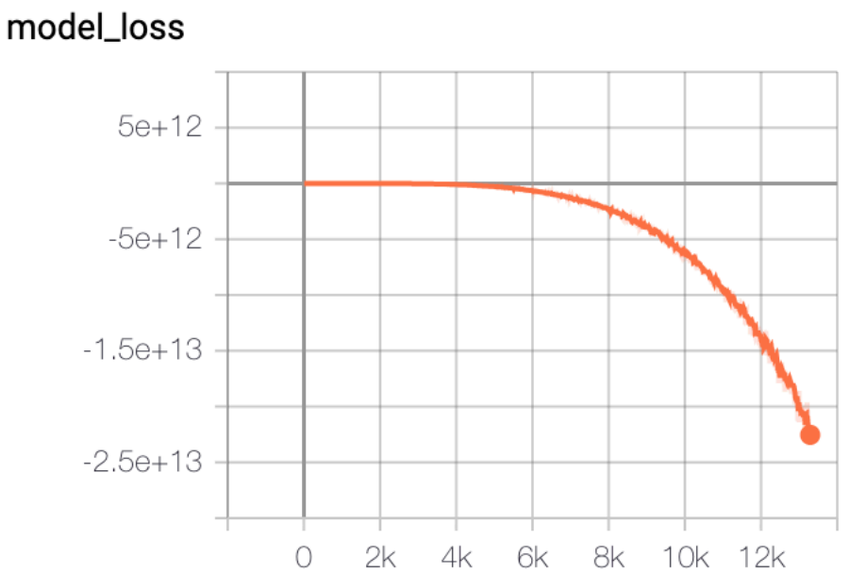
D3G Control Experiments

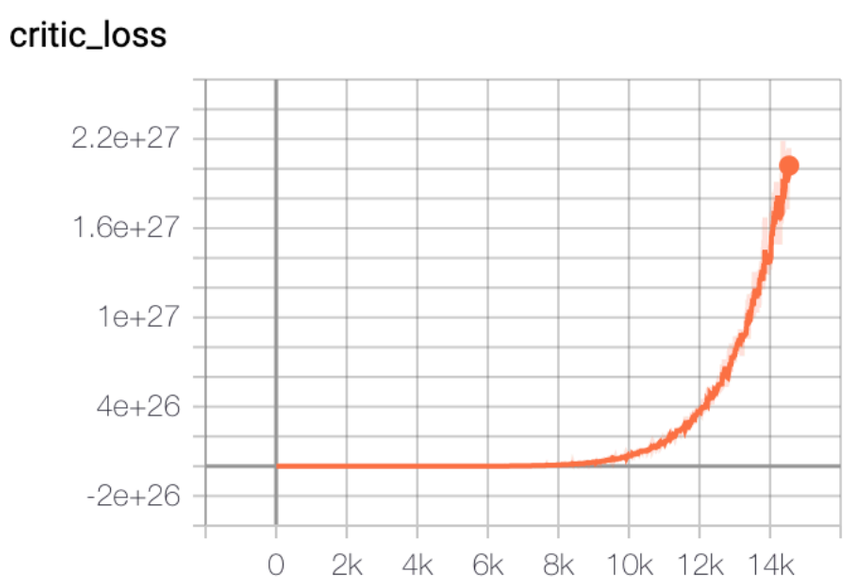






D3G Control Experiments



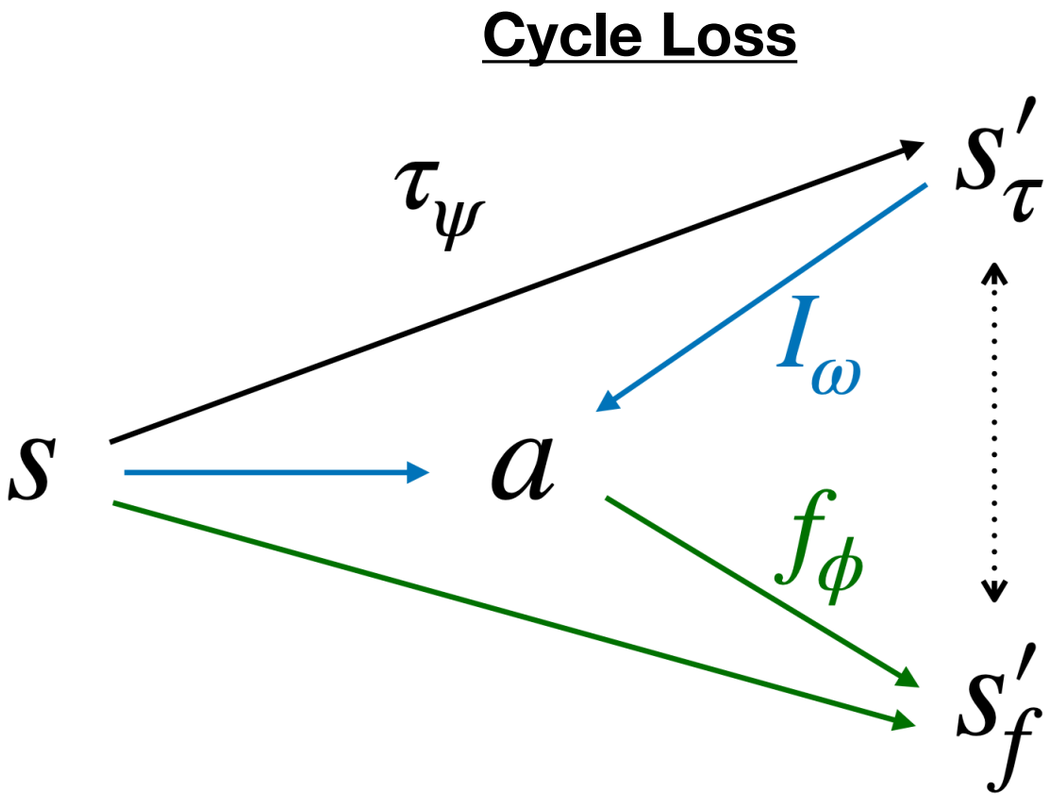


$$\mathcal{L}_\psi = -Q_\theta(s, \tau_\psi(s))$$

$$\mathcal{L}_\theta = \sum \|y - Q_{\theta_i}(s, s')\|$$

$$y = r + \gamma \min_{i=1,2} Q_{\theta'_i}(s', \tau_{\psi'}(s'))]$$

D3G Control Experiments

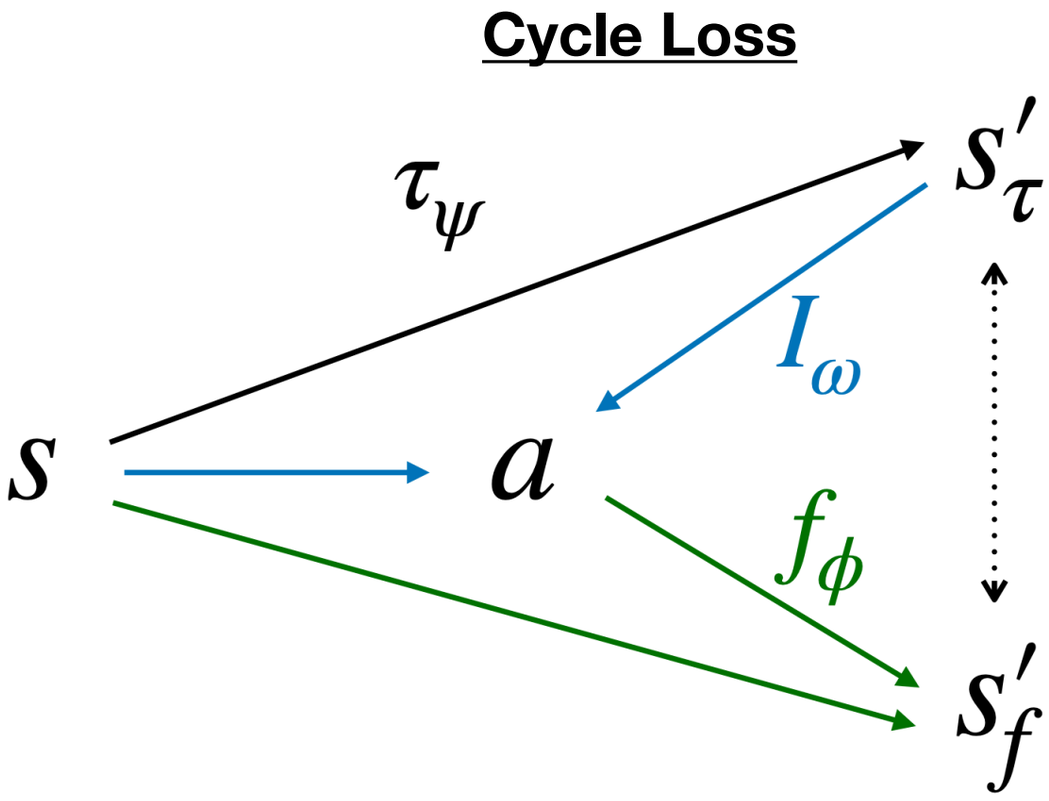


Model $\tau_\psi(s) = s'_\tau$

Inverse dynamics $I_\omega(s, s'_\tau) = a$

Forward dynamics $f_\phi(s, a) = s'_f$

D3G Control Experiments



Model $\tau_\psi(s) = s'_\tau$

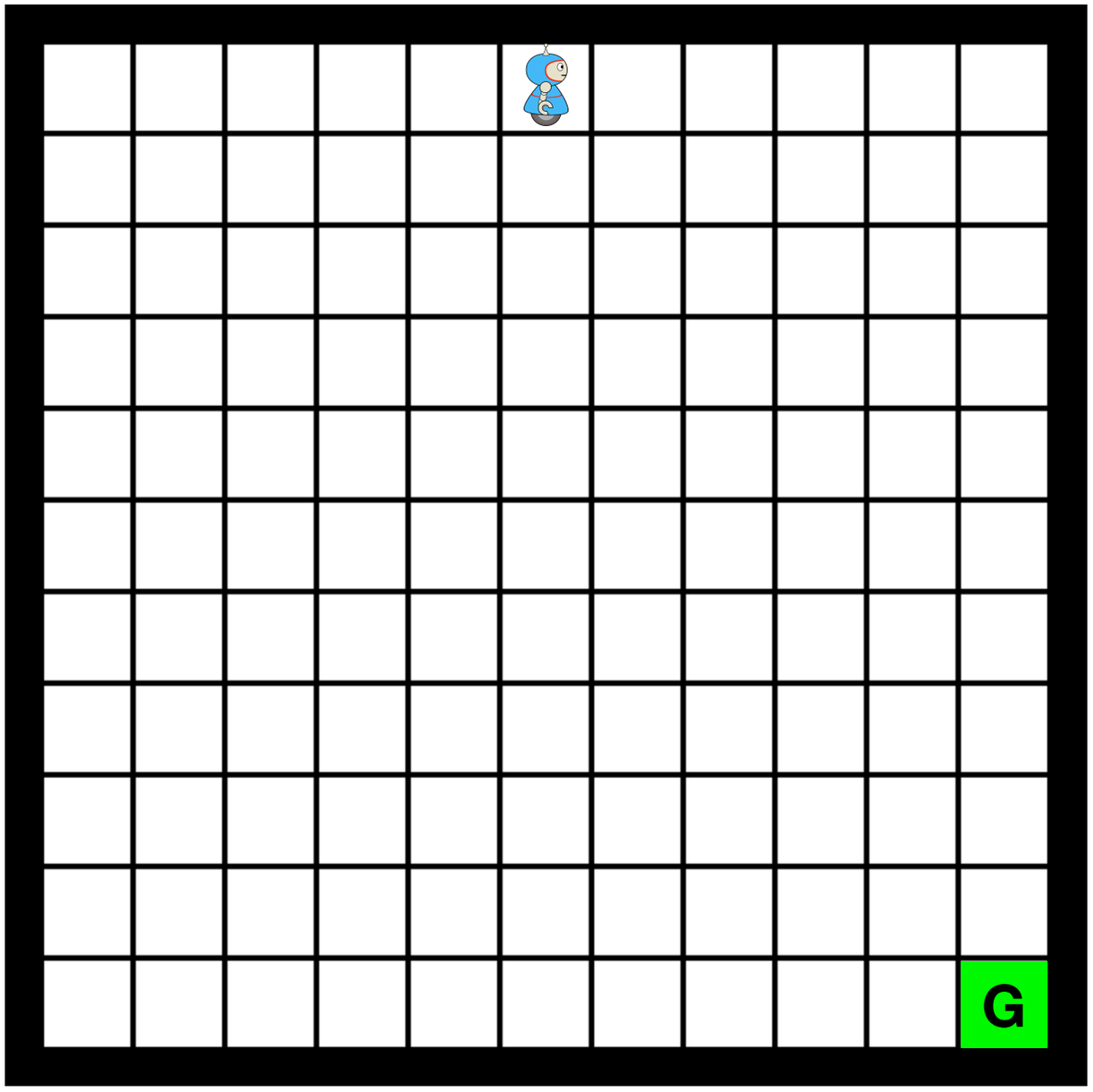
Inverse dynamics $I_\omega(s, s'_\tau) = a$

Forward dynamics $f_\phi(s, a) = s'_f$

$$\mathcal{L}_\psi = -Q_\theta(s, C(s, \tau_\psi(s)) + \beta \| \tau_\psi(s) - C(s, \tau_\psi(s)) \|$$

$$y = r + \gamma \min_{i=1,2} Q_{\theta'_i}(s', C(s', \tau_{\psi'}(s')))$$

D3G Control Experiments

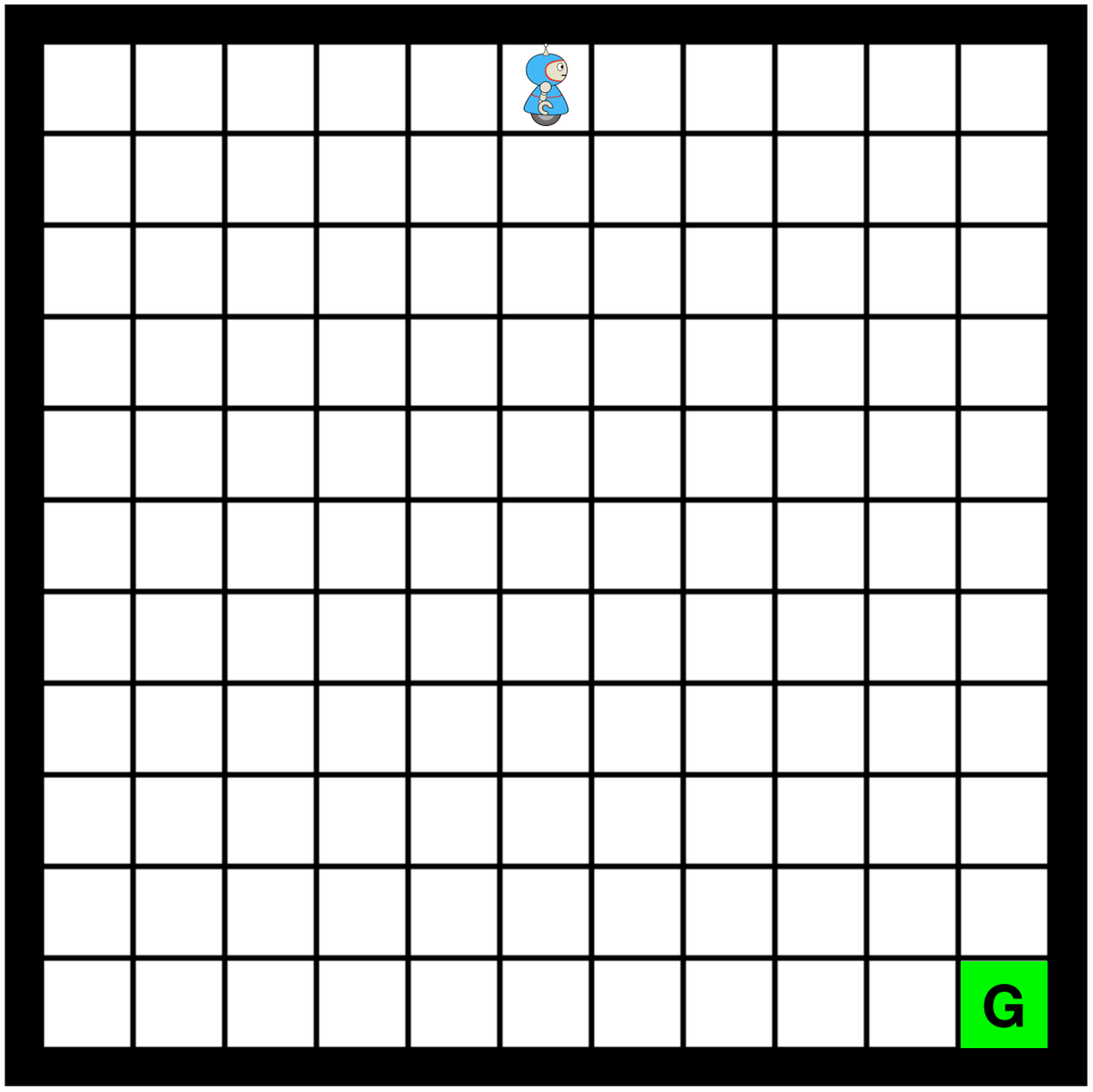


Training

1) Select state with largest value
2) Determine action
3) Predict next state

$$s'_\tau \leftarrow \tau(s)$$

D3G Control Experiments

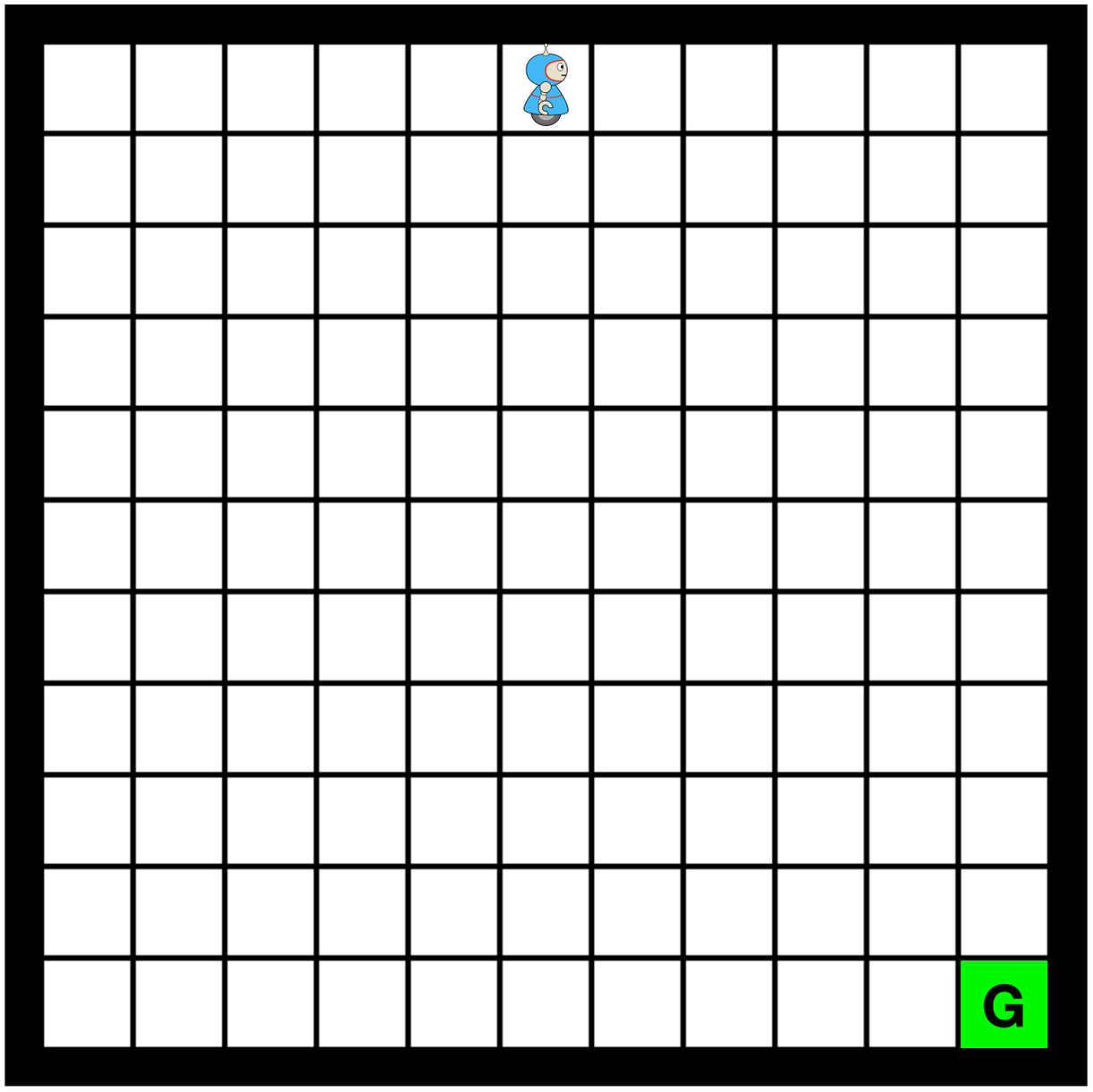


Training

1) Select state with largest value
2) Determine action
3) Predict next state

$$s'_\tau \leftarrow \tau(s)$$

D3G Control Experiments

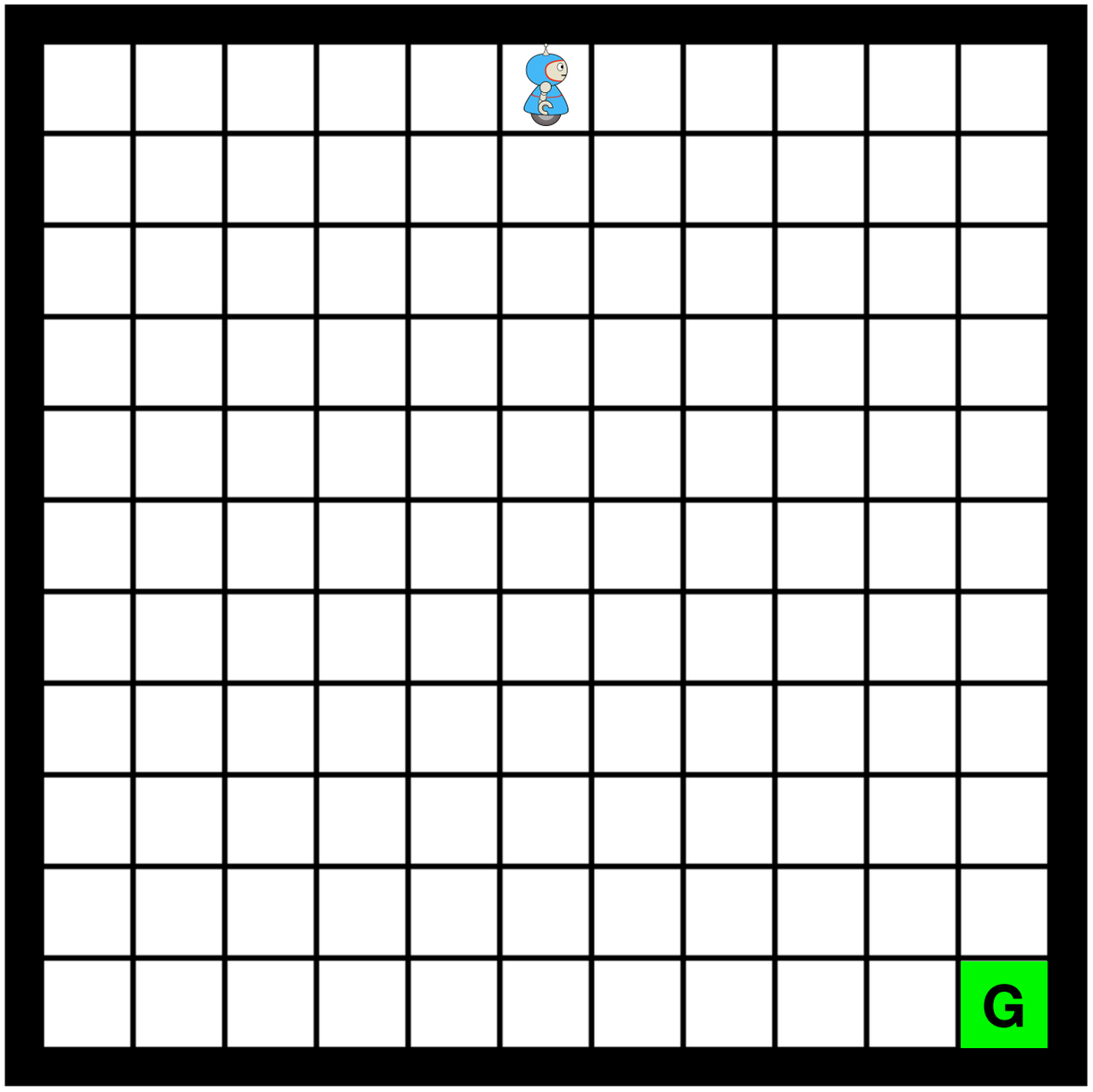


Training

1) Select state with largest value
2) Determine action
3) Predict next state

$$a \leftarrow I(s, s'_\tau)$$

D3G Control Experiments

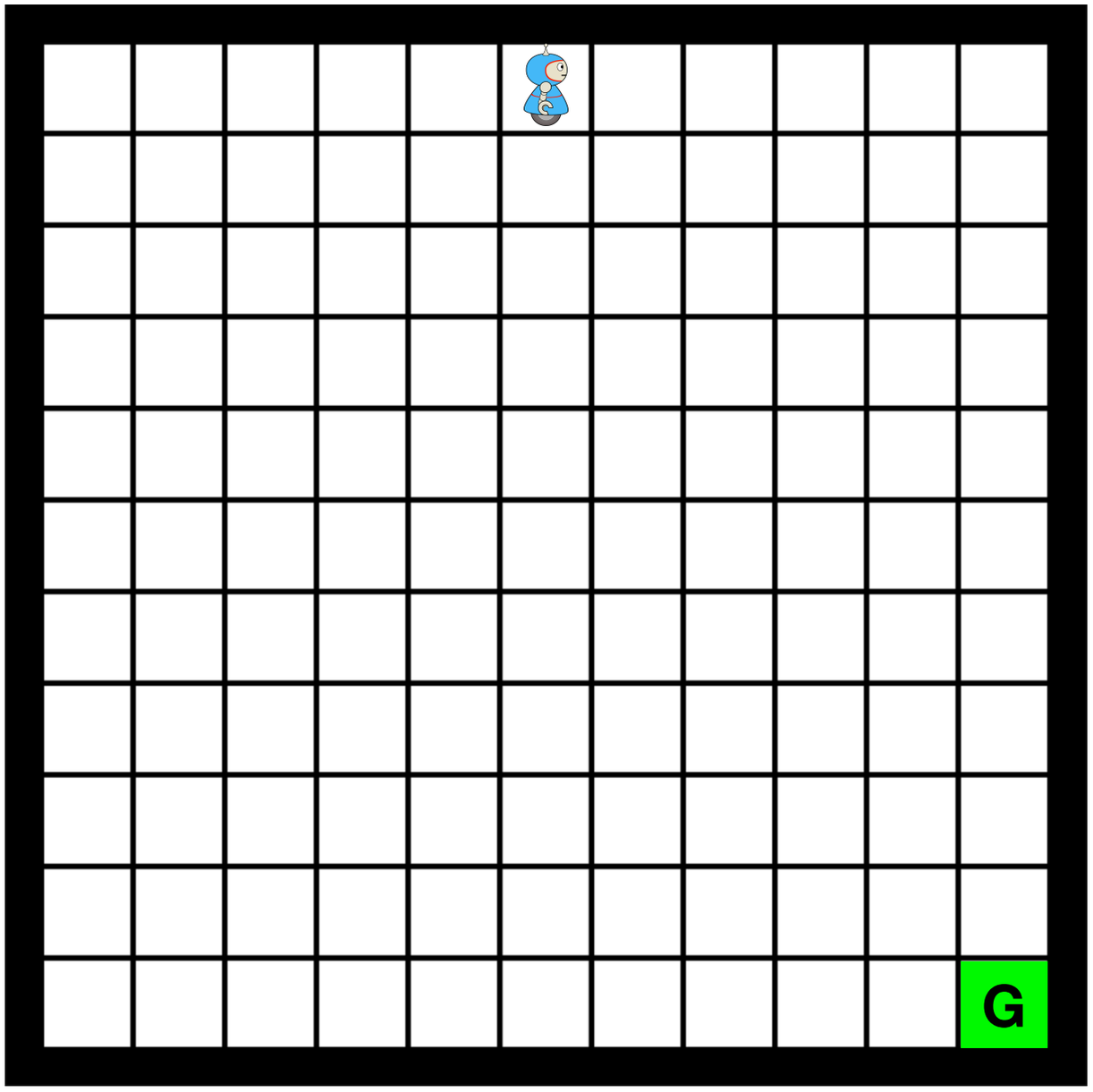


Training

1) Select state with largest value
2) Determine action
3) Predict next state

$$s'_f \leftarrow f(s, a)$$

D3G Control Experiments

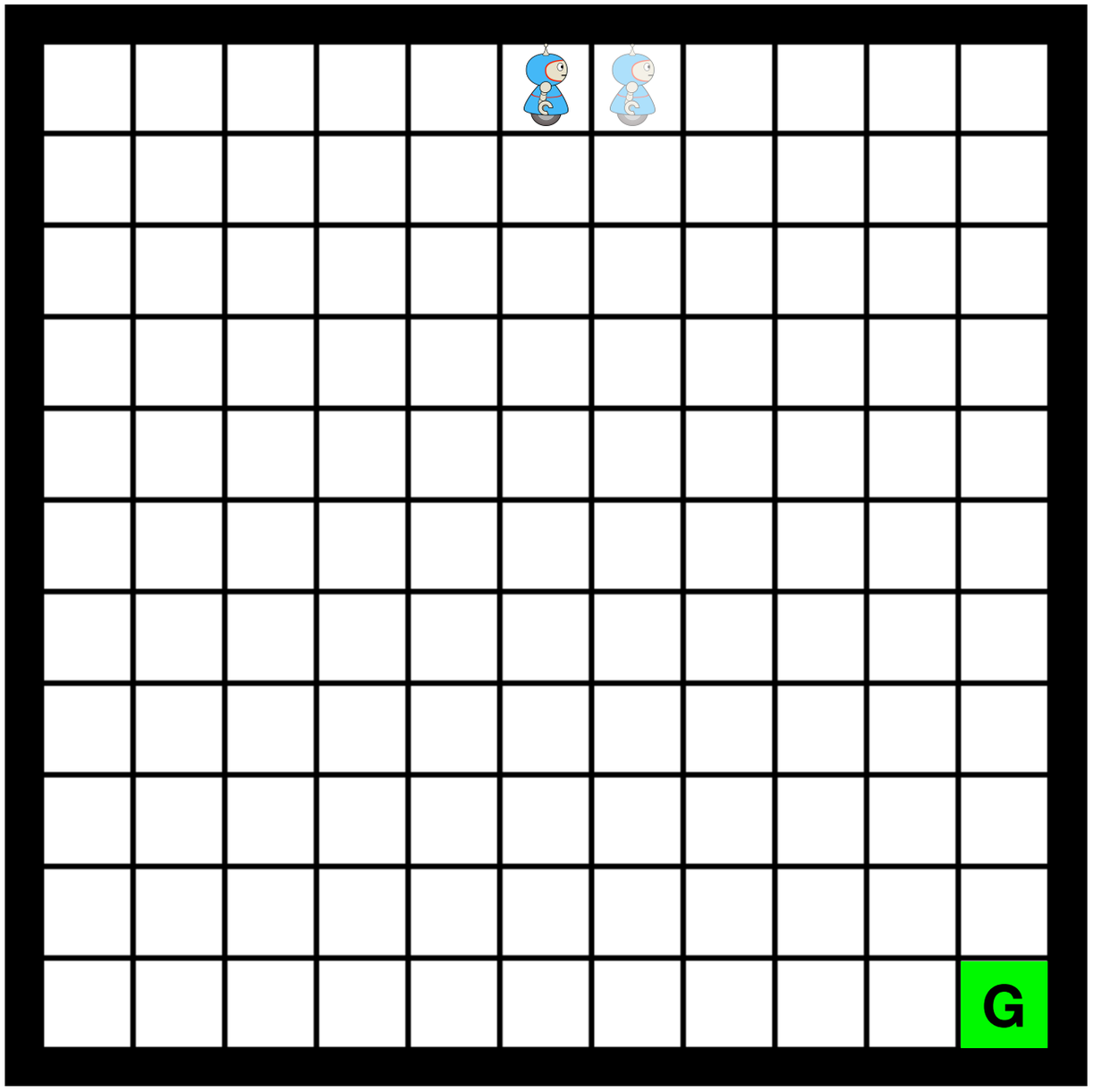


Training

1) Select state with largest value
2) Determine action
3) Predict next state

$$s'_\tau \leftarrow \tau(s)$$

D3G Control Experiments

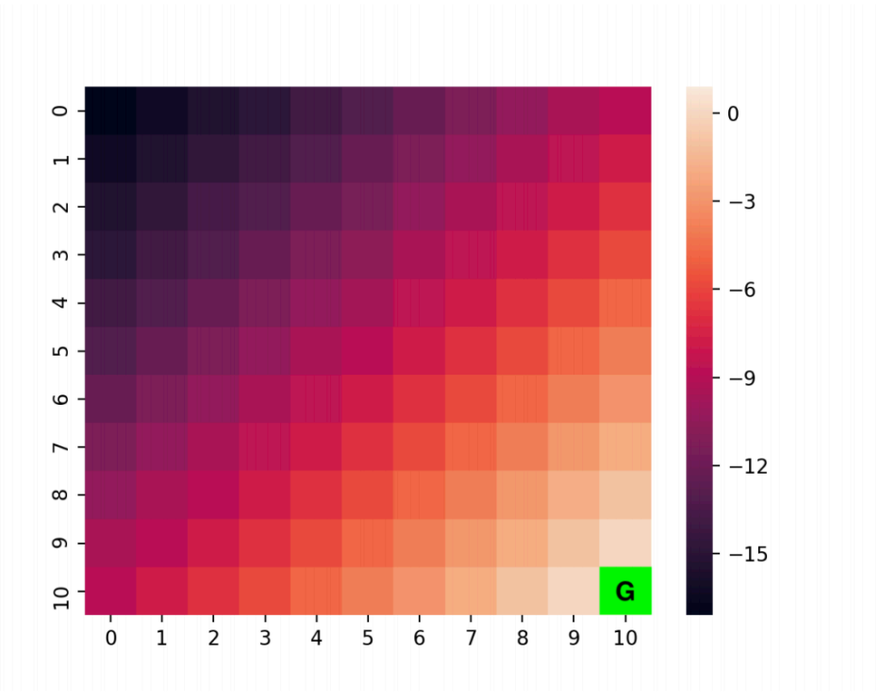


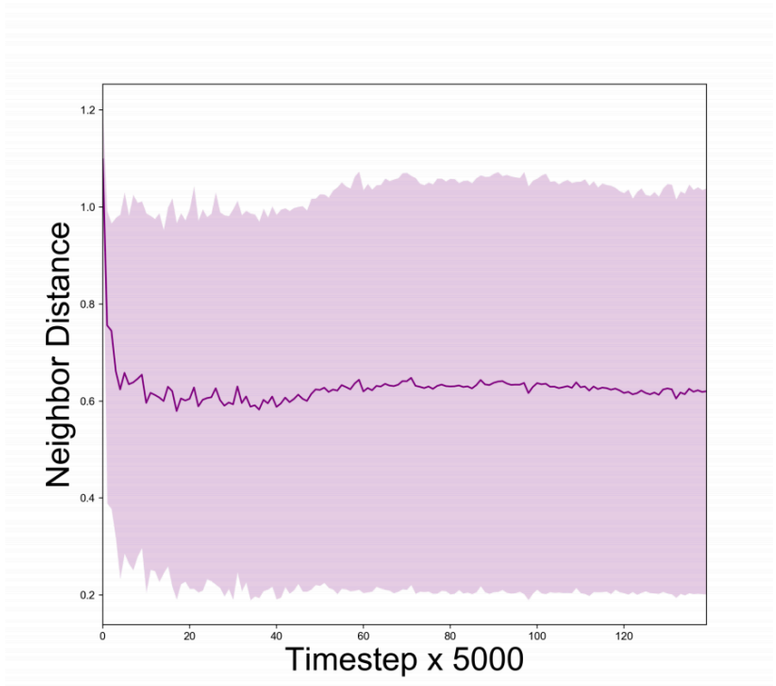
Training

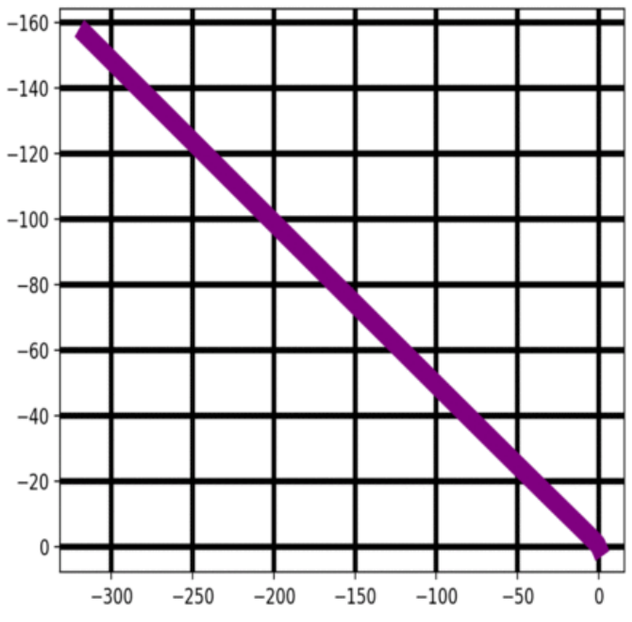
1) Select state with largest value
2) Determine action
3) Predict next state

$$s'_\tau \leftarrow \tau(s)$$

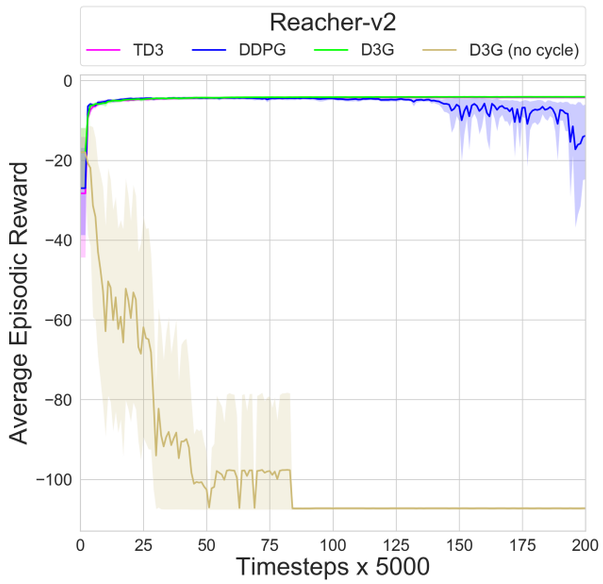
D3G Control Experiments

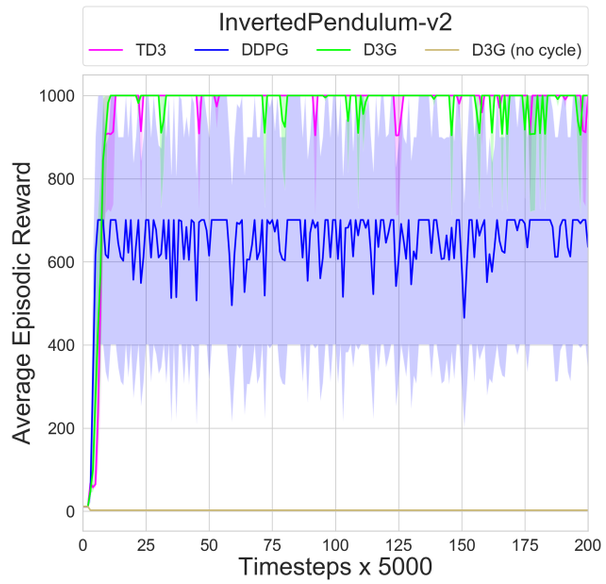


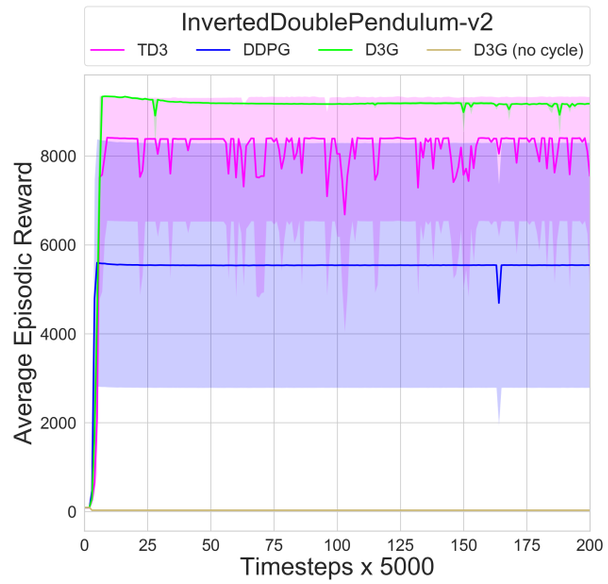


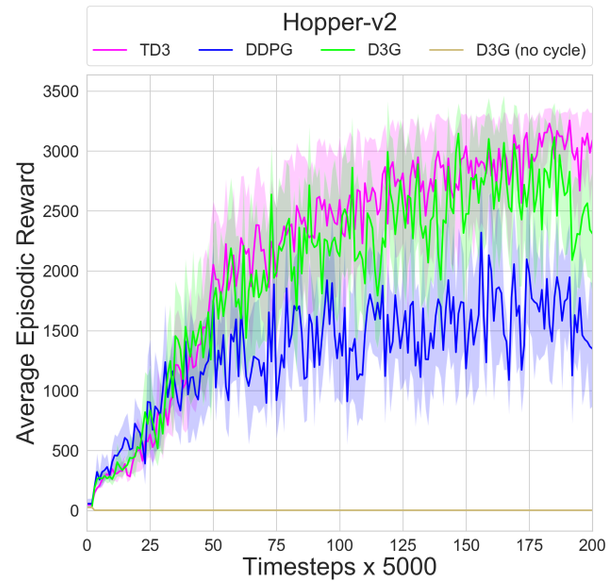


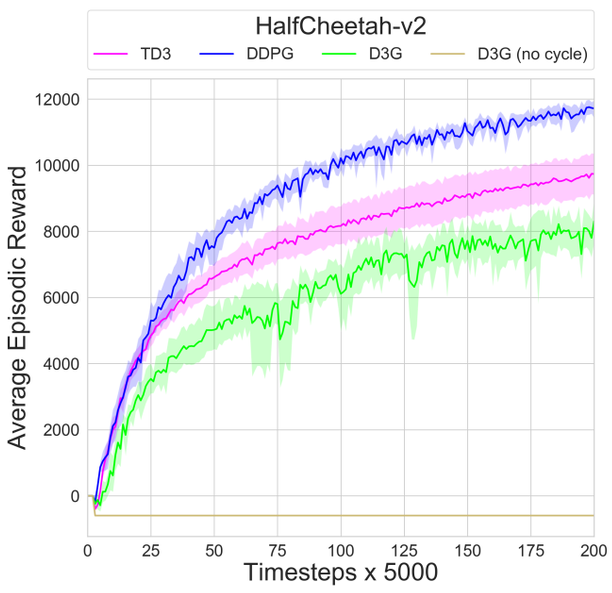
D3G Control Experiments

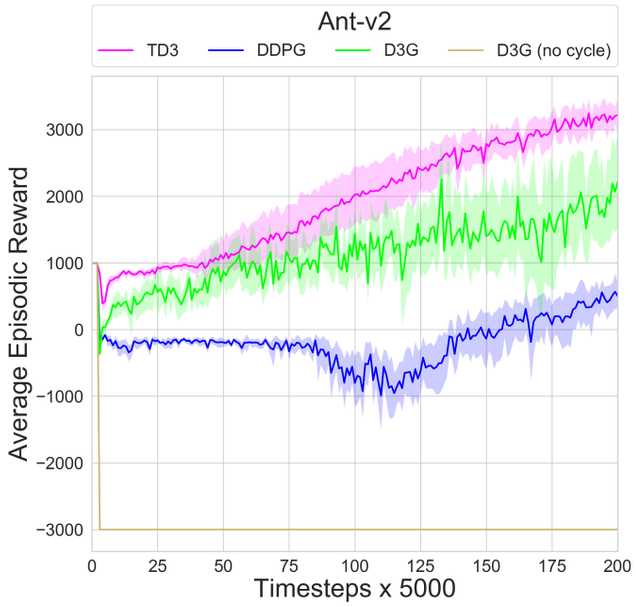


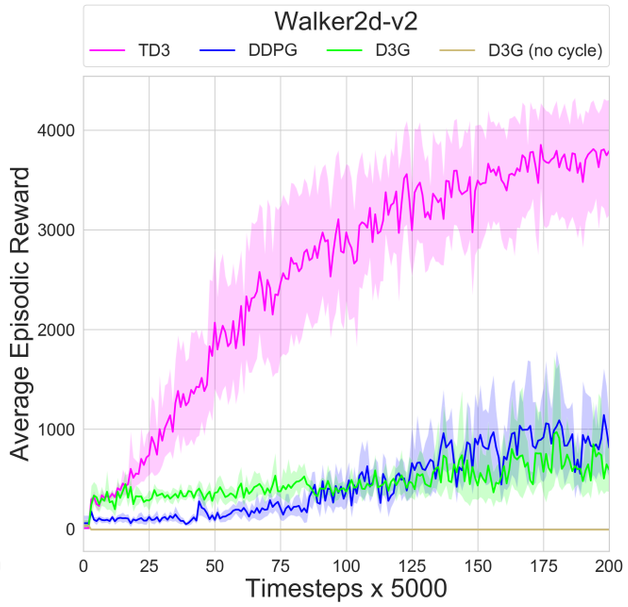


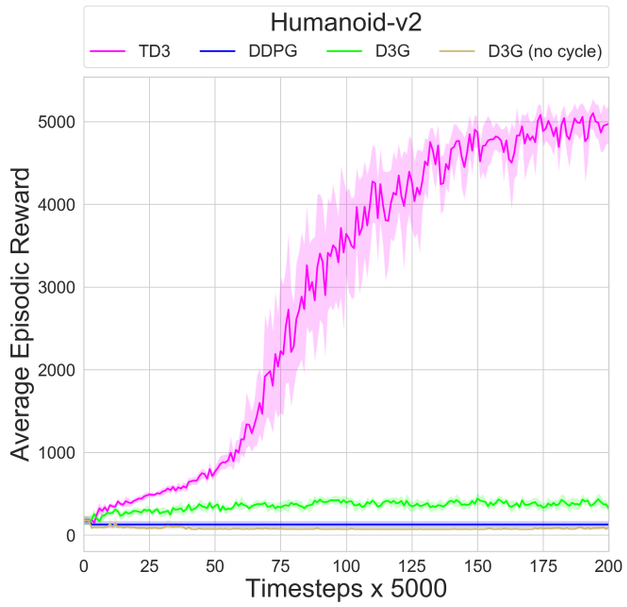




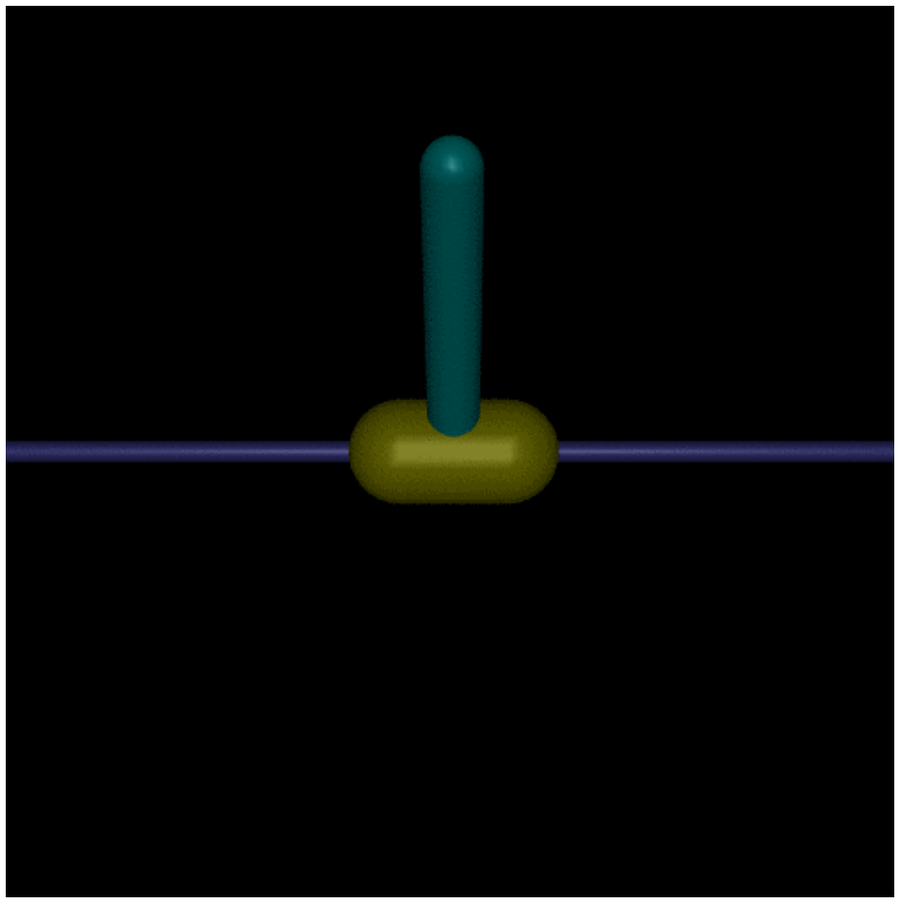


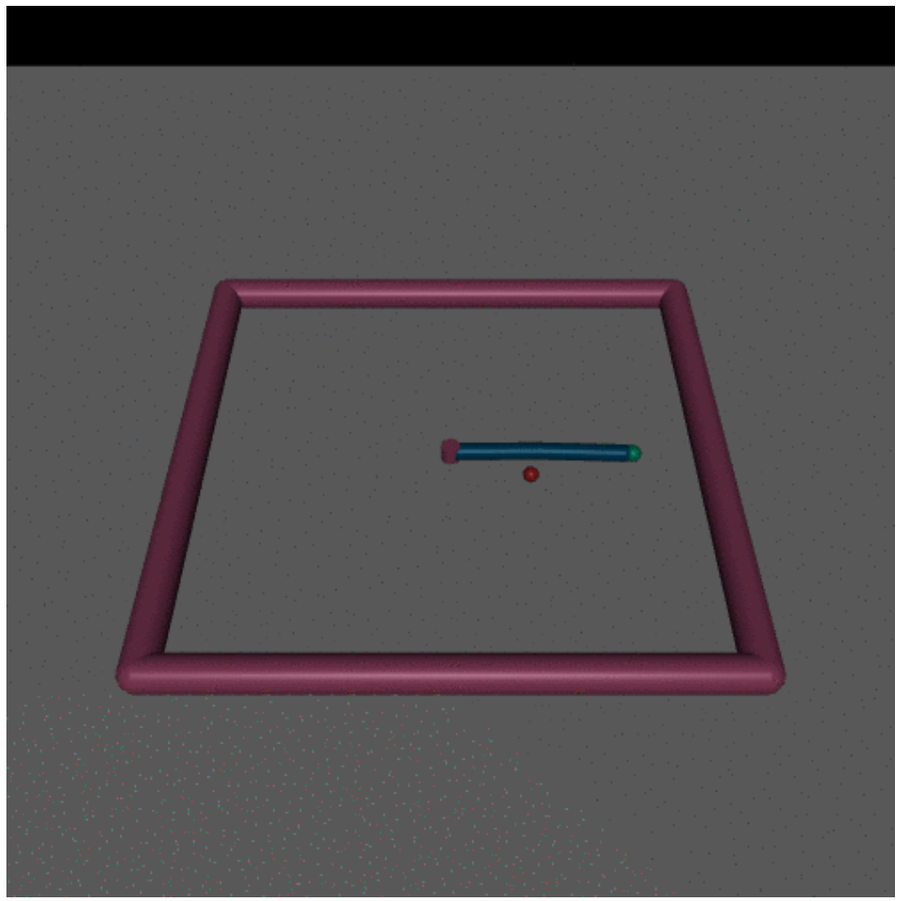




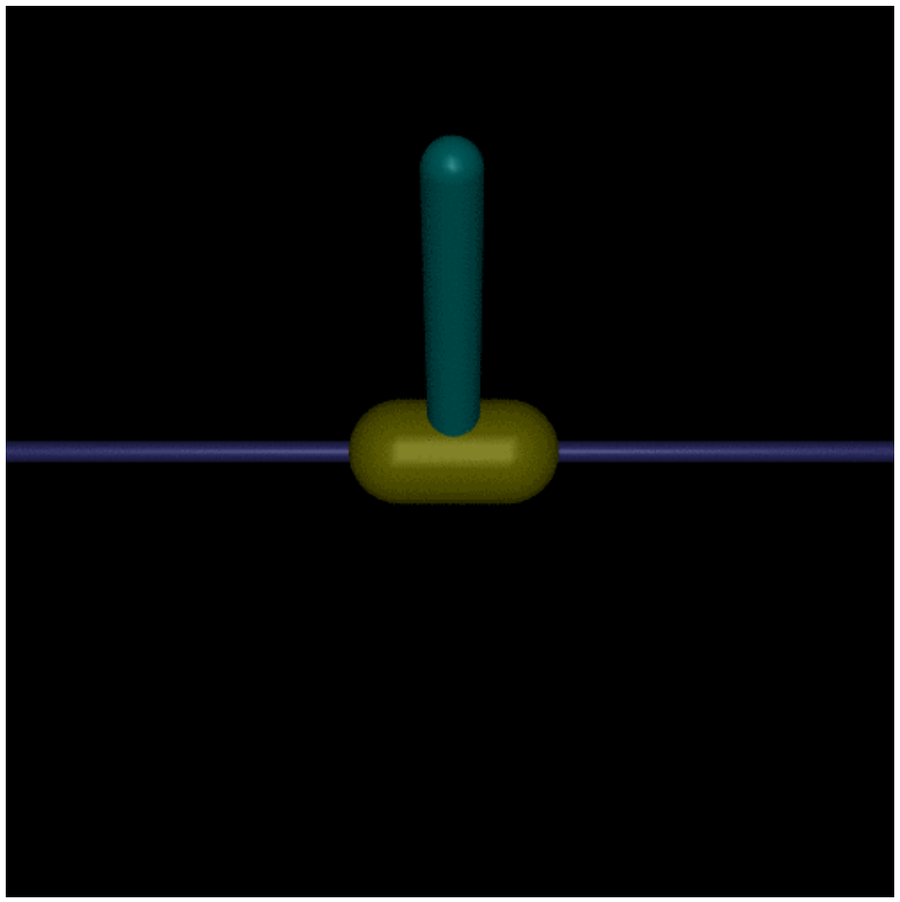


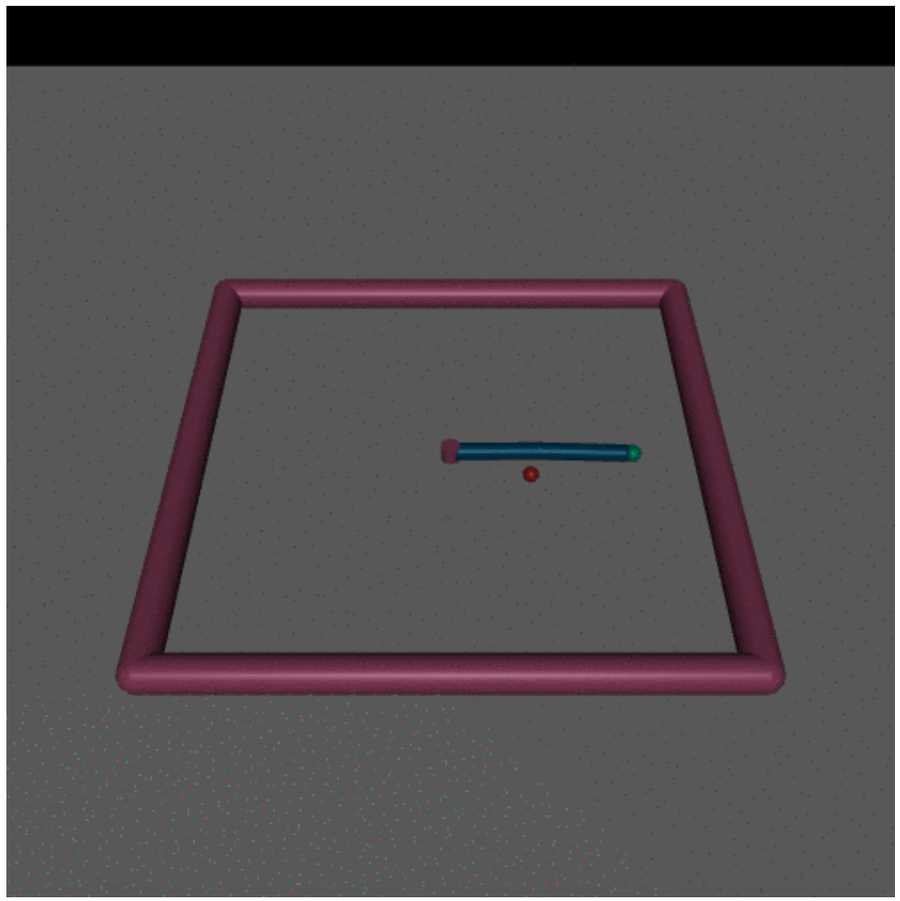
Learning from Observation





Learning from Observation



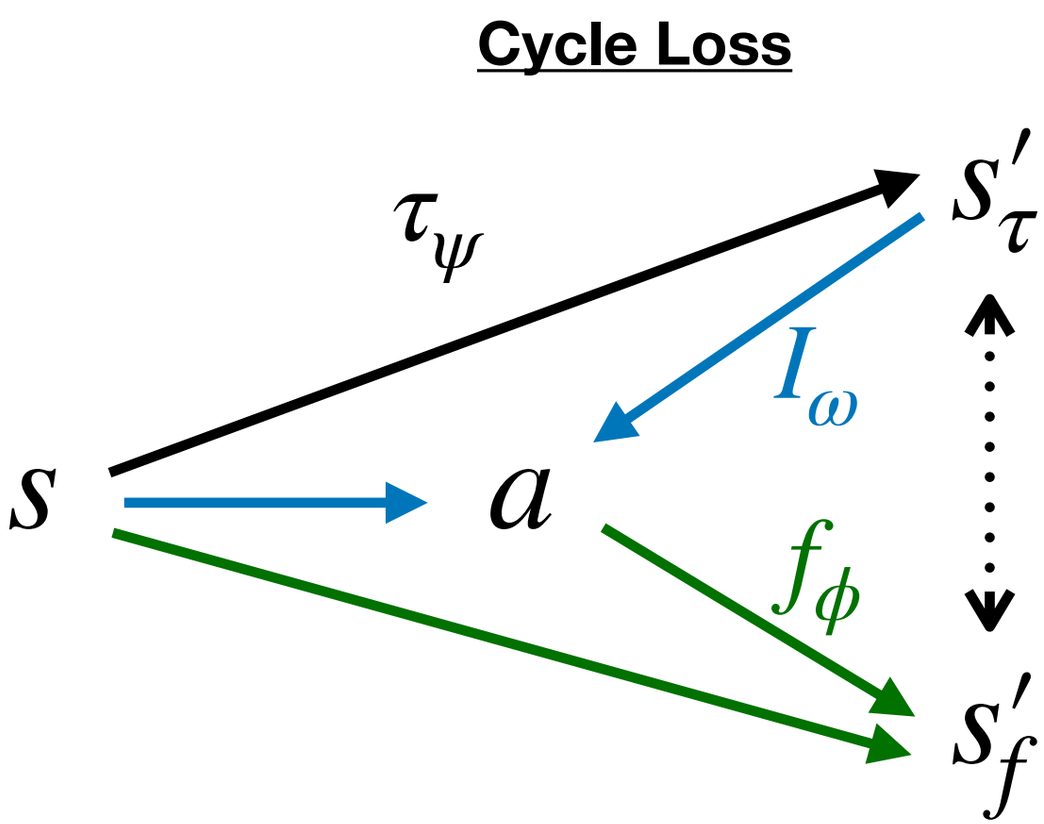


$$Q(s, s') = r + \gamma Q(s, \tau(s))$$

Learning from Observation

- Given demonstration data $\langle s_1, r_1, d_1, s_2, r_2, d_2, \ldots \rangle$
- Train $Q(s, s')$ and $\tau(s)$

Learning from Observation



Model $\tau_\psi(s) = s'_\tau$

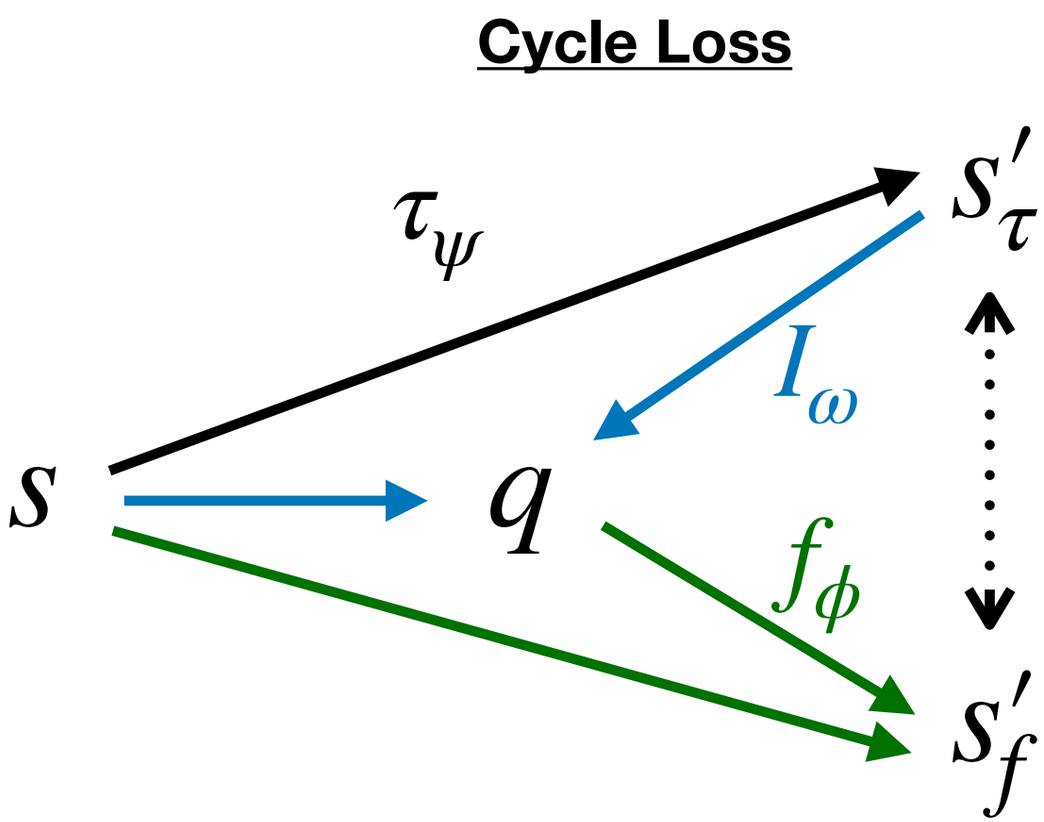
Inverse dynamics $I_\omega(s, s'_\tau) = a$

Forward dynamics $f_\phi(s, a) = s'_f$

$$\mathcal{L}_\psi = -Q_\theta(s, C(s, \tau_\psi(s)) + \beta \| \tau_\psi(s) - C(s, \tau_\psi(s)) \|$$

$$y = r + \gamma \min_{i=1,2} Q_{\theta'_i}(s', C(s', \tau_{\psi'}(s')))$$

Learning from Observation



Model $\tau_\psi(s) = s'_\tau$

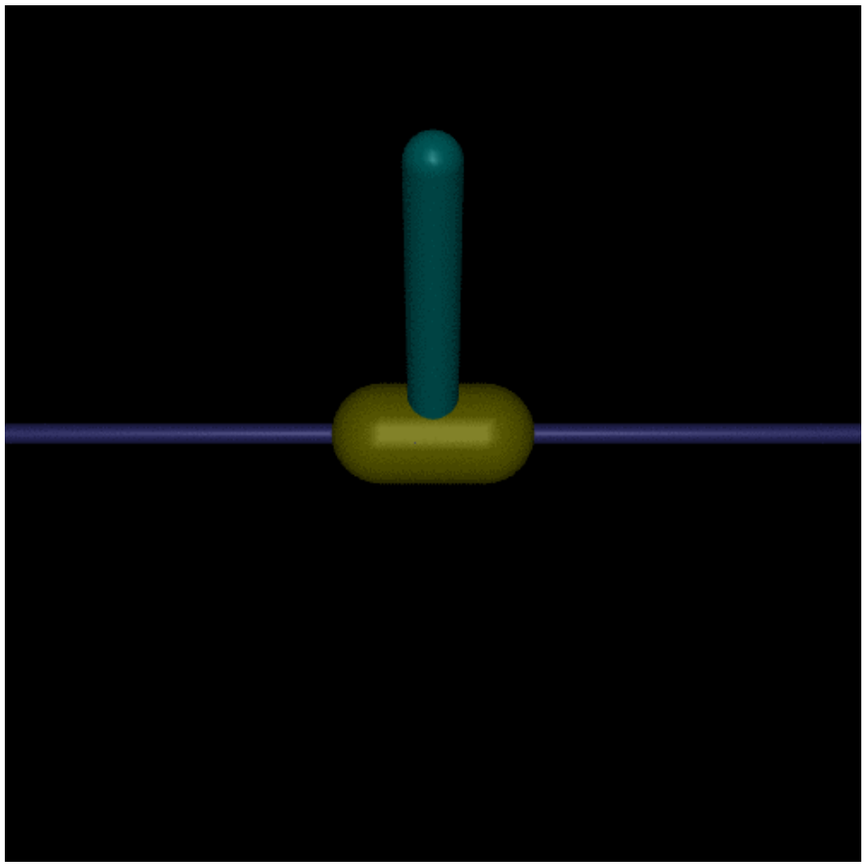
QSS $Q(s, s'_\tau) = q$

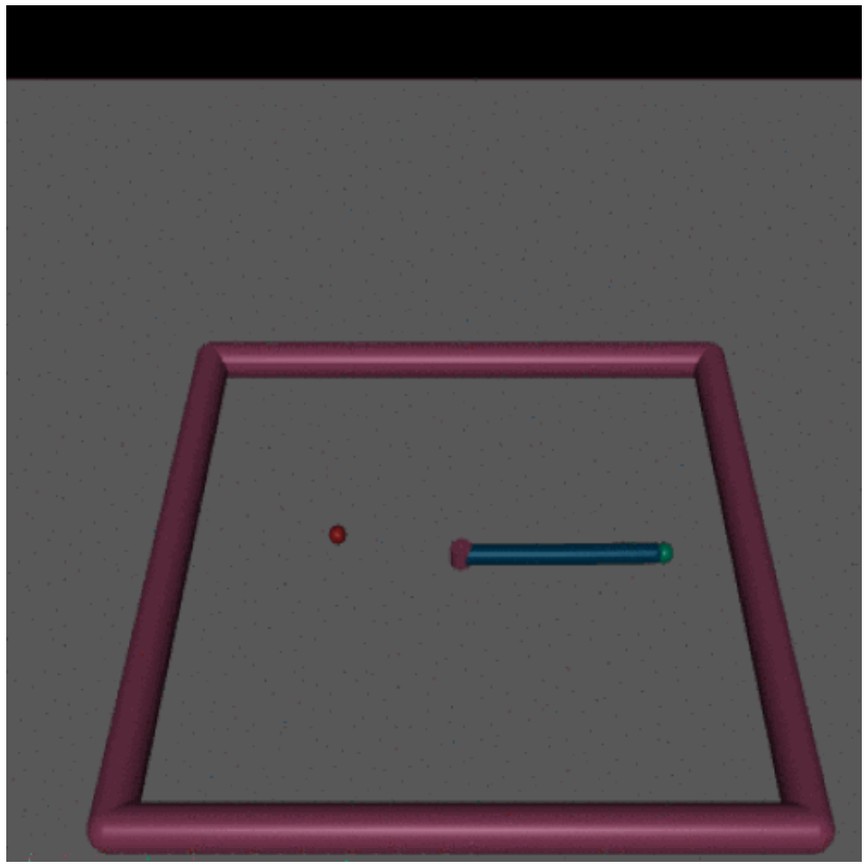
Forward dynamics $f_\phi(s, q) = s'_f$

$$\mathcal{L}_\psi = -Q_\theta(s, C(s, \tau_\psi(s)) + \beta \| \tau_\psi(s) - C(s, \tau_\psi(s)) \|$$

$$y = r + \gamma \min_{i=1,2} Q_{\theta'_i}(s', C(s', \tau_{\psi'}(s')))$$

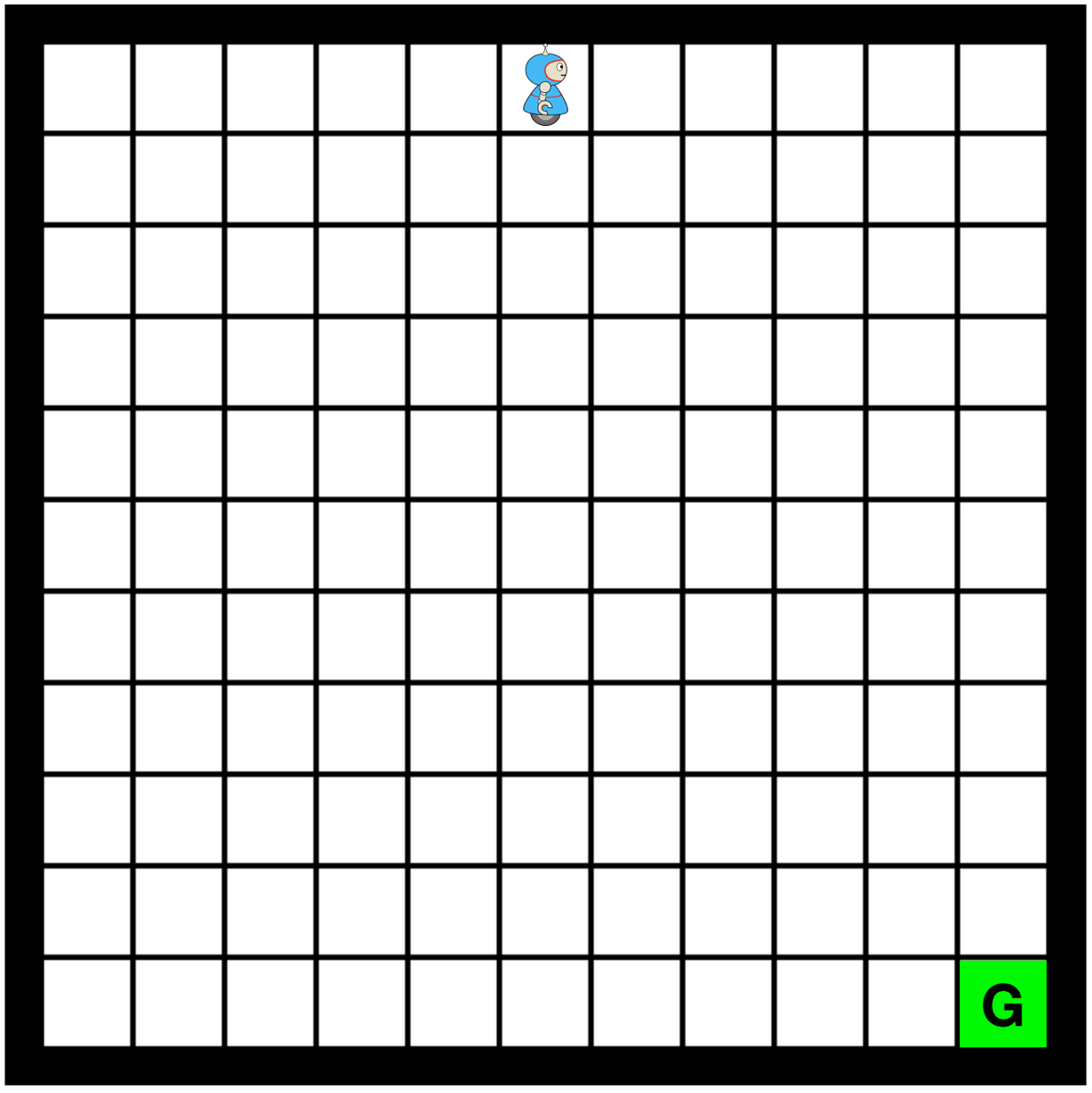
Learning from Observation





$$s \rightarrow C(s, \tau(s)) \rightarrow \ldots$$

Learning from Observation

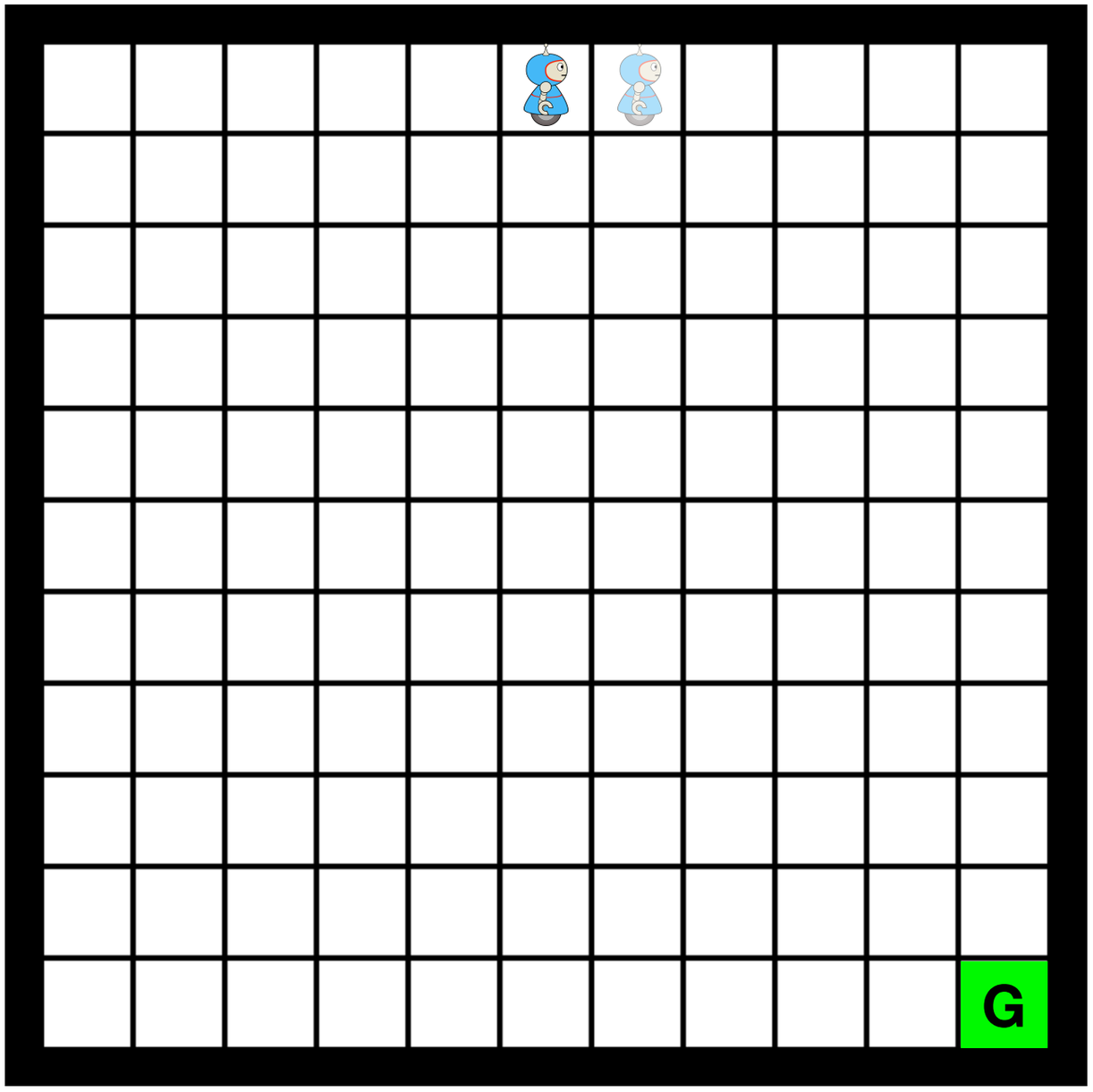


Control

1) Select state with largest value
2) Determine action

$$s'_\tau \leftarrow \tau(s)$$

Learning from Observation

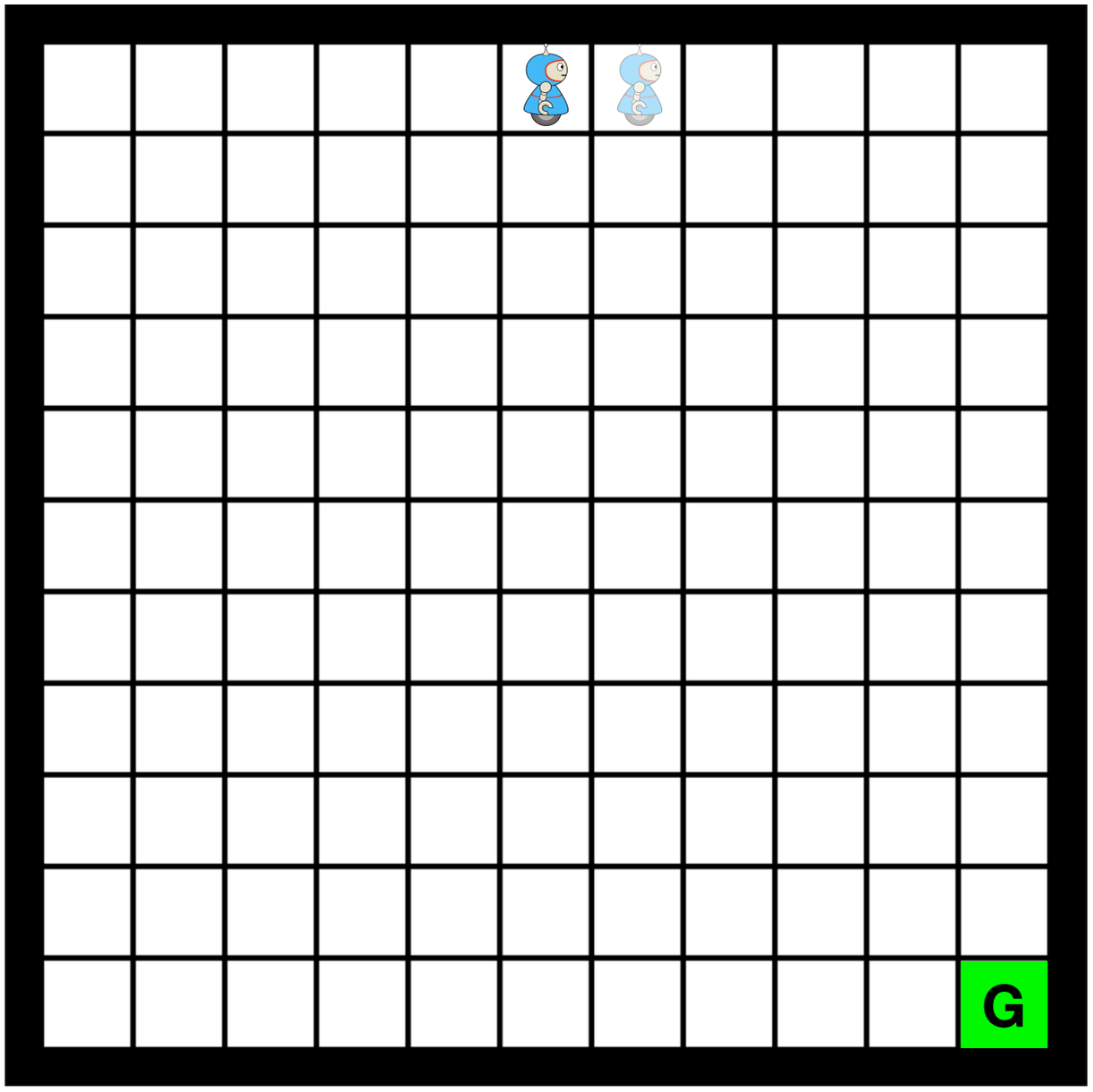


Control

1) Select state with largest value
2) Determine action

$$s'_\tau \leftarrow \tau(s)$$

Learning from Observation



Control

1) Select state with largest value
2) Determine action

$$\pi(s) \leftarrow I(s, s'_T)$$

Learning from Observation

Table: Learning from observation with noise injected into policies

	Reacher-v2			InvertedPendulum-v2		
% Random	π_o	BCO	D3G	π_o	BCO	D3G
0	-4.1 $\pm$ 0.7	**-4.2 $\pm$ 0.6**	-14.7 $\pm$ 30.5	1000 $\pm$ 0	**1000 $\pm$ 0**	3.0 $\pm$ 0.9
25	-12.5 $\pm$ 1.0	-4.3 $\pm$ 0.6	**-4.2 $\pm$ 0.6**	52.3 $\pm$ 3.7	**1000 $\pm$ 0**	602.1 $\pm$ 487.4
50	-22.6 $\pm$ 0.9	-4.9 $\pm$ 0.7	**-4.2 $\pm$ 0.6**	18.0 $\pm$ 2.4	12.1 $\pm$ 8.3	**900.2 $\pm$ 299.2**
75	-32.6 $\pm$ 0.4	-6.6 $\pm$ 1.3	**-4.6 $\pm$ 0.6**	11.4 $\pm$ 1.3	12.1 $\pm$ 8.3	**1000 $\pm$ 0**
100	-40.6 $\pm$ 0.5	-9.7 $\pm$ 0.8	**-6.4 $\pm$ 0.7**	8.6 $\pm$ 0.3	31.0 $\pm$ 4.7	**1000 $\pm$ 0**

Conclusion

- Introduced new value function, Q(s,s’)
- Described predictive model for maximizing these values
- Showed benefits in redundant action spaces, transfer across permuted actions, and learning from observation
- Code: https://github.com/uber-research/D3G