Abstraction in Reinforcement Learning

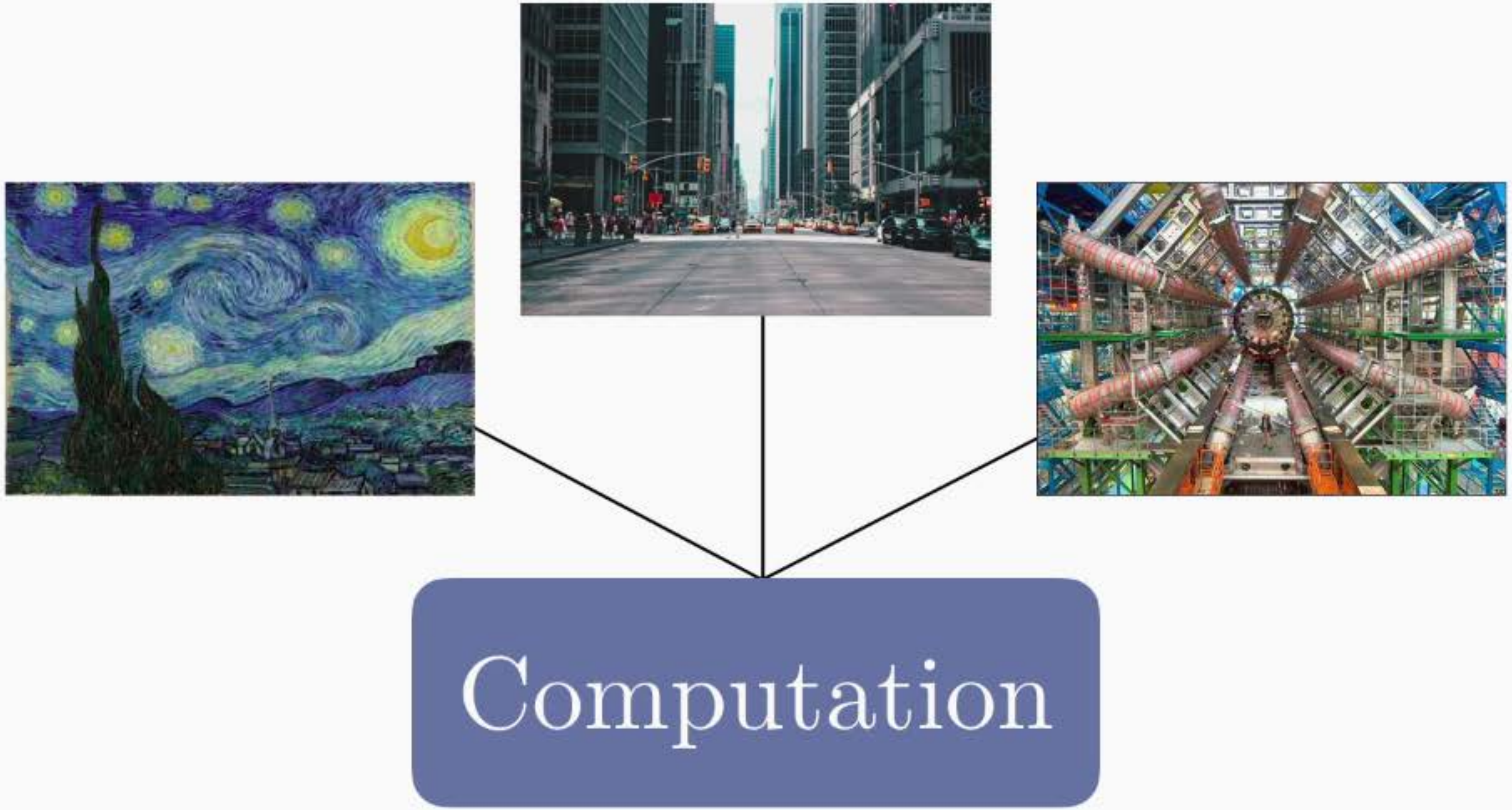
David Abel
Brown University
david_abel@brown.edu

Microsoft Research, February 24th, 2020

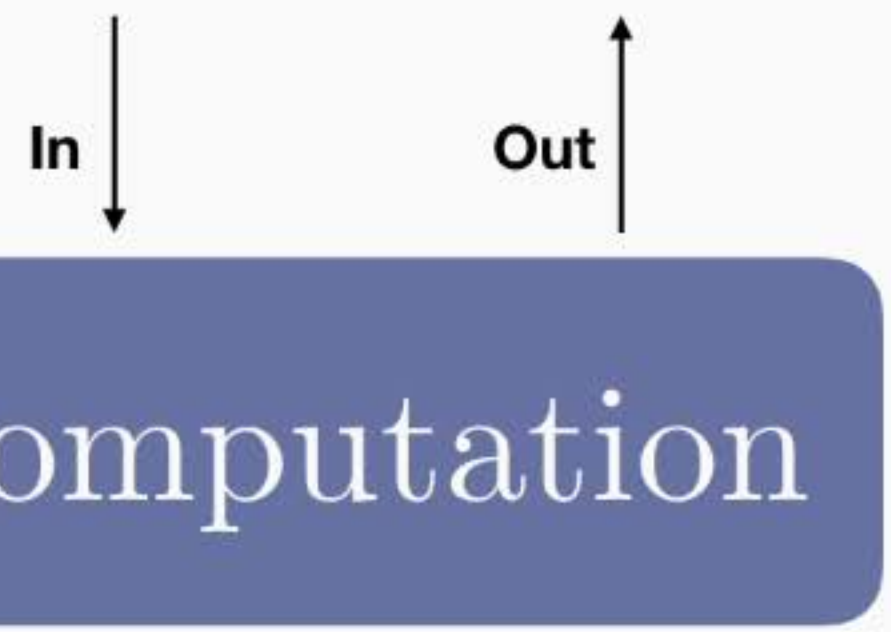
Systematized Problem Solving

Computation

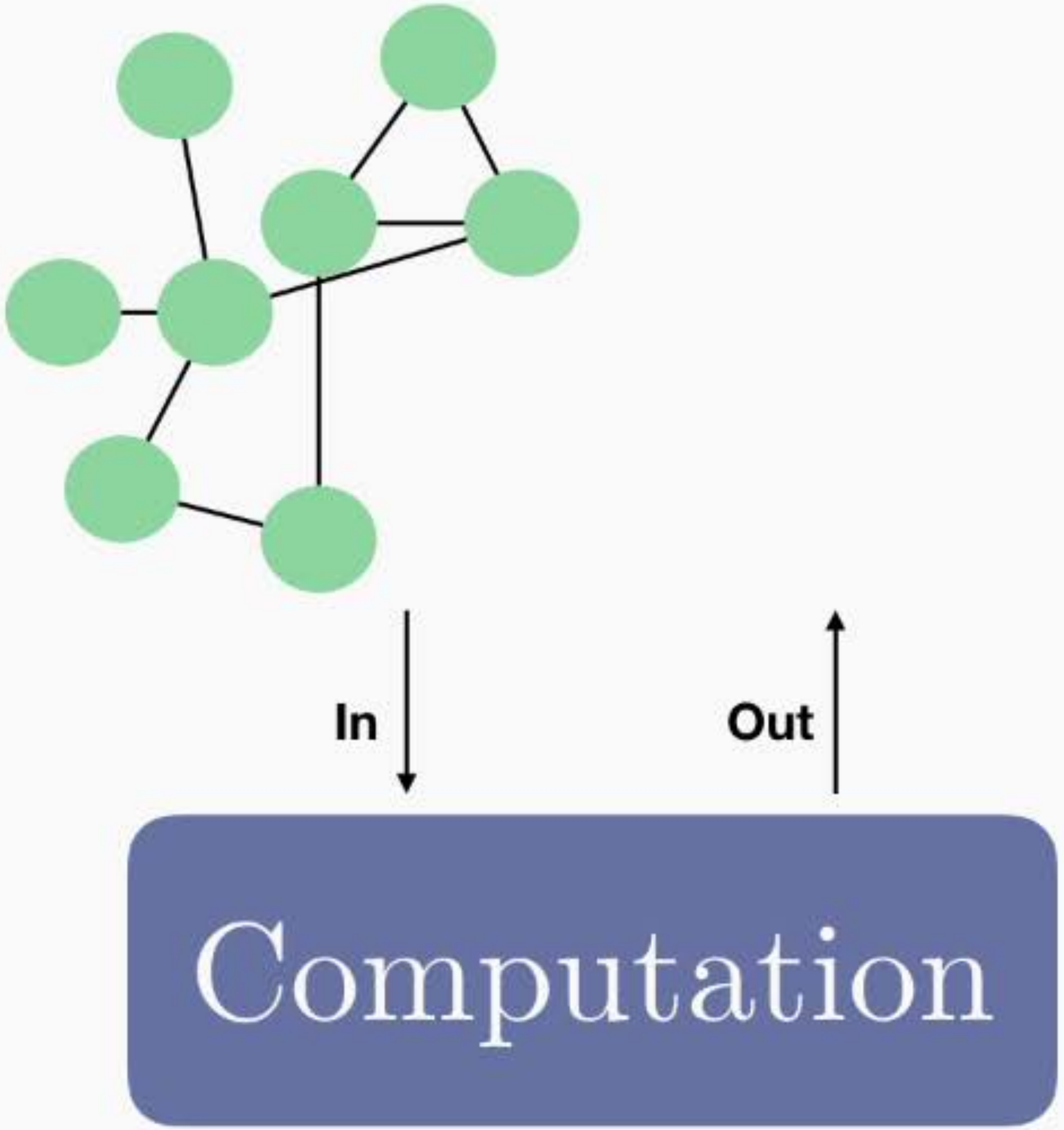
Systematized Problem Solving



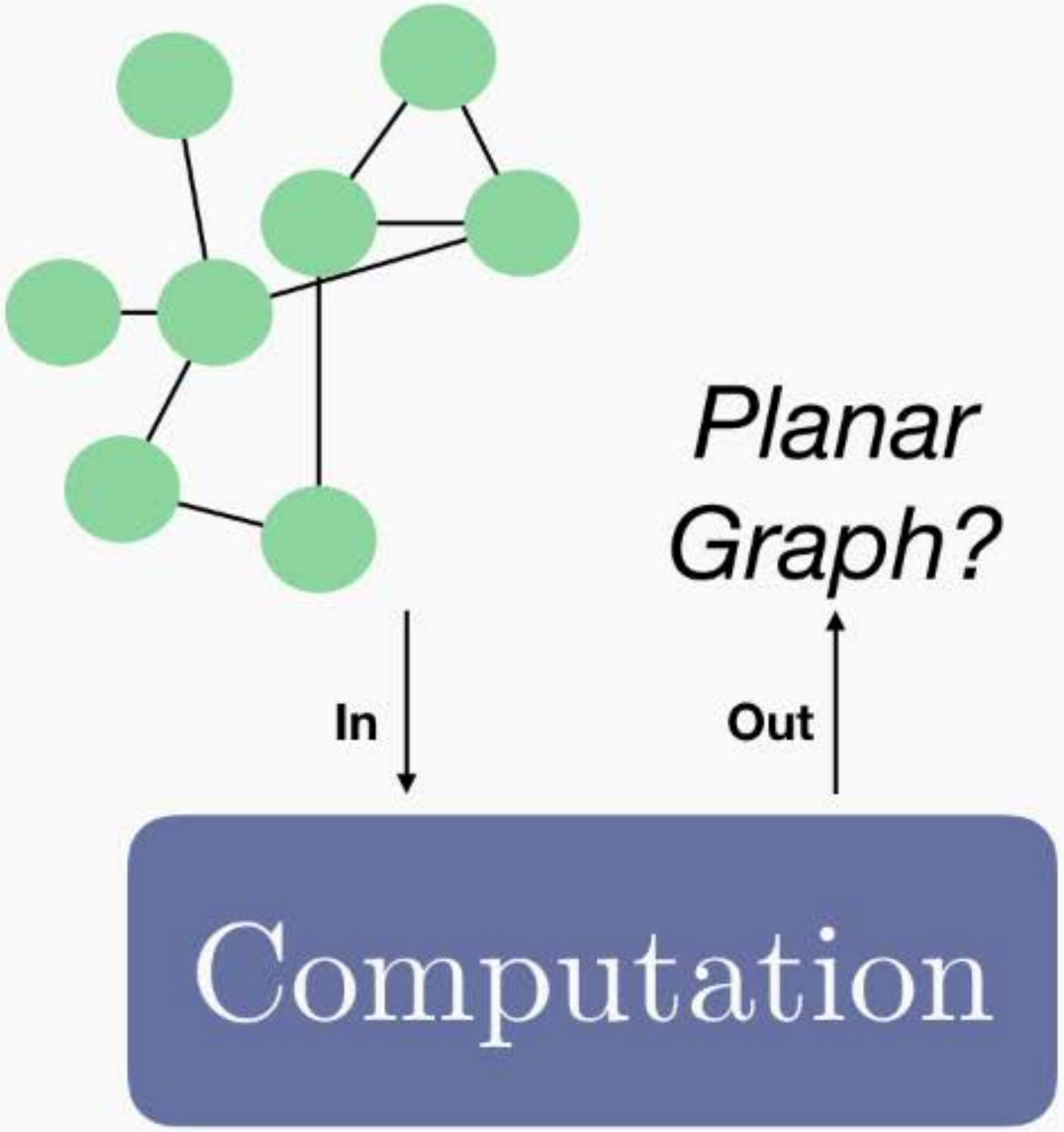
Systematized Problem Solving



Systematized Problem Solving



Systematized Problem Solving



Systematized Problem Solving

Planar Graph?

In

Out

Computation

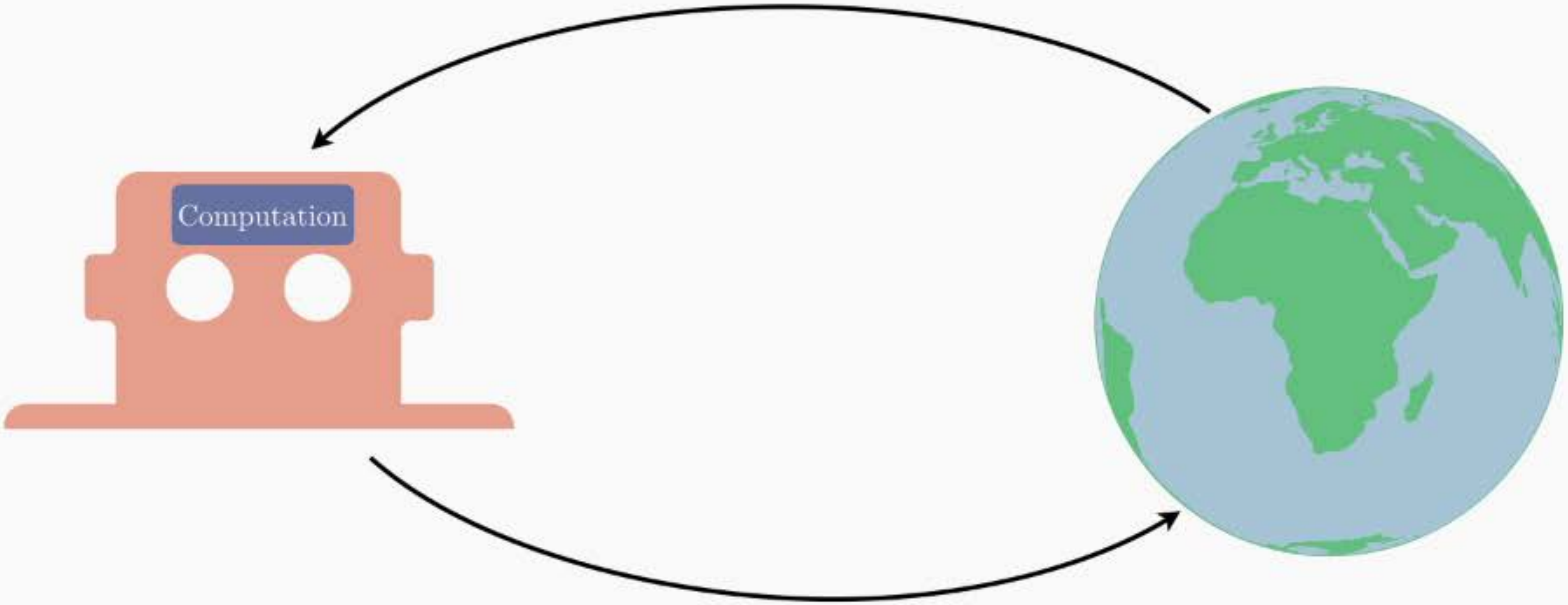
Systematized Problem Solving

Motor Controls

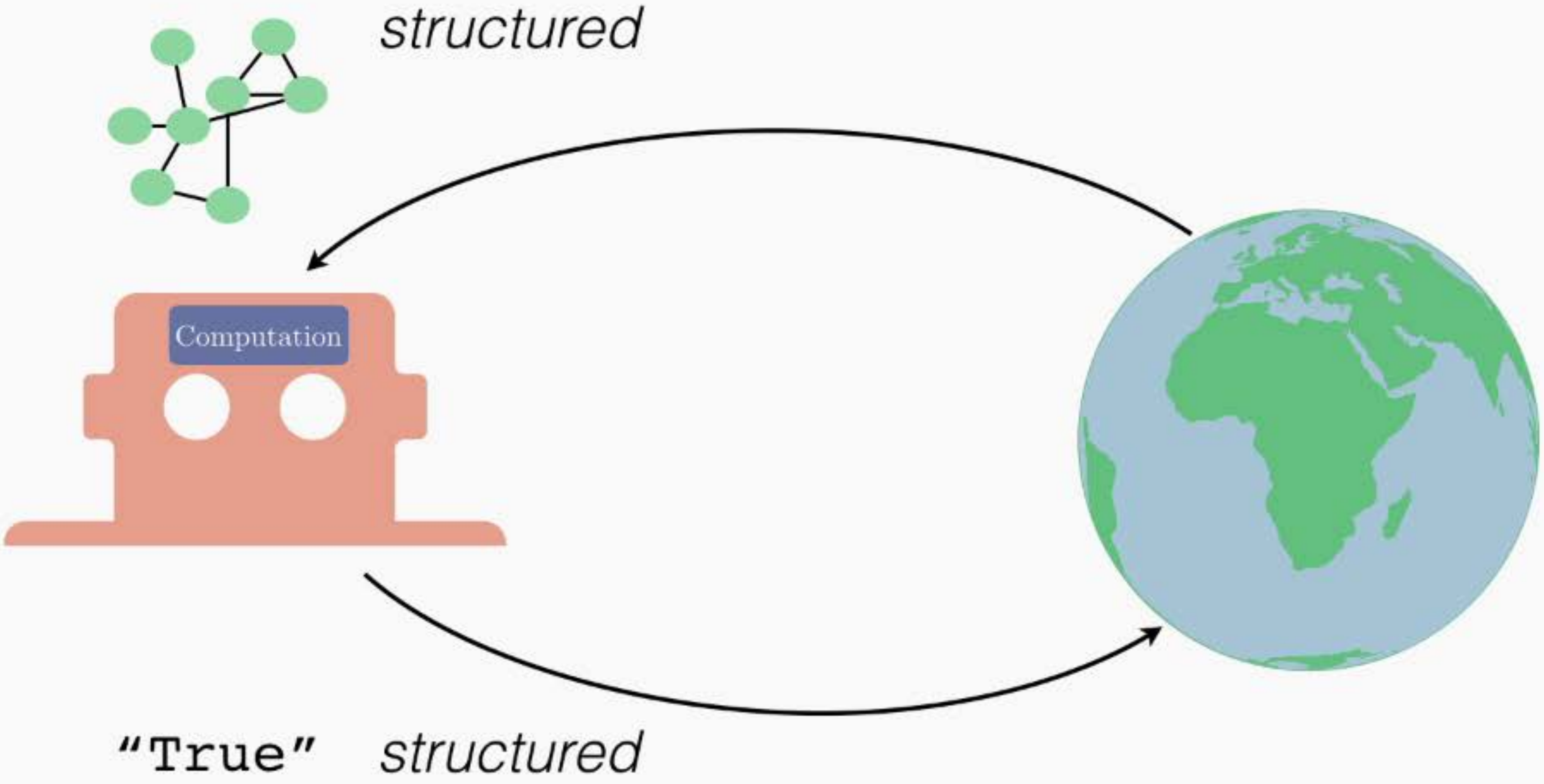
In ↓ **Out** ↑

Computation

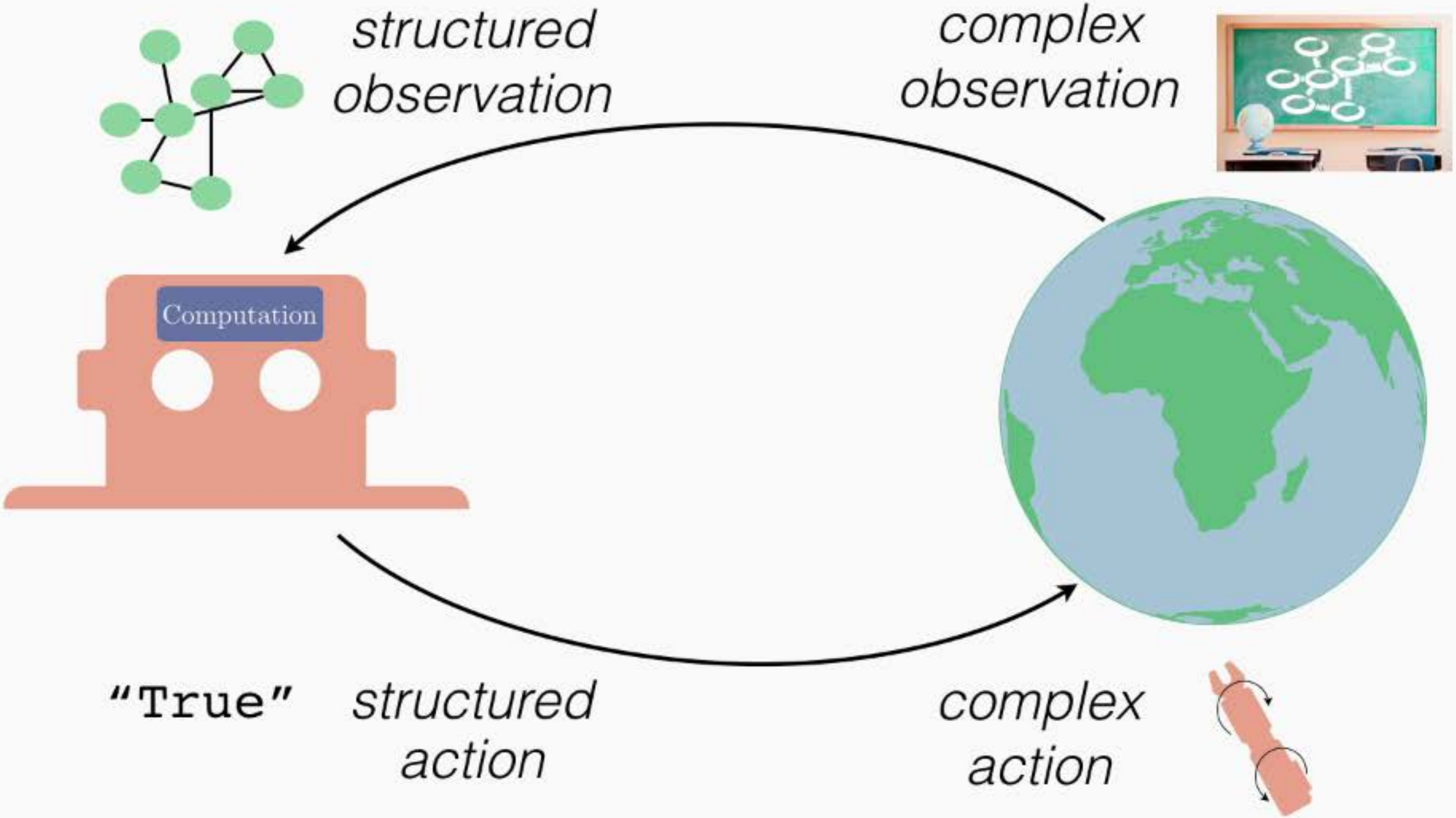
RL as Learning to Solve Problems



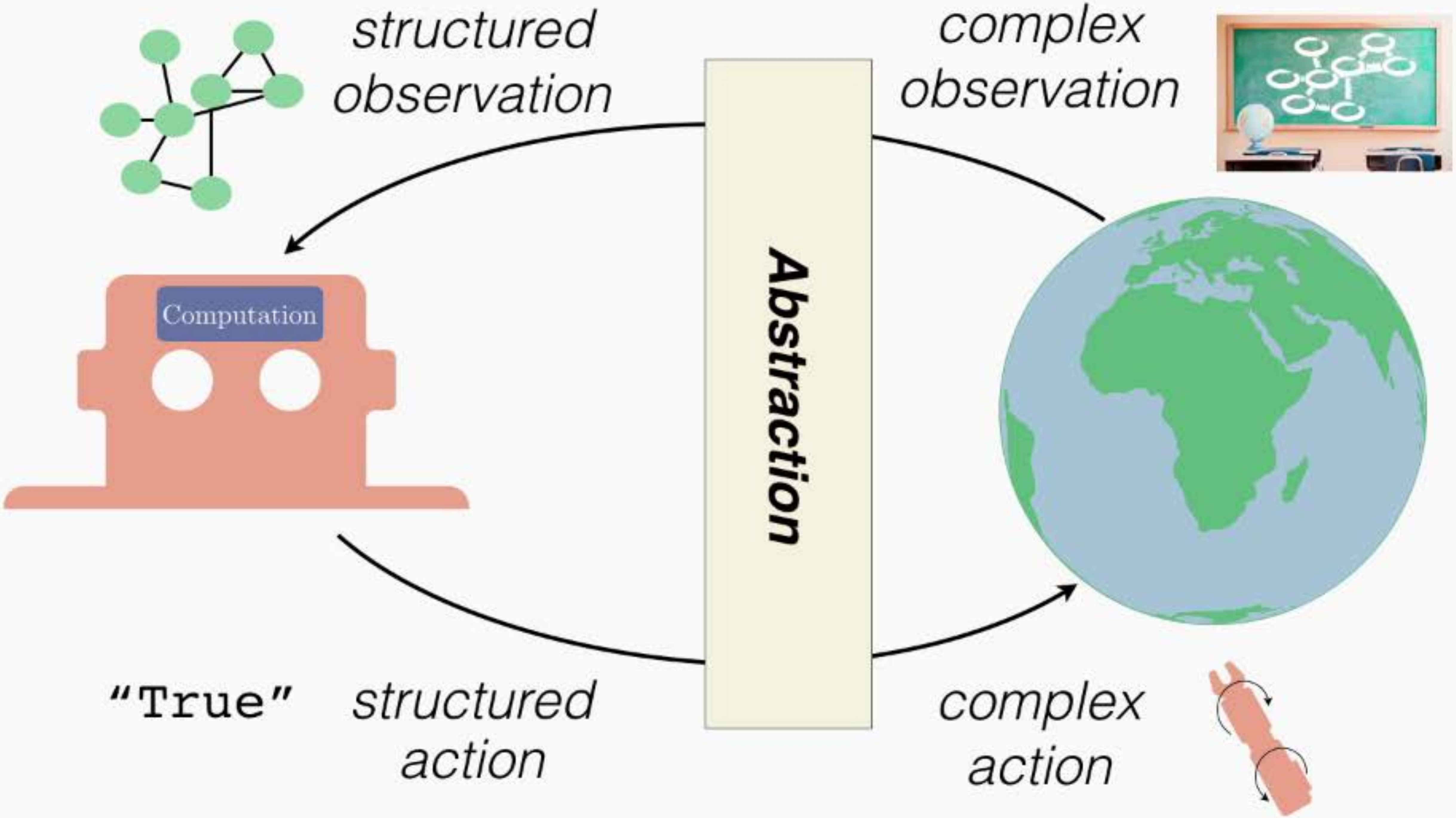
RL as Learning to Solve Problems



RL as Learning to Solve Problems

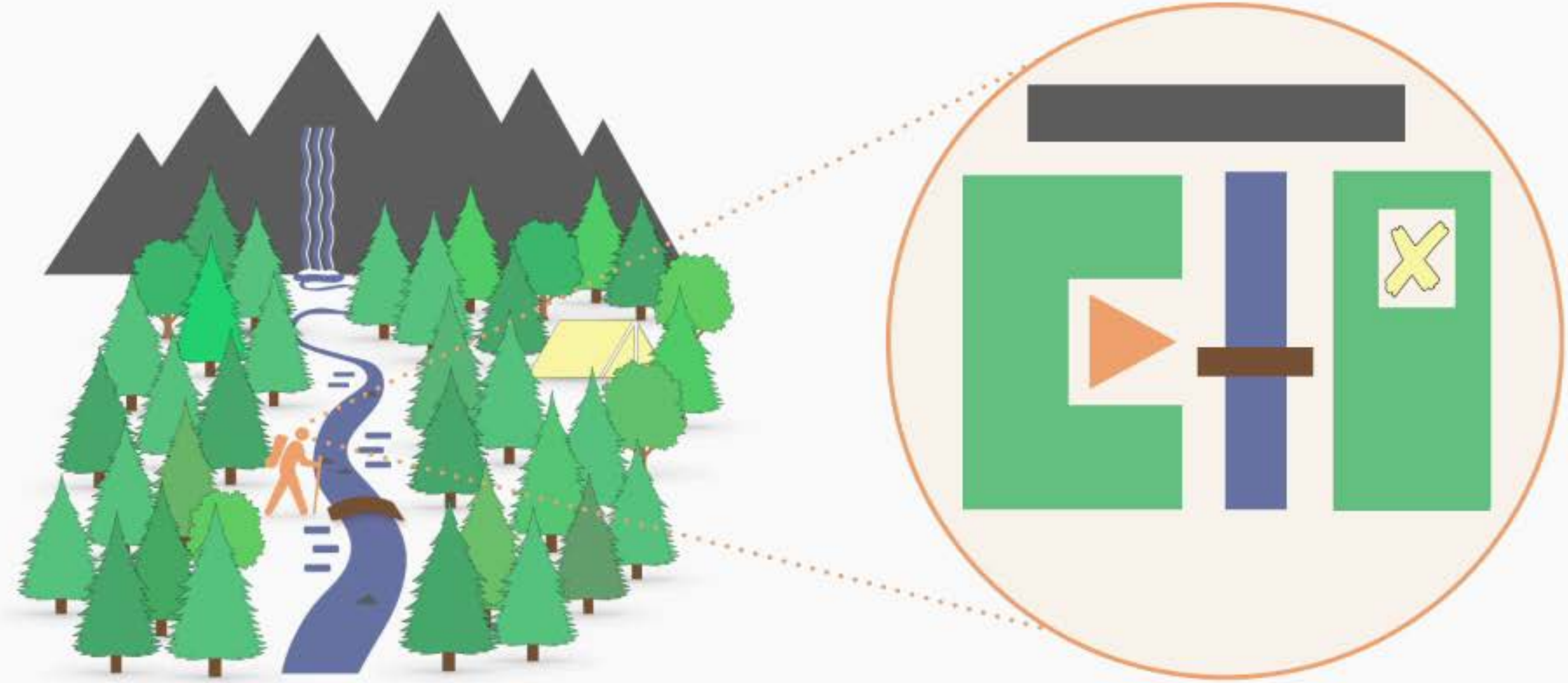


RL as Learning to Solve Problems

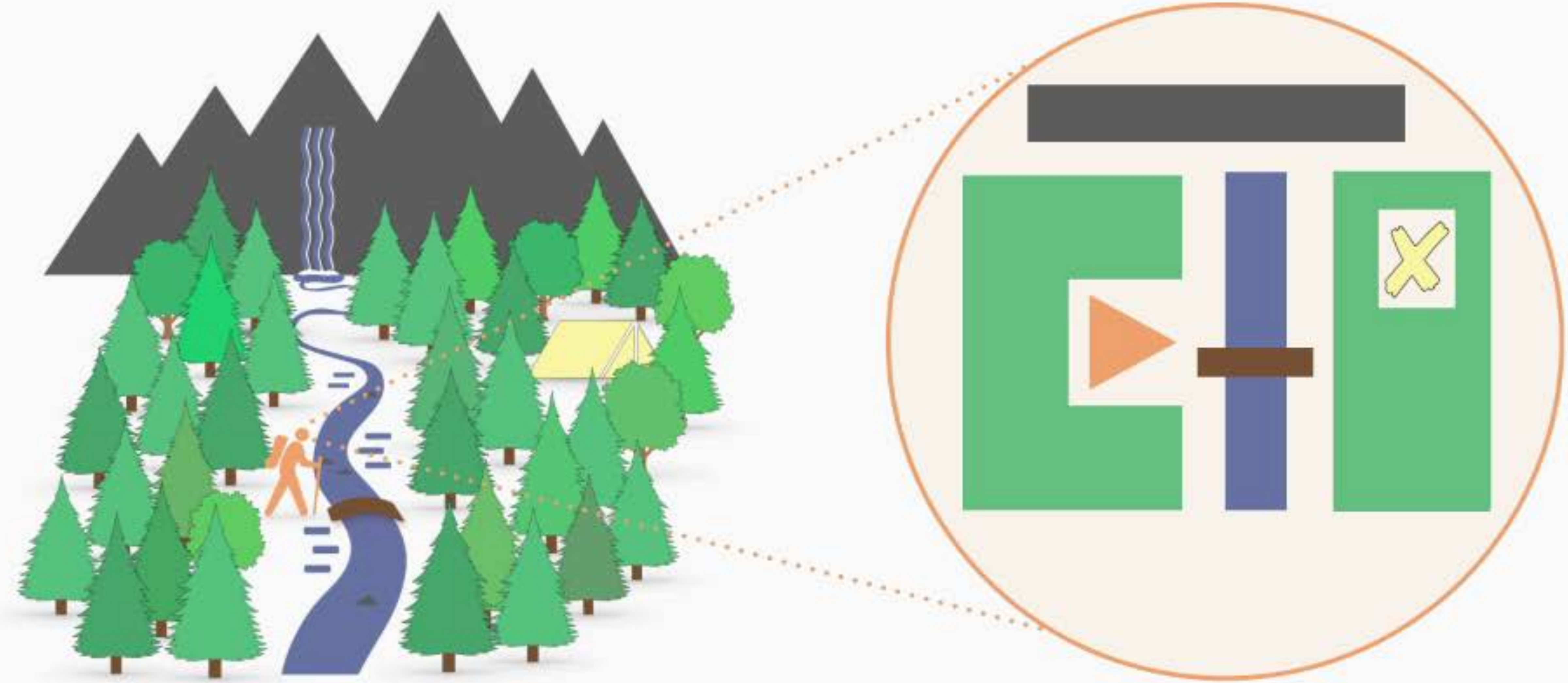


Abstraction

Abstraction

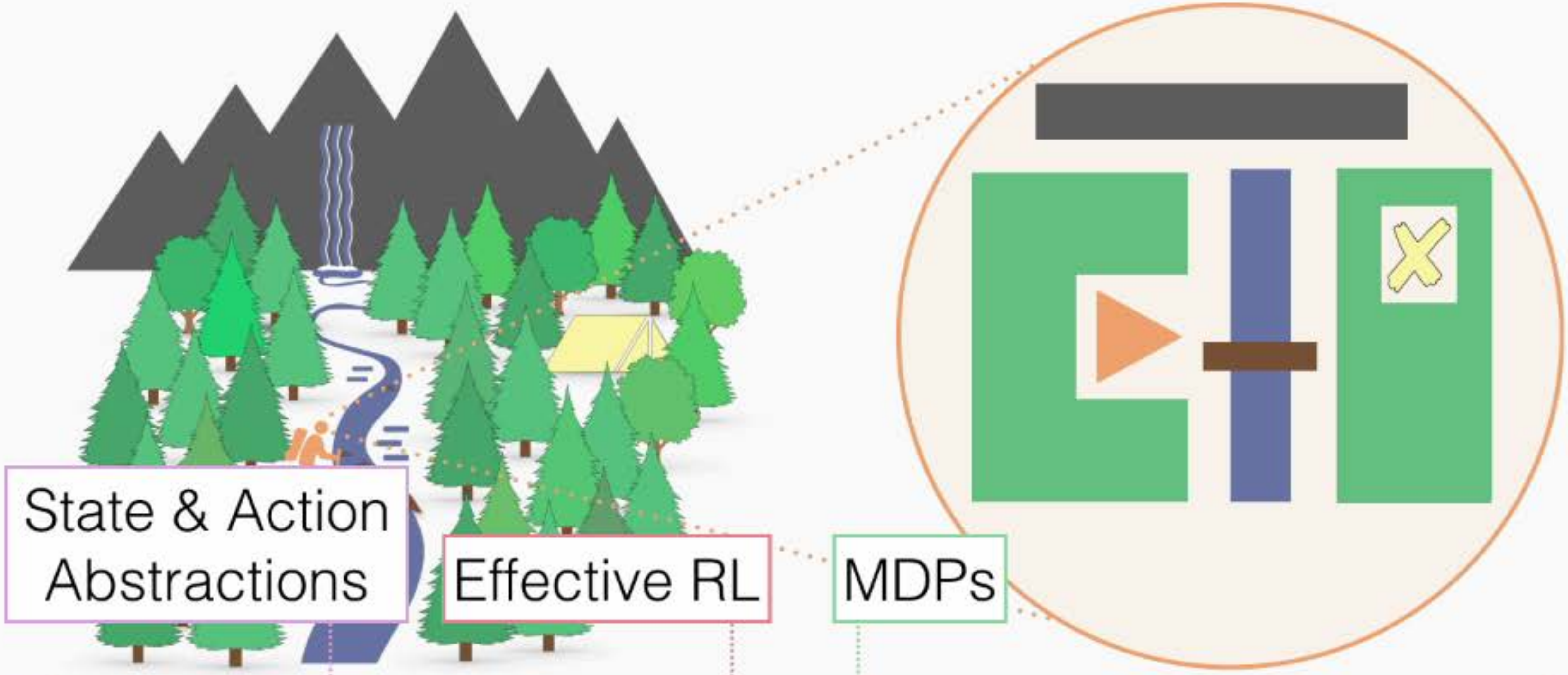


Abstraction



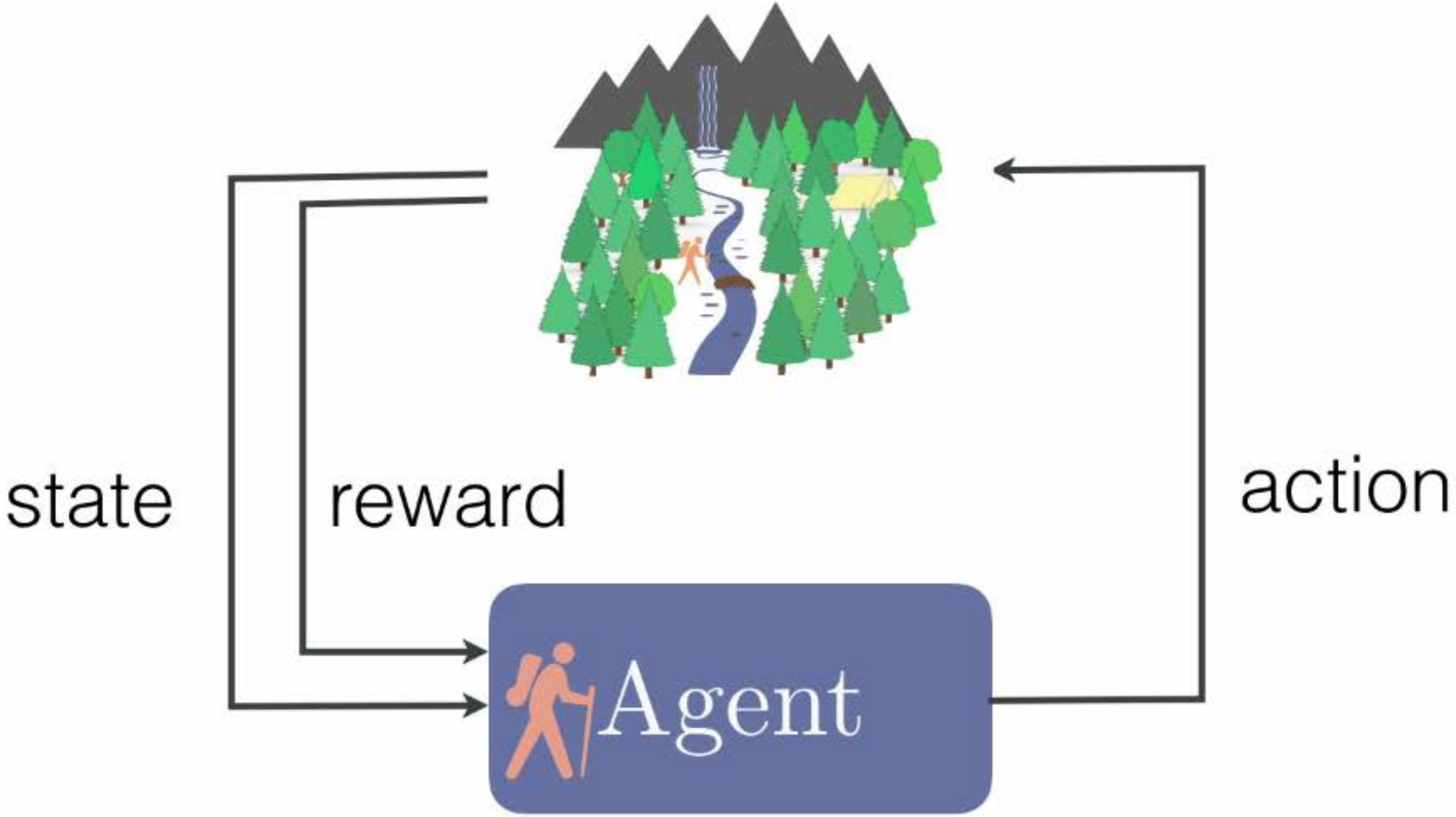
Question: *How do intelligent agents come up with the right abstract understanding of the worlds they inhabit?*

Abstraction

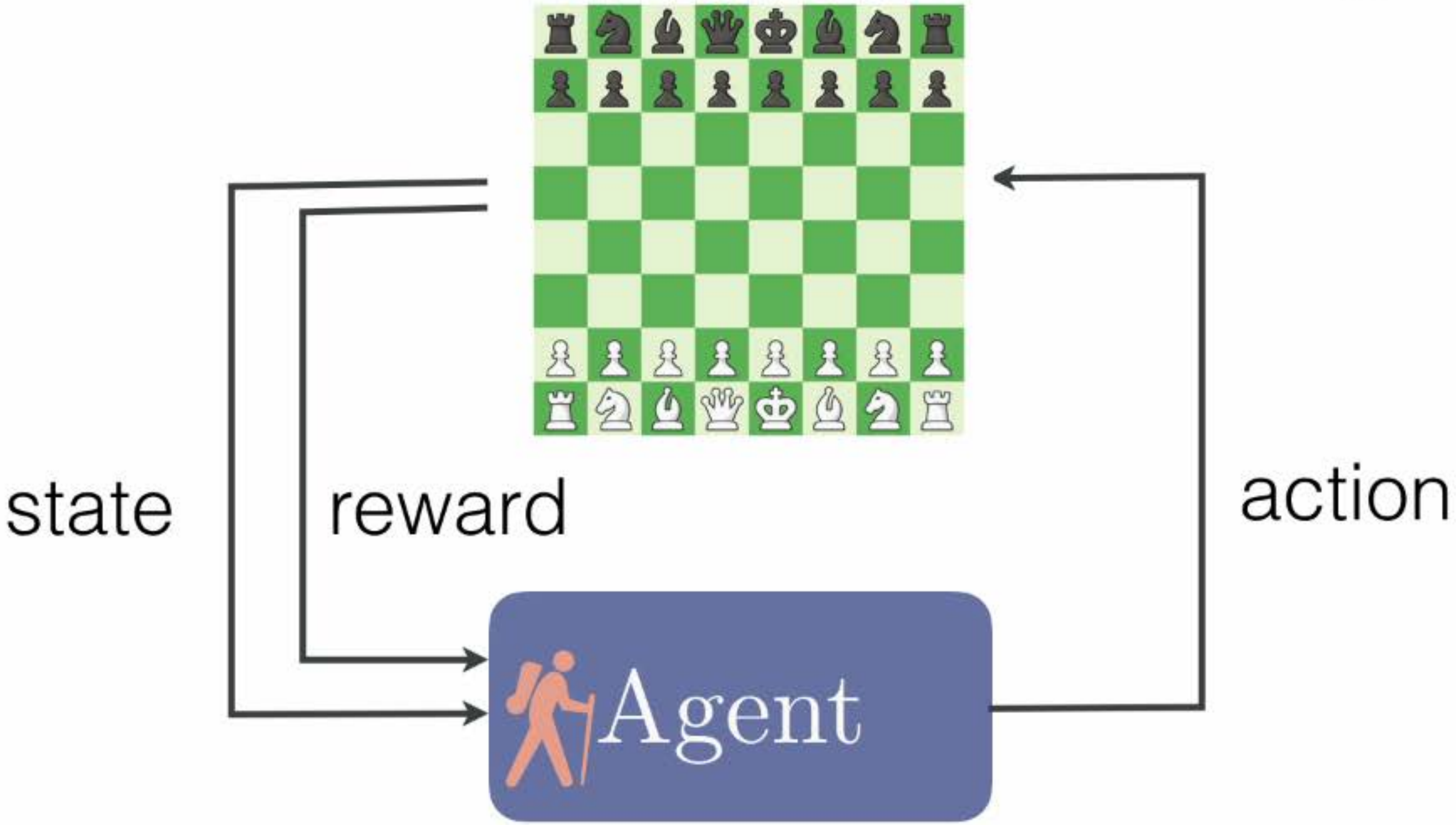


Question: *How do intelligent agents come up with the right abstract understanding of the worlds they inhabit?*

Reinforcement Learning



Reinforcement Learning



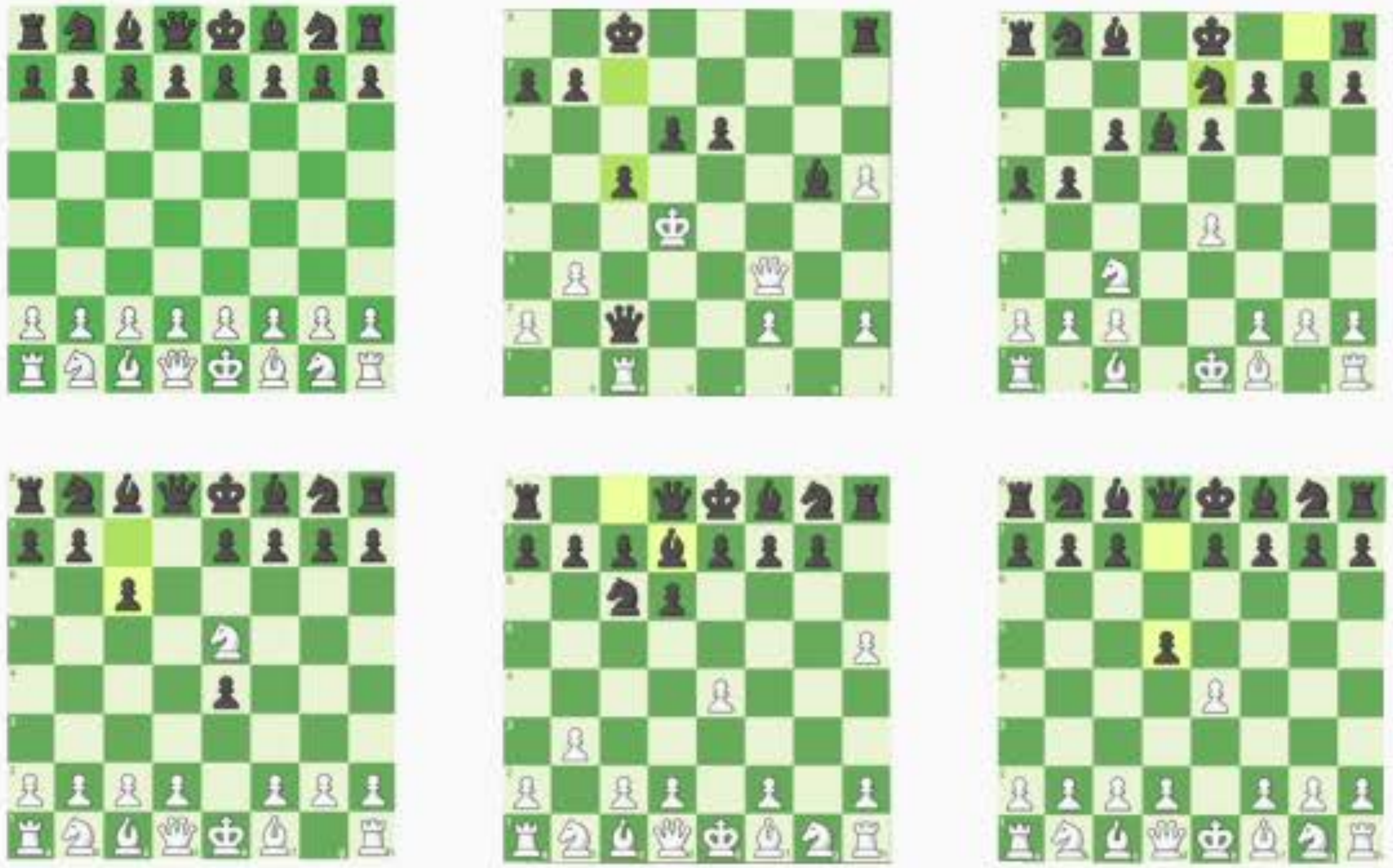
Reinforcement Learning

Central Formalism: ***Markov Decision Process*** *(MDP):*

Reinforcement Learning

Central Formalism: *Markov Decision Process (MDP):*

$\mathcal{S}$ A set of states.



$\cdots$

Reinforcement Learning

Central Formalism: *Markov Decision Process (MDP):*

$\mathcal{S}$ A set of states.

$\mathcal{A}$ A set of actions.

Reinforcement Learning

Central Formalism: *Markov Decision Process (MDP):*

$\mathcal{S}$	A set of states.
$\mathcal{A}$	A set of actions.
$\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$	A reward function.

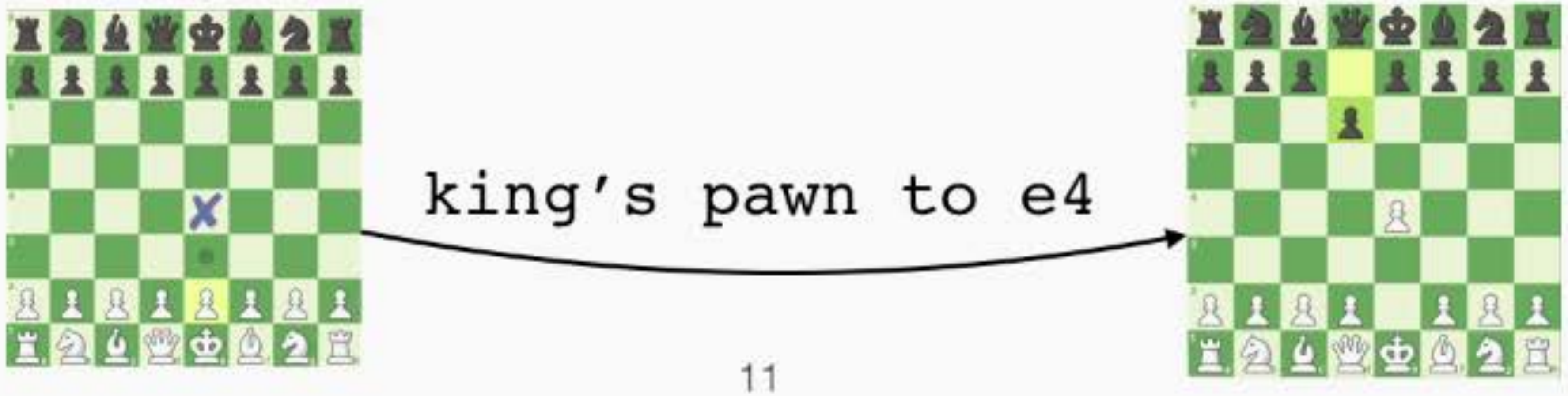
Win!

Loss!

Reinforcement Learning

Central Formalism: *Markov Decision Process (MDP):*

$\mathcal{S}$	A set of states.
$\mathcal{A}$	A set of actions.
$\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$	A reward function.
$\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \Pr(\mathcal{S})$	A transition function.

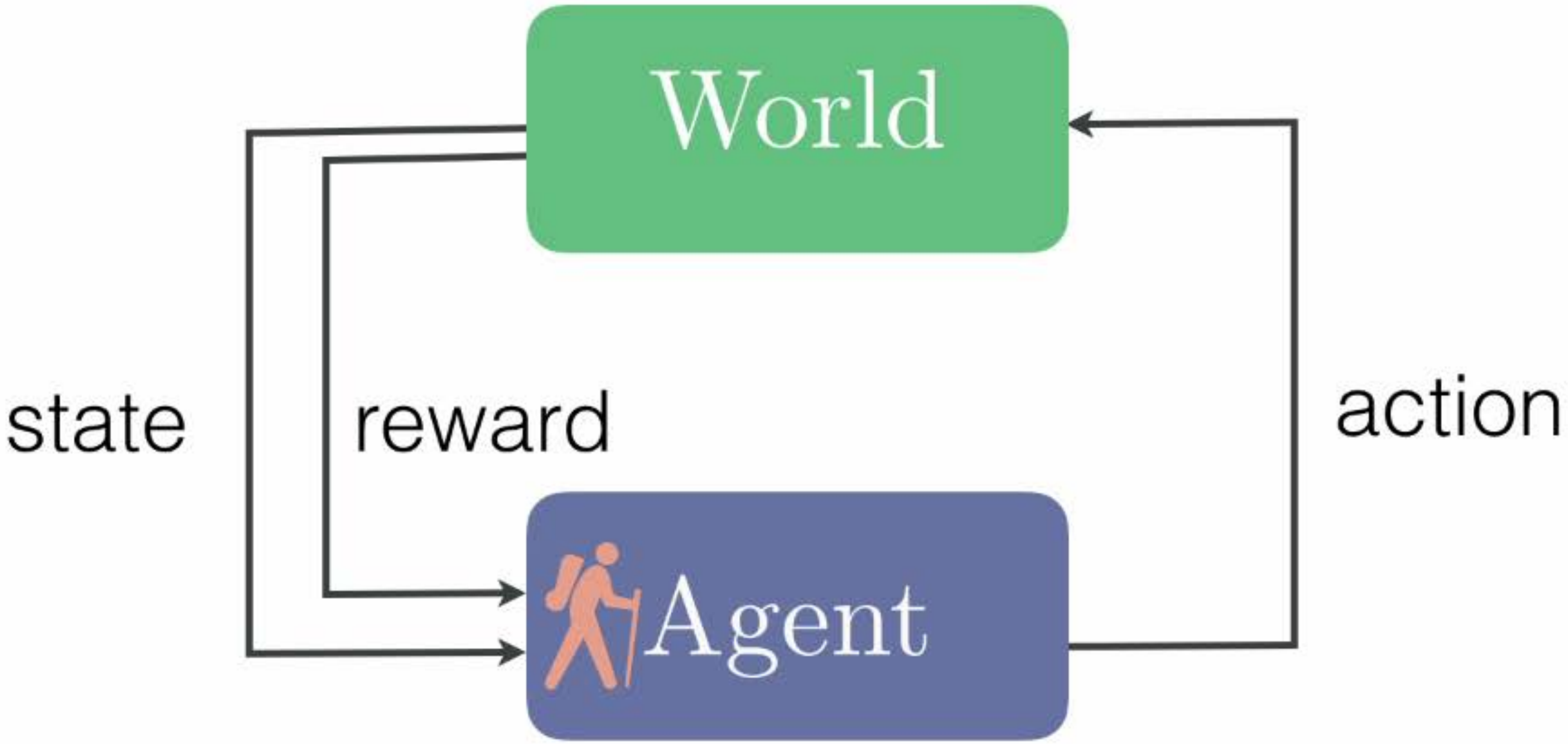


Reinforcement Learning

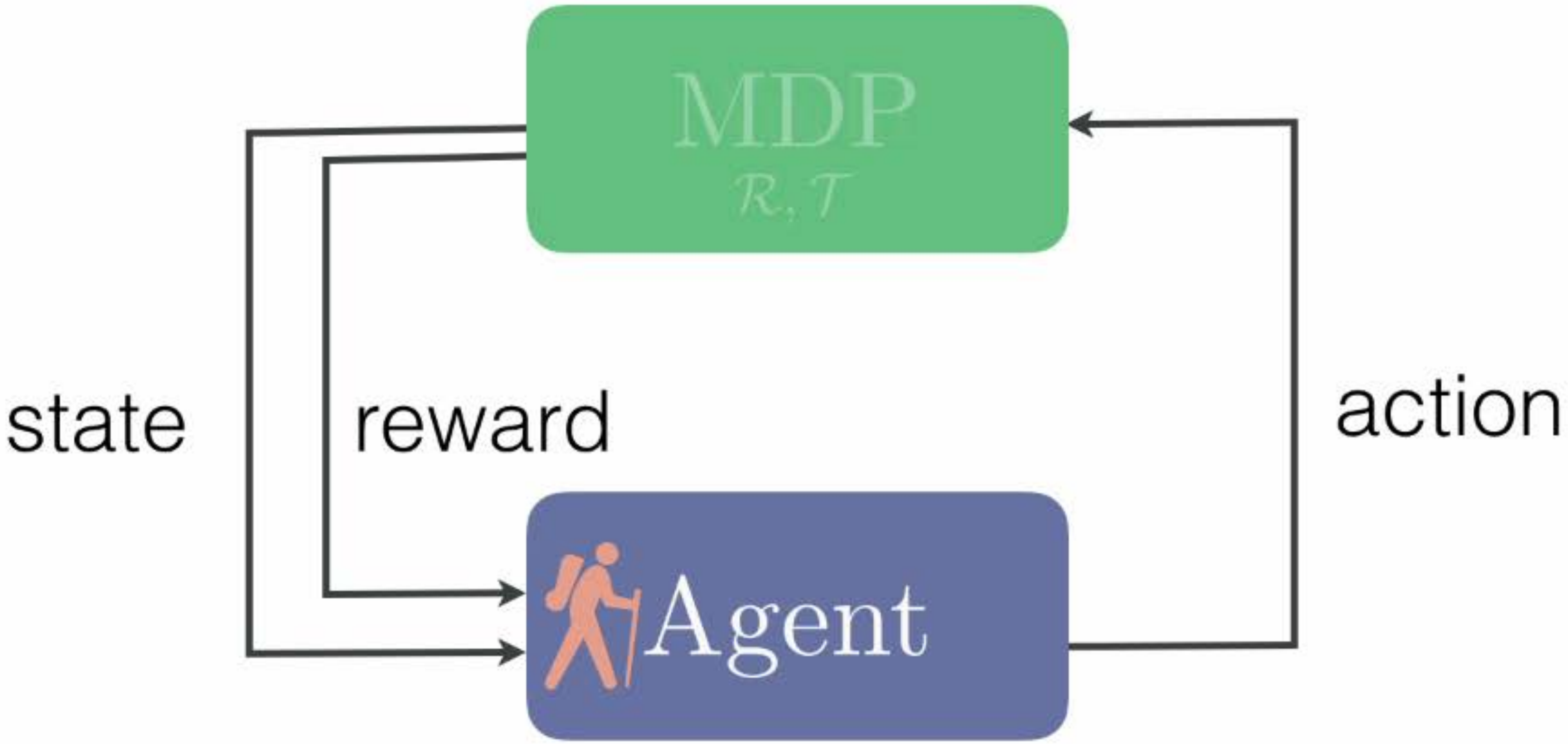
Central Formalism: *Markov Decision Process (MDP):*

$\mathcal{S}$	A set of states.
$\mathcal{A}$	A set of actions.
$\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$	A reward function.
$\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \Pr(\mathcal{S})$	A transition function.
$\gamma \in [0, 1)$	A discount factor.

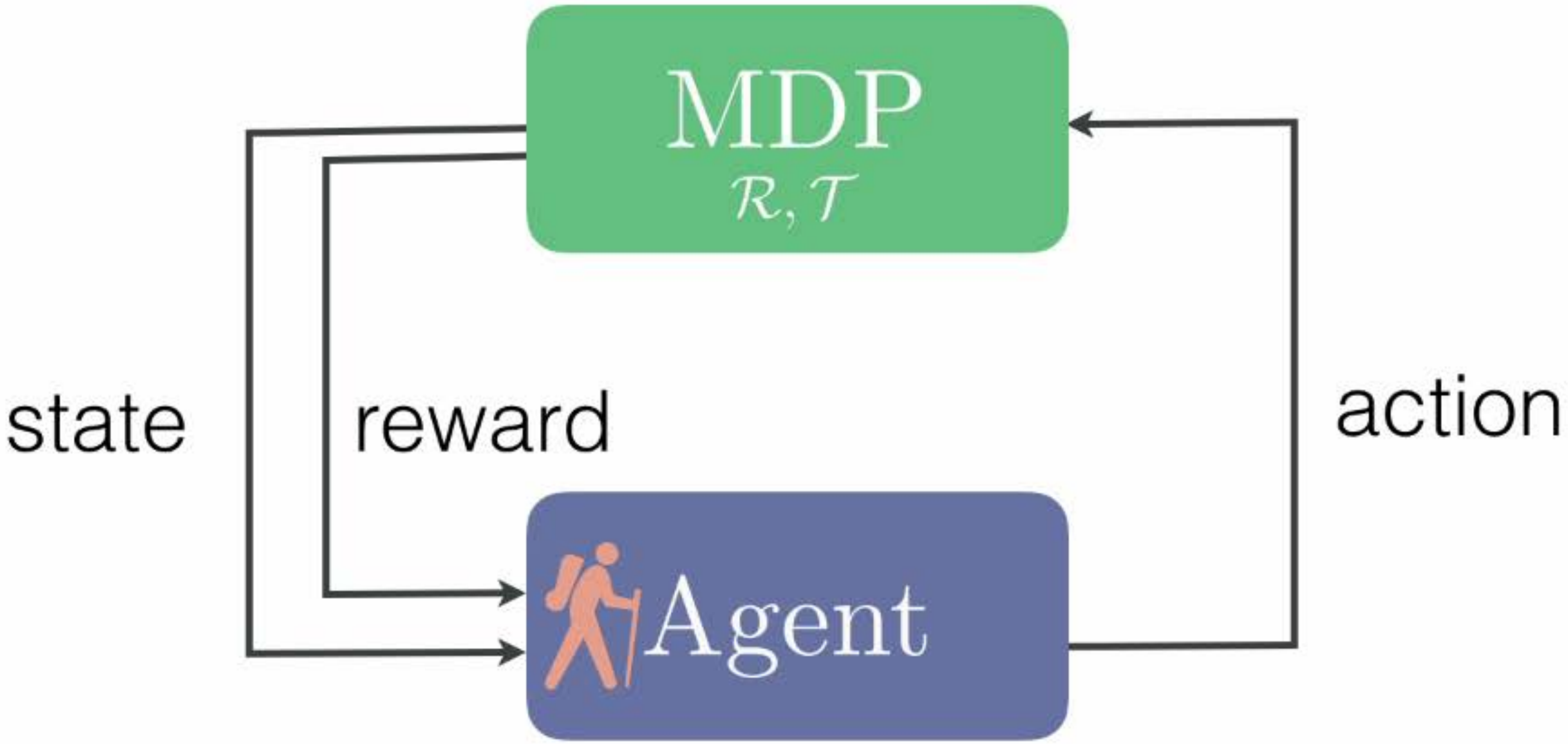
Reinforcement Learning



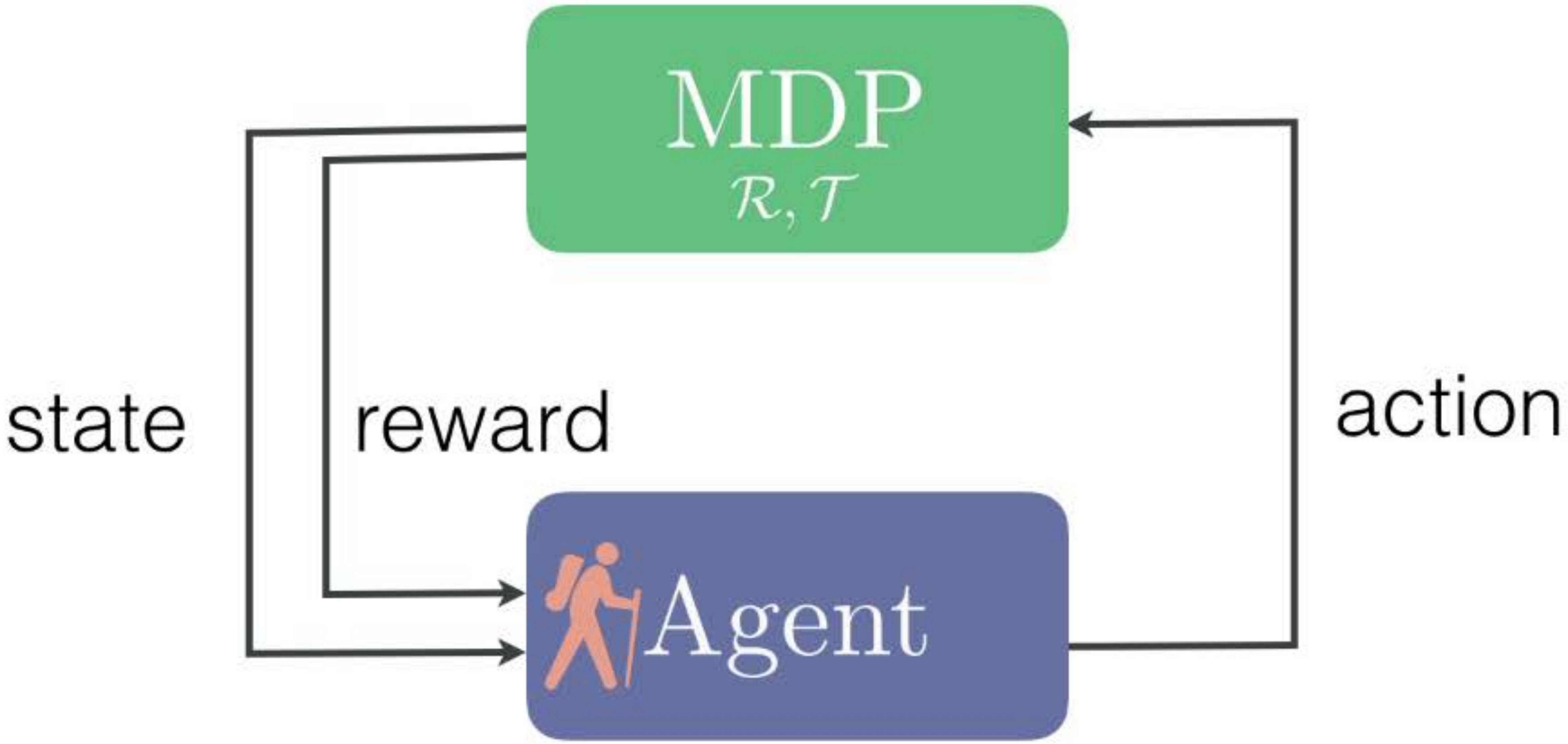
Reinforcement Learning



Reinforcement Learning



Reinforcement Learning



Goal: Maximize long term expected reward.

Reinforcement Learning

Policy: $\pi : \mathcal{S} \to \mathcal{A}$

Goal: Maximize long term expected reward.

Reinforcement Learning

Policy: $$\pi : \mathcal{S} \to \mathcal{A}$$

Value: $$V^{\pi}(s) = \mathcal{R}(s, \pi(s)) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{T}(s, \pi(s), s') V^{\pi}(s')$$

Goal: Maximize long term expected reward.

Reinforcement Learning

Policy: $\pi : \mathcal{S} \rightarrow \mathcal{A}$

Value: $V^{\pi}(s) = \mathcal{R}(s, \pi(s)) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{T}(s, \pi(s), s') V^{\pi}(s')$

Immediate Reward

Goal: Maximize long term expected reward.

Reinforcement Learning

Policy: $$\pi : \mathcal{S} \rightarrow \mathcal{A}$$

Value: $$V^{\pi}(s) = \mathcal{R}(s, \pi(s)) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{T}(s, \pi(s), s') V^{\pi}(s')$$

Immediate Reward

Discounted Expected Future Reward

Goal: Maximize long term expected reward.

Reinforcement Learning

Policy: $$\pi : \mathcal{S} \to \mathcal{A}$$

Value: $$V^{\pi}(s) = \mathcal{R}(s, \pi(s)) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{T}(s, \pi(s), s') V^{\pi}(s')$$

Action Value: $$Q^{\pi}(s, a) = \mathcal{R}(s, a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{T}(s, a, s') V^{\pi}(s')$$

Goal: Maximize long term expected reward.

Reinforcement Learning

Policy: $$\pi : \mathcal{S} \rightarrow \mathcal{A}$$

Value: $$V^{\pi}(s) = \mathcal{R}(s, \pi(s)) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{T}(s, \pi(s), s') V^{\pi}(s')$$

Action Value: $$Q^{\pi}(s, a) = \mathcal{R}(s, a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{T}(s, a, s') V^{\pi}(s')$$

Immediate Reward

Goal: Maximize long term expected reward.

Reinforcement Learning

Policy: $\pi : \mathcal{S} \to \mathcal{A}$

Value: $V^{\pi}(s) = \mathcal{R}(s, \pi(s)) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{T}(s, \pi(s), s') V^{\pi}(s')$

Action Value: $Q^{\pi}(s, a) = \mathcal{R}(s, a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{T}(s, a, s') V^{\pi}(s')$

Immediate Reward ($\mathcal{R}(s, a)$)

Discounted Expected Future Reward ($\gamma \sum_{s' \in \mathcal{S}} \mathcal{T}(s, a, s') V^{\pi}(s')$)

Goal: Maximize long term expected reward.

Reinforcement Learning

Policy: $$\pi : \mathcal{S} \to \mathcal{A}$$

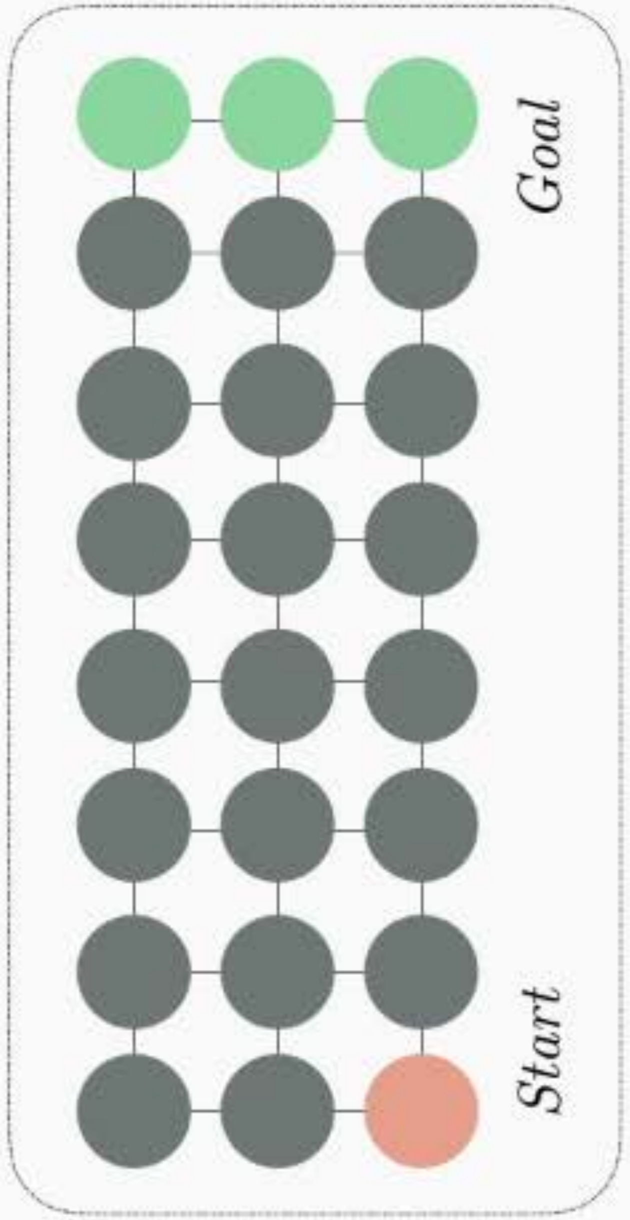
Value: $$V^\pi(s) = \mathcal{R}(s, \pi(s)) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{T}(s, \pi(s), s') V^\pi(s')$$

Action Value: $$Q^\pi(s, a) = \mathcal{R}(s, a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{T}(s, a, s') V^\pi(s')$$

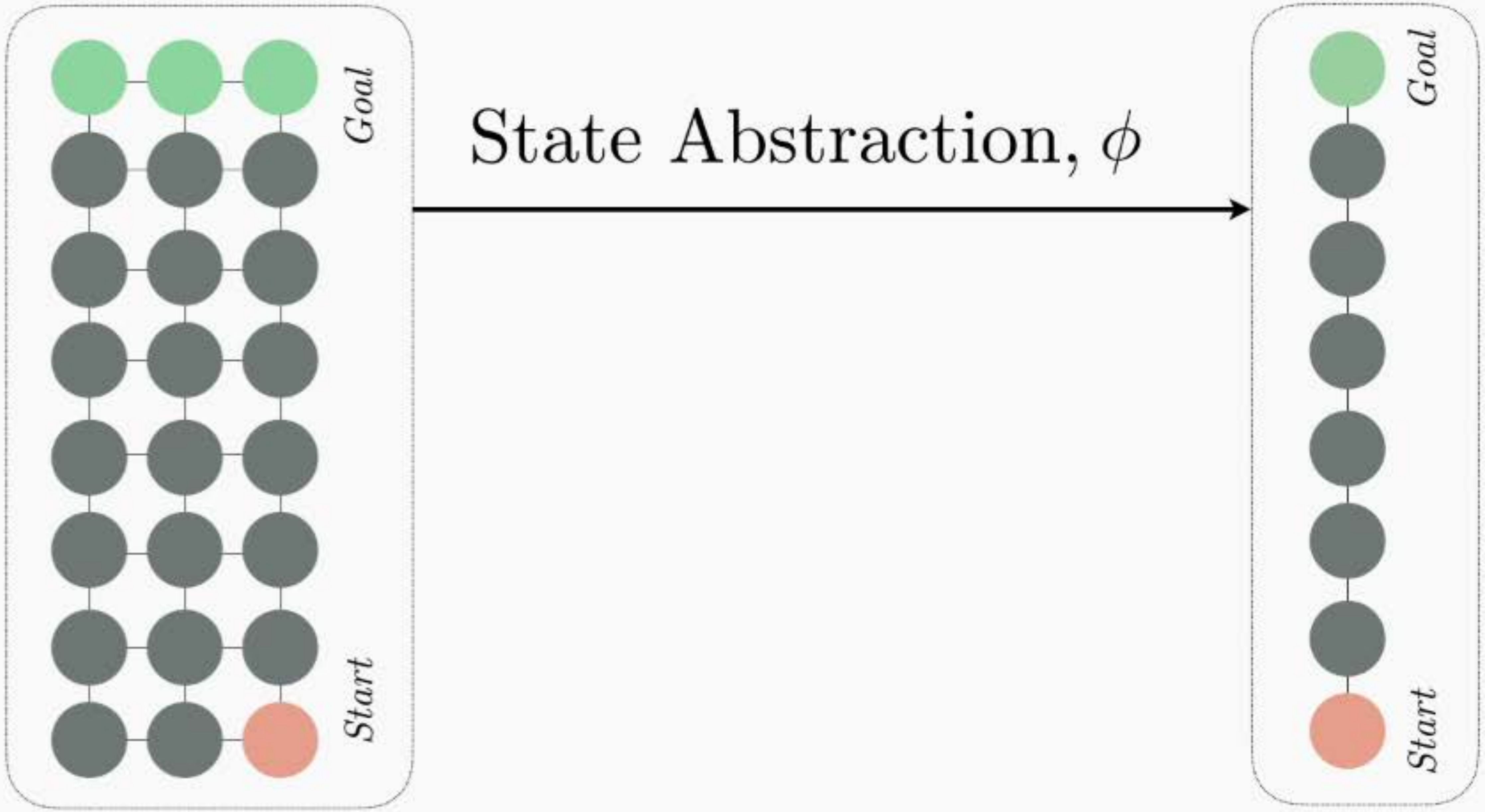
Optimal: $$V^*(s) = \max_\pi V^\pi(s) \qquad Q^*(s, a) = \max_\pi Q^\pi(s, a)$$

Goal: Maximize long term expected reward.

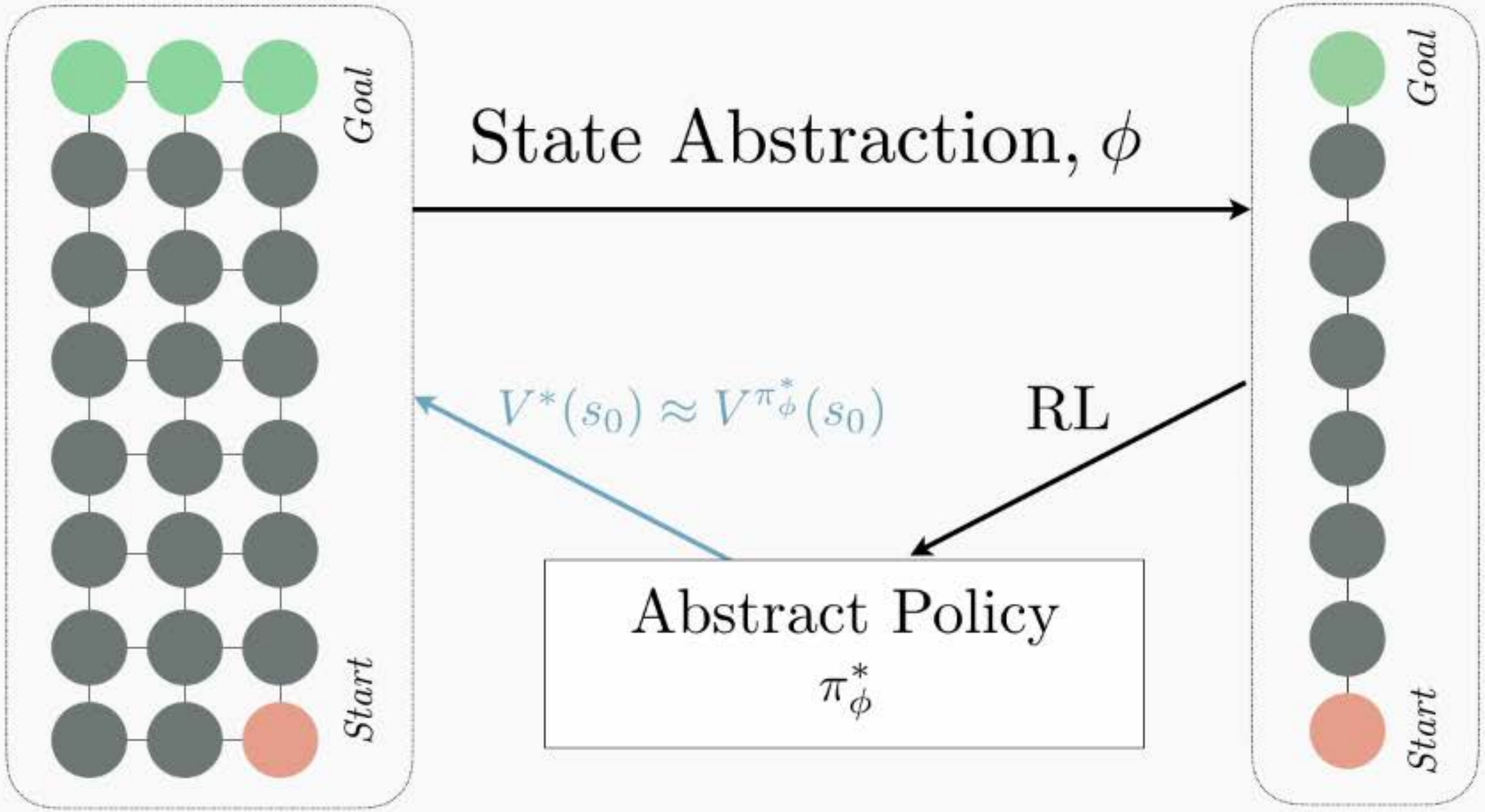
State Abstraction



State Abstraction



State Abstraction

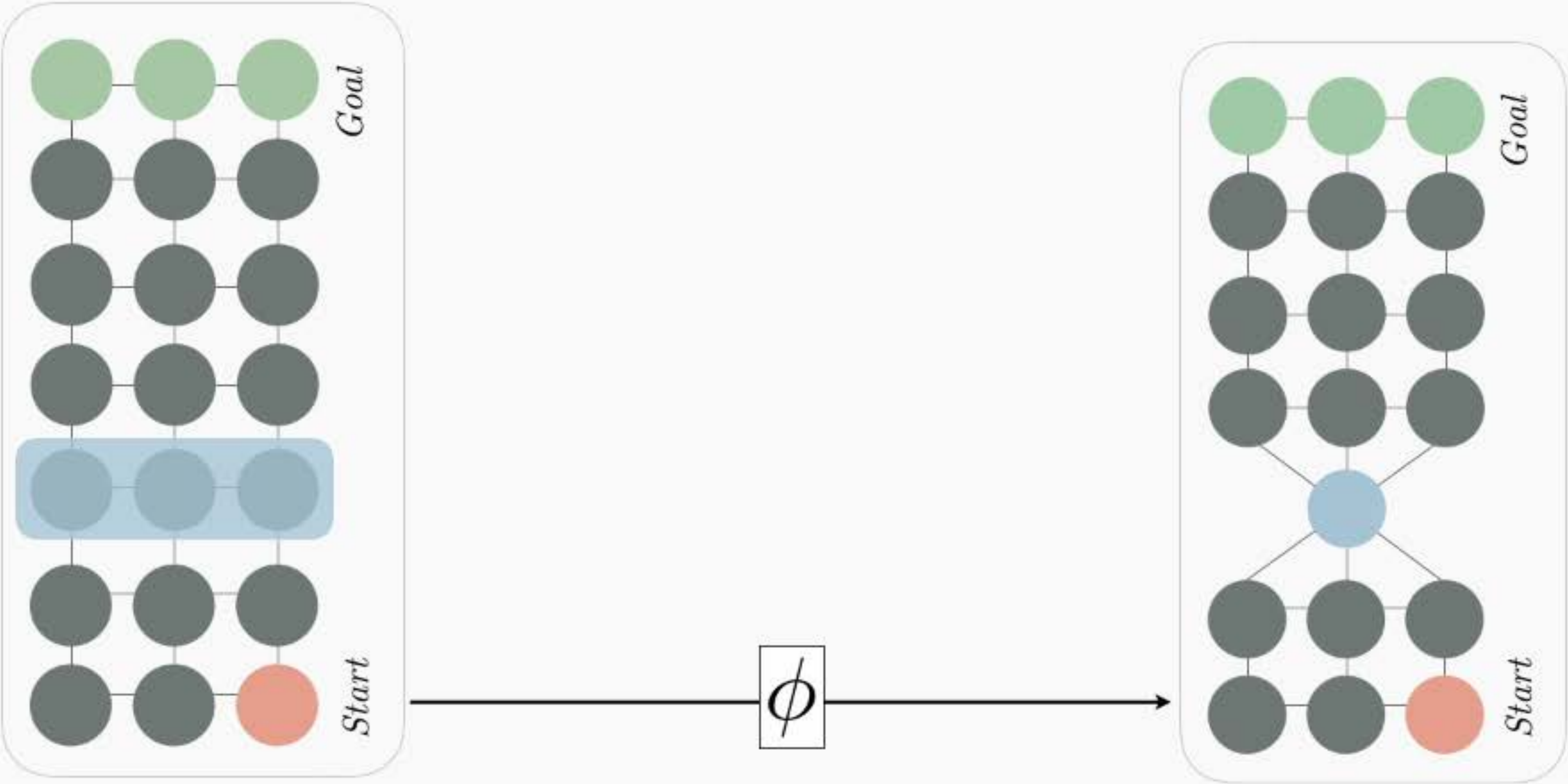


State Abstraction

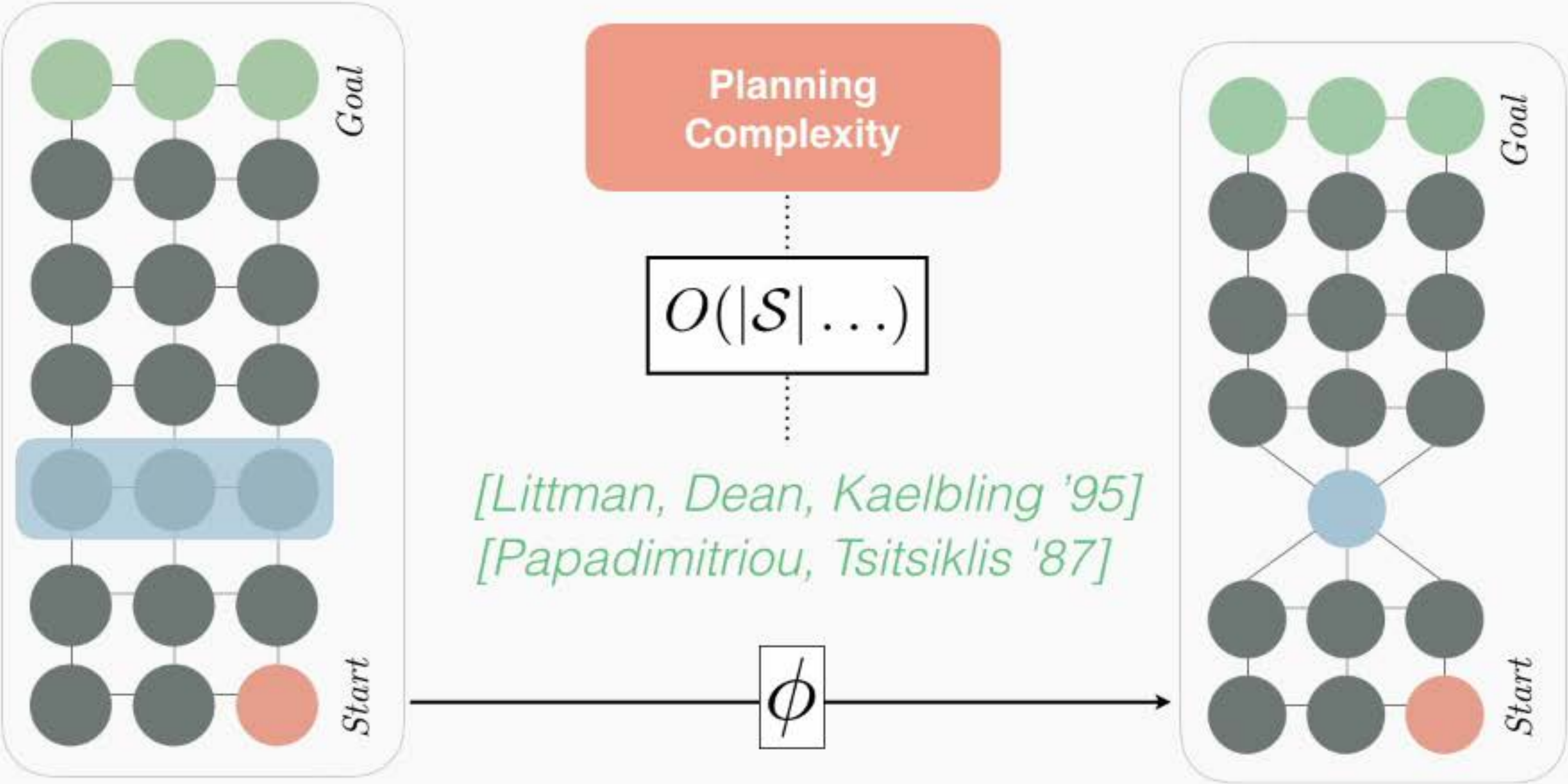
Definition (State Abstraction): A state abstraction, $\phi : \mathcal{S} \to \mathcal{S}_\phi$, is a function that maps every ground state to an abstract state.

[Fox '73]
[Whitt '78]
[Singh et al. '95]
[Dean, Givan '97]
[Dieterrich '00]
[Andre, Russell '02]
[Ravindran, Barto '03, '04]
[Jong, Stone '05]
[Ferns et al., '04, '06]
[Li et. al '06]
[Whiteson et al.'07]
[Castro, Precup '09]
[Van Seijen et al. '14]
[Ortner et al. '07, '14, '19]
[Hutter '14, '16, '19]
[Jiang et al., ''14, 15]
[Akrour et al., '18]
[Menashe, Stone '18]
[Misra et al. '19]
[Hostetler et al. '14, '15, '17]

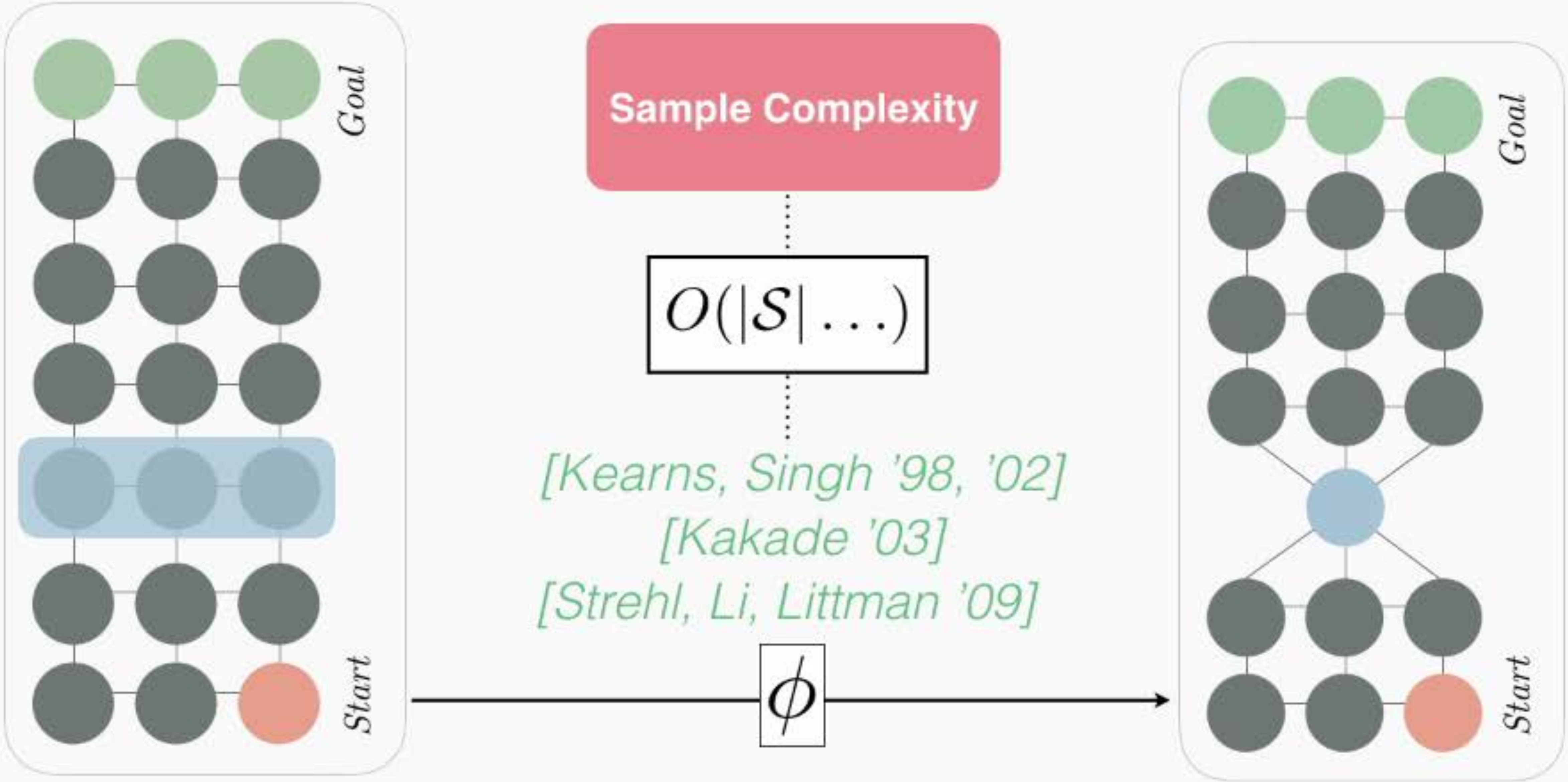
State Abstraction



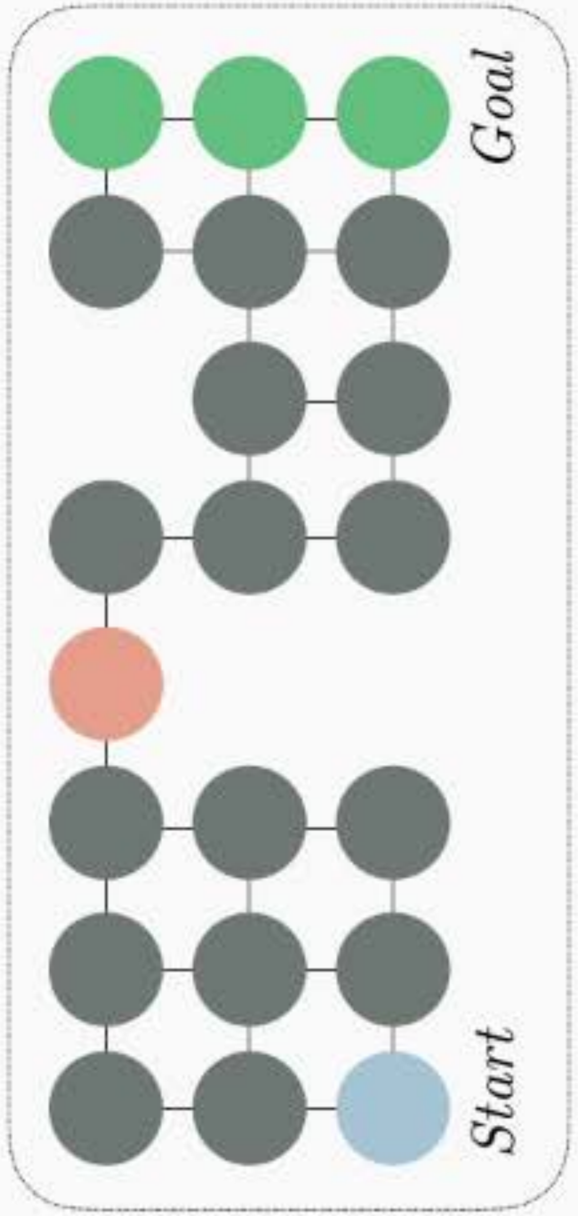
State Abstraction



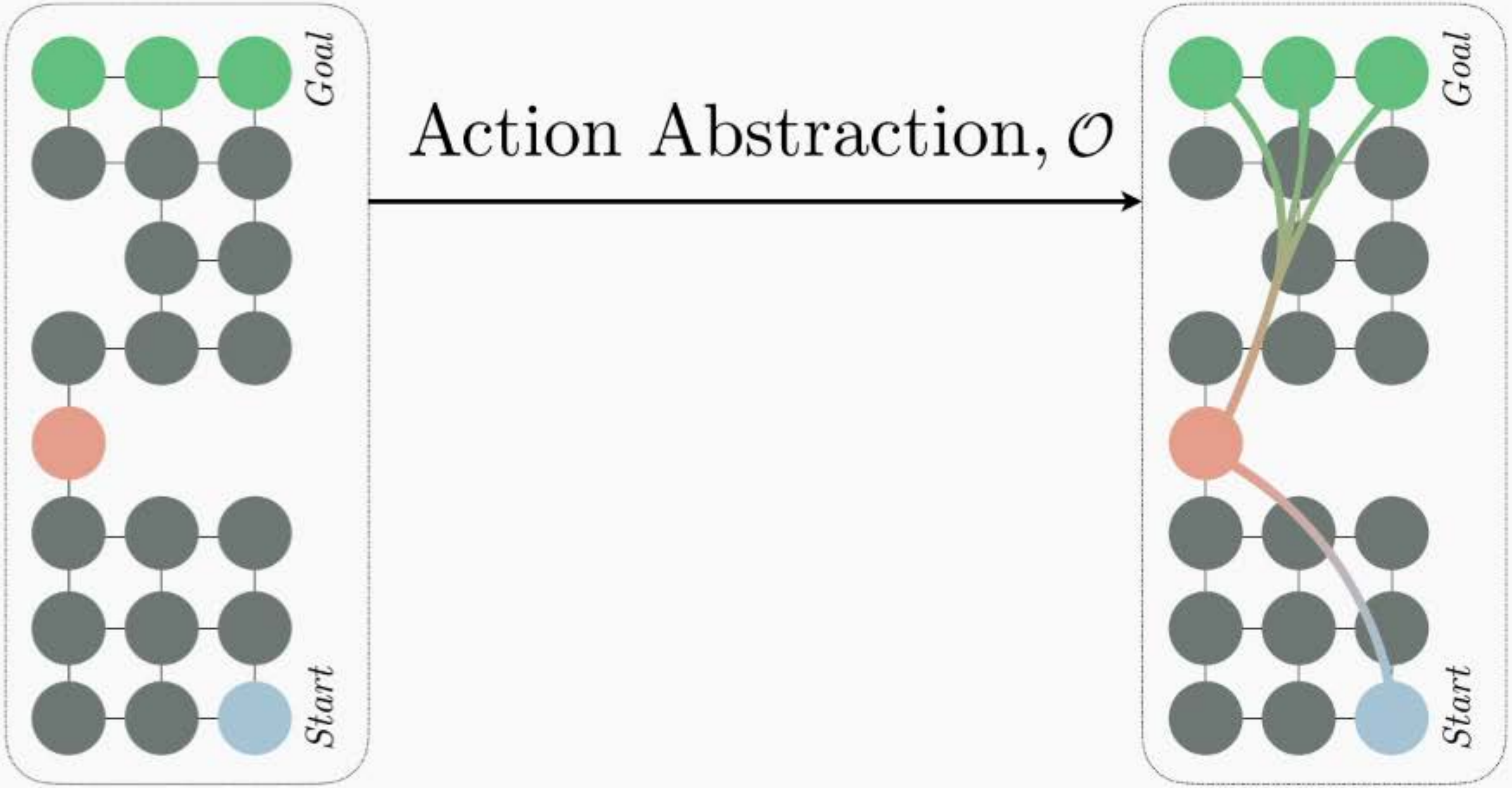
State Abstraction



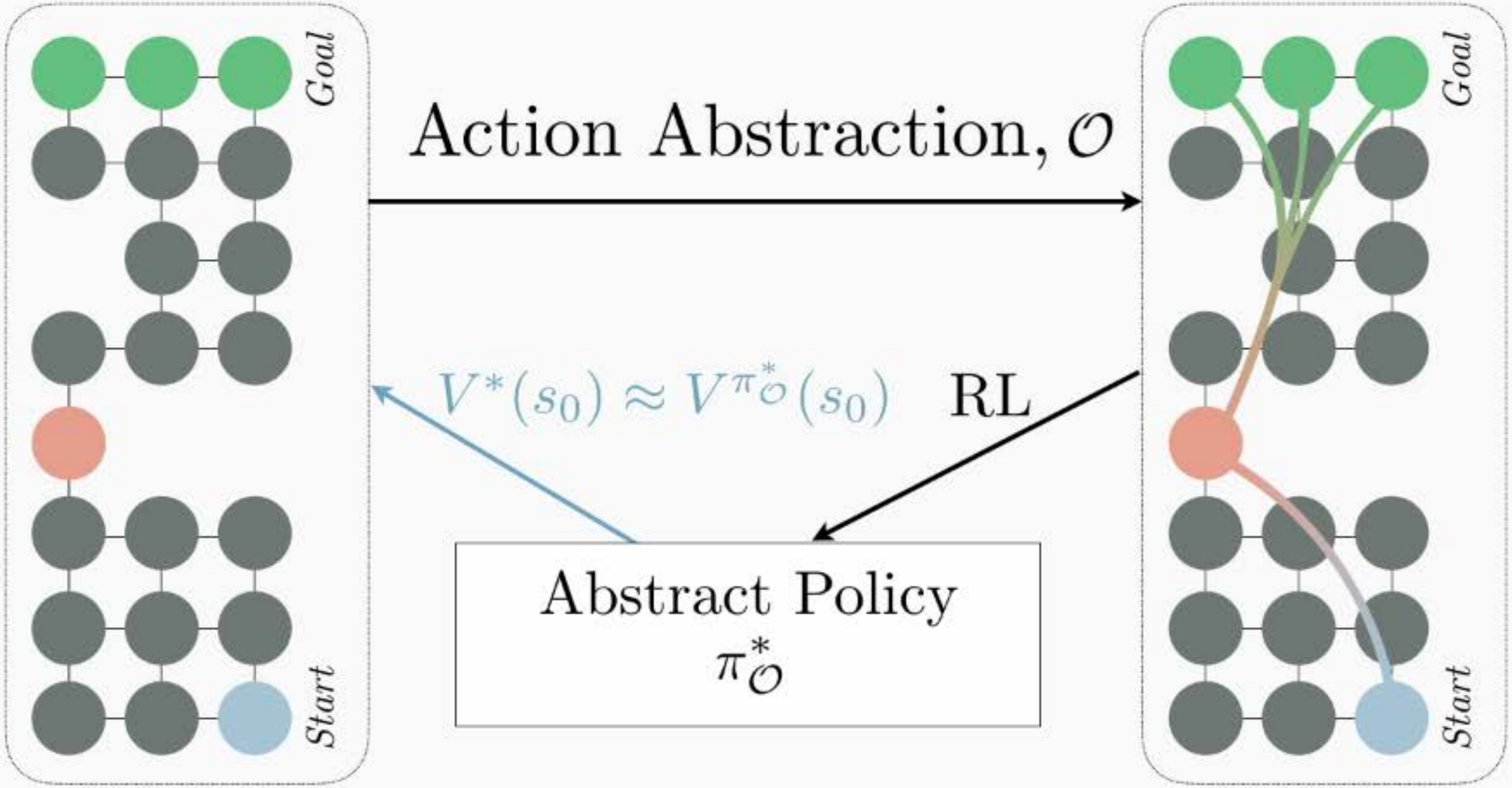
Action Abstraction



Action Abstraction



Action Abstraction



Action Abstraction

[Sutton, Precup, Singh 1999]

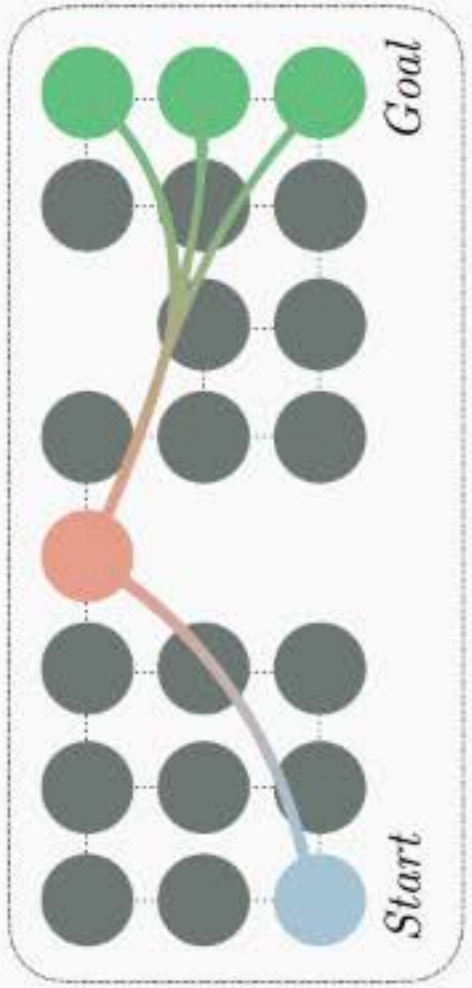
Definition (Option): A start condition, end condition, and a policy.

Action Abstraction

Example:

$$o_1 = (\quad , \quad , \pi)$$

$$o_2 = (\quad , \quad , \pi)$$



[Sutton, Precup, Singh 1999]

Definition (Option): A start condition, end condition, and a policy.

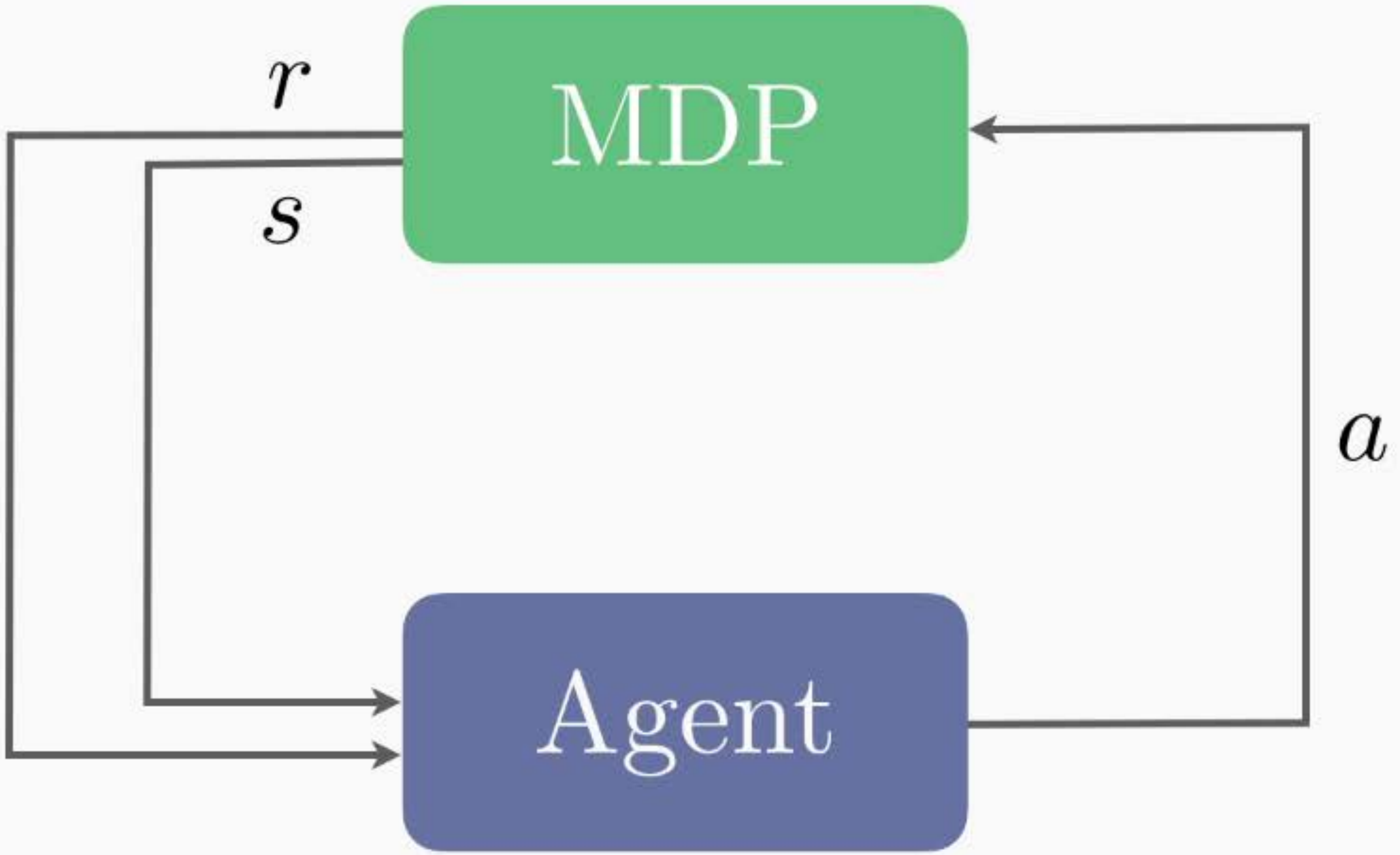
Action Abstraction

Definition (Action Abstraction): An action abstraction extends the primitive actions with the option set $\mathcal{O}$.

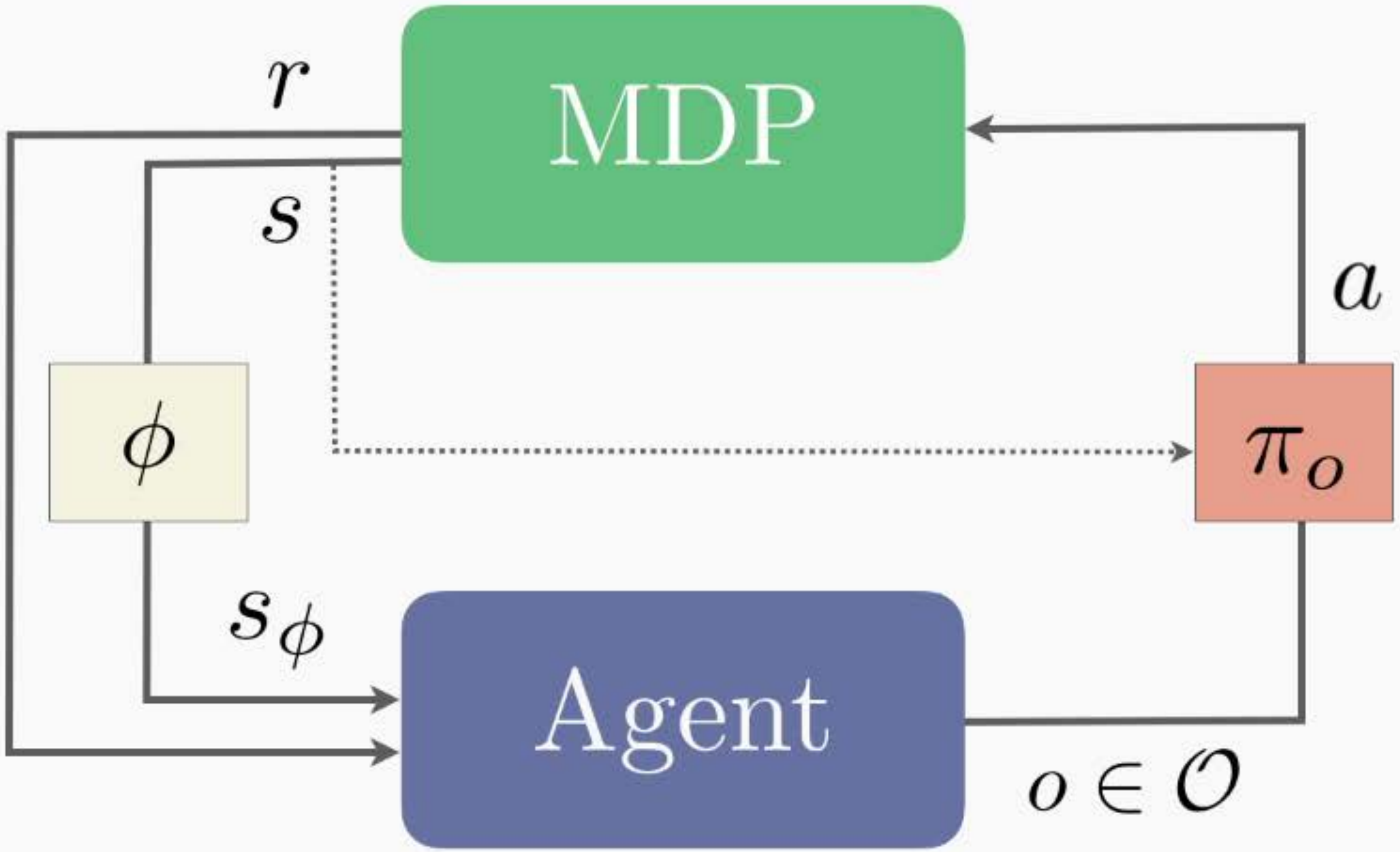
[McGovern et. al. '97]
[Sutton, Precup, Singh '99]
[Konidaris, Barto '05, '06]
[Jong, Hester, Stone '08]
[Mugan, Kuipers '09, '12]
[Brunskill, Li '14]
[Ciosek, Silver '15]
[Durugkar et al. '16]
[Bacon et al. '17]
[Fruit et al. '17, '17]
[Machado et al. '17]
[Harutyunyan et al. '18]
[Eysenbach et al. '18]
[Barreto et al. '19]

…and more!

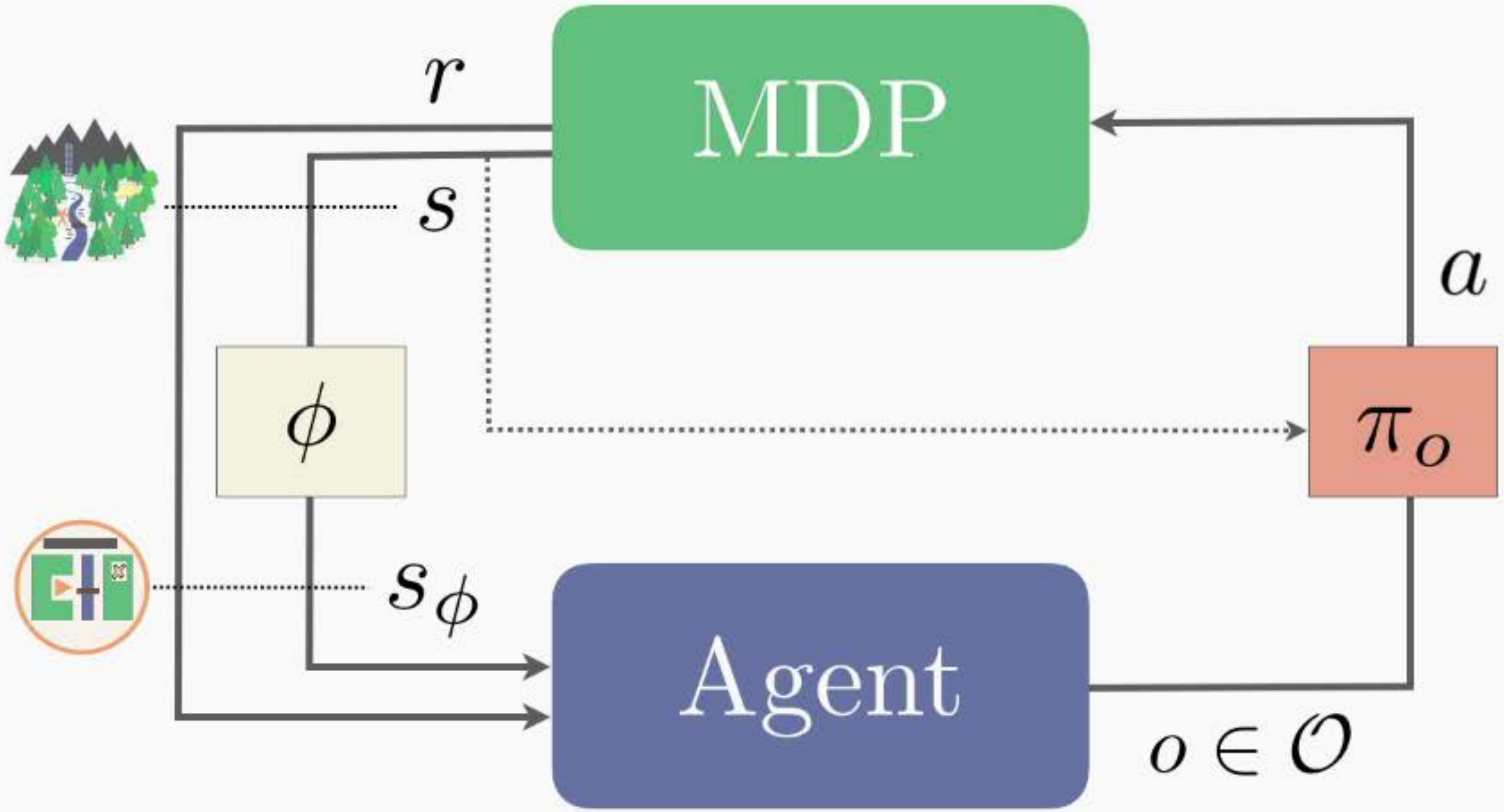
Abstraction in RL



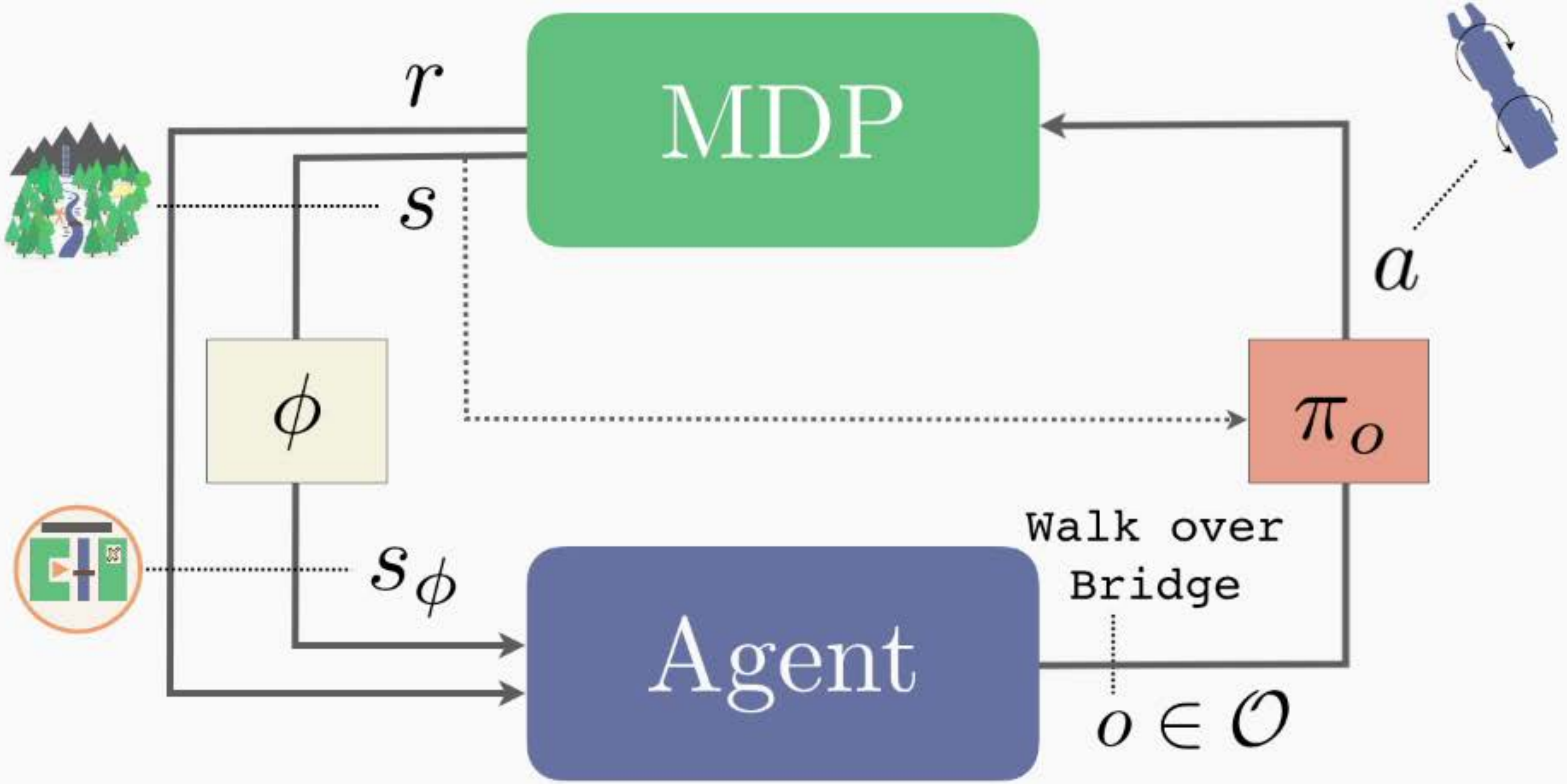
Abstraction in RL



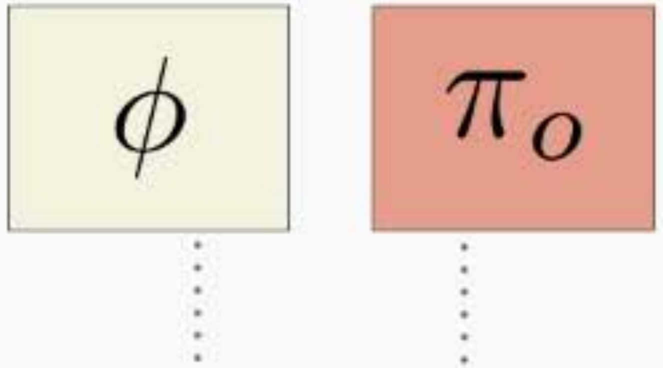
Abstraction in RL



Abstraction in RL



Desirable Abstractions



Q: Which kinds of abstractions are desirable?

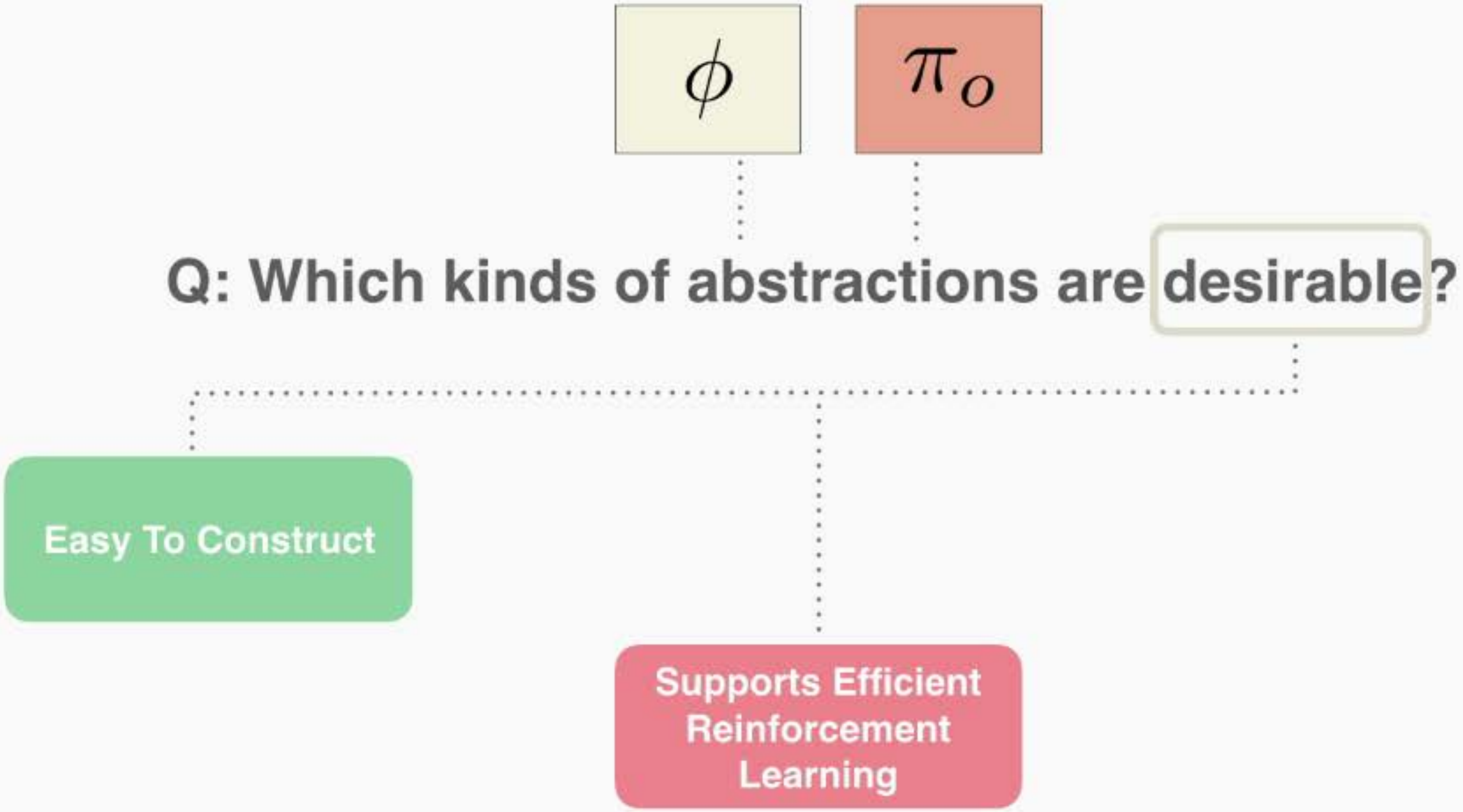
Desirable Abstractions

ϕ π_o

Q: Which kinds of abstractions are desirable?

Easy To Construct

Desirable Abstractions



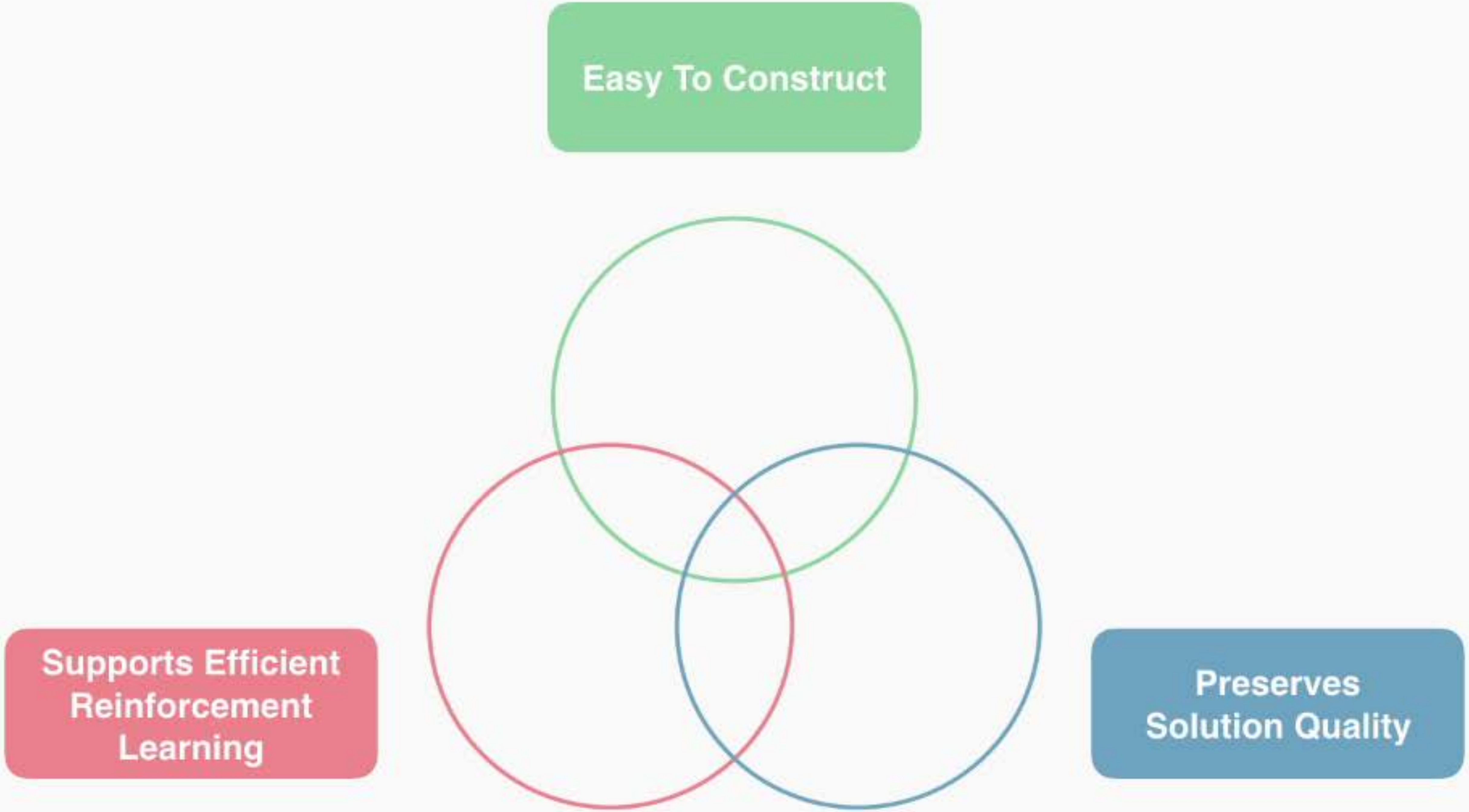
Desirable Abstractions

ϕ π_o

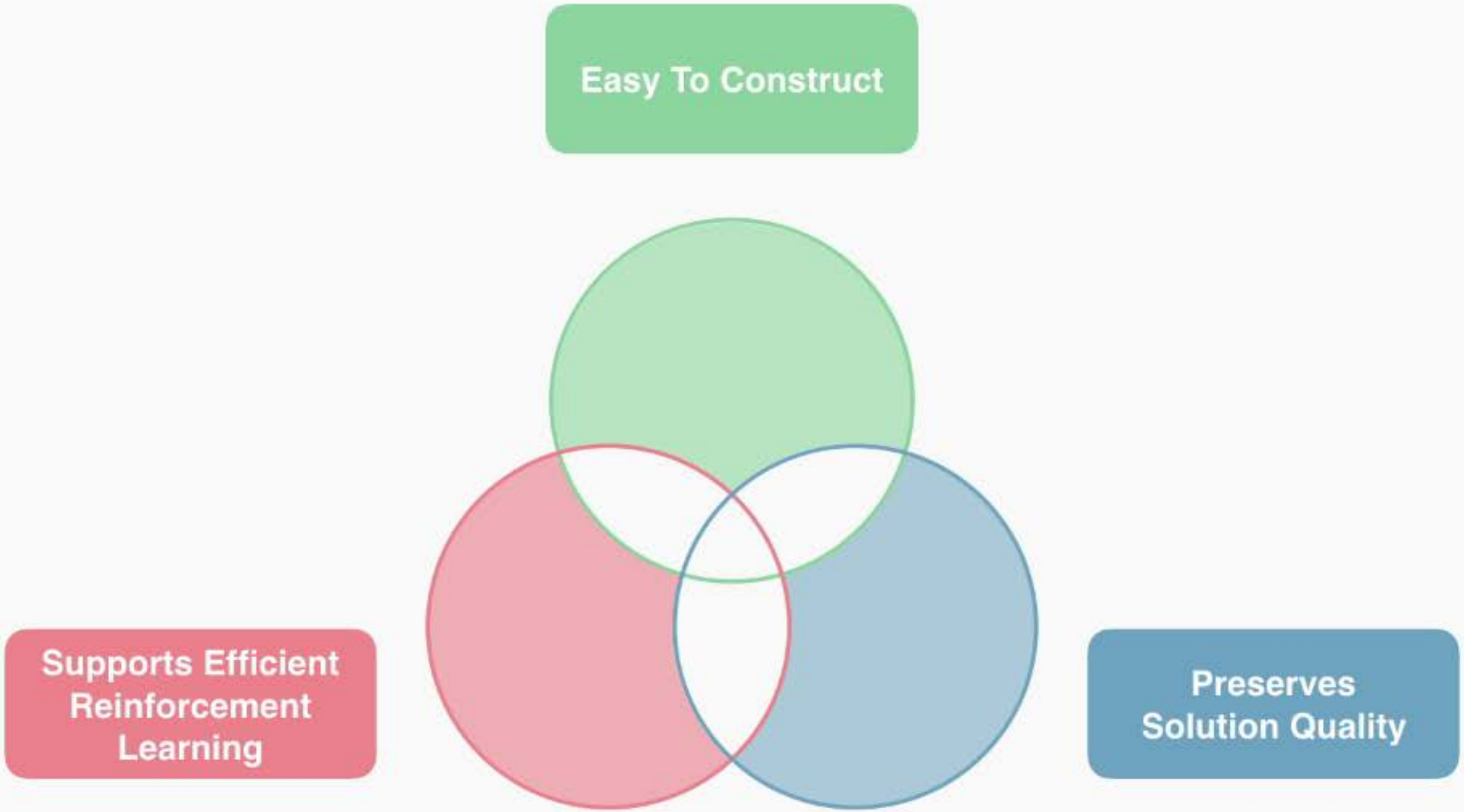
Q: Which kinds of abstractions are desirable?

- Easy To Construct
- Supports Efficient Reinforcement Learning
- Preserves Solution Quality

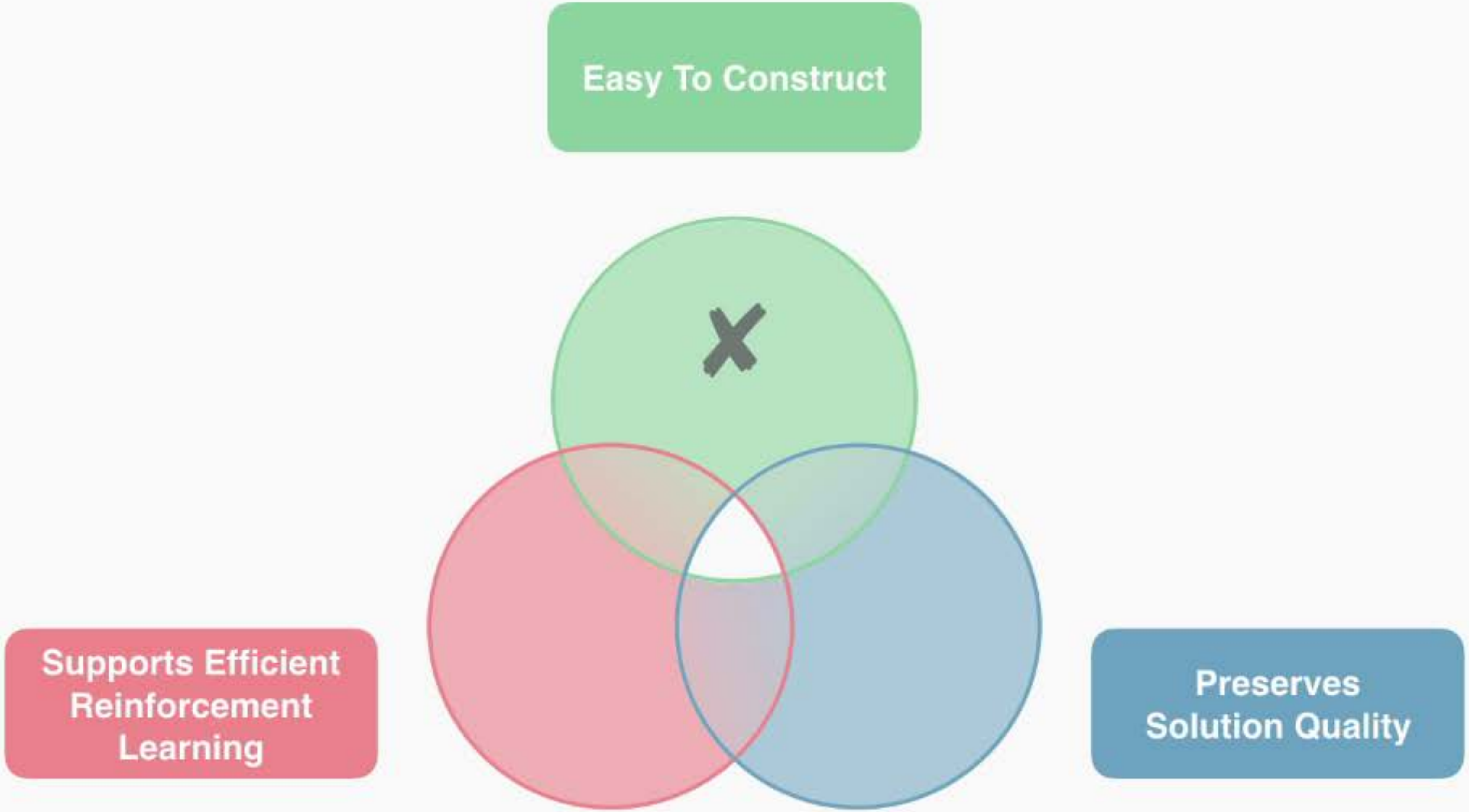
Abstraction Desiderata



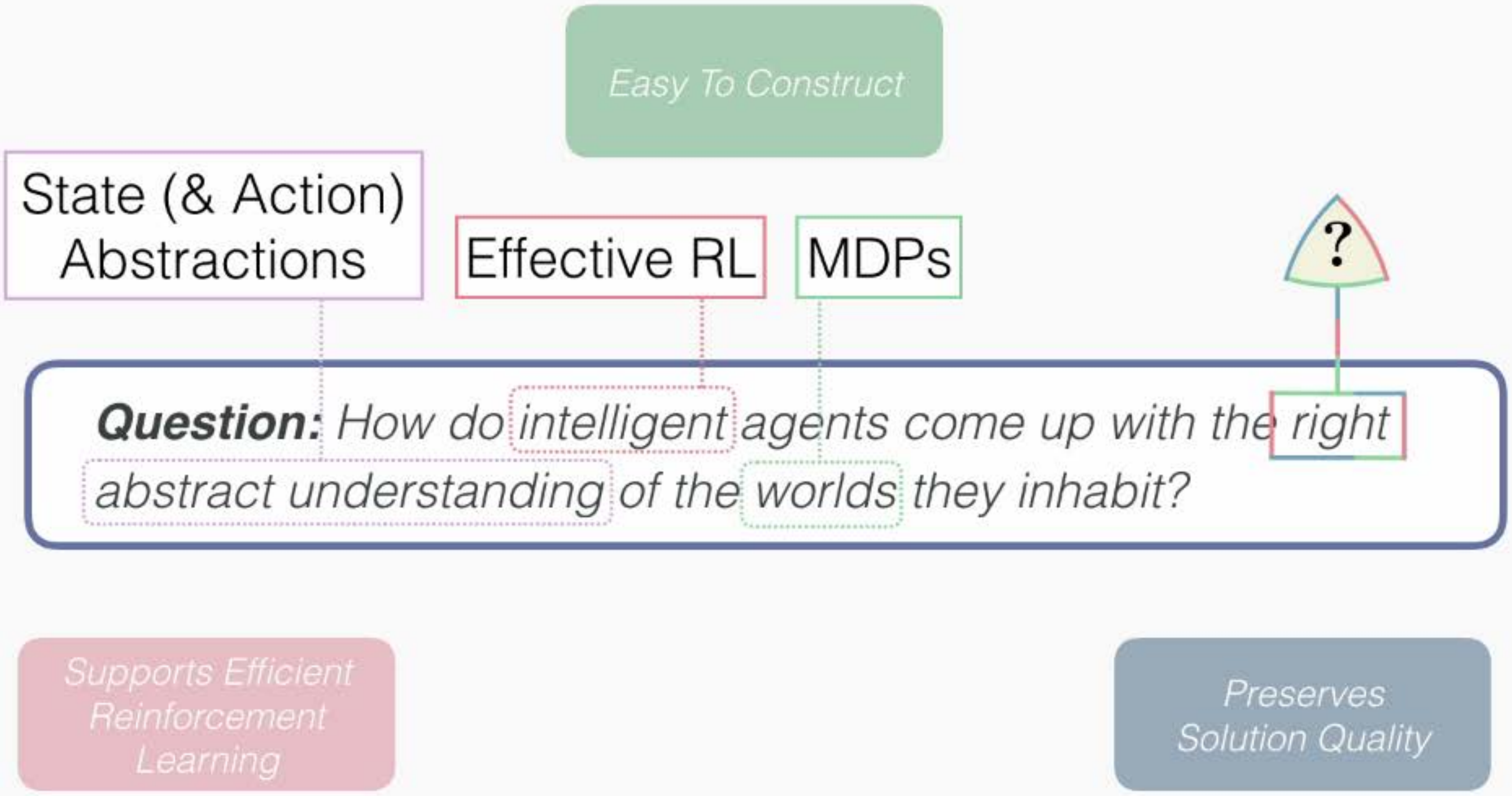
Abstraction Desiderata



Abstraction Desiderata

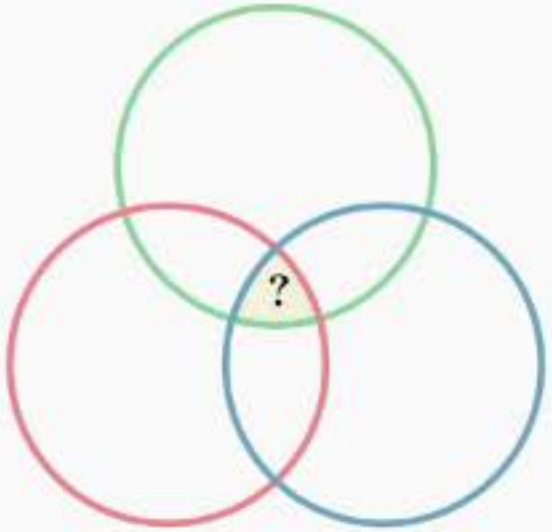


Abstraction Desiderata



Talk Overview

Easy To Construct



Supports Efficient Reinforcement Learning

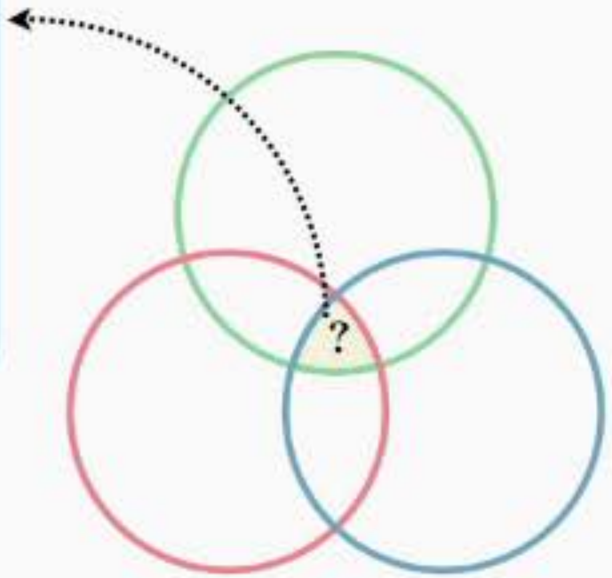
Preserves Solution Quality

Talk Overview

Easy To Construct

1 **State Abstraction**

*[**A**AAJLW AAAI '19]*



Supports Efficient Reinforcement Learning

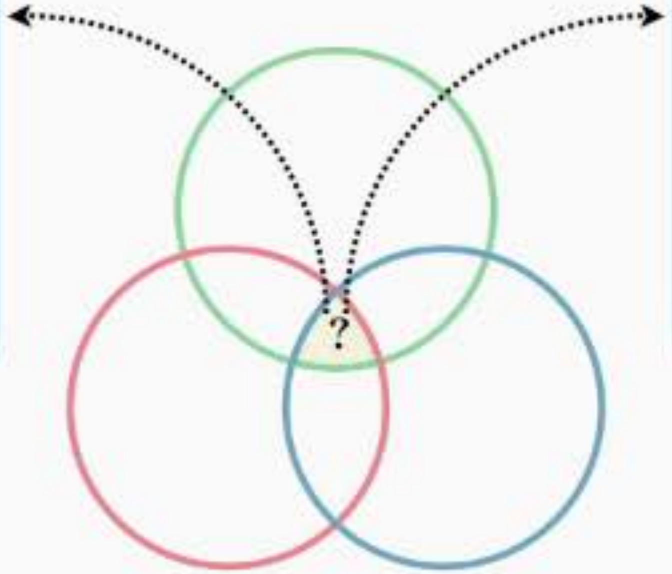
Preserves Solution Quality

Talk Overview

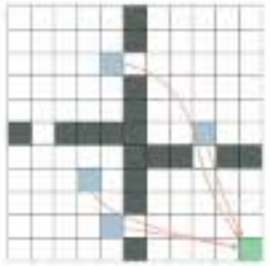
Easy To Construct

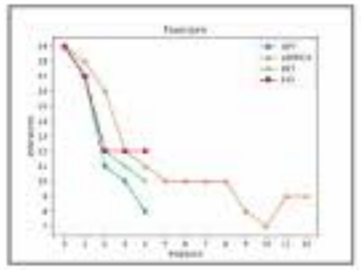
1 State Abstraction

*[**A**AAJLW AAAI '19]*



2 Action Abstraction



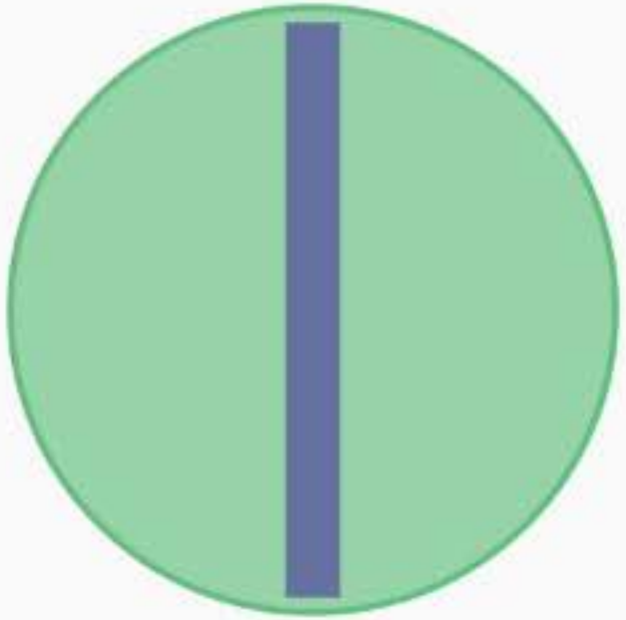


*[J**A**HLK ICML '19]*

Supports Efficient Reinforcement Learning

Preserves Solution Quality

Compression vs. Value



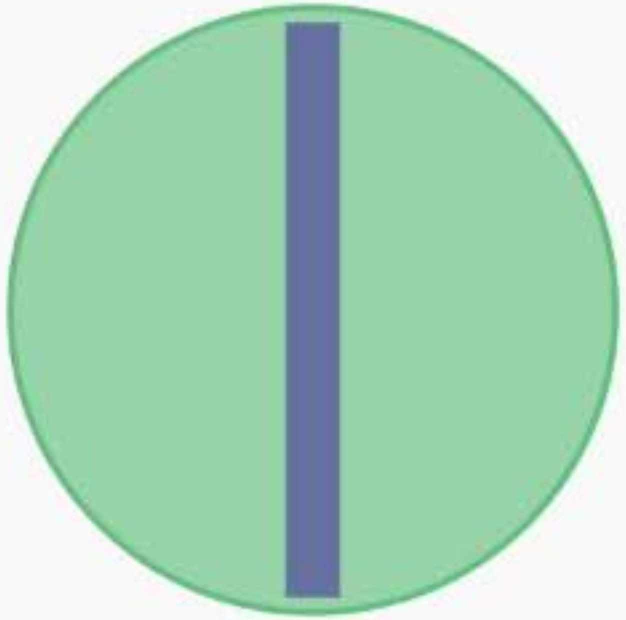
[Abel, Arumugam, Asadi, Jinnai, Littman, Wong; AAAI 2019]

Compression vs. Value

High Value
No Compression

Some Value
Some Compression

No Value
High Compression



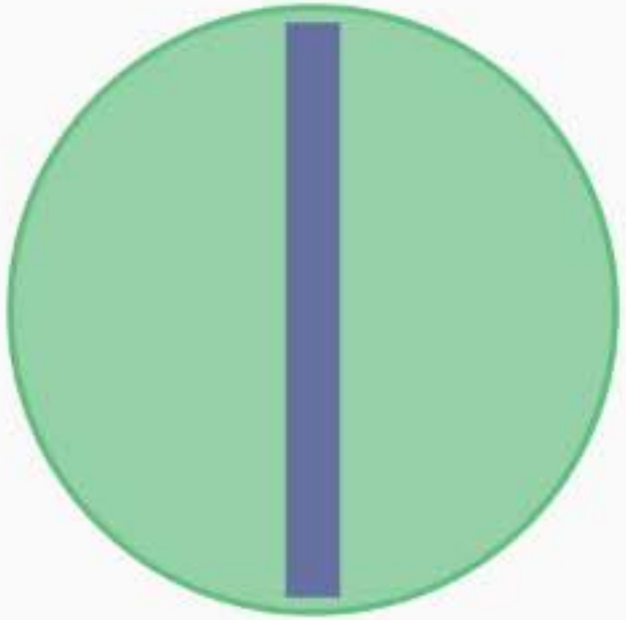
Question: *Can we find state abstractions that minimize* $|\mathcal{S}_\phi|$ *while still representing good policies?*

Compression vs. Value

High Value
No Compression

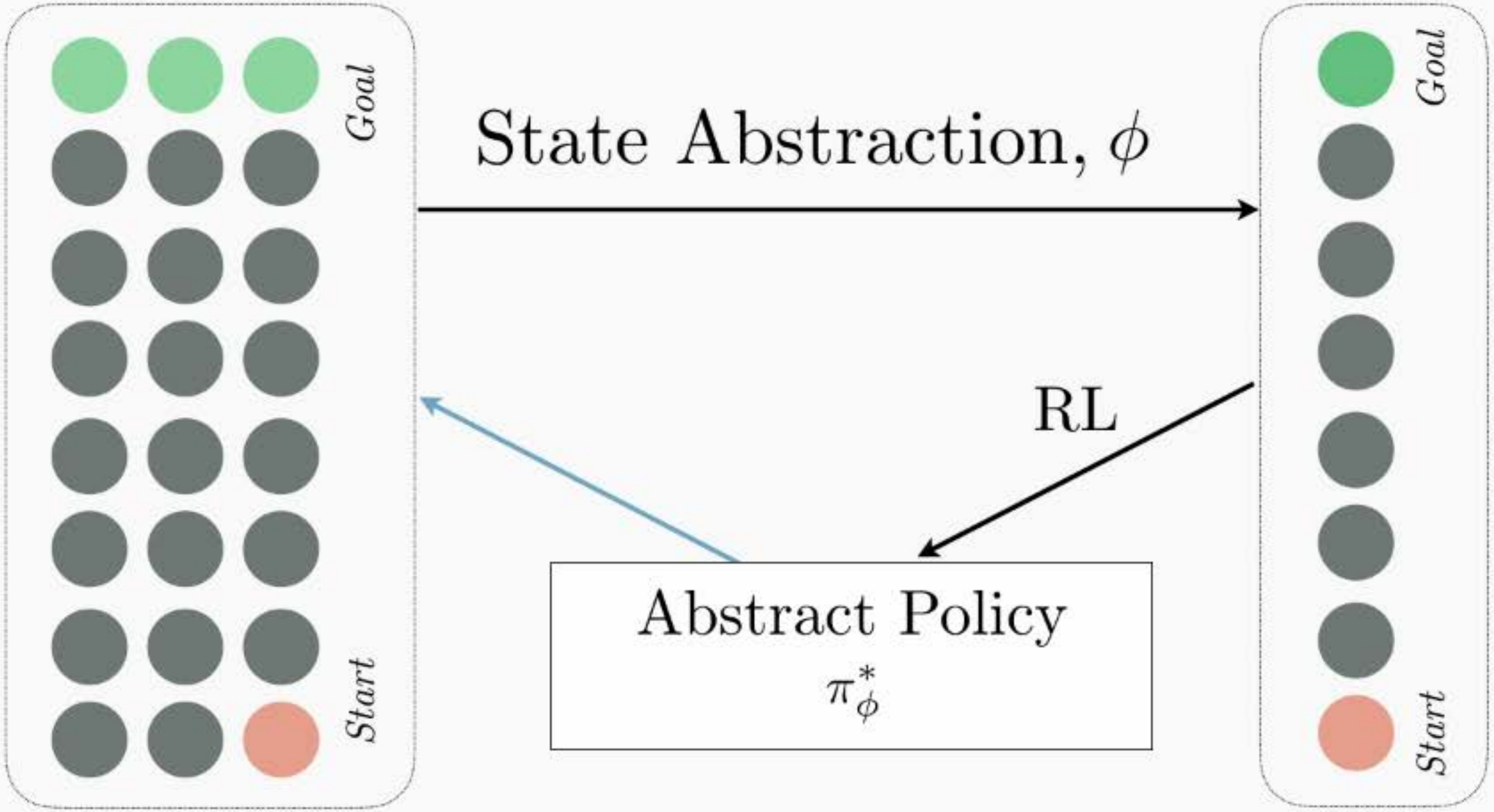
Some Value
Some Compression

No Value
High Compression

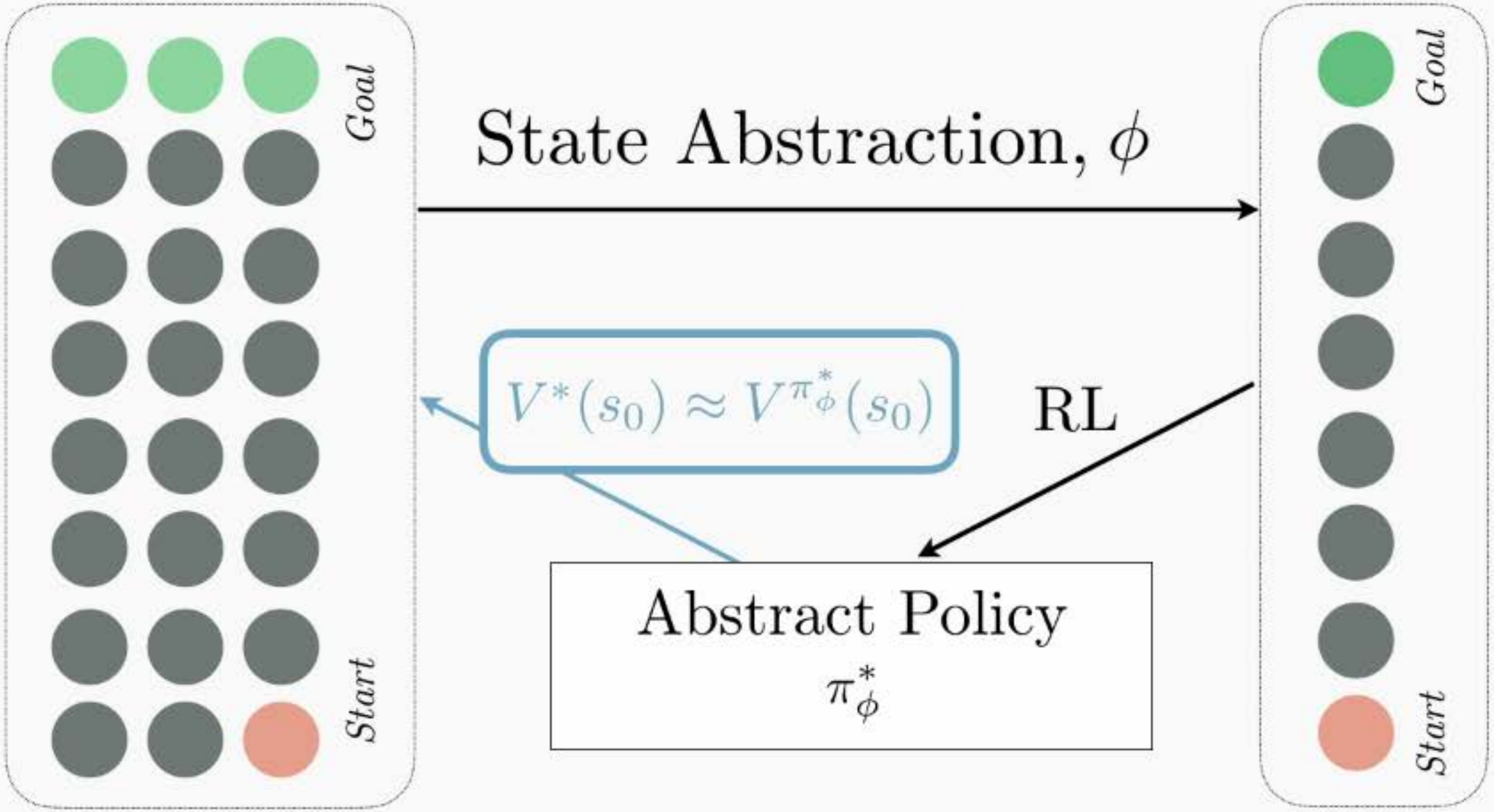


Question: *Can we find state abstractions that minimize* $|\mathcal{S}_\phi|$ *while still representing good policies?* **Preserves Solution Quality**

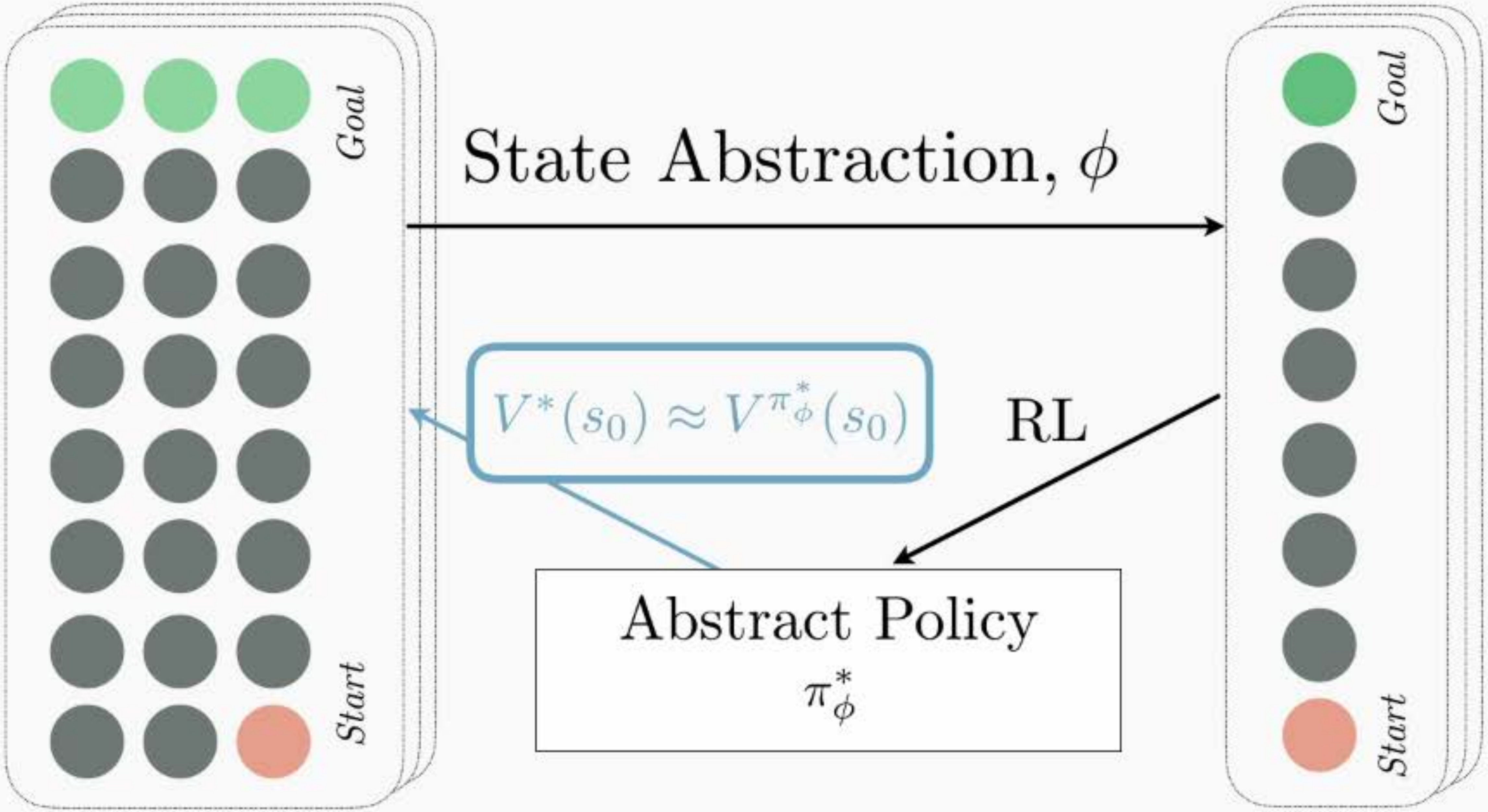
Preserves Solution Quality



Preserves Solution Quality



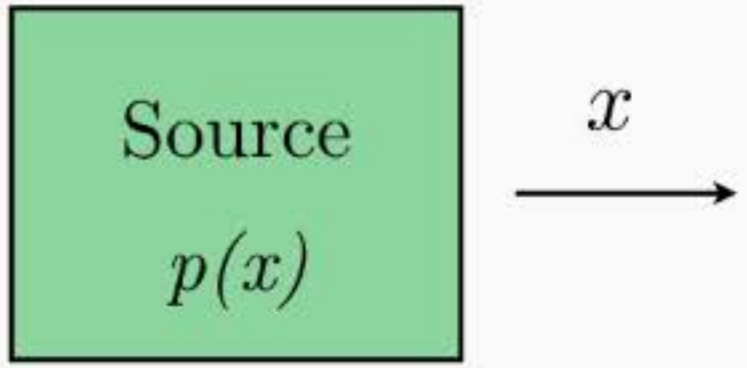
Preserves Solution Quality



Rate-Distortion Theory

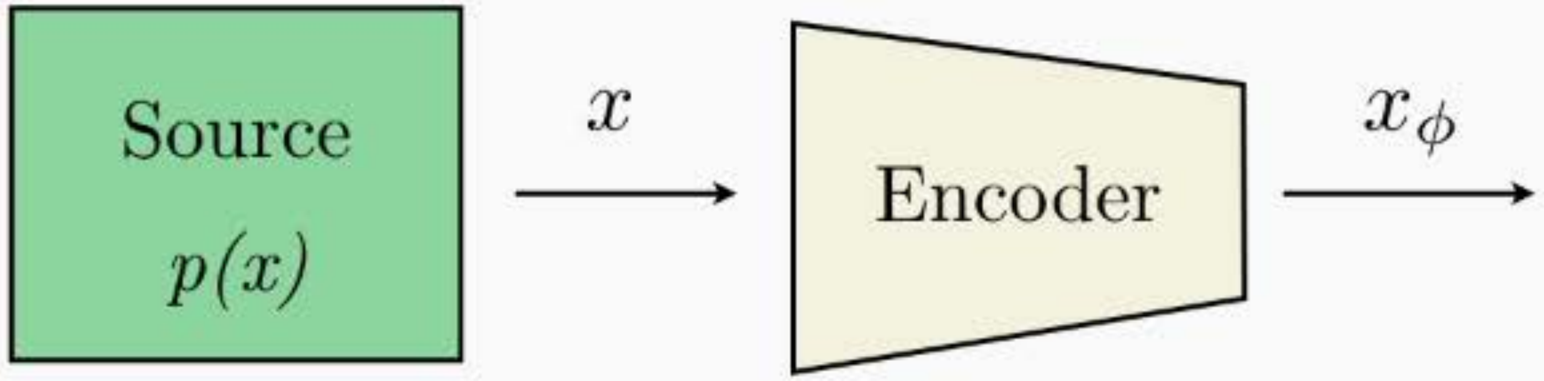
[Shannon '48, Berger '03]

Rate-Distortion Theory



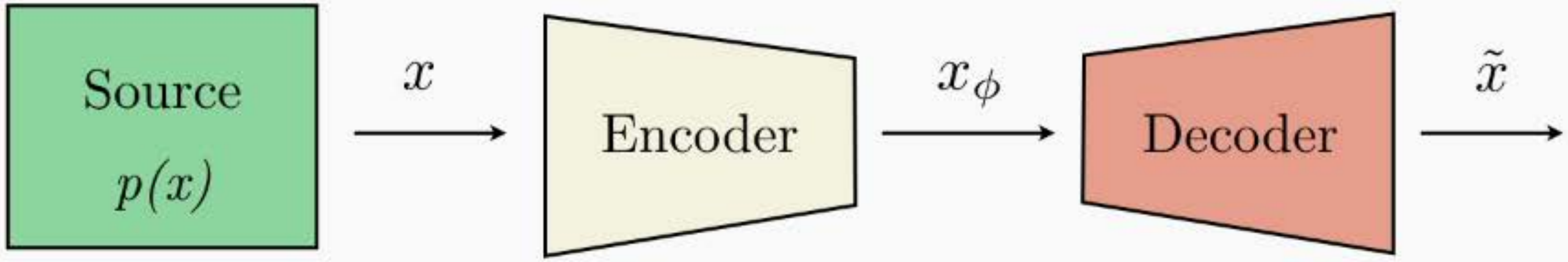
[Shannon '48, Berger '03]

Rate-Distortion Theory



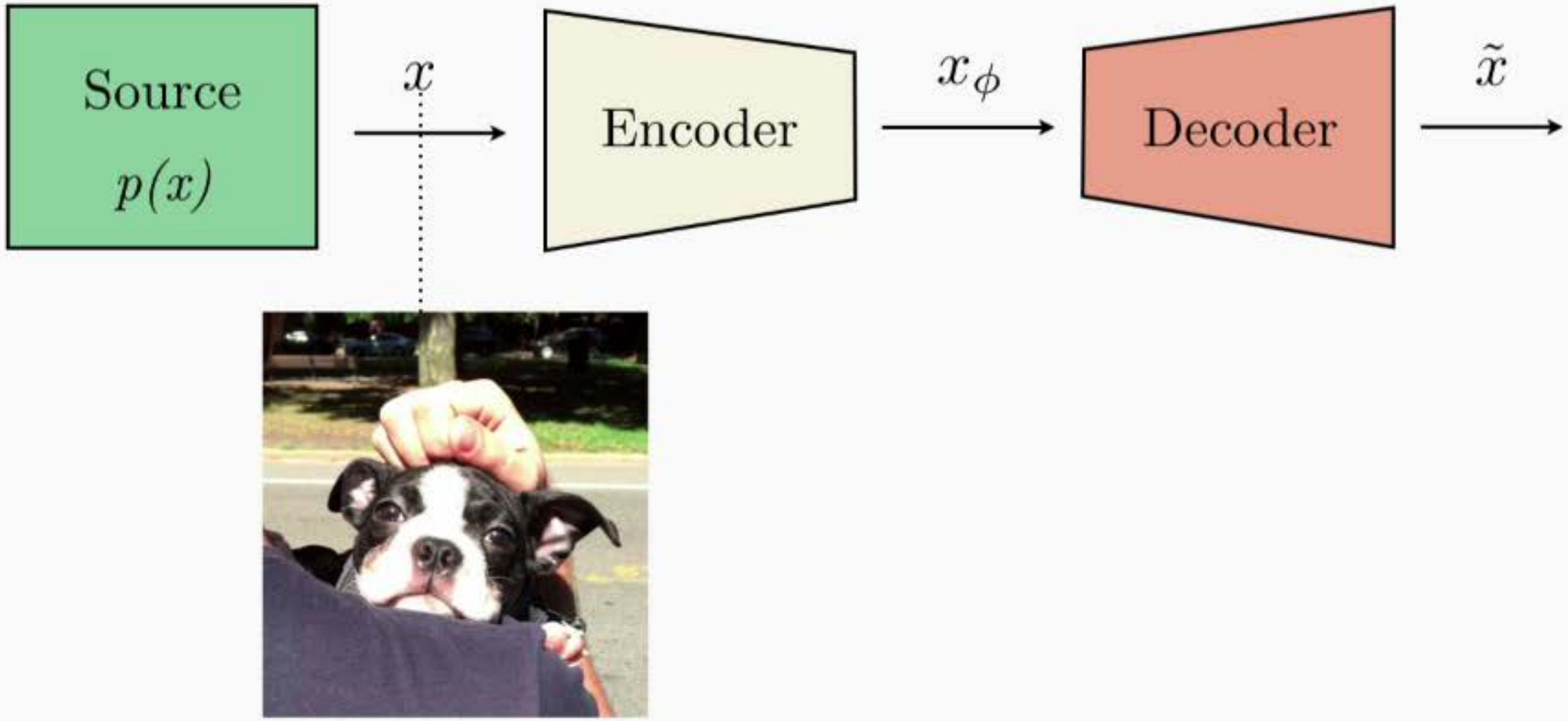
[Shannon '48, Berger '03]

Rate-Distortion Theory



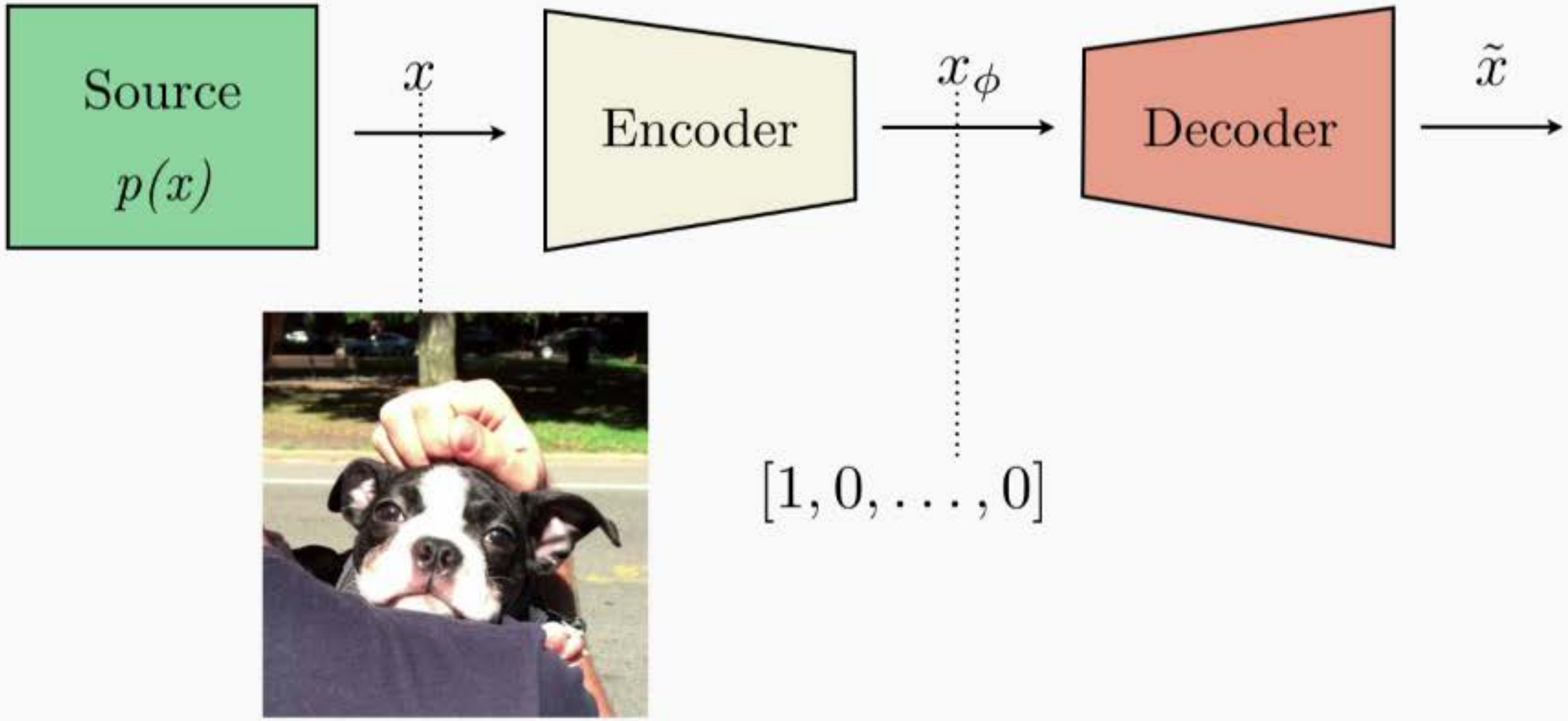
[Shannon '48, Berger '03]

Rate-Distortion Theory



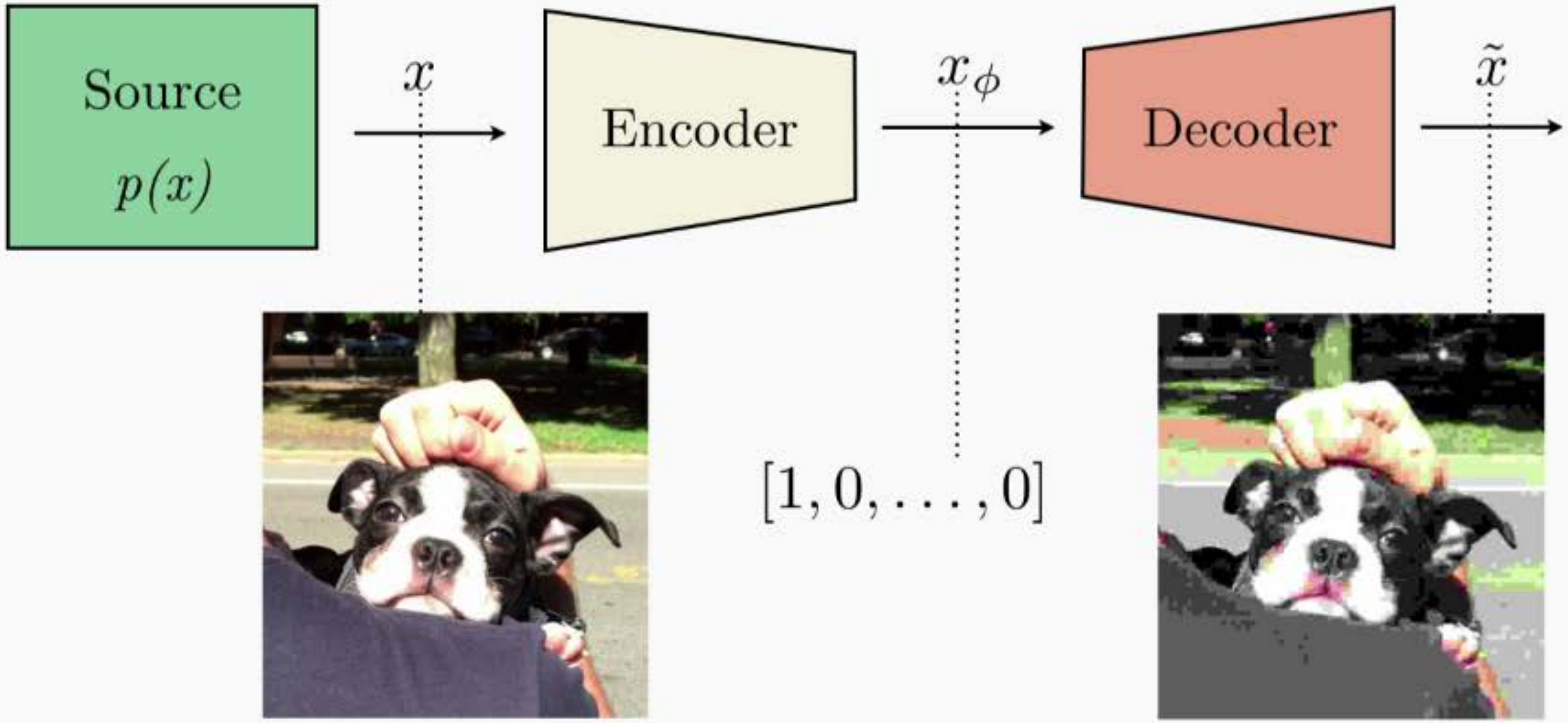
[Shannon '48, Berger '03]

Rate-Distortion Theory



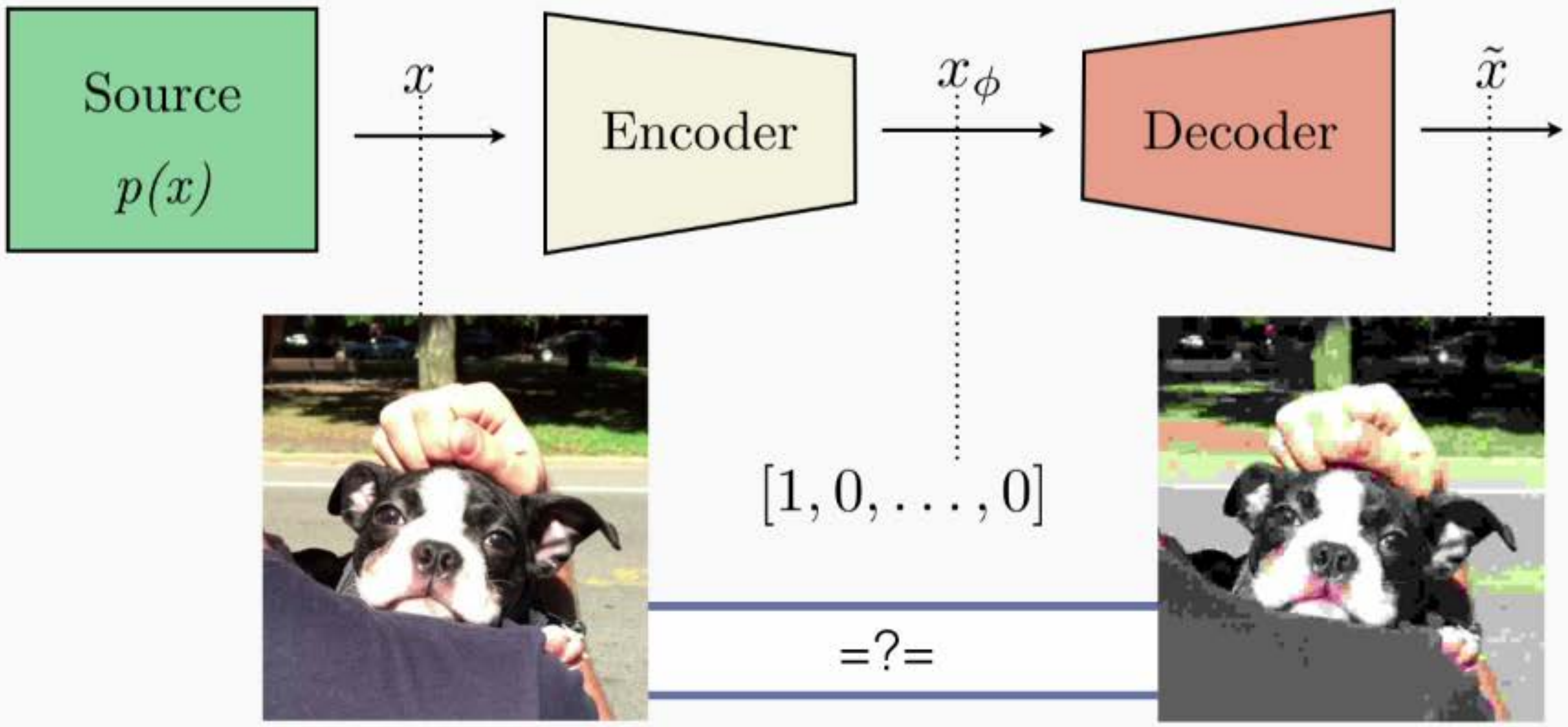
[Shannon '48, Berger '03]

Rate-Distortion Theory



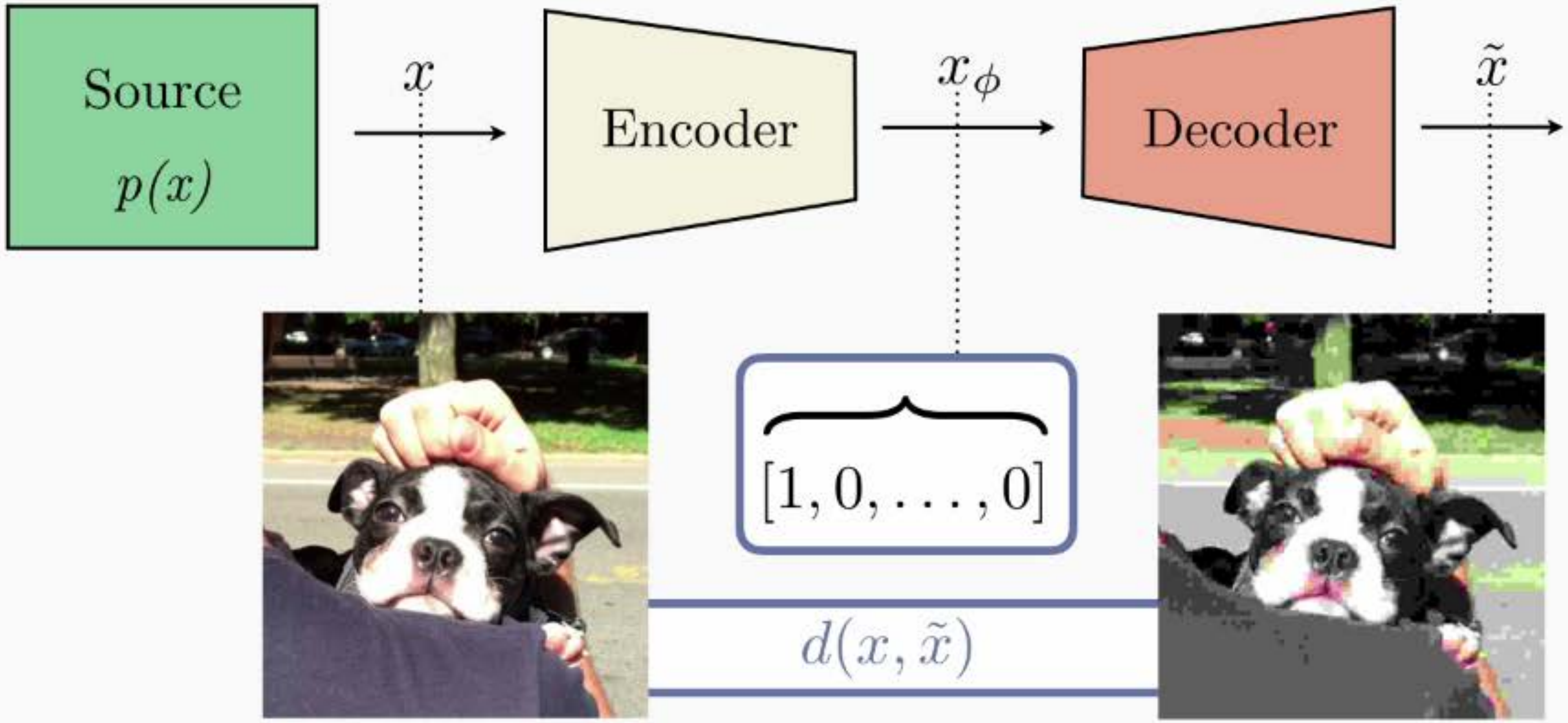
[Shannon '48, Berger '03]

Rate-Distortion Theory



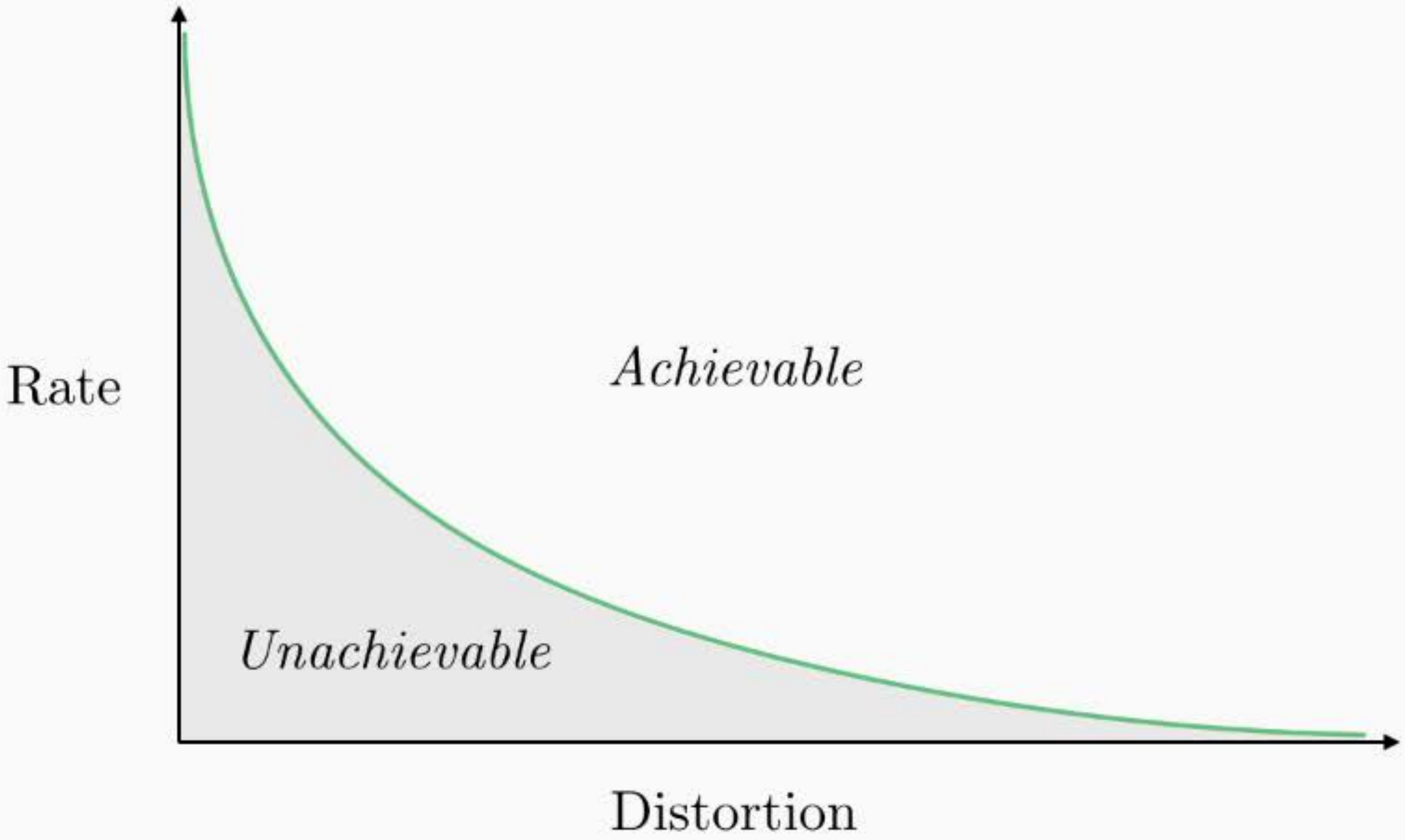
[Shannon '48, Berger '03]

Rate-Distortion Theory

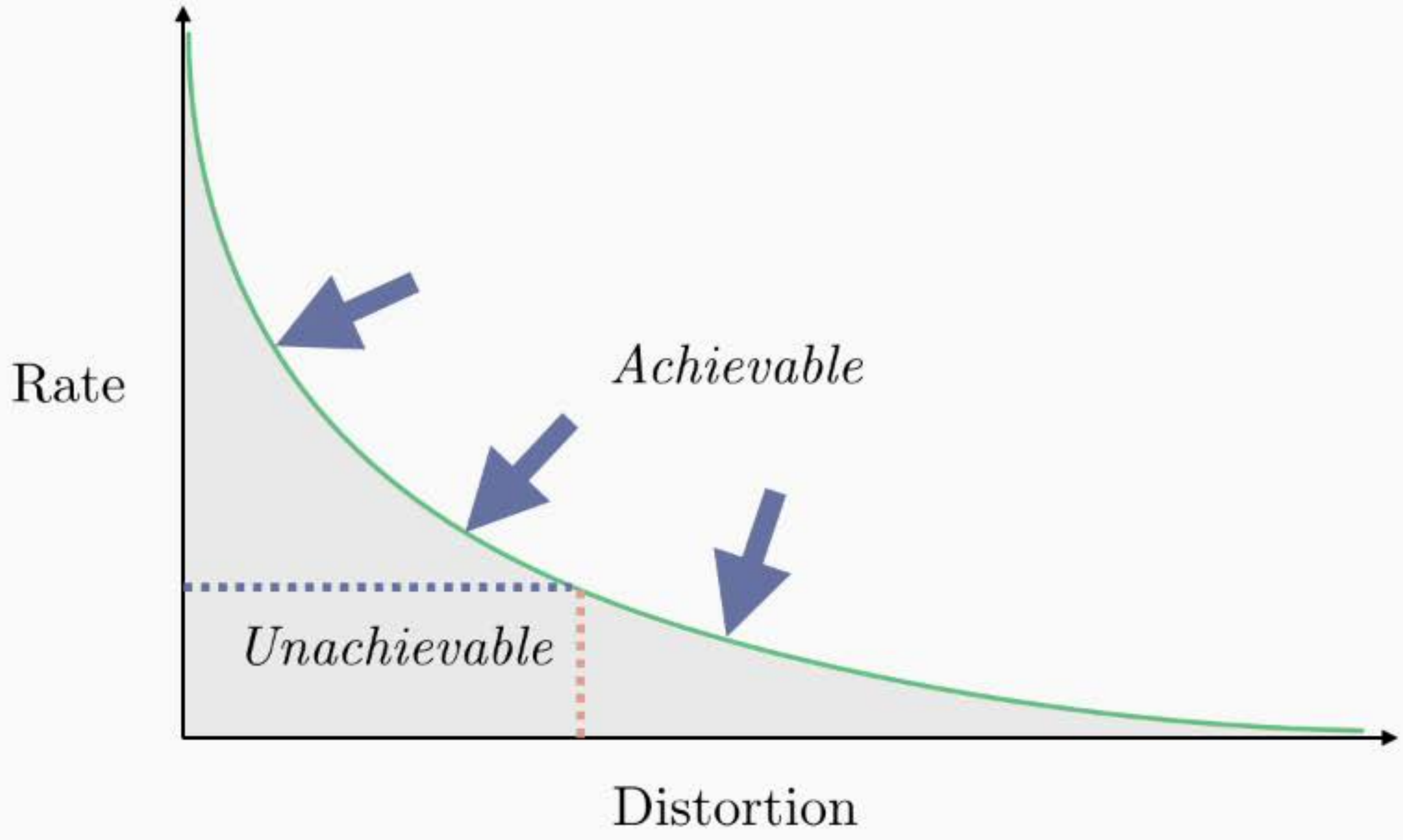


[Shannon '48, Berger '03]

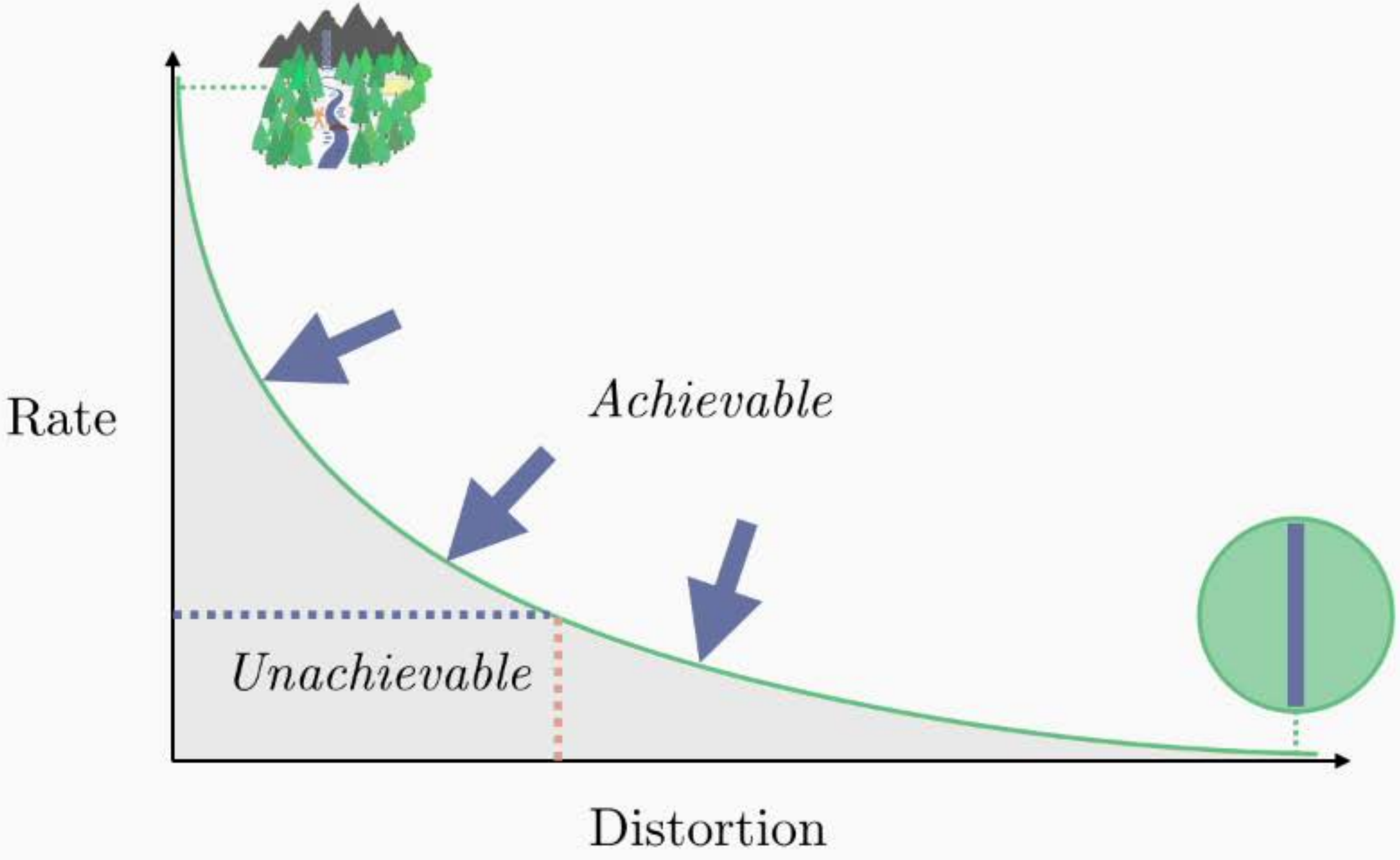
Rate-Distortion Theory



Rate-Distortion Theory



Rate-Distortion Theory



Blahut-Arimoto

Problem: $$\min_{p(\tilde{x}|x)} \underbrace{I(X;\tilde{X})}_{Rate} + \beta \underbrace{\mathbb{E}\left[d(x,\tilde{x})\right]}_{Distortion}.$$

[Blahut 1972, Arimoto, 1972]

Blahut-Arimoto

Problem: $$\min_{p(\tilde{x}|x)} \underbrace{I(X;\tilde{X})}_{Rate} + \beta \underbrace{\mathbb{E}\left[d(x,\tilde{x})\right]}_{Distortion}.$$

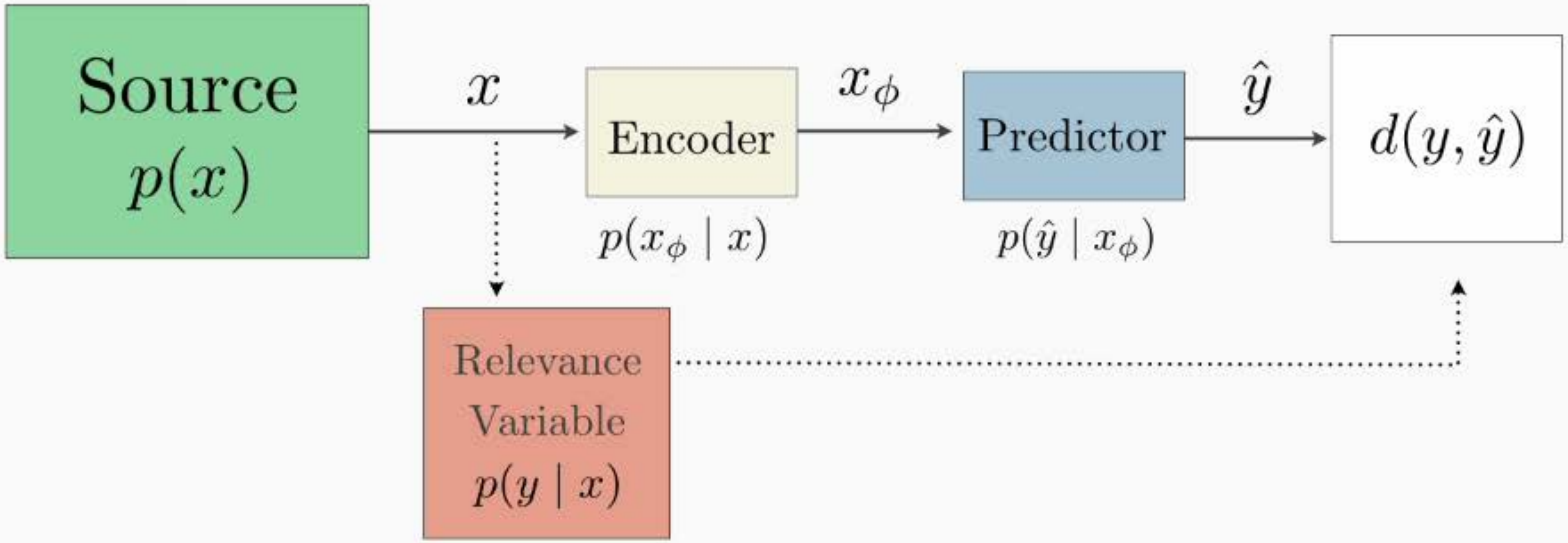
Solution:

$$p_{t+1}(\tilde{x}) = \sum_x p_t(x) p_t(\tilde{x} \mid x)$$

$$p_{t+1}(\tilde{x} \mid x) = \frac{p_t(\tilde{x}) \exp\{-\beta d(x,\tilde{x})\}}{\sum_x p_t(x) \exp\{-\beta d(x,\tilde{x})\}}$$

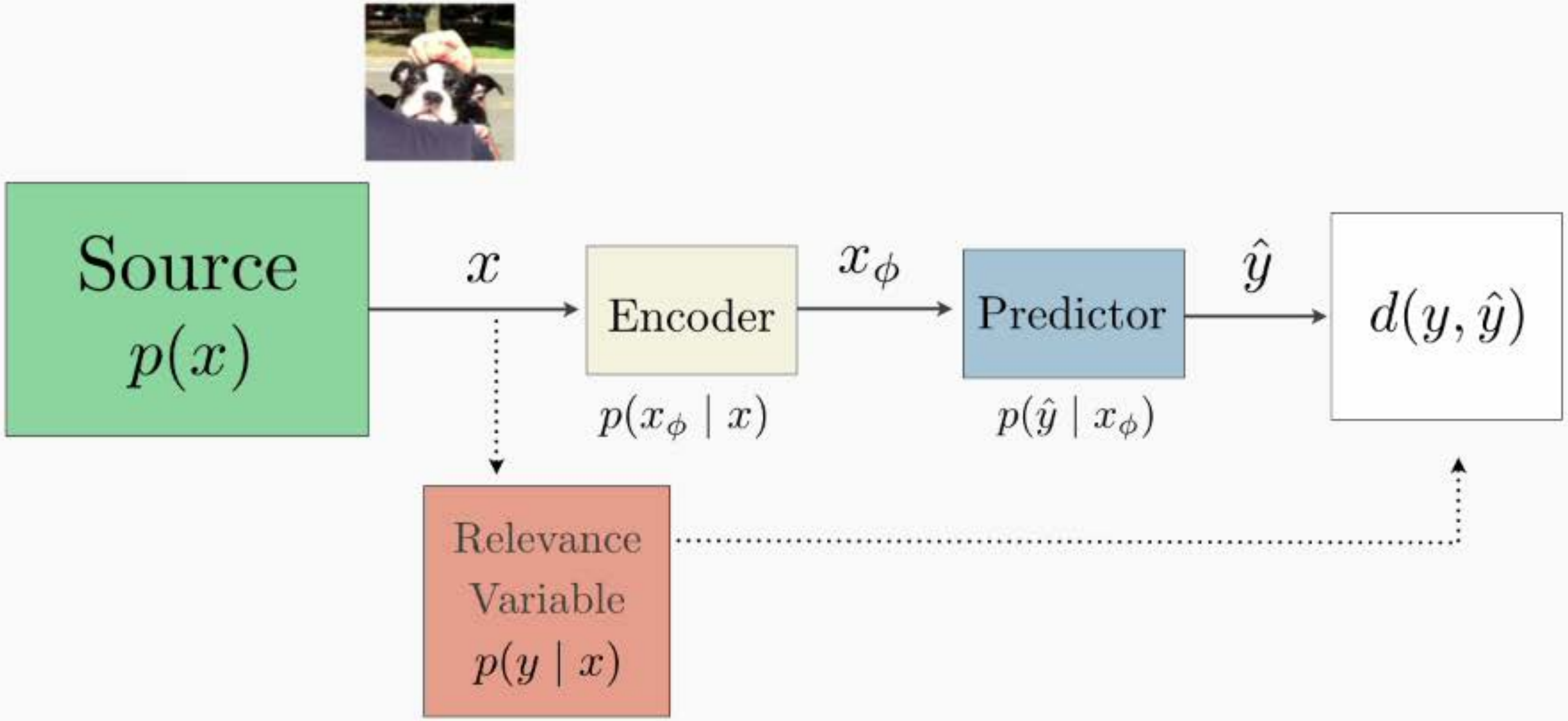
[Blahut 1972, Arimoto, 1972]

Information Bottleneck



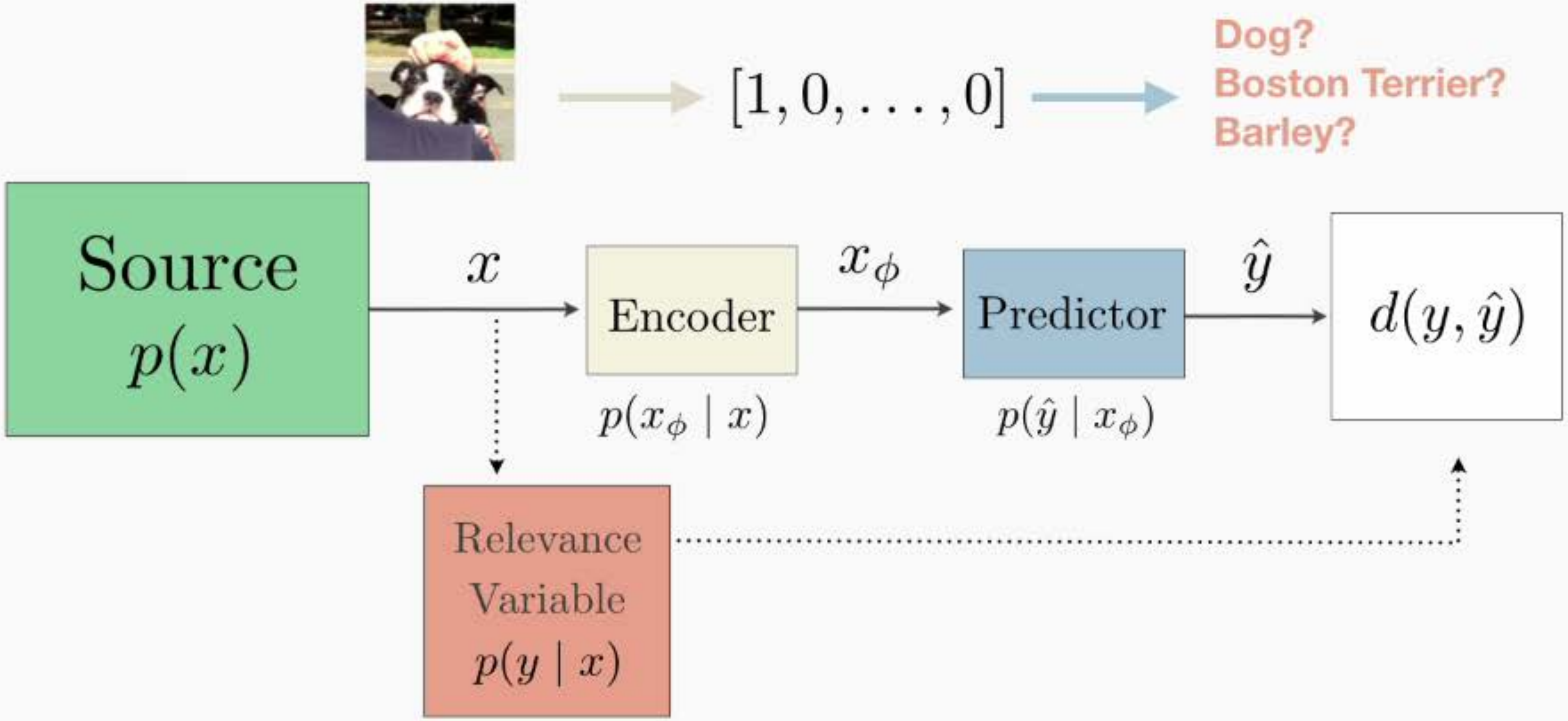
[Tishby, Pereira, Bialek '99]

Information Bottleneck



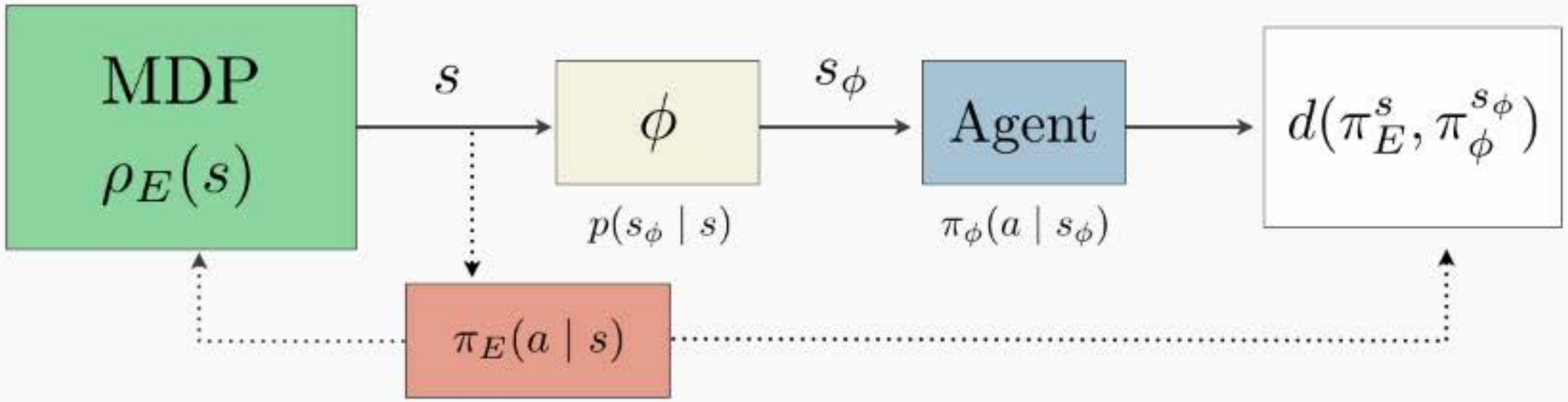
[Tishby, Pereira, Bialek '99]

Information Bottleneck

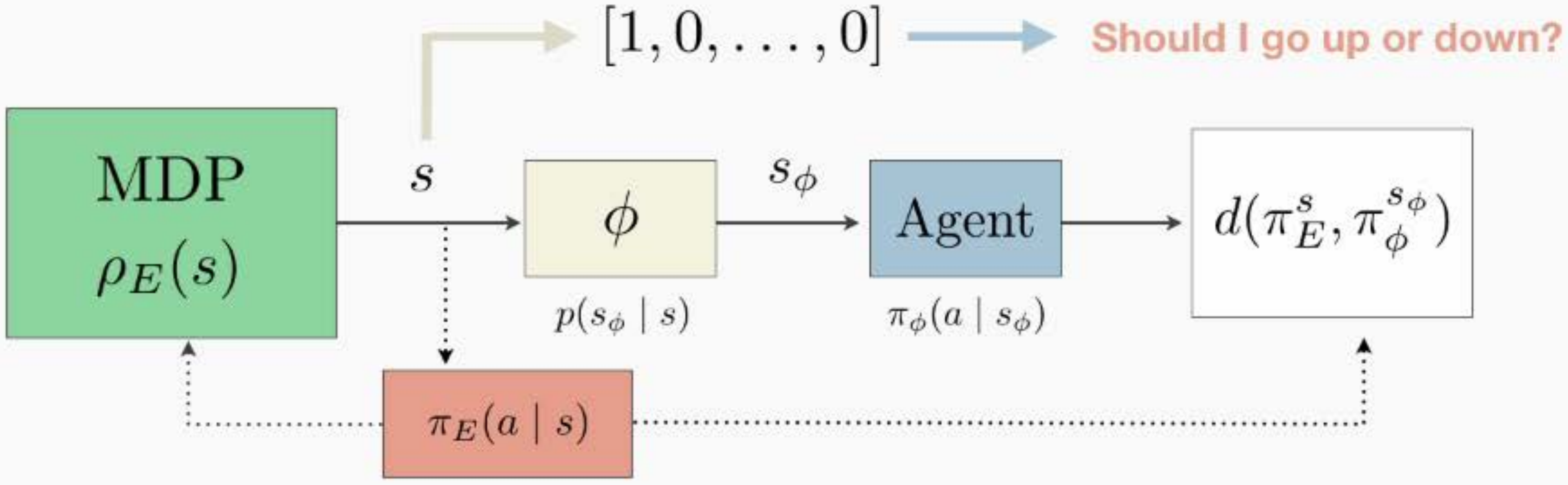


[Tishby, Pereira, Bialek '99]

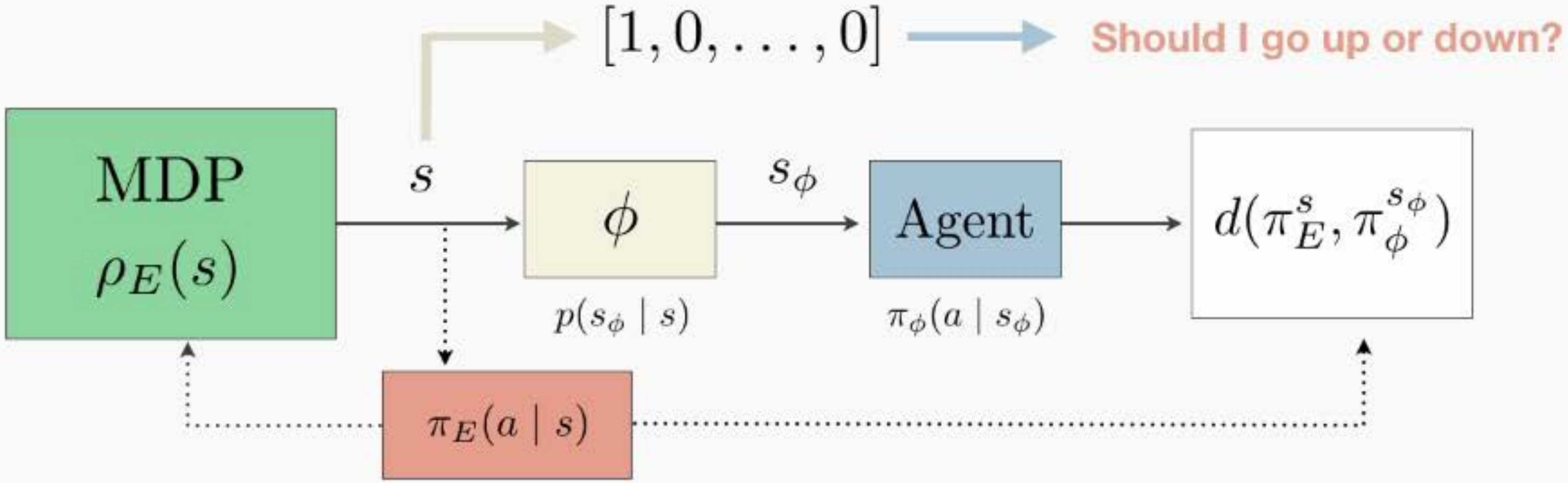
State Abstraction as Compression



State Abstraction as Compression



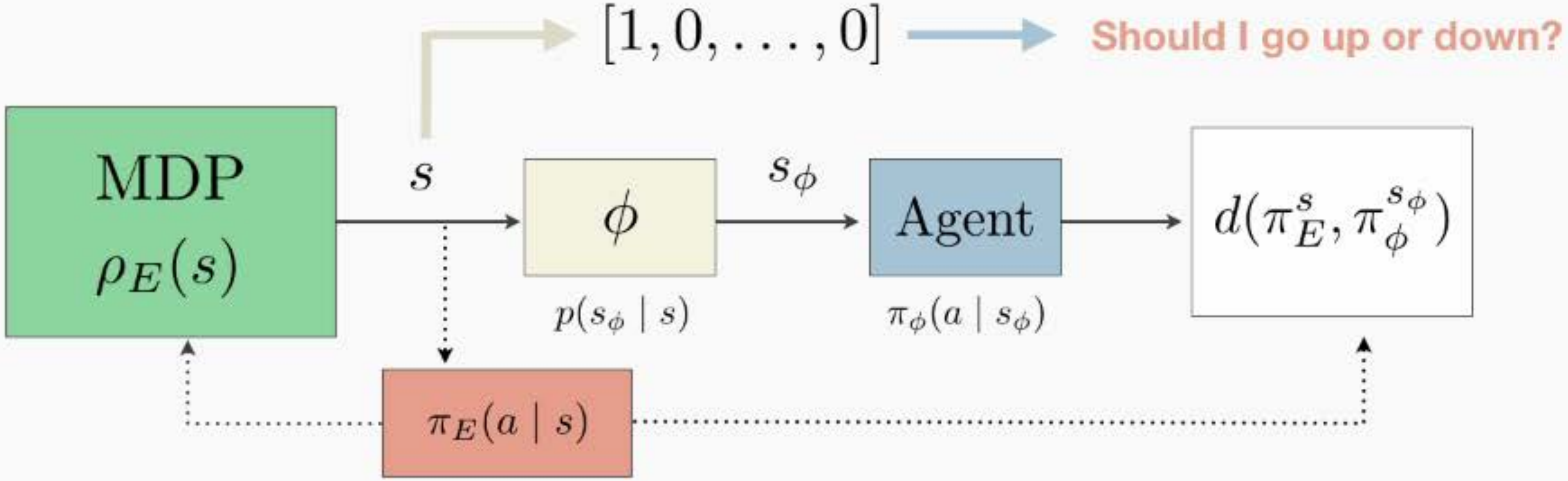
State Abstraction as Compression



$$\min_\phi \left(|\mathcal{S}_\phi| + \beta \mathop{\mathbb{E}}_{\rho_E(s)} \left[V^{\pi_E}(s) - V^{\pi_\phi^*}(s) \right] \right)$$

Our Objective

State Abstraction as Compression

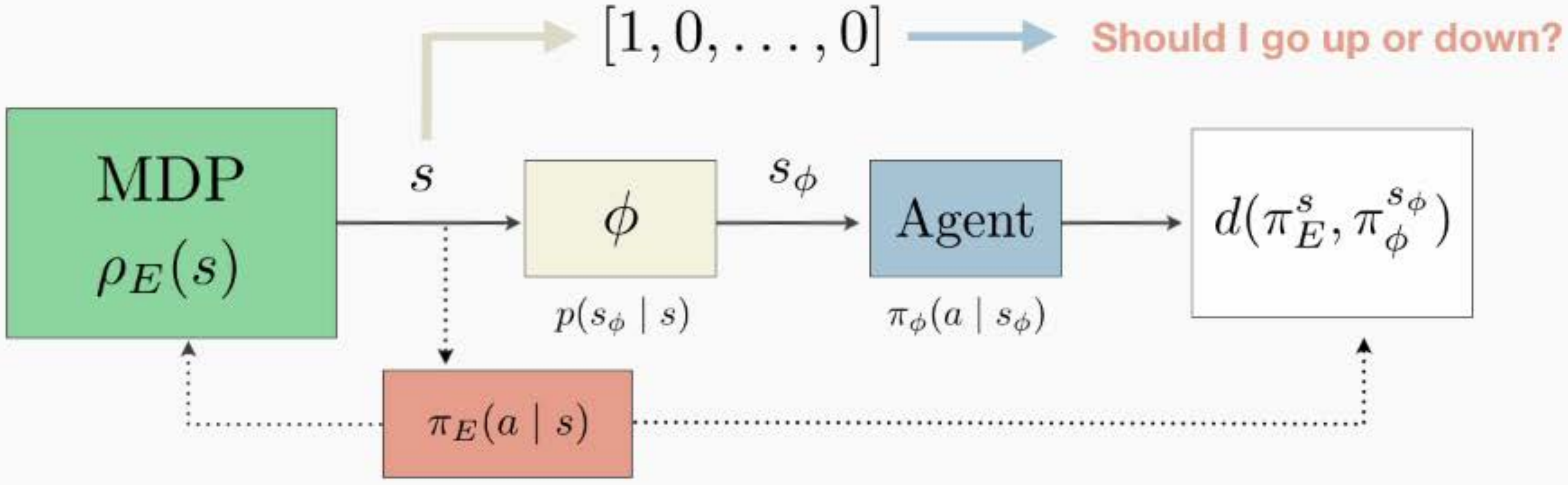


$$\min_\phi \left(|\mathcal{S}_\phi| + \beta \mathop{\mathbb{E}}_{\rho_E(s)} \left[V^{\pi_E}(s) - V^{\pi_\phi^*}(s) \right] \right)$$

Our Objective

Preserves Solution Quality

State Abstraction as Compression



$$\min_{\phi} \left(|\mathcal{S}_\phi| + \beta \mathop{\mathbb{E}}_{\rho_E(s)} \left[V^{\pi_E}(s) - V^{\pi_\phi^*}(s) \right] \right)$$

Our Objective

Preserves Solution Quality

State Abstraction as Compression

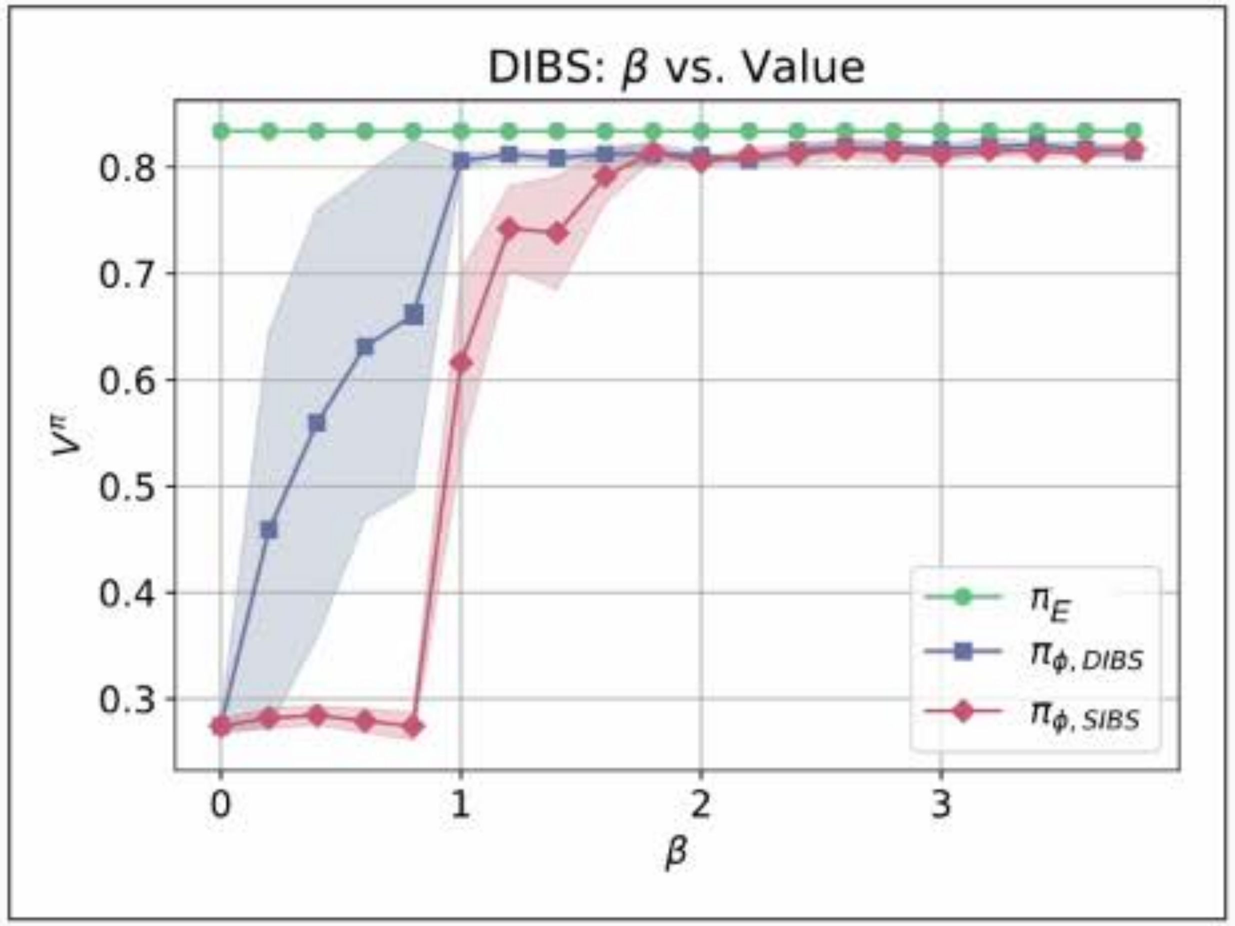
Theorem.

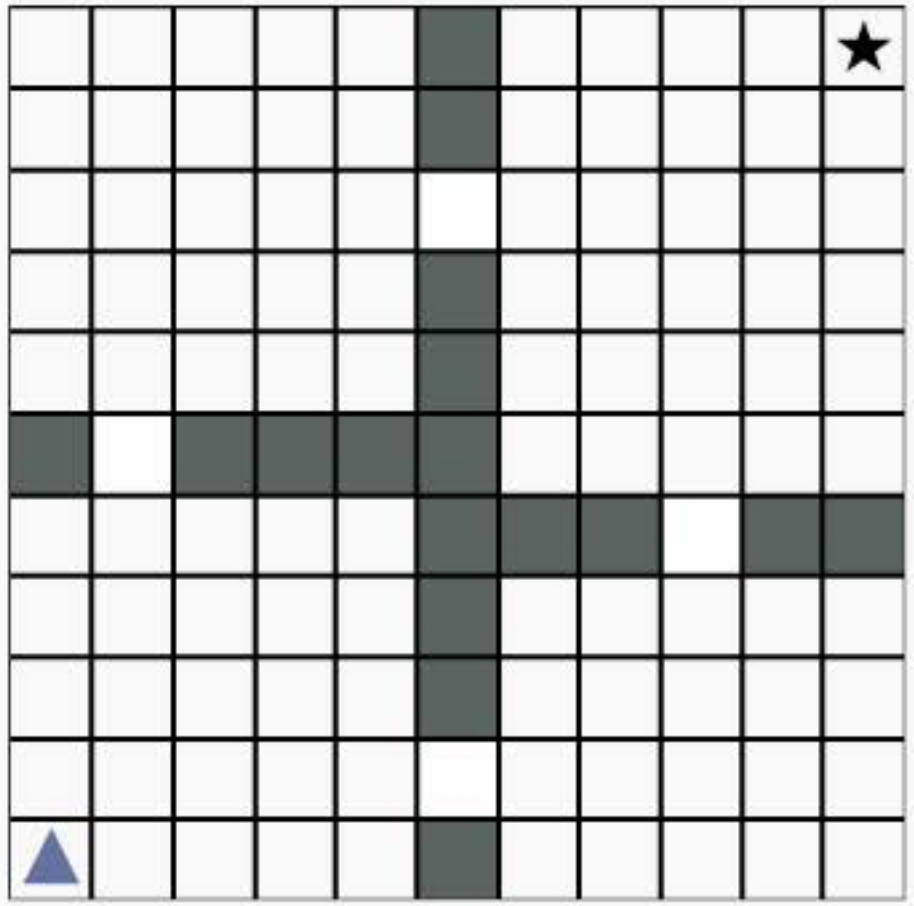
$$\min_{\phi} \left(|\mathcal{S}_\phi| + \beta \underset{\rho_E(s)}{\mathbb{E}} \left[V^{\pi_E}(s) - V^{\pi^*_\phi}(s) \right] \right)$$

Our Objective $\leq$ *DIB Objective*

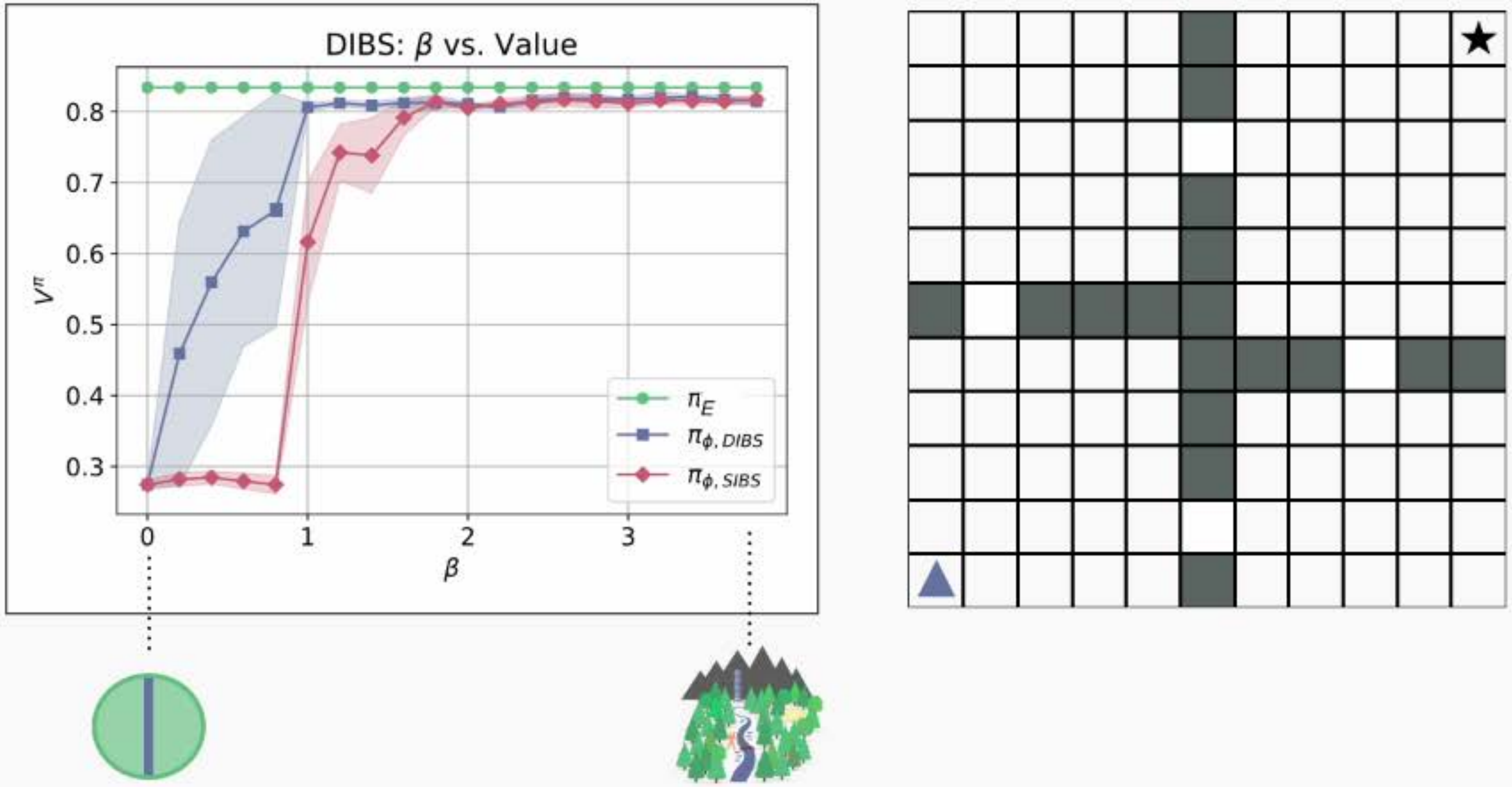
$$\min_{\phi} \left(C_1 H(\rho_\phi) + C_2 \beta \underset{\rho_E(s)}{\mathbb{E}} \left[D_{\mathrm{KL}} \left(\pi_E(s) \,||\, \pi^*_\phi(s) \right) \right] \right)$$

State Abstraction as Compression

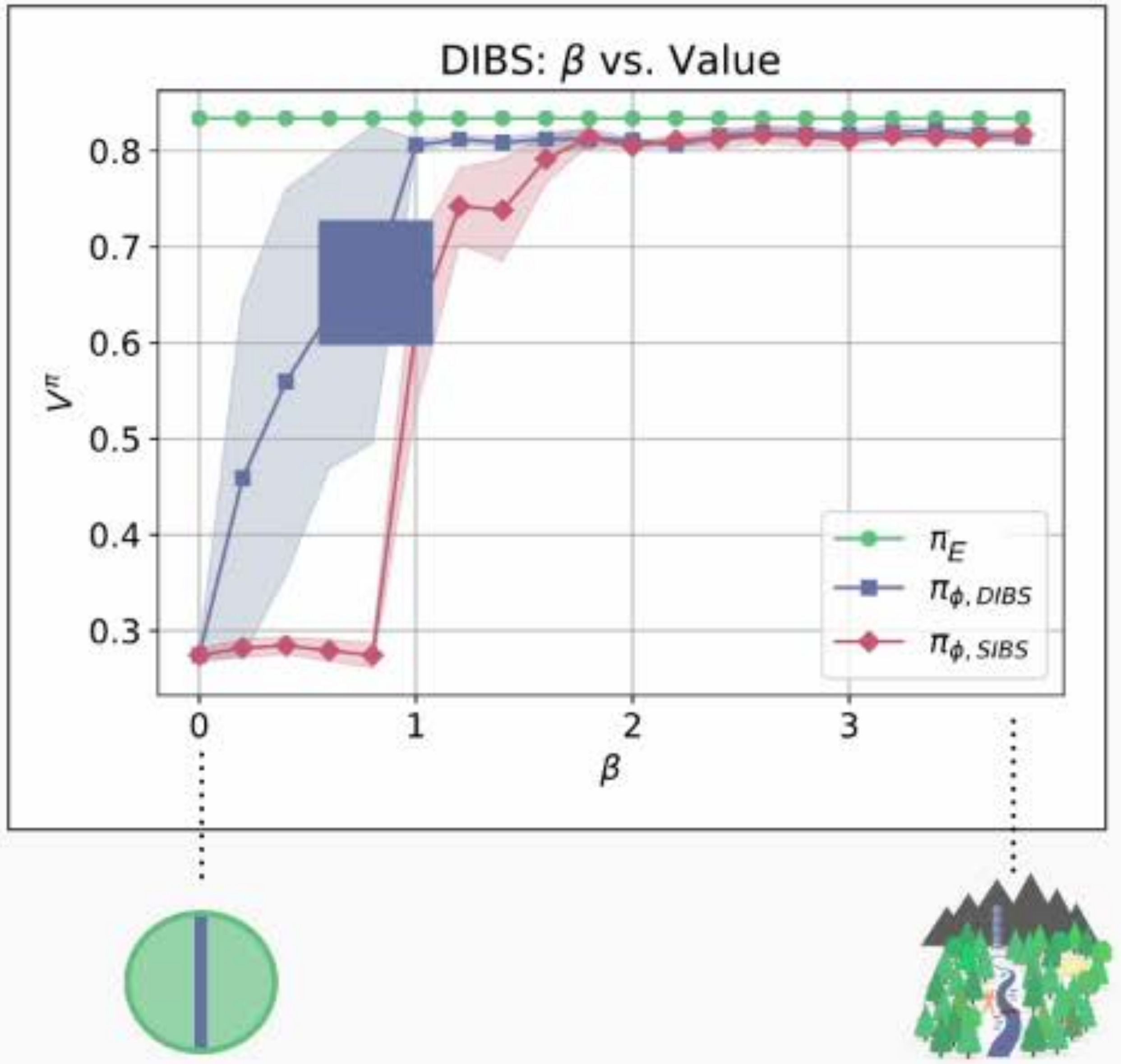


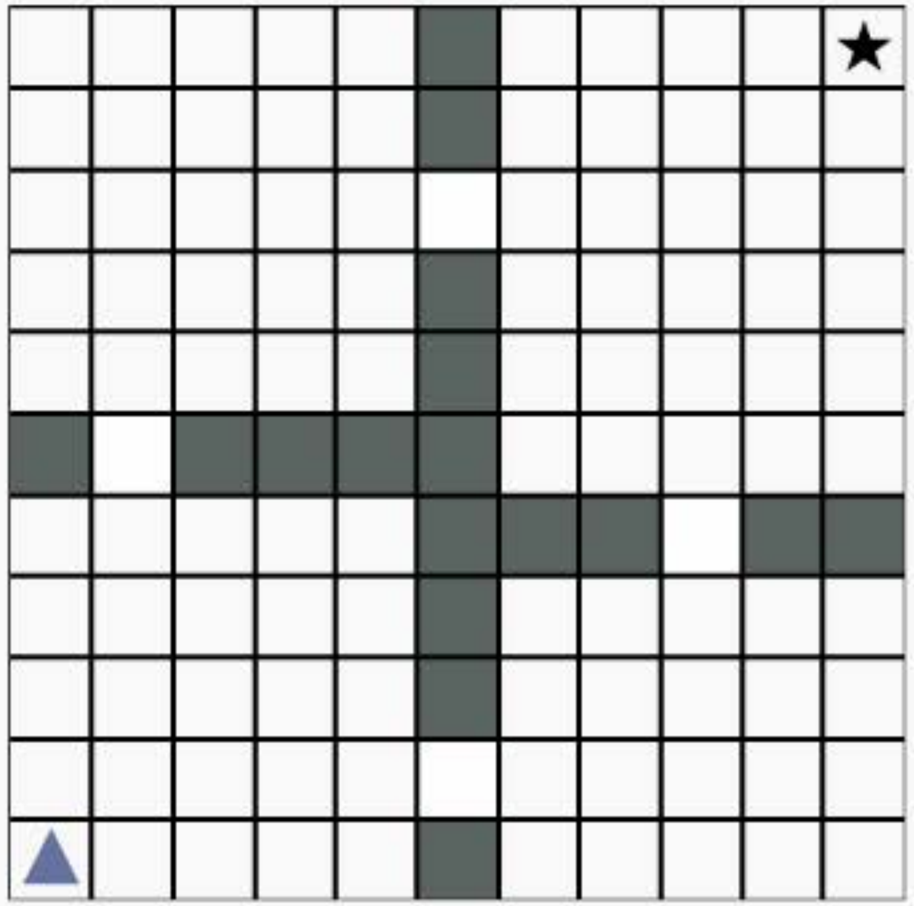


State Abstraction as Compression



State Abstraction as Compression





State Abstraction as Compression

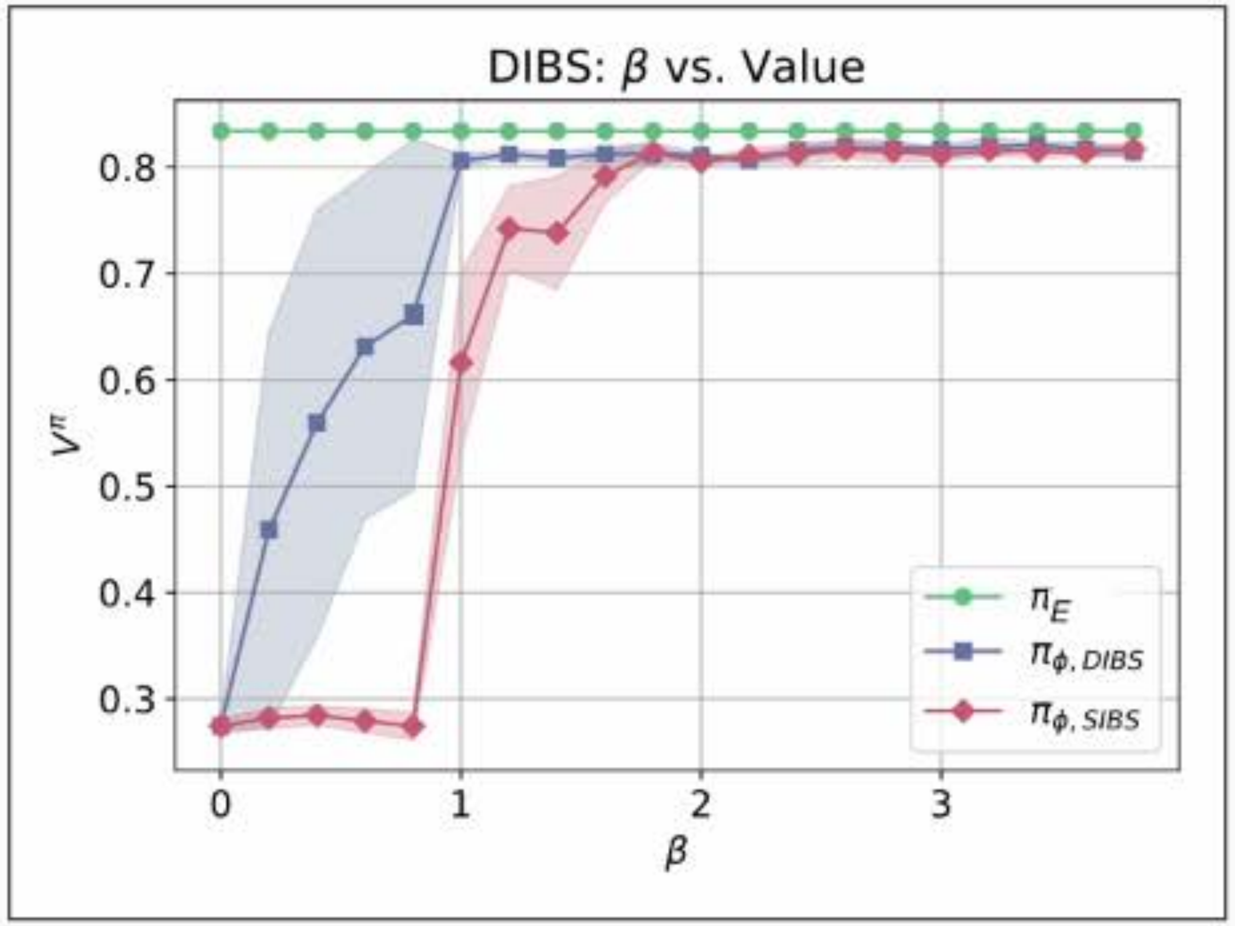
Theorem.

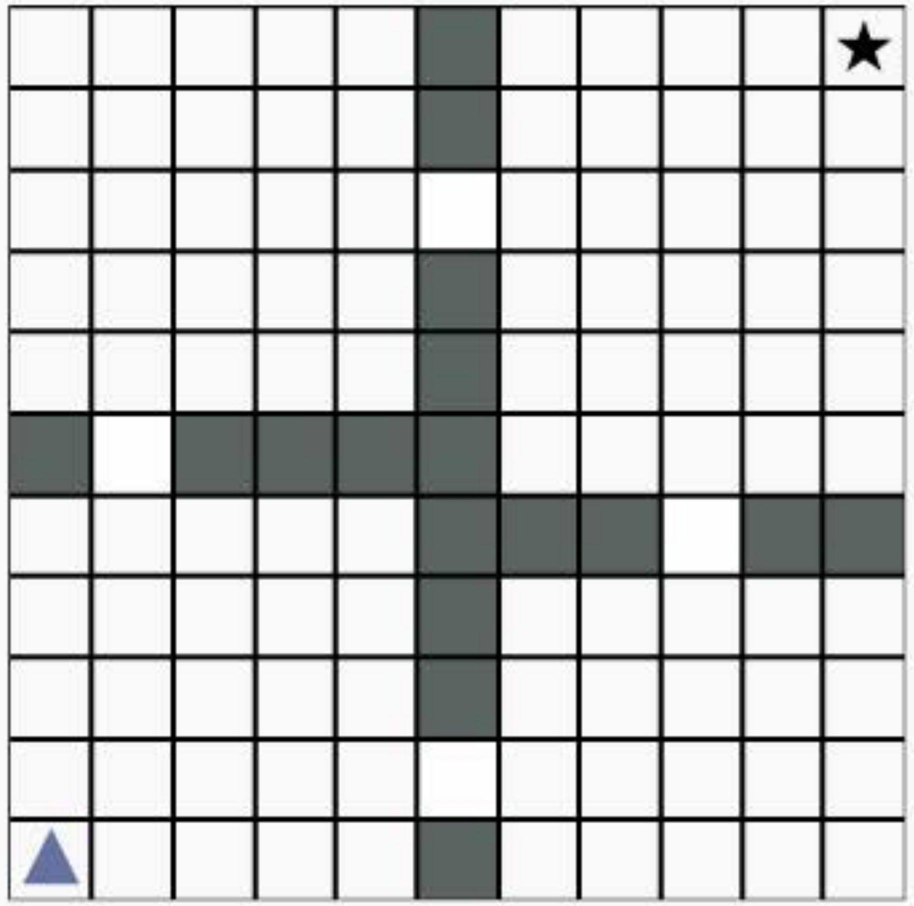
$$\min_{\phi} \left(|\mathcal{S}_\phi| + \beta \mathop{\mathbb{E}}_{\rho_E(s)} \left[V^{\pi_E}(s) - V^{\pi^*_\phi}(s) \right] \right)$$

Our Objective $\leq$ *DIB Objective*

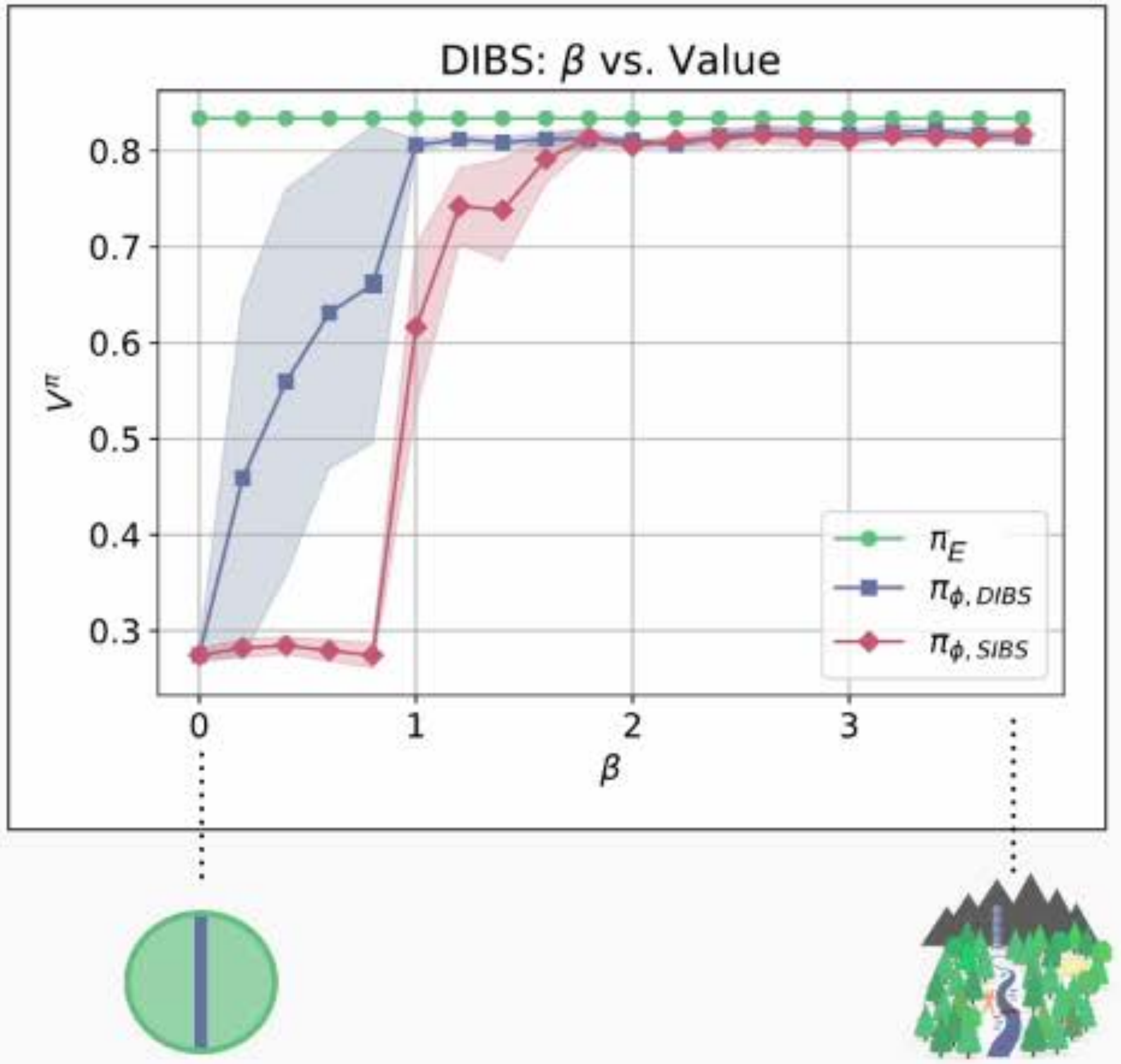
$$\min_{\phi} \left(C_1 H(\rho_\phi) + C_2 \beta \mathop{\mathbb{E}}_{\rho_E(s)} \left[D_{\mathrm{KL}} \left(\pi_E(s) \,||\, \pi^*_\phi(s) \right) \right] \right)$$

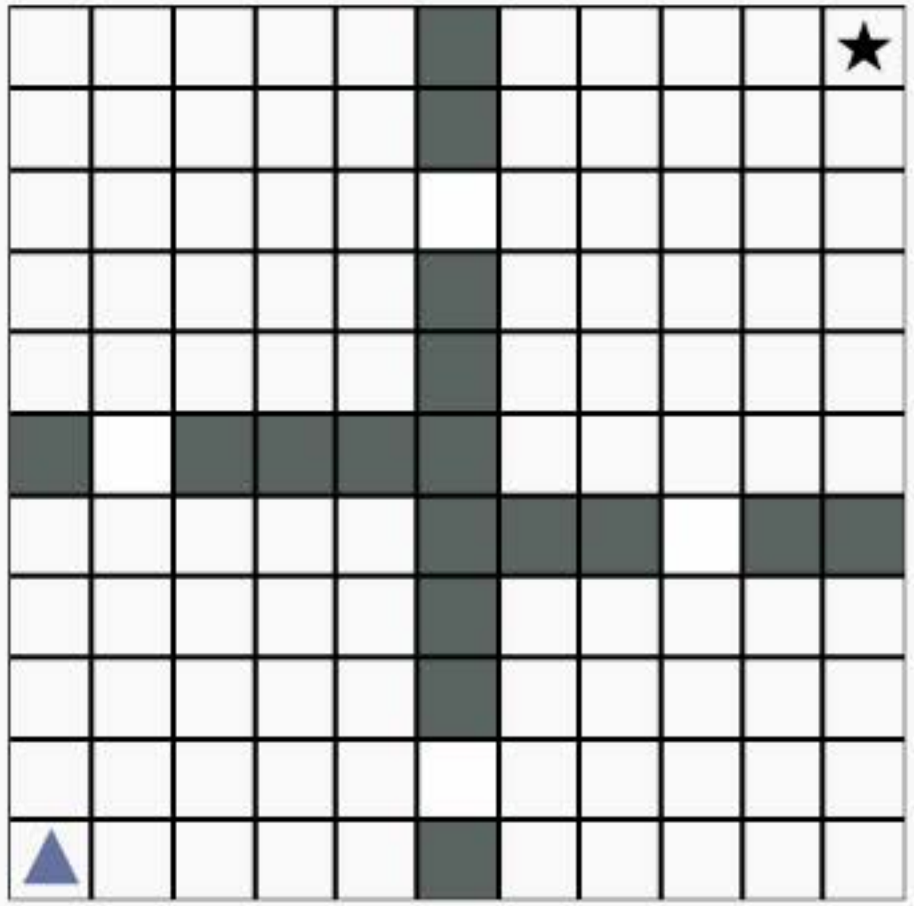
State Abstraction as Compression



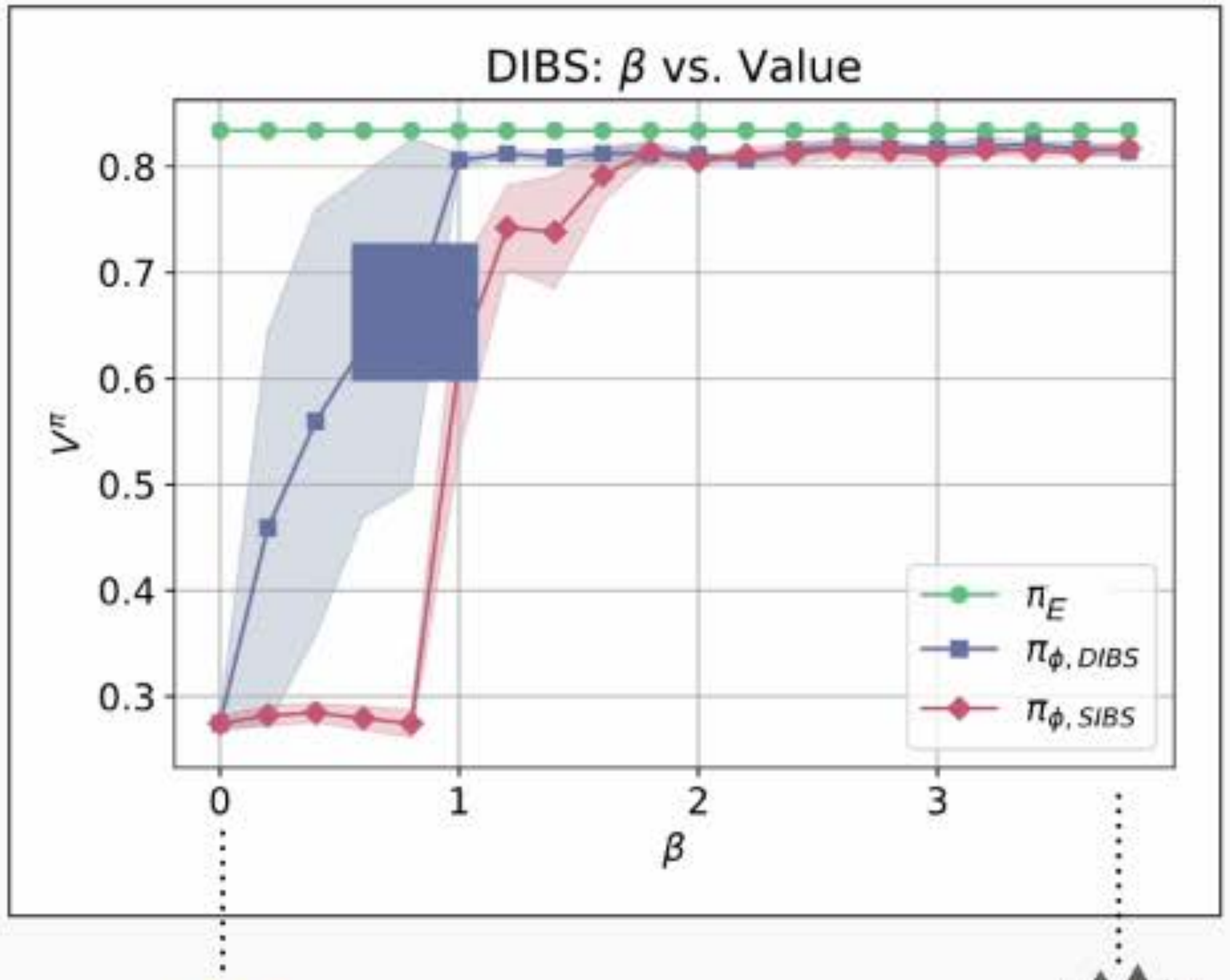


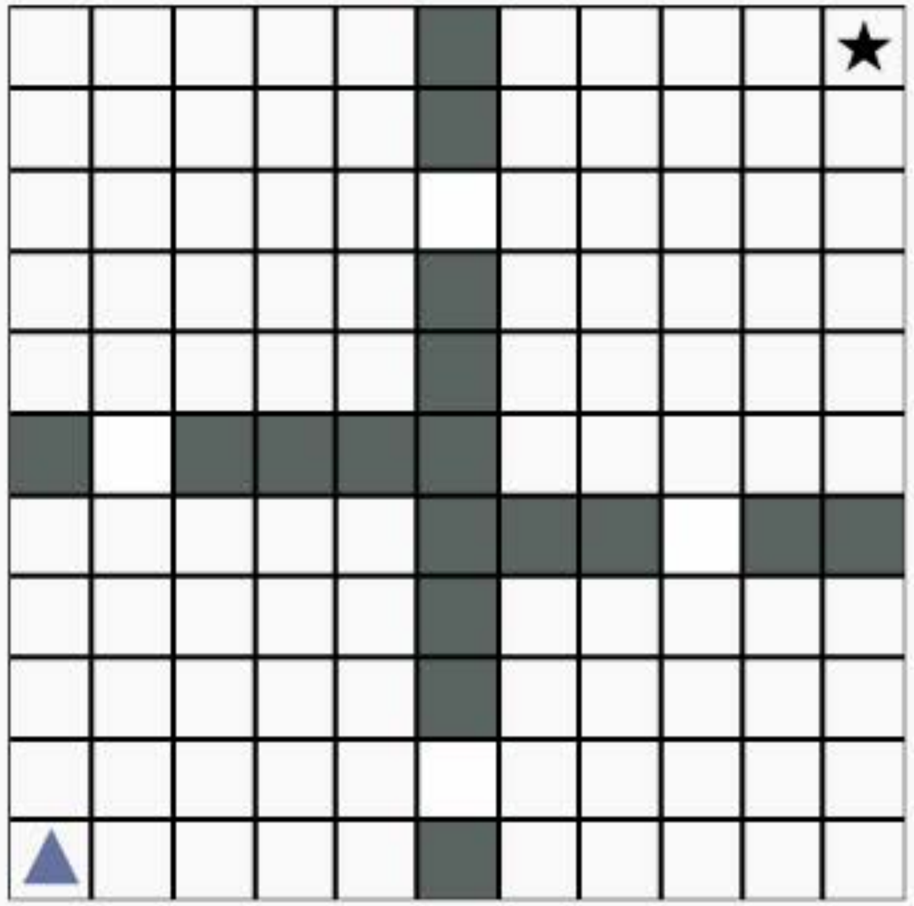
State Abstraction as Compression



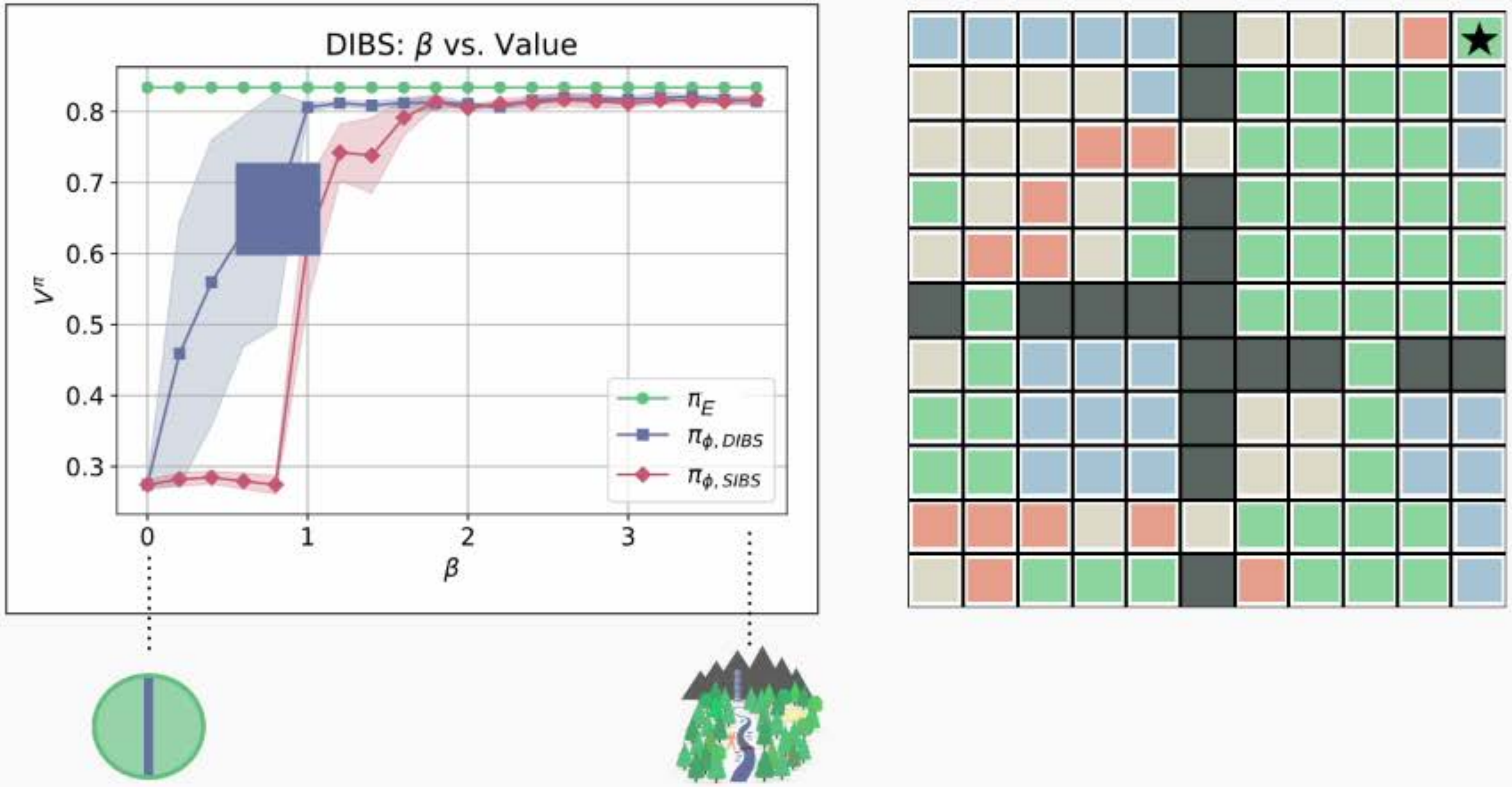


State Abstraction as Compression

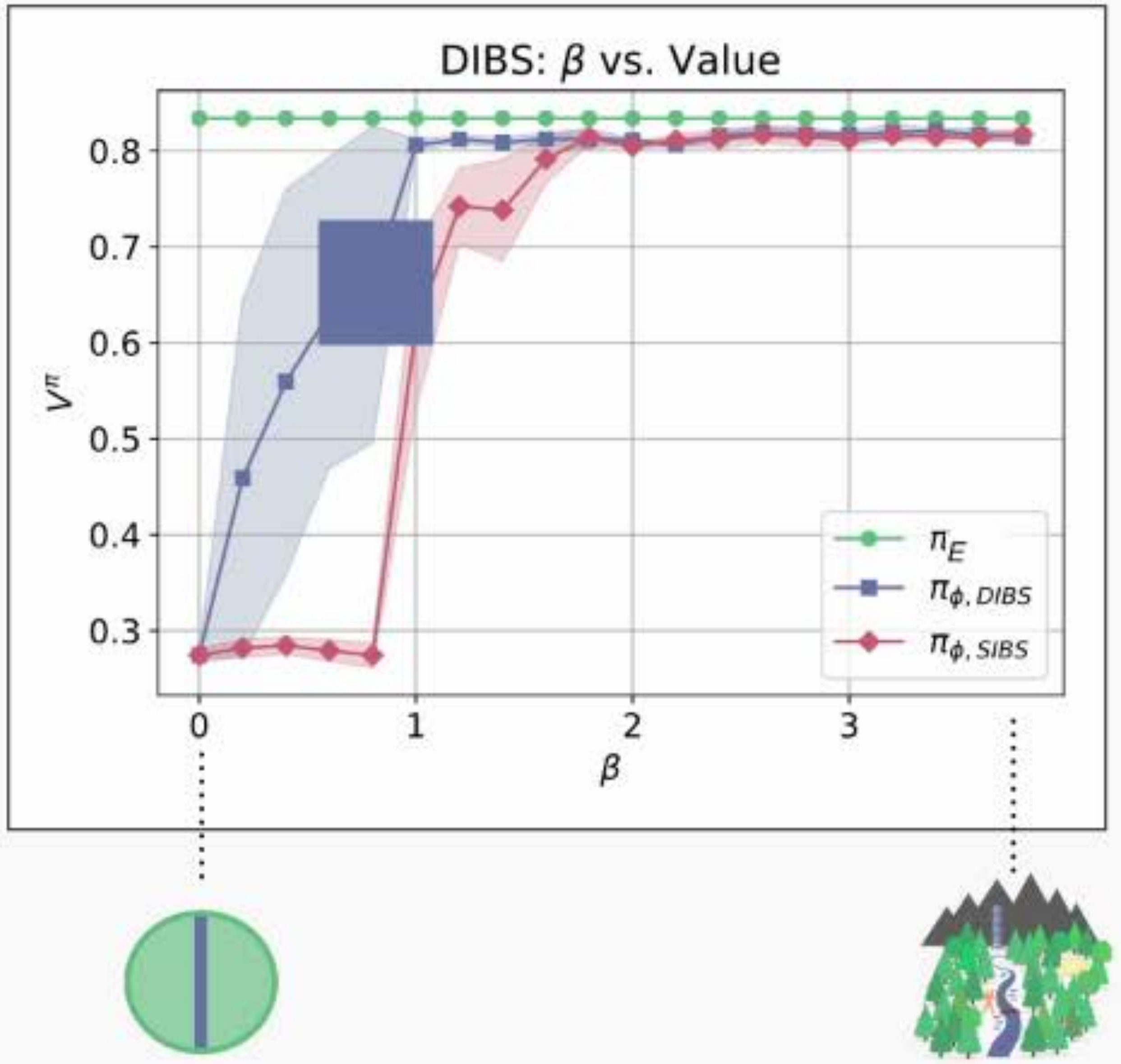


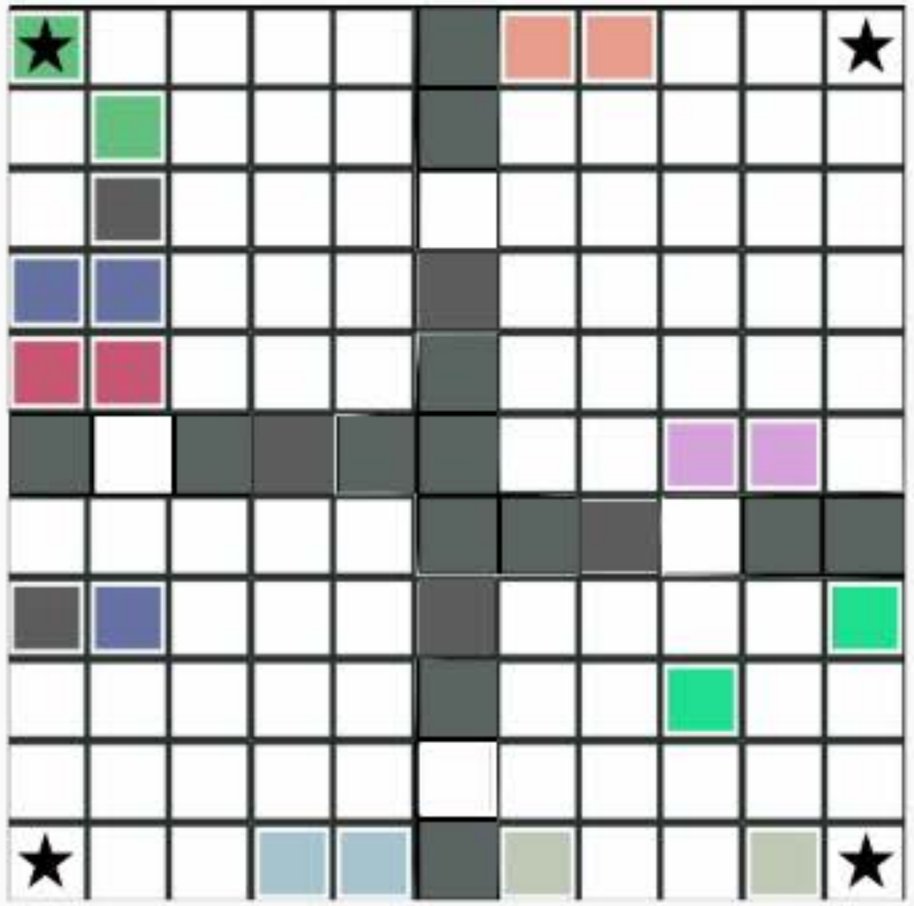


State Abstraction as Compression



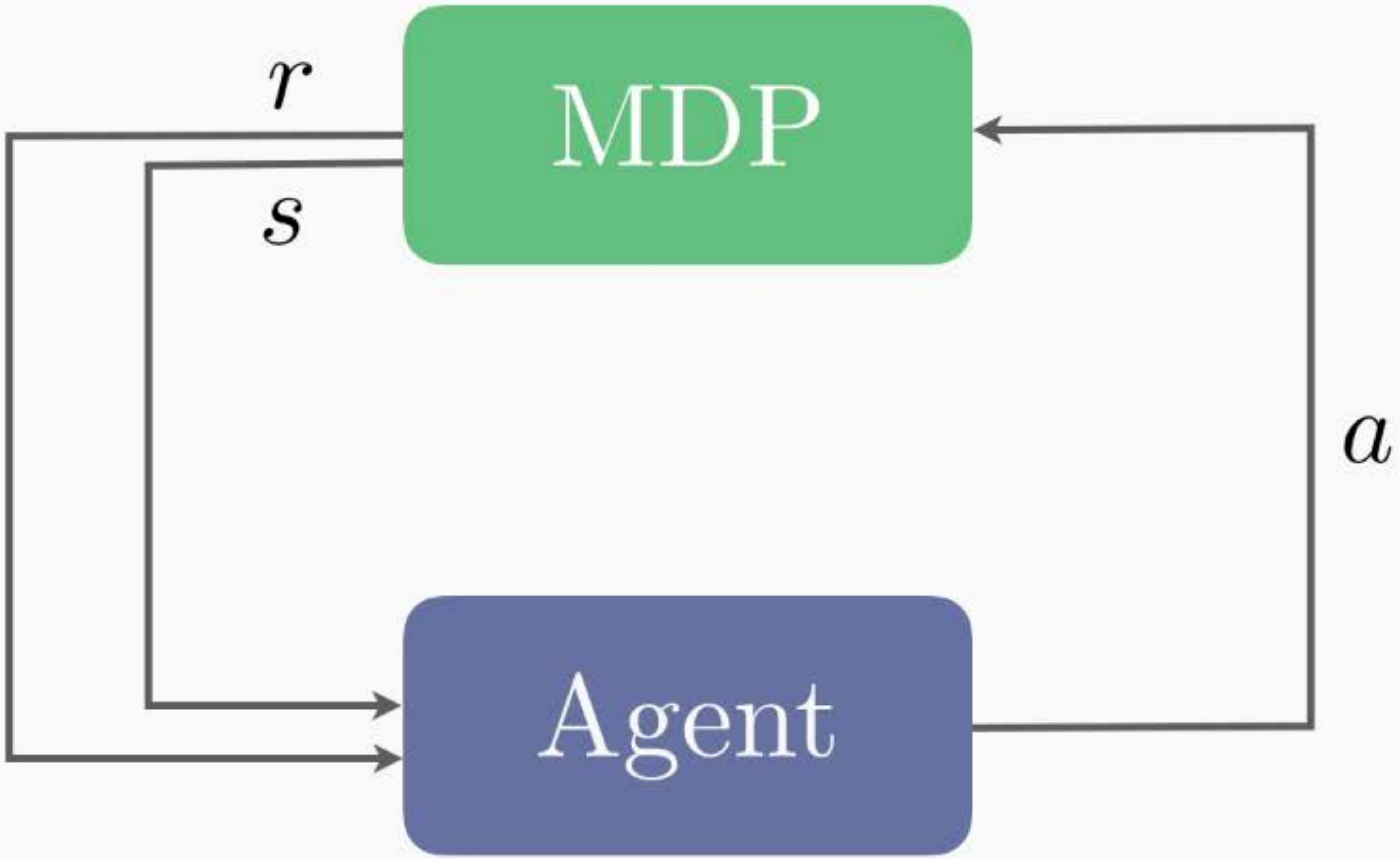
State Abstraction as Compression



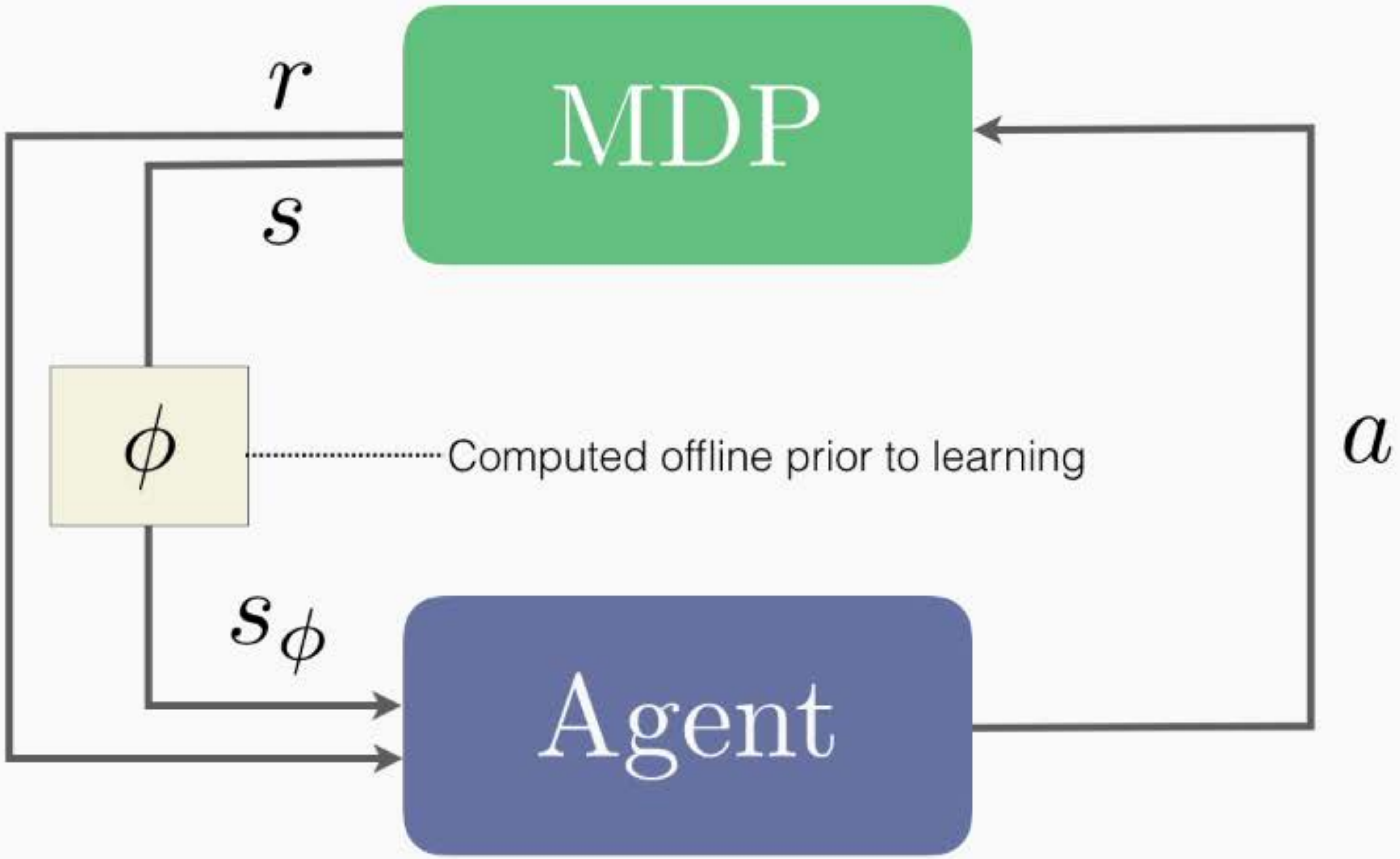


Multitask Abstraction

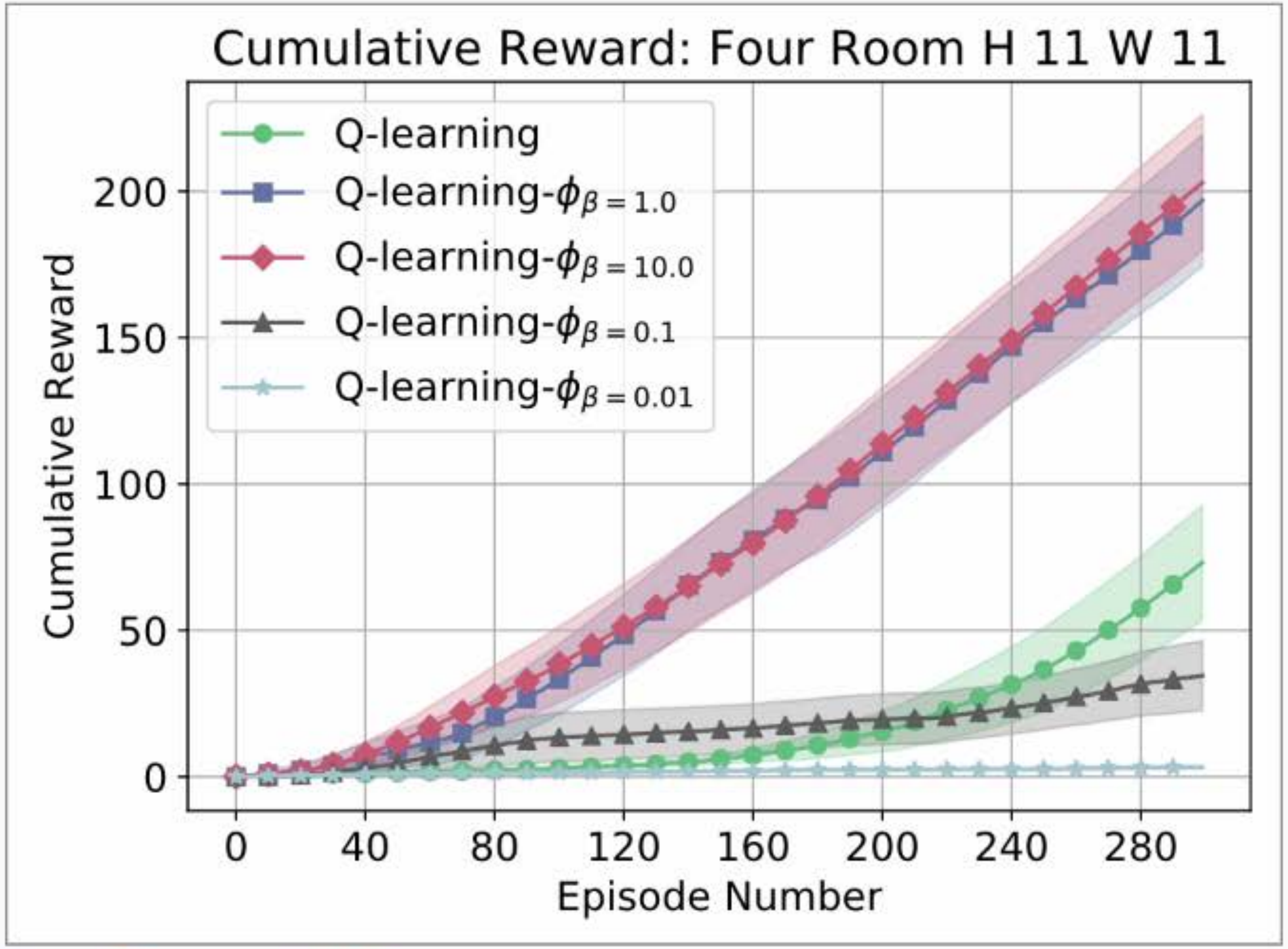
Learning Experiments



Learning Experiments

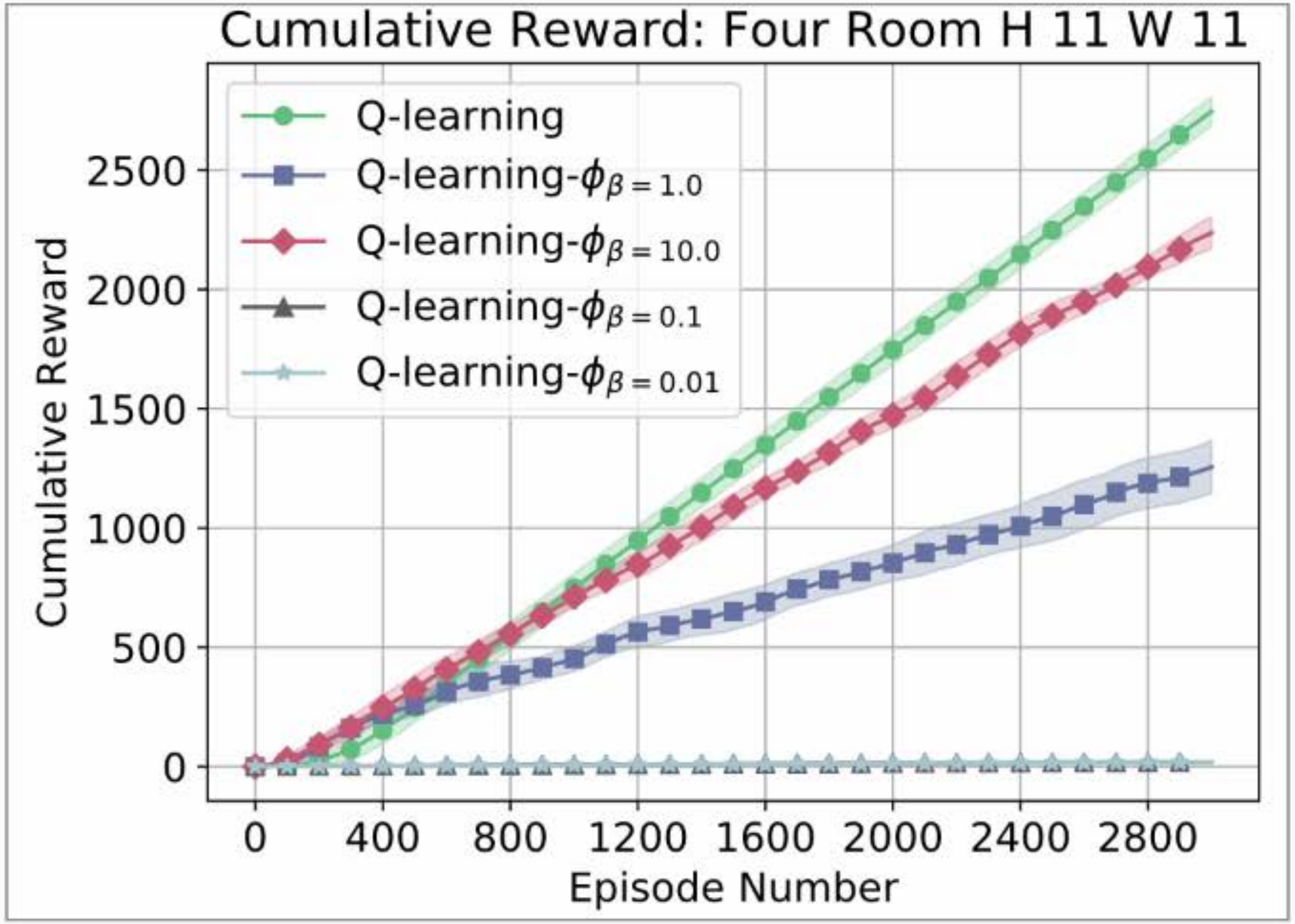


Experiments: Four Rooms

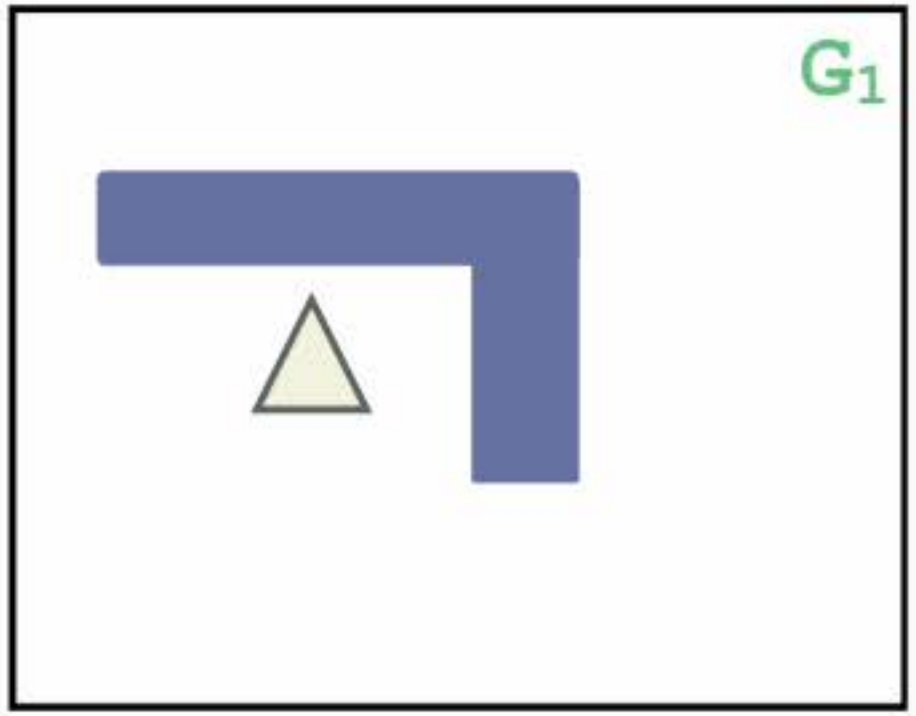


[Watkins, Dayan '92]

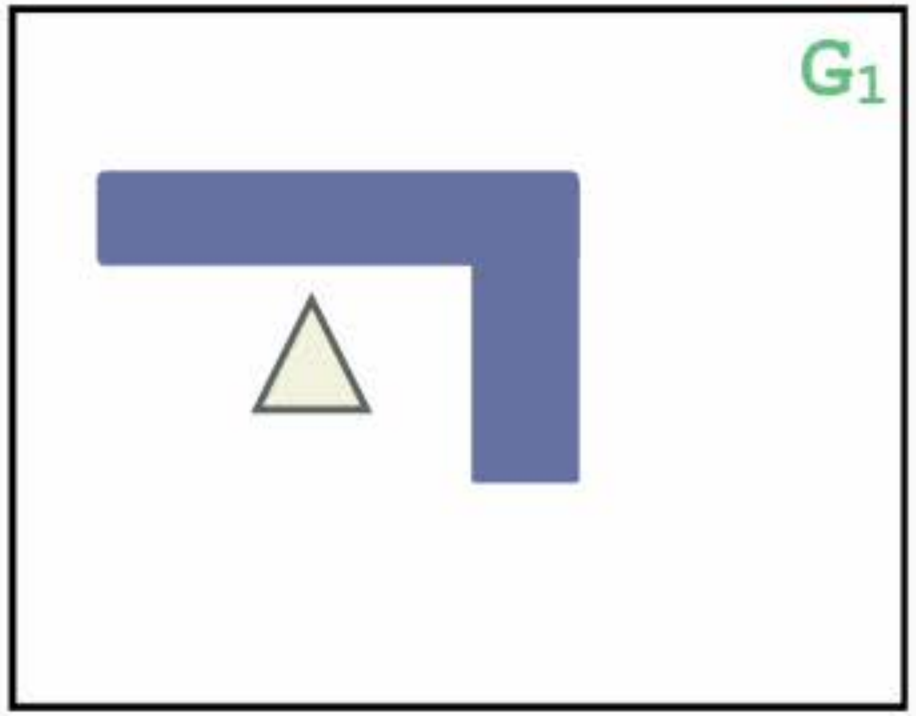
Experiments: Four Rooms

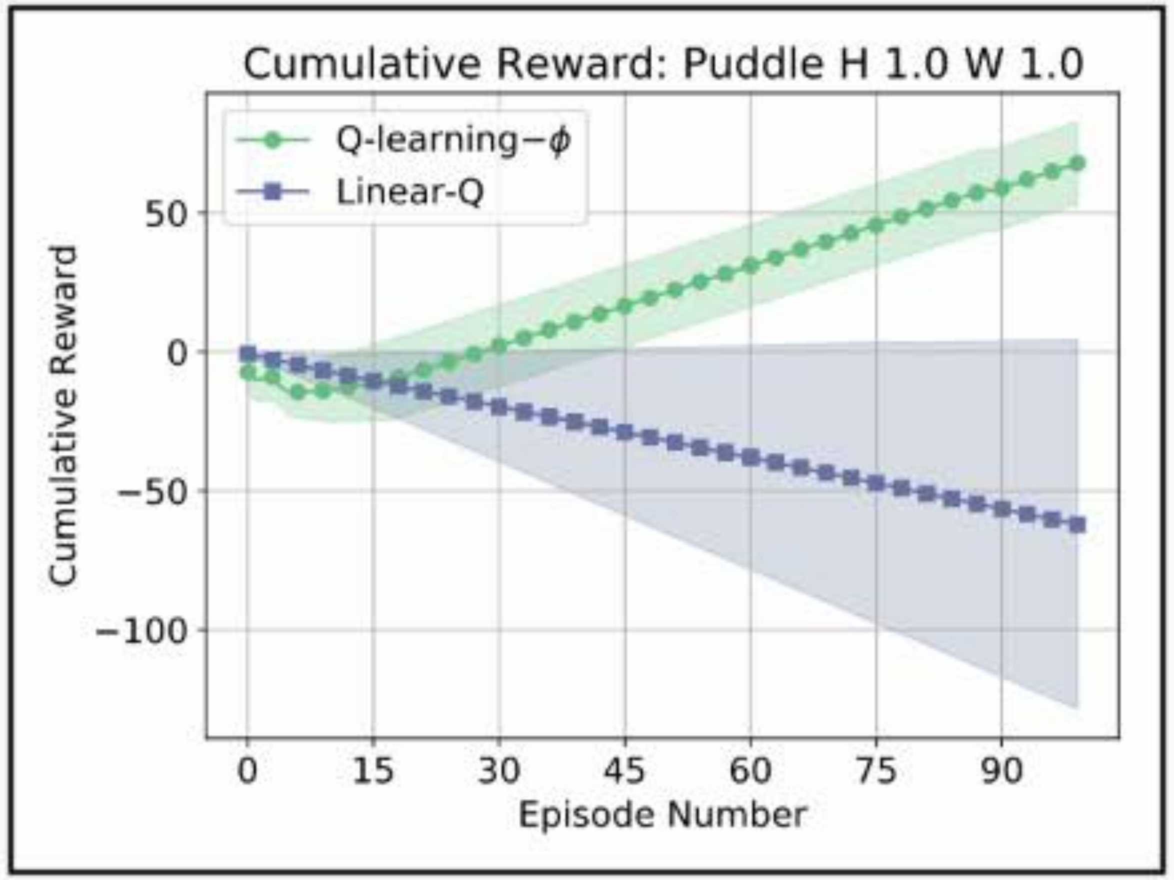


Continuous State



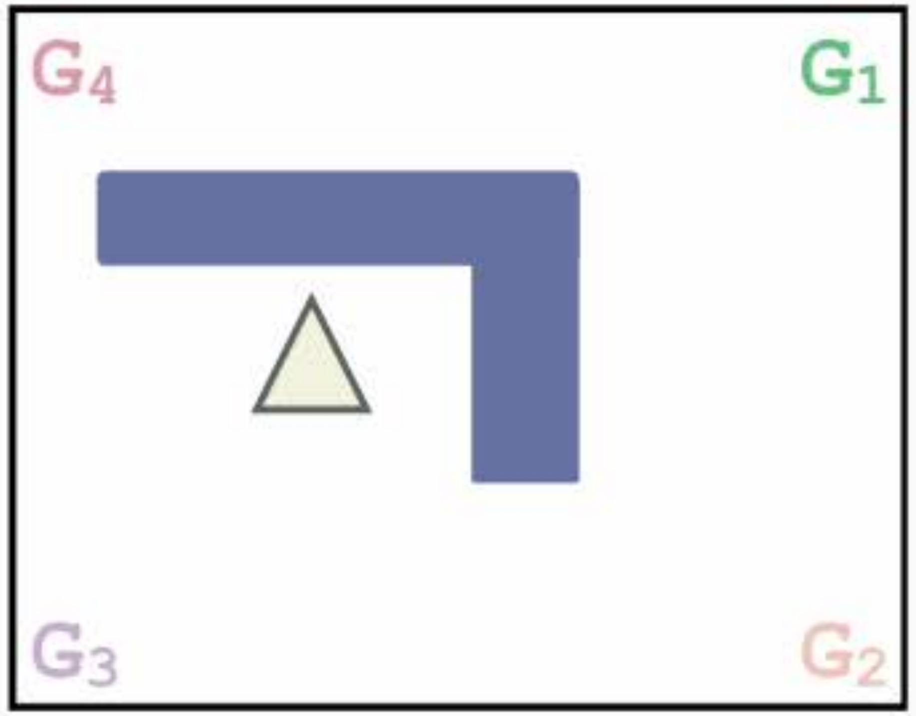
Continuous State



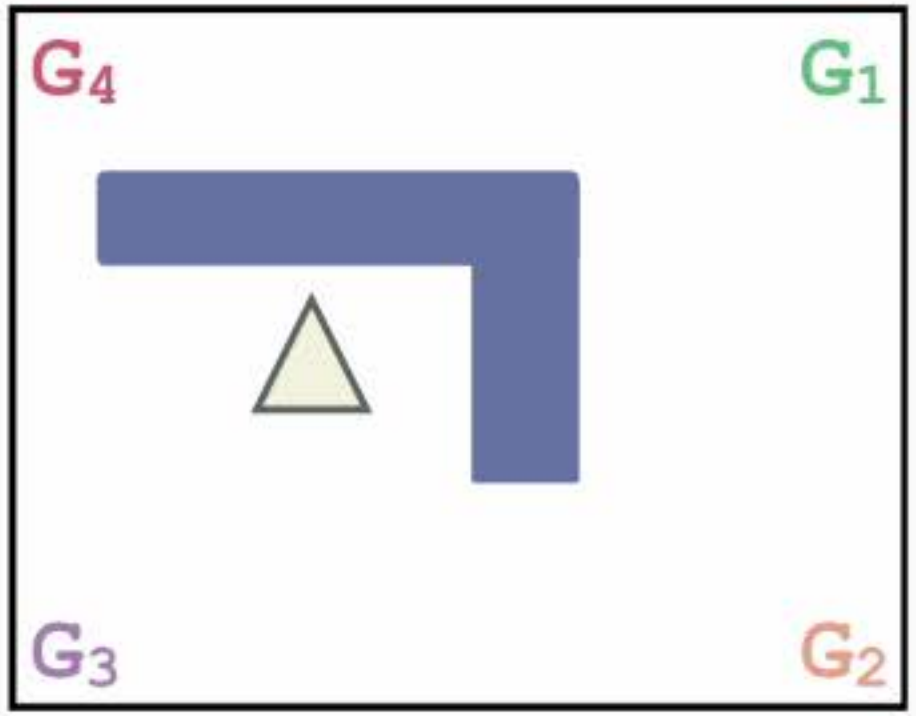


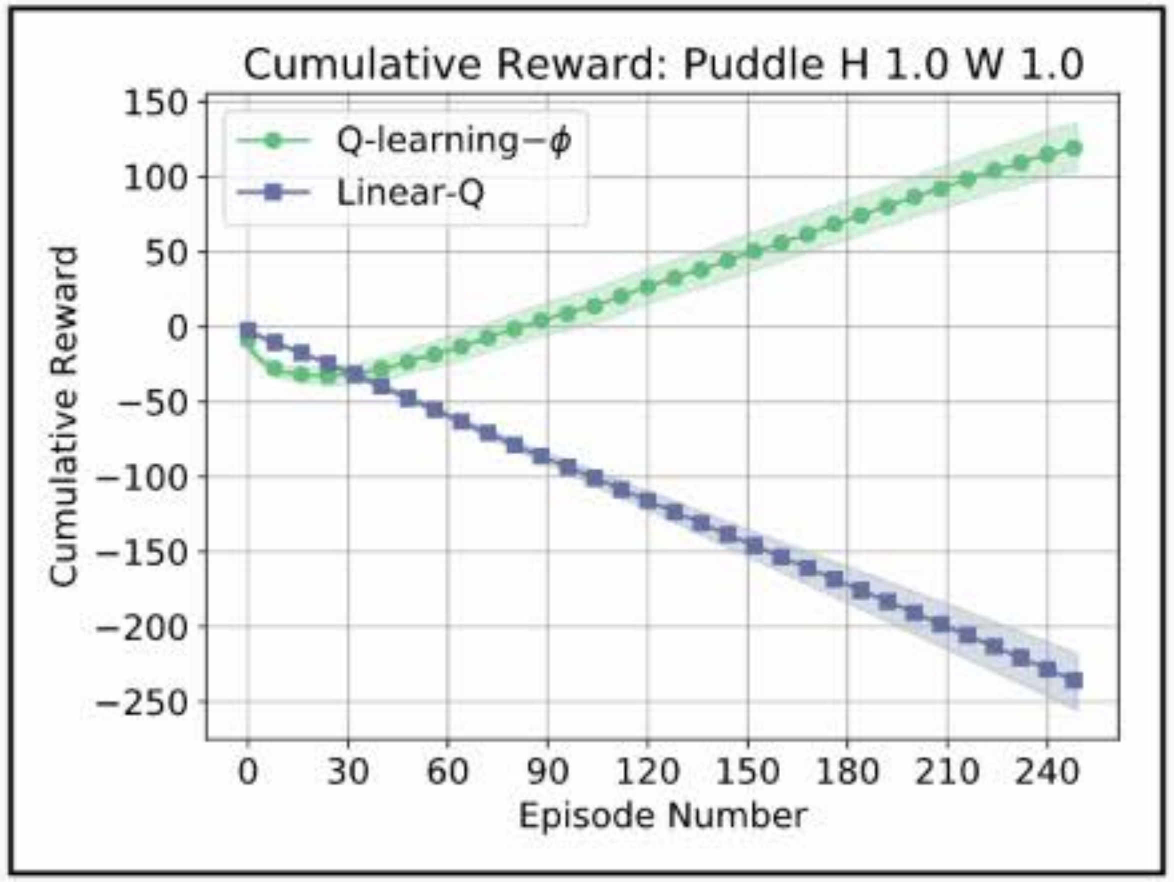
Single Task

Continuous State



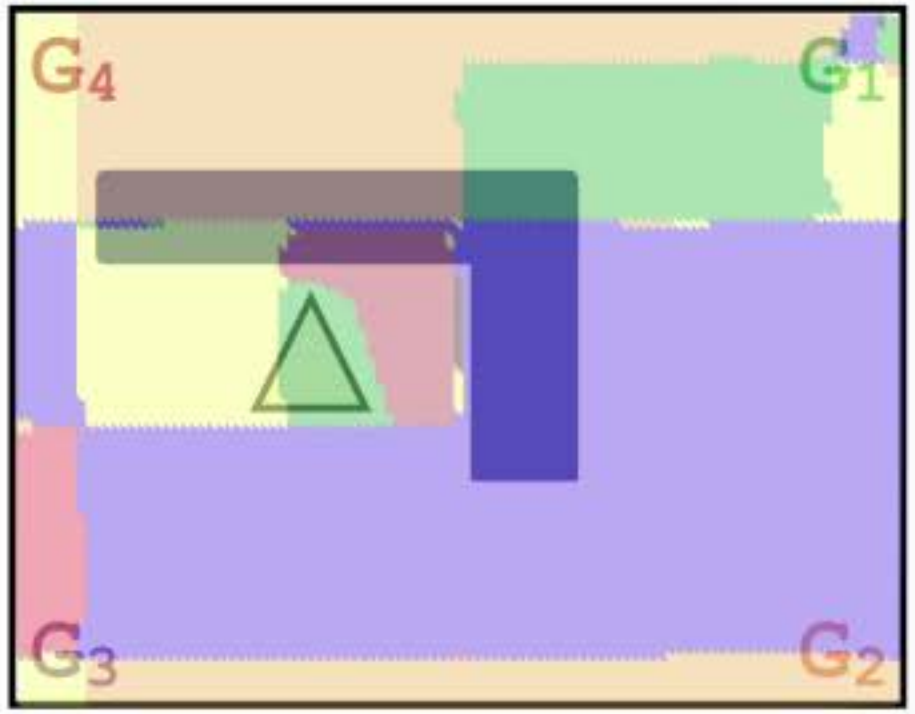
Continuous State

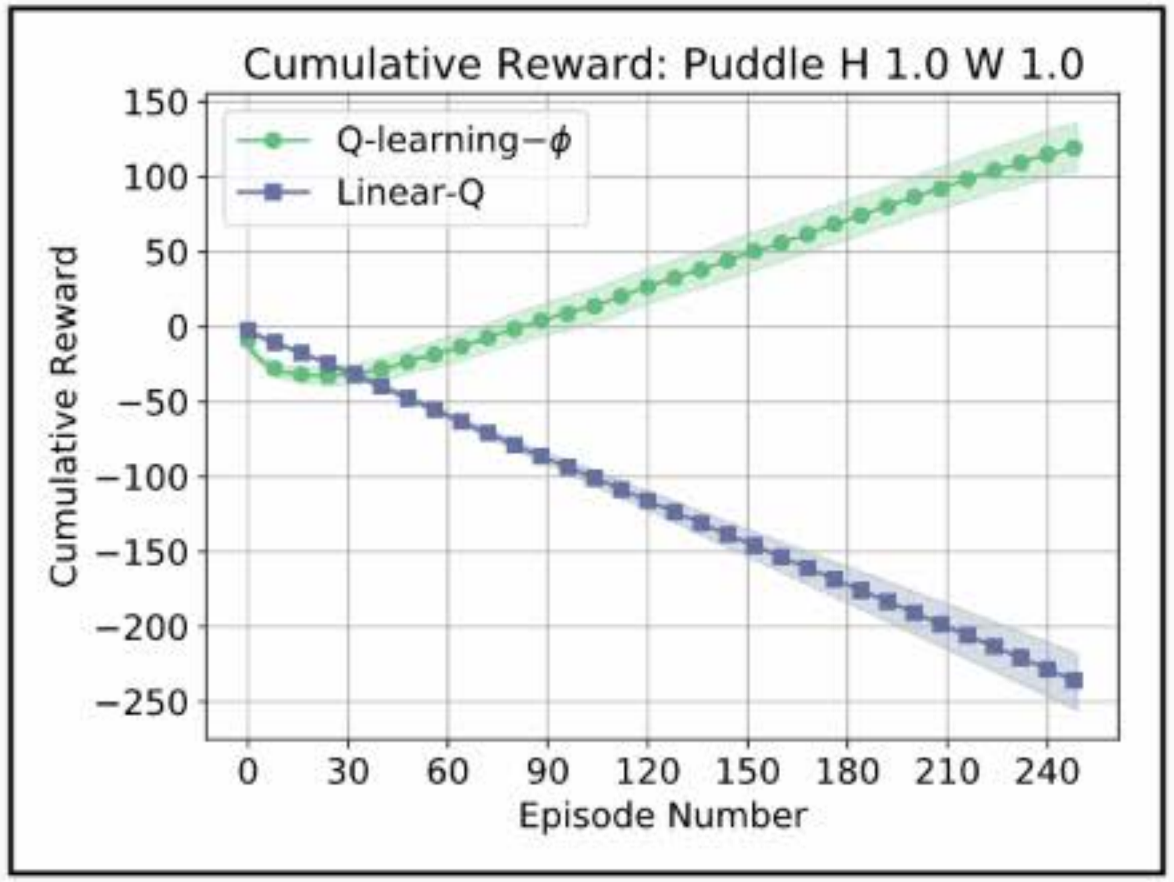




Transfer

Continuous State





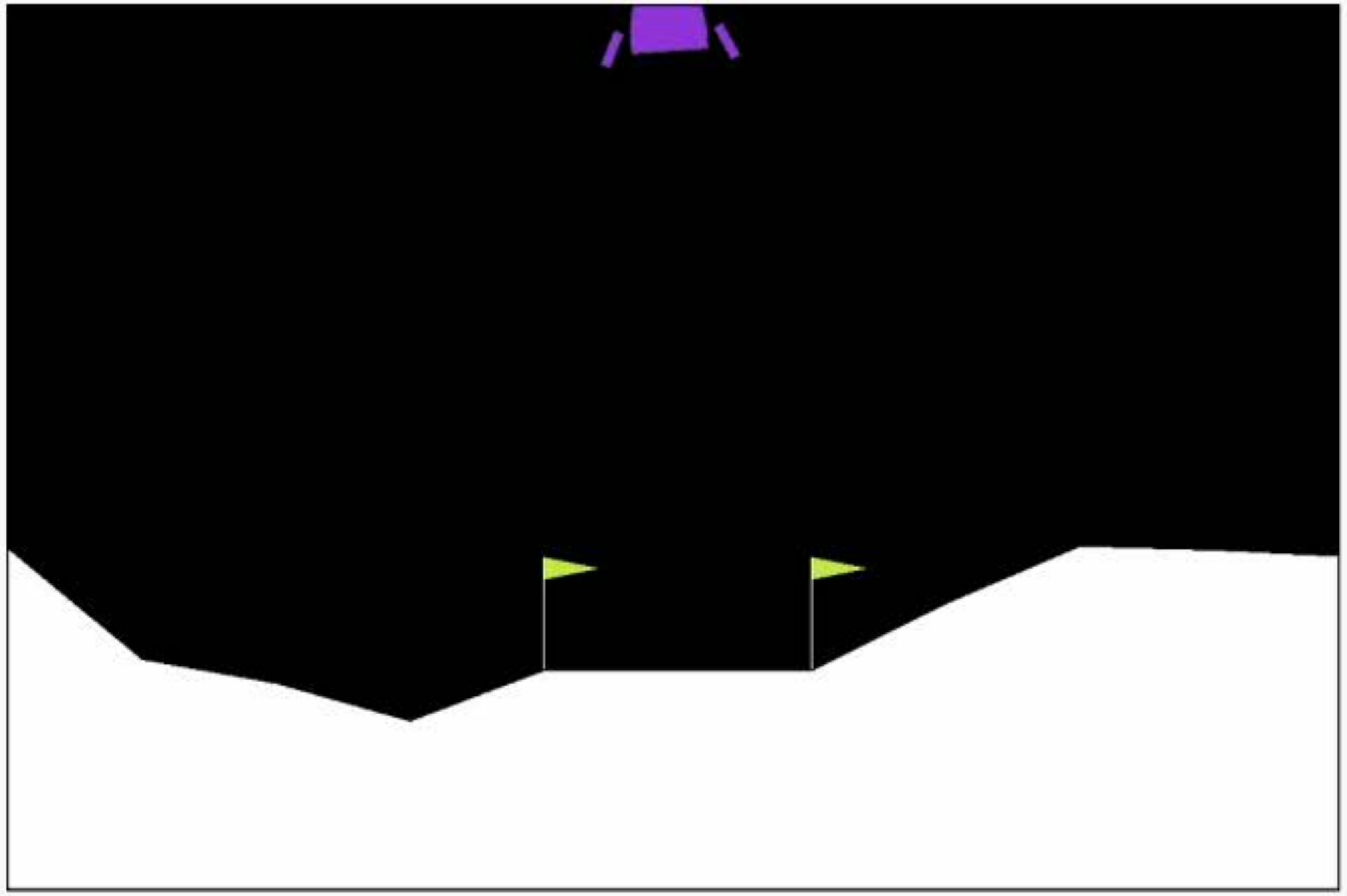
Transfer

Continuous State

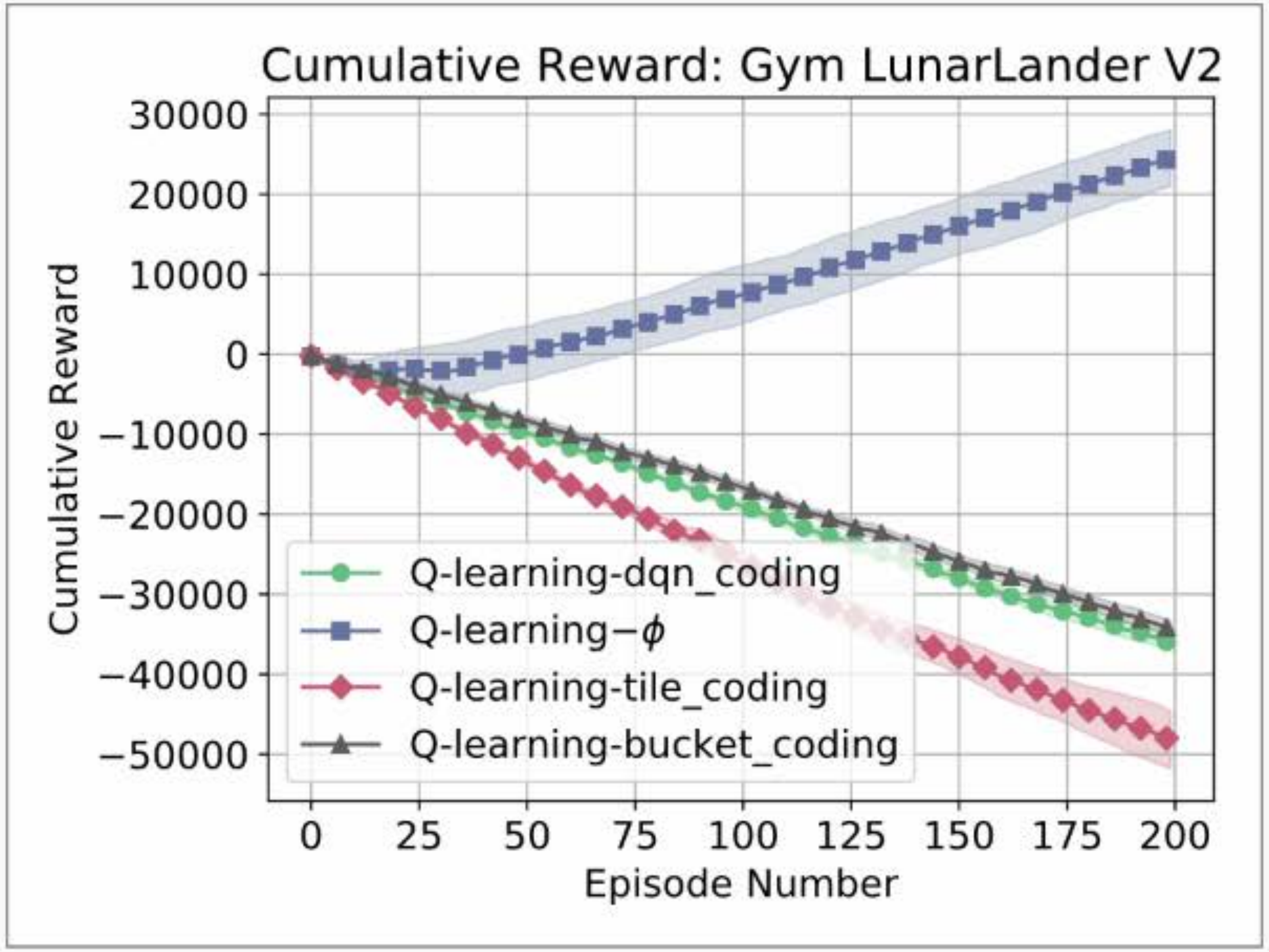
Theorem. *With probability at least* $1-\delta$*, for any* $\delta \in (0,1)$*:*

$$\mathbb{E}_s \left|\left|\left(\pi^*(\cdot \mid s) - \pi^*_\phi(\cdot \mid \phi(s))\right)\right|\right|_1 \leq \frac{\Delta}{2} + 2\sqrt{2}\mathrm{Rad}\left(\Phi\right) + \sqrt{\frac{2\ln\frac{1}{\delta}}{n}}$$

Lunar Lander



Experiments: Lunar Lander



Kavosh
Asadi

Results Summary

Question: *Can we find state abstractions that minimize* $|\mathcal{S}_\phi|$ *while still representing good policies?*

Results Summary

Question: *Can we find state abstractions that minimize* $|\mathcal{S}_\phi|$ *while still representing good policies?*

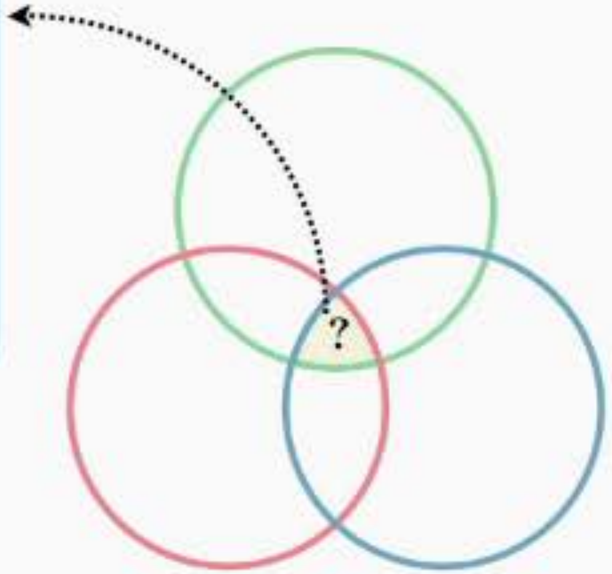
Answer: *Yes! But, perhaps state space size isn't the full story.*

Results Summary

Easy To Construct

1 **State Abstraction**

*[**A**AAJLW AAAI '19]*



Supports Efficient Reinforcement Learning

Preserves Solution Quality

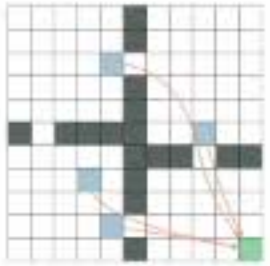
Results Summary

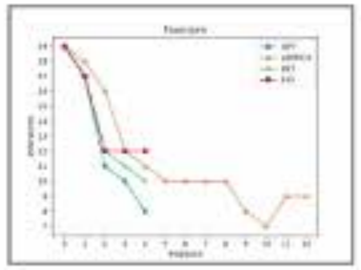
Easy To Construct

1 **State Abstraction**

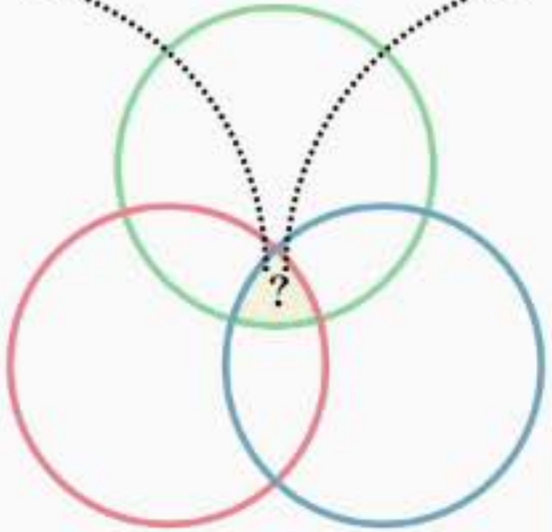
*[**A**AAJLW AAAI '19]*

2 **Action Abstraction**





*[J**A**HLK ICML '19]*



Supports Efficient Reinforcement Learning

Yuu Jinnai

Preserves Solution Quality

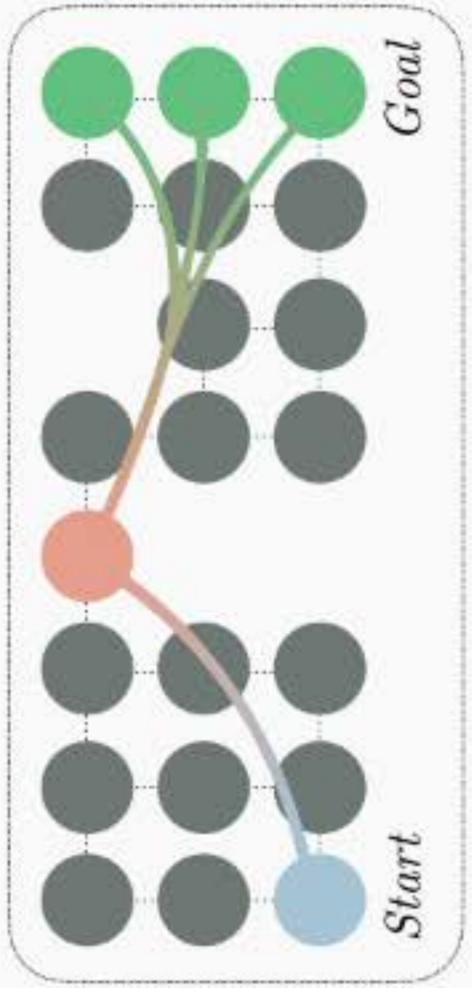
Action Abstraction

[Sutton, Precup, Singh 1999]

Definition (Option): A start condition, end condition, and a policy.

Action Abstraction

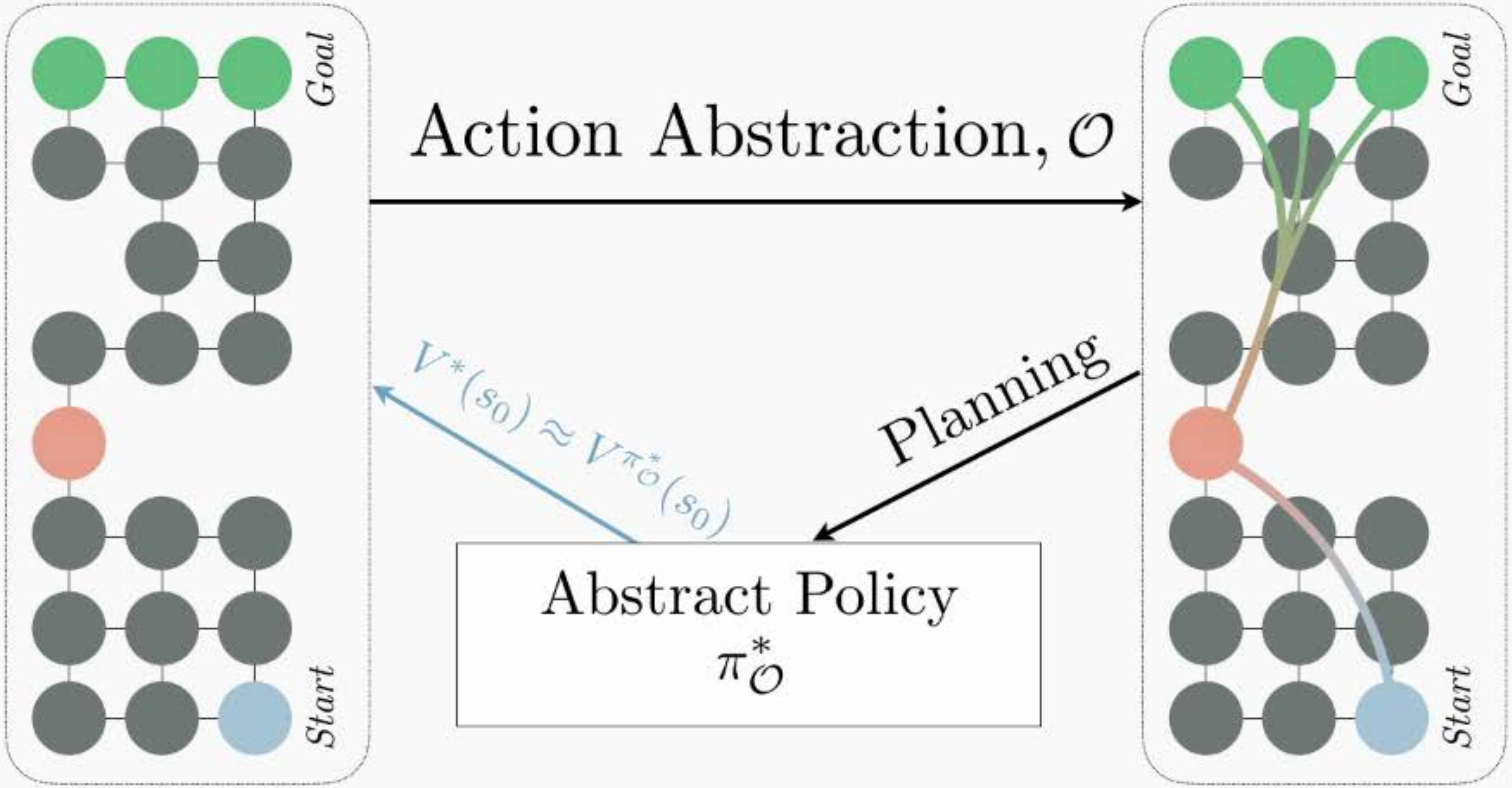
Example:



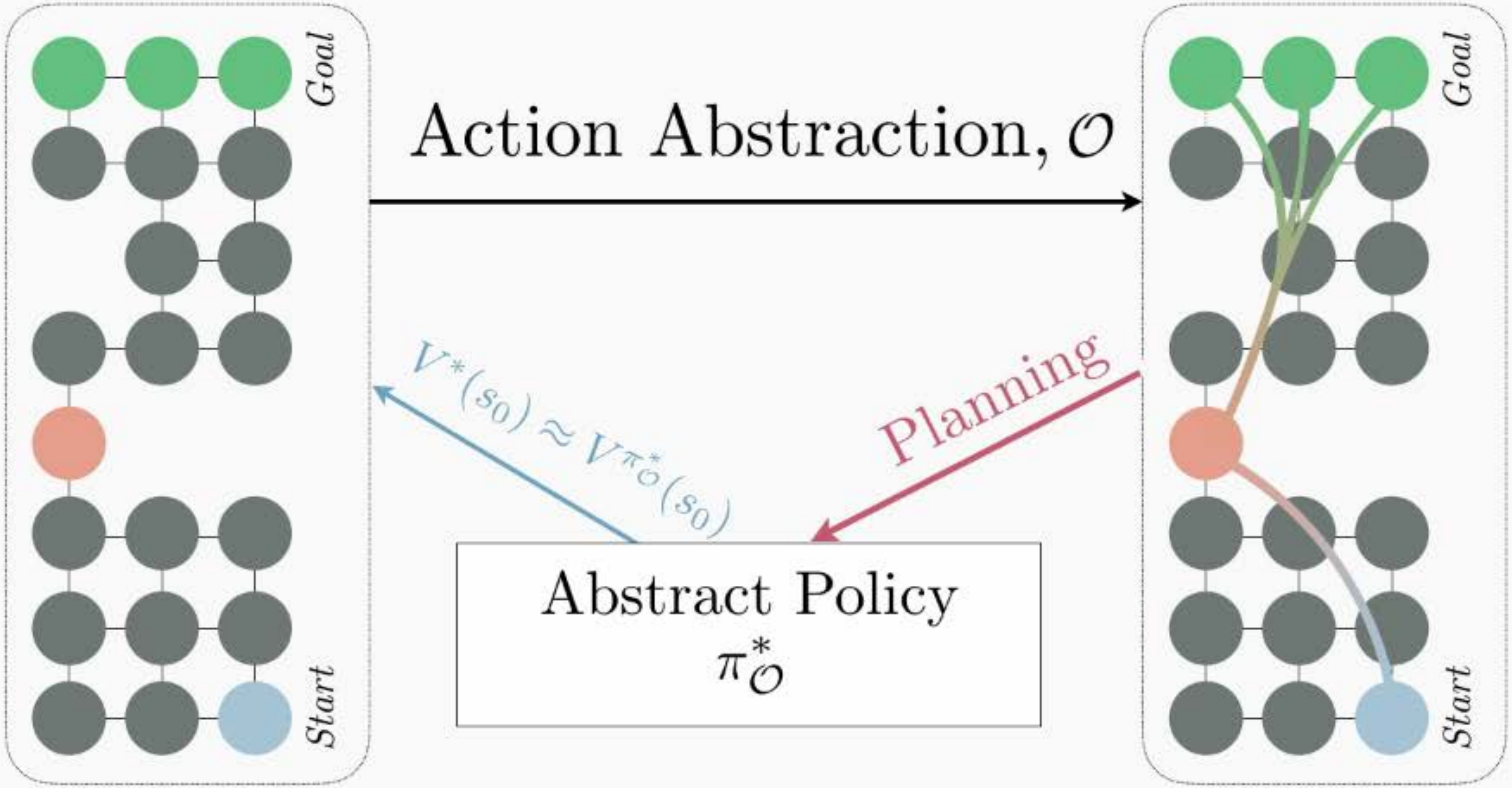
[Sutton, Precup, Singh 1999]

Definition (Option): A start condition, end condition, and a policy.

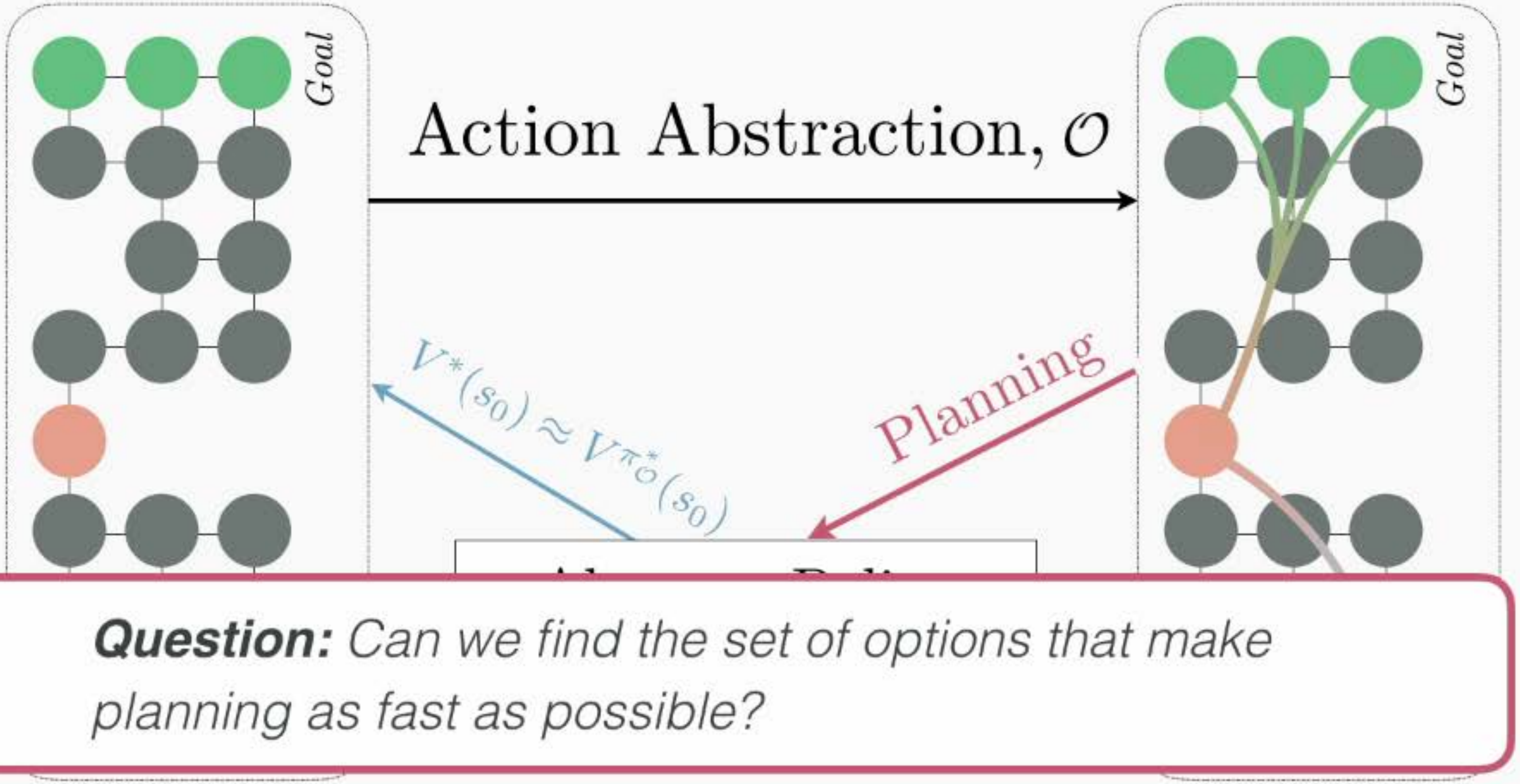
Finding Options for Planning



Finding Options for Planning



Finding Options for Planning



Question: *Can we find the set of options that make planning as fast as possible?*

Finding Options for Planning

Question: *Can we find the set of options that make planning as fast as possible?*

Finding Options for Planning

Definition (Value-Planning Problem): **Given** *an MDP M and* $\varepsilon \in \mathbb{R}_{>0}$, **return** *a value function, V such that* $|V(s) - V^*(s)| < \varepsilon$ *for all* $s \in \mathcal{S}$.

Question: *Can we find the set of options that make planning as fast as possible?*

Finding Options for Planning

Definition (Value-Planning Problem): **Given** *an MDP* M *and* $\varepsilon \in \mathbb{R}_{>0}$, **return** *a value function,* V *such that* $|V(s) - V^*(s)| < \varepsilon$ *for all* $s \in \mathcal{S}$.

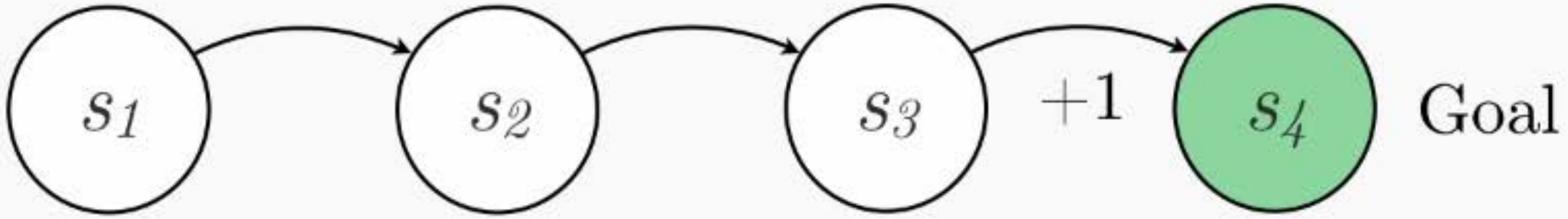
Iterations needed for value iteration to converge

Question: *Can we find the set of options that make planning as fast as possible?*

Thanks to Yuu Jinnai for the example

Value Iteration

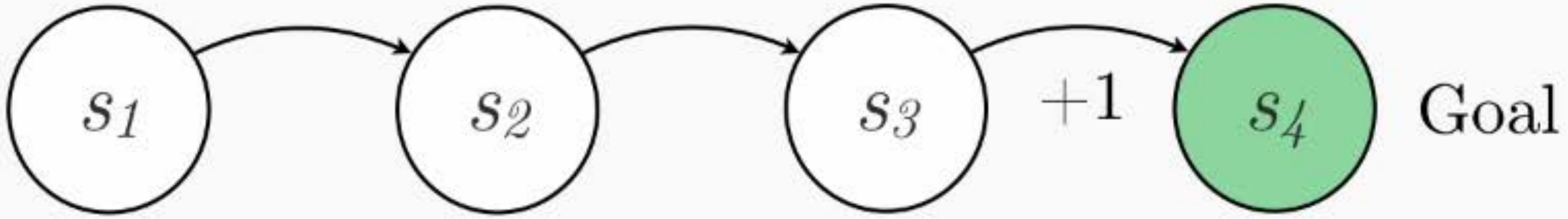
[Bellman 1957]



Thanks to Yuu Jinnai for the example

Value Iteration

[Bellman 1957]

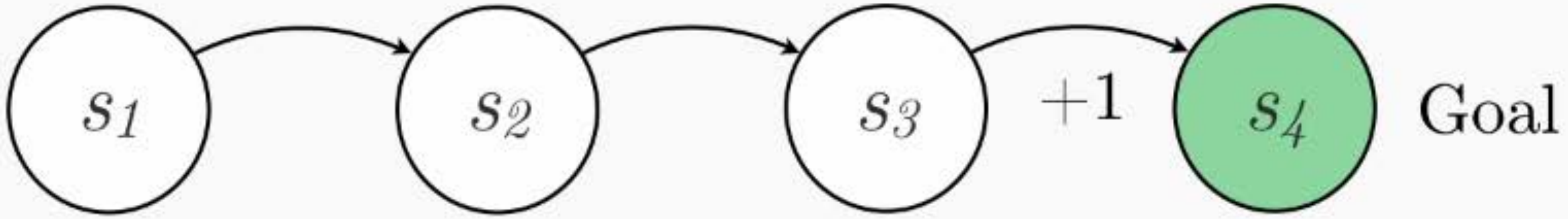


$V_0(s)$	0	0	0	0

Thanks to Yuu Jinnai for the example

Value Iteration

[Bellman 1957]

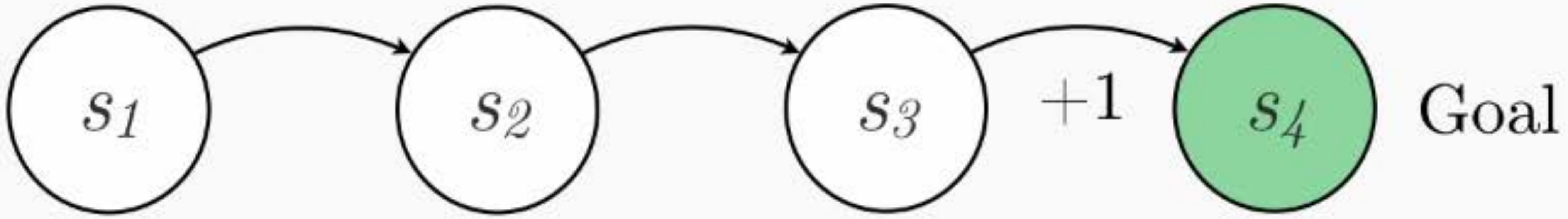


$V_0(s)$	0	0	0	0
$V_1(s)$	0	0	1	0

Thanks to Yuu Jinnai for the example

Value Iteration

[Bellman 1957]

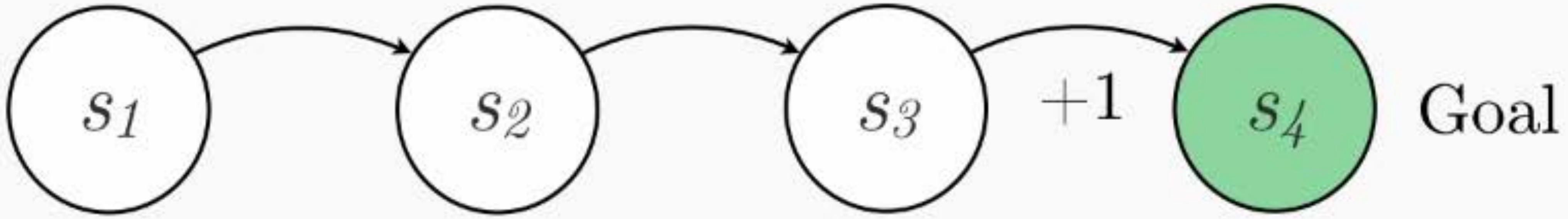


$V_0(s)$	0	0	0	0
$V_1(s)$	0	0	1	0
$V_2(s)$	0	γ	1	0

Thanks to Yuu Jinnai for the example

Value Iteration

[Bellman 1957]

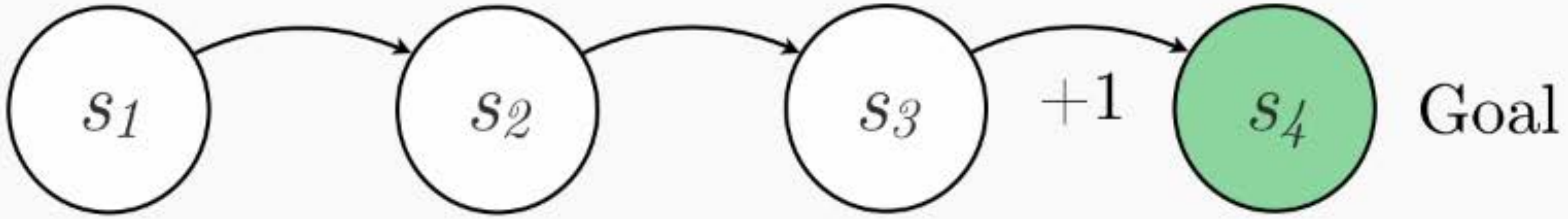


$V_0(s)$	0	0	0	0
$V_1(s)$	0	0	1	0
$V_2(s)$	0	γ	1	0
$V_3(s)$	γ^2	γ	1	0

Thanks to Yuu Jinnai for the example

Value Iteration

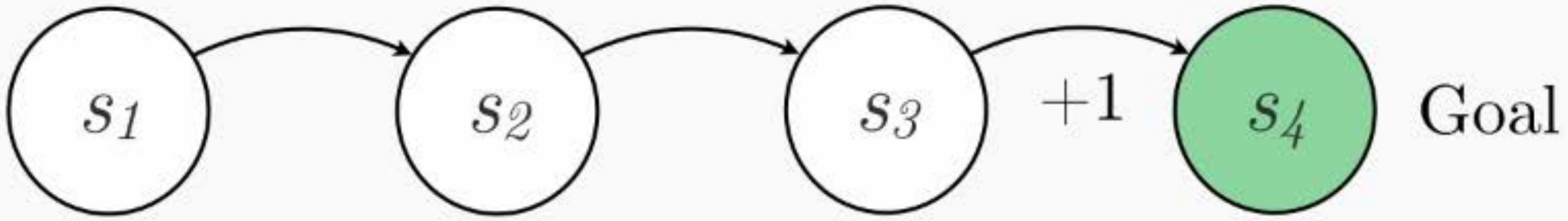
[Bellman 1957]



$V_0(s)$	0	0	0	0
$V_1(s)$	0	0	1	0
$V_2(s)$	0	γ	1	0
$V_3(s)$	γ^2	γ	1	0

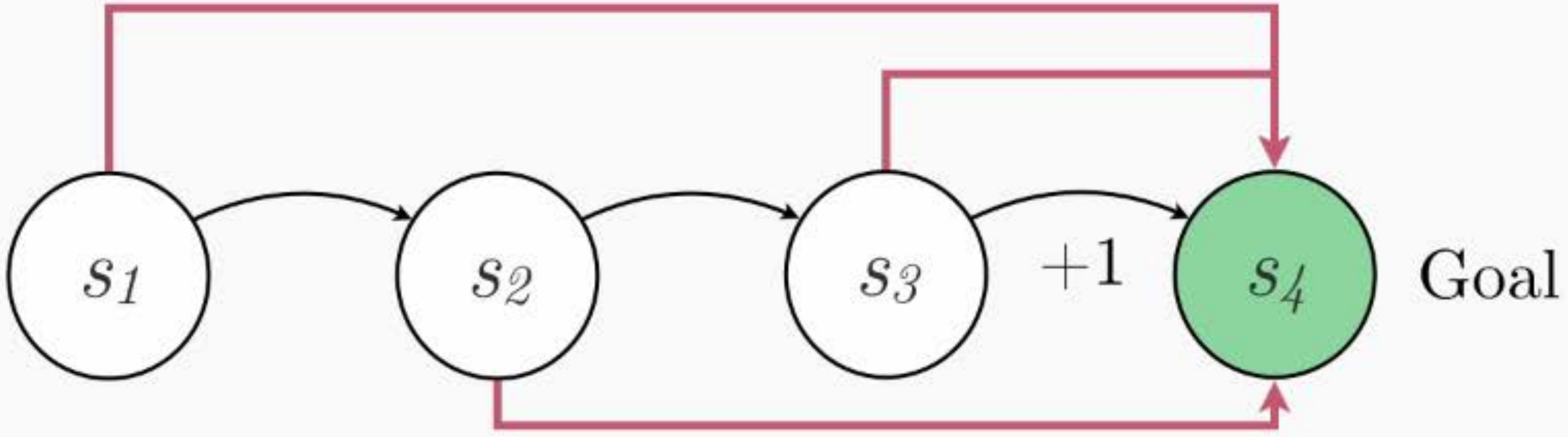
Thanks to Yuu Jinnai for the example

Value Iteration with Options



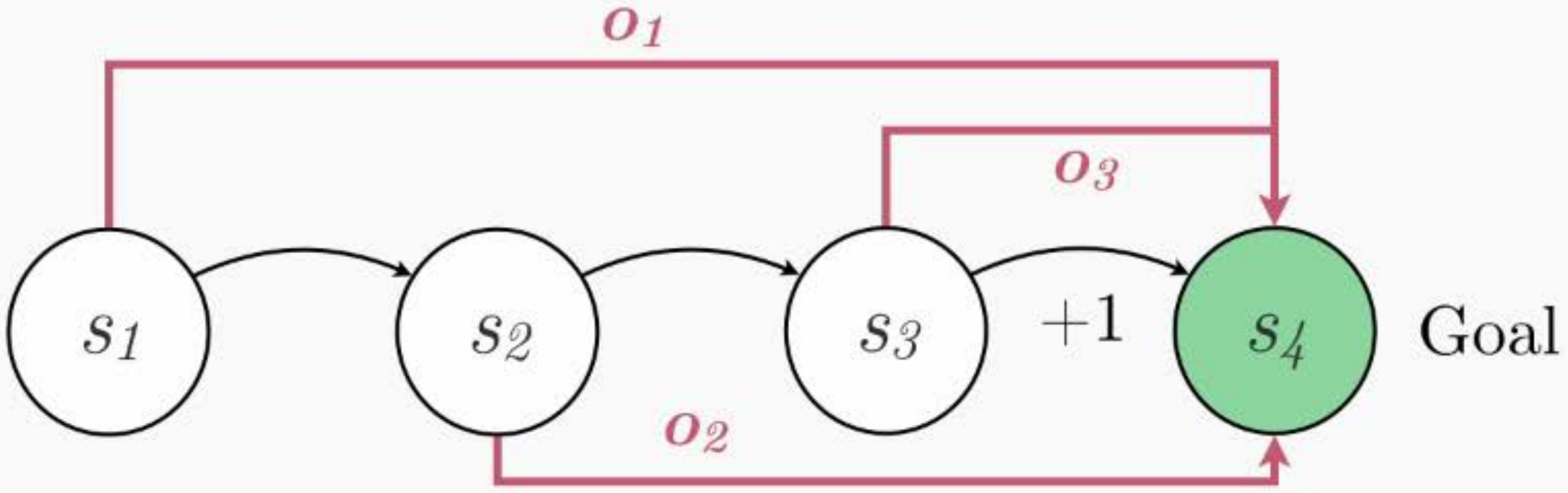
Thanks to Yuu Jinnai for the example

Value Iteration with Options



Thanks to Yuu Jinnai for the example

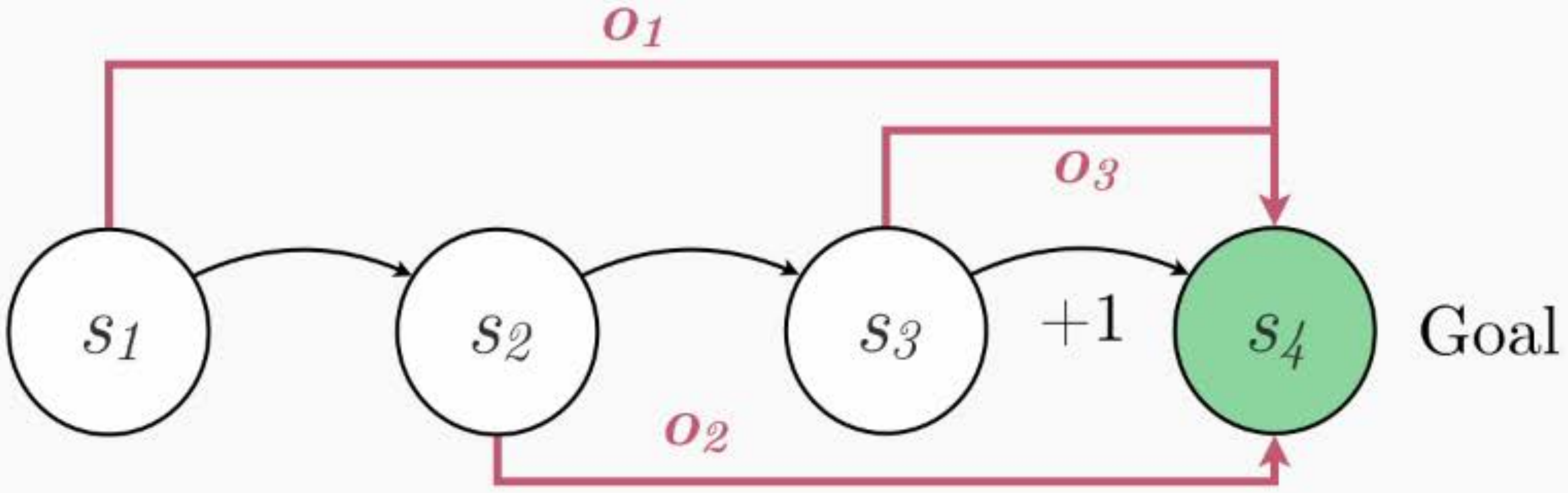
Value Iteration with Options



$V_0(s)$ 0 0 0 0

Thanks to Yuu Jinnai for the example

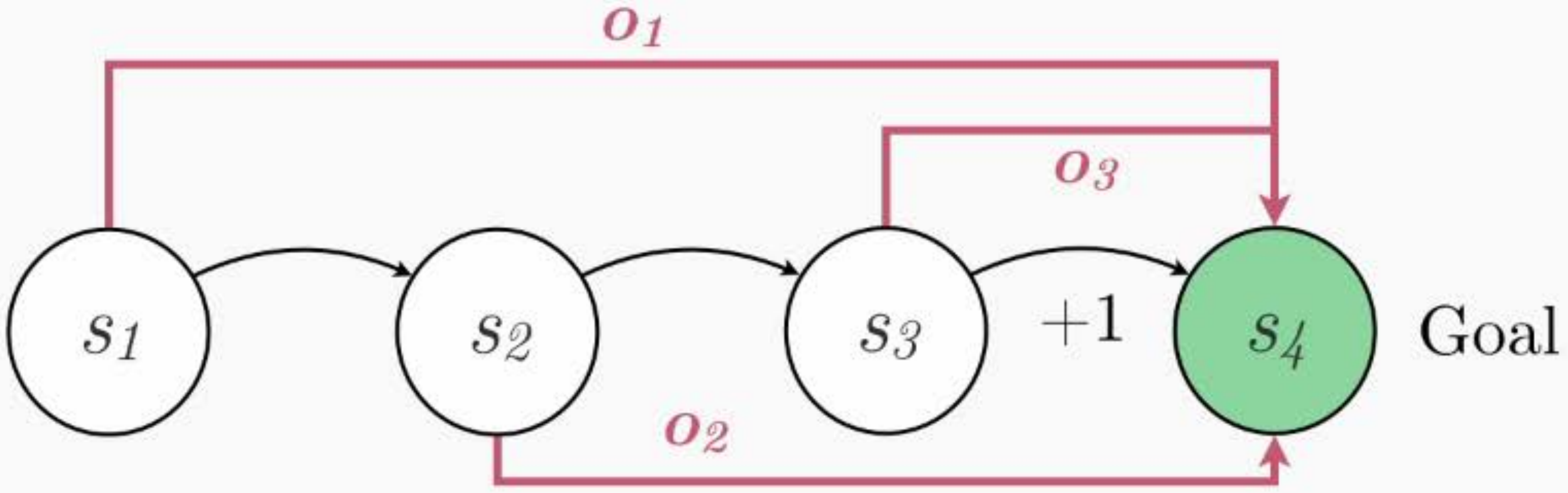
Value Iteration with Options



$V_0(s)$	0	0	0	0
$V_1(s)$	γ^2	γ	1	0

Thanks to Yuu Jinnai for the example

Value Iteration with Options



$V_0(s)$	0	0	0	0
$V_1(s)$	γ^2	γ	1	0

Value Iteration with Options

$$V_{i+1}(s) = \max_{o \in \mathcal{A} \cup \mathcal{O}} \left(R_\gamma(s, o) + \sum_{s' \in \mathcal{S}} T_\gamma(s' \mid s, o) V_i(s') \right)$$

Value Iteration with Options

$$V_{i+1}(s) = \max_{o \in \mathcal{A} \cup \mathcal{O}} \left(R_\gamma(s, o) + \sum_{s' \in \mathcal{S}} T_\gamma(s' \mid s, o) V_i(s') \right)$$

Planning with options *and* primitives

[Ciosek and Silver 2015; Sutton, Precup, Singh 1999]

Value Iteration with Options

$$V_{i+1}(s) = \max_{o \in \mathcal{A} \cup \mathcal{O}} \left(R_\gamma(s, o) + \sum_{s' \in \mathcal{S}} T_\gamma(s' \mid s, o) V_i(s') \right)$$

Planning with options *and* primitives

Multi-time model

[Ciosek and Silver 2015; Sutton, Precup, Singh 1999]

Finding Options for Planning

Definition (Value-Planning Problem): **Given** *an MDP* M *and* $\varepsilon \in \mathbb{R}_{>0}$, **return** *a value function,* V *such that* $|V(s) - V^*(s)| < \varepsilon$ *for all* $s \in \mathcal{S}$.

Iterations needed for value iteration to converge

Question: *Can we find the set of options that make planning as fast as possible?*

Finding Options for Planning

Theorem. Finding the set of options that minimizes planning time is:

1) NP-hard in general.

2) $2^{\log^{1-\varepsilon} n}$-hard to approximate.[1]

[1]Unless $NP \subseteq DTIME(n^{\text{poly} \log n})$ [Dinitz et al. 2012]

Question: *Can we find the set of options that make planning as fast as possible?*

Finding Options for Planning

Theorem. Finding the set of options that minimizes planning time is:

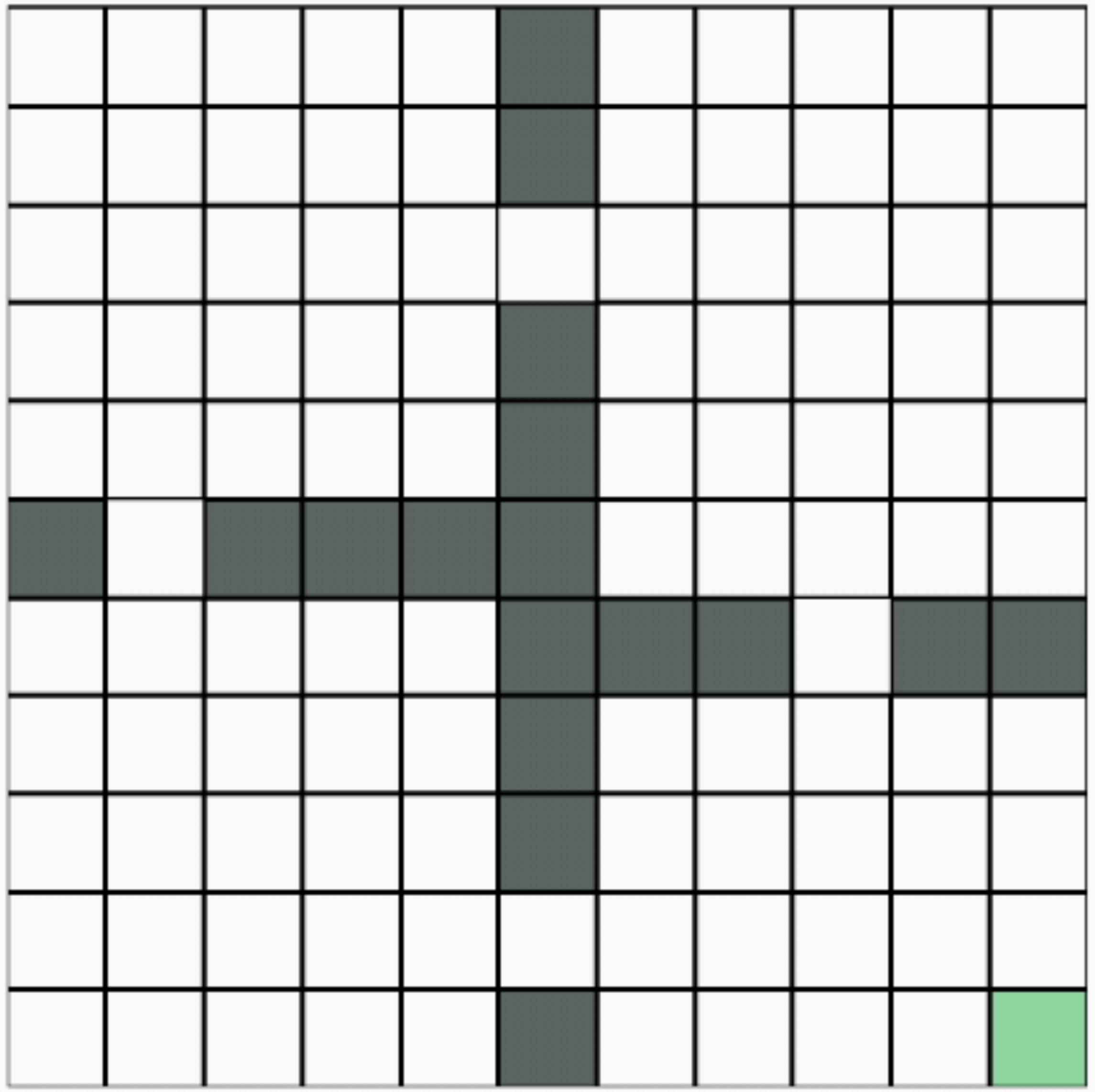
1) NP-hard in general.

2) $2^{\log^{1-\varepsilon} n}$-hard to approximate.[1]

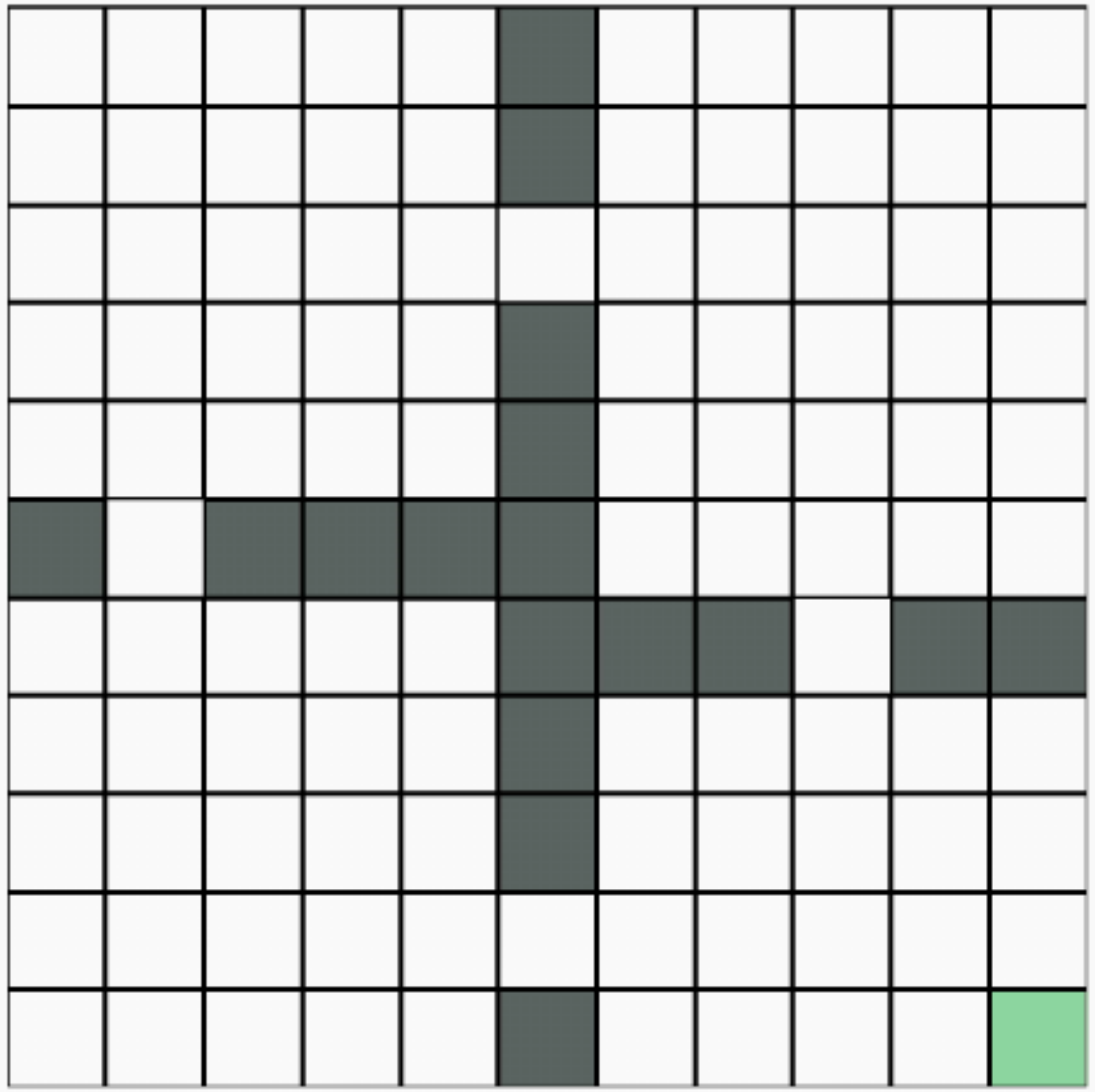
[1]Unless $NP \subseteq DTIME(n^{\text{poly} \log n})$ [Dinitz et al. 2012]

Corollary. Also holds for distributions of tasks.

Visuals: $K = 1$

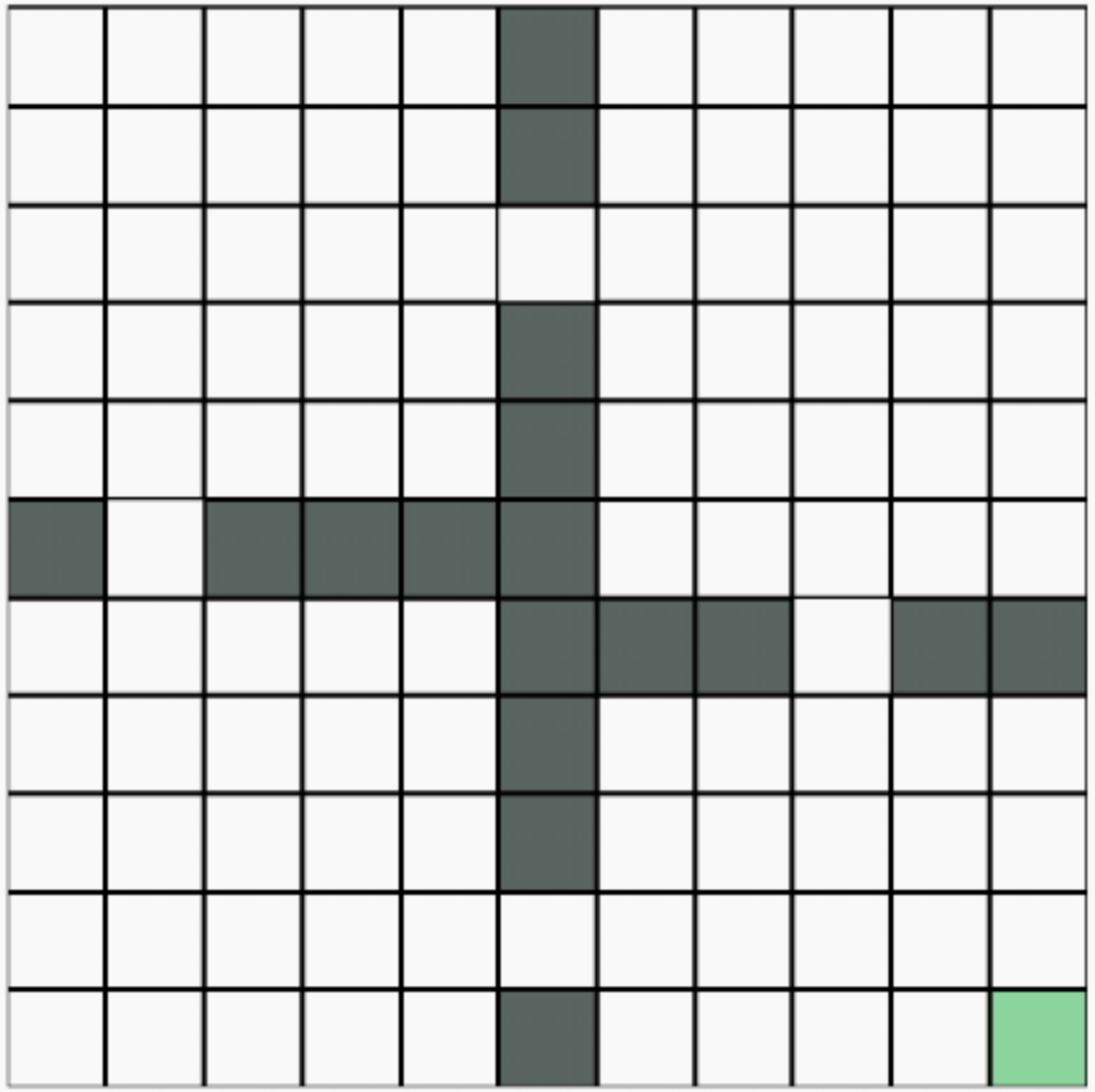


Optimal

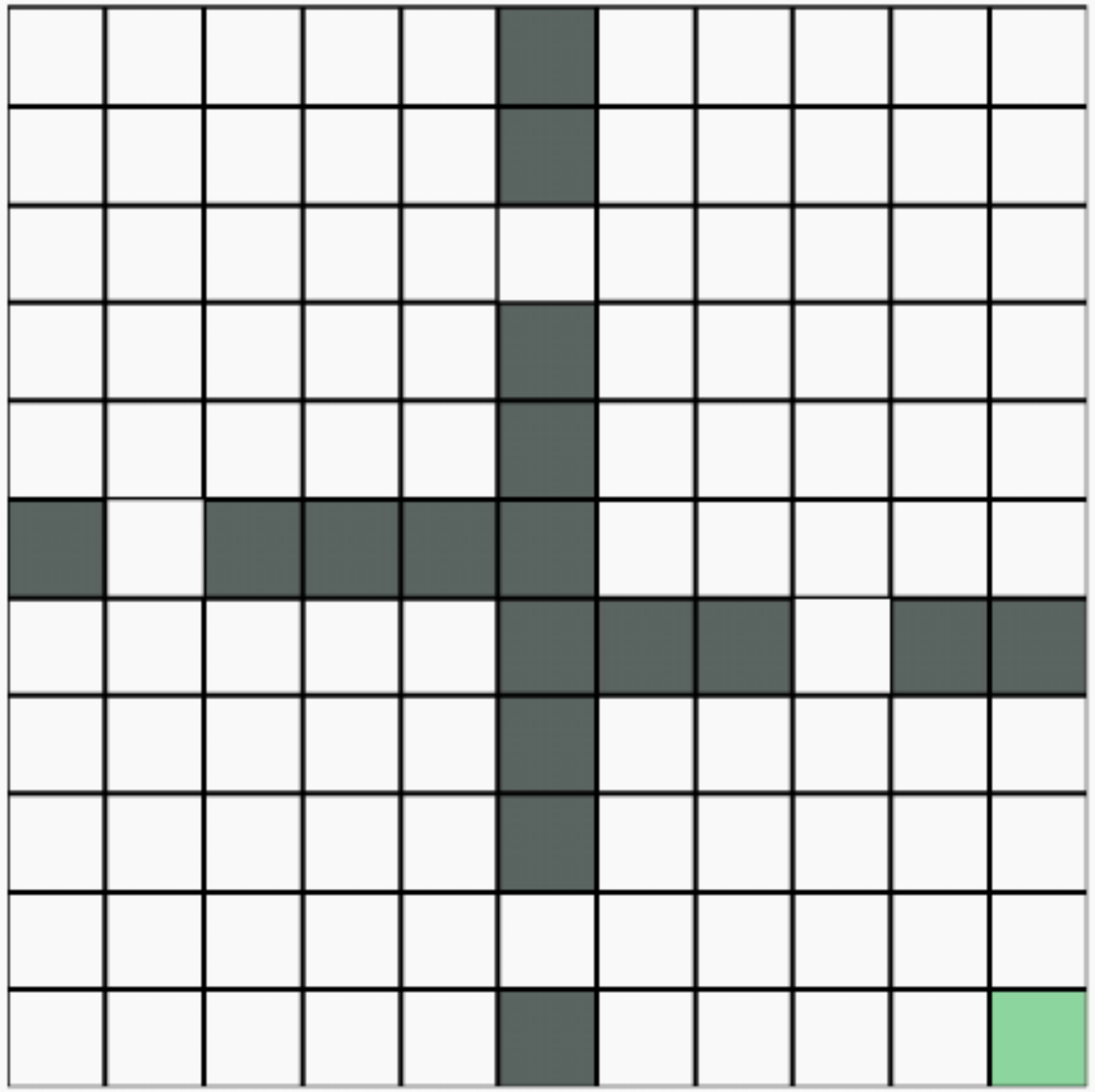


Approximation

Visuals: $K = 2$

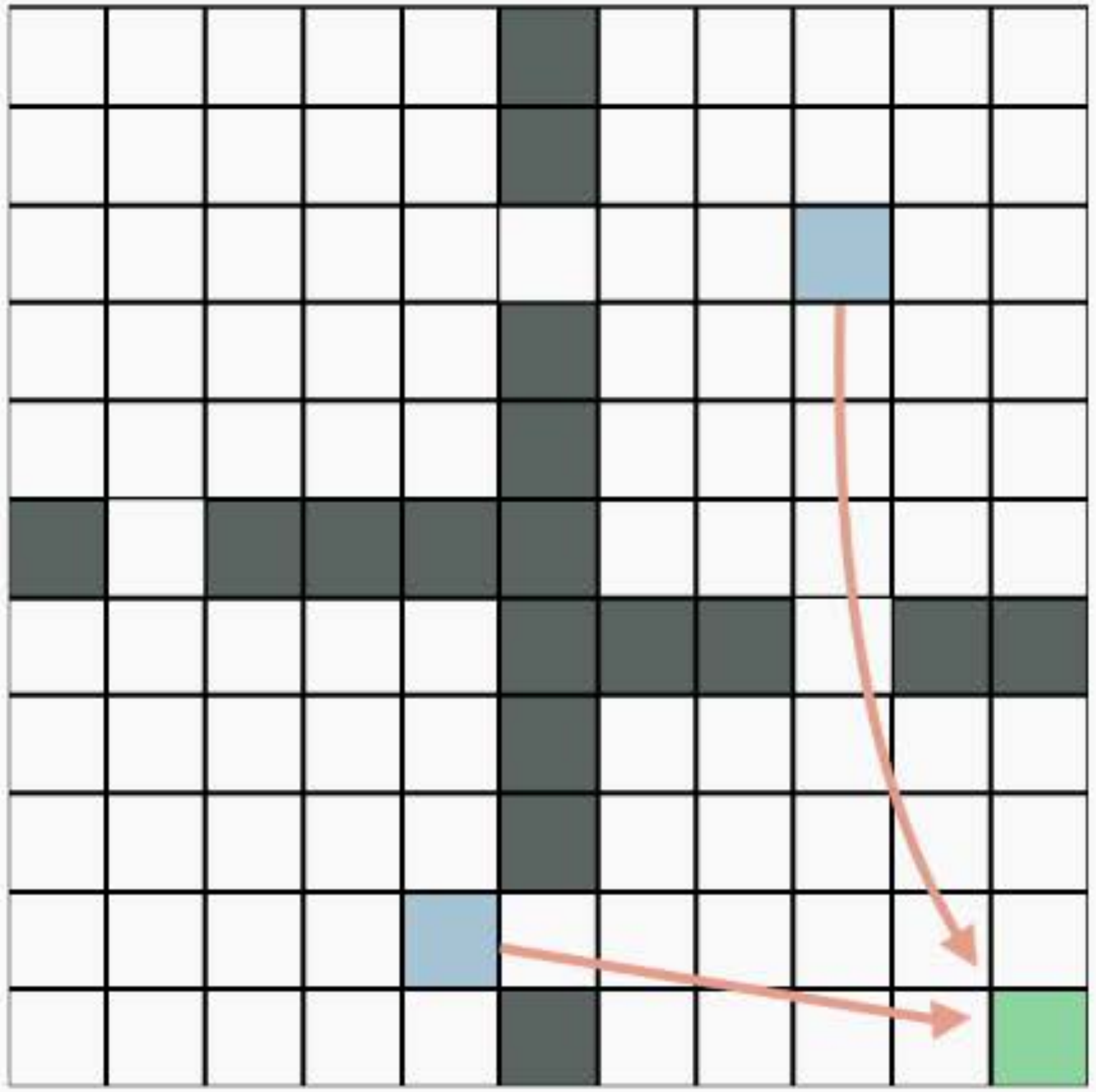


Optimal

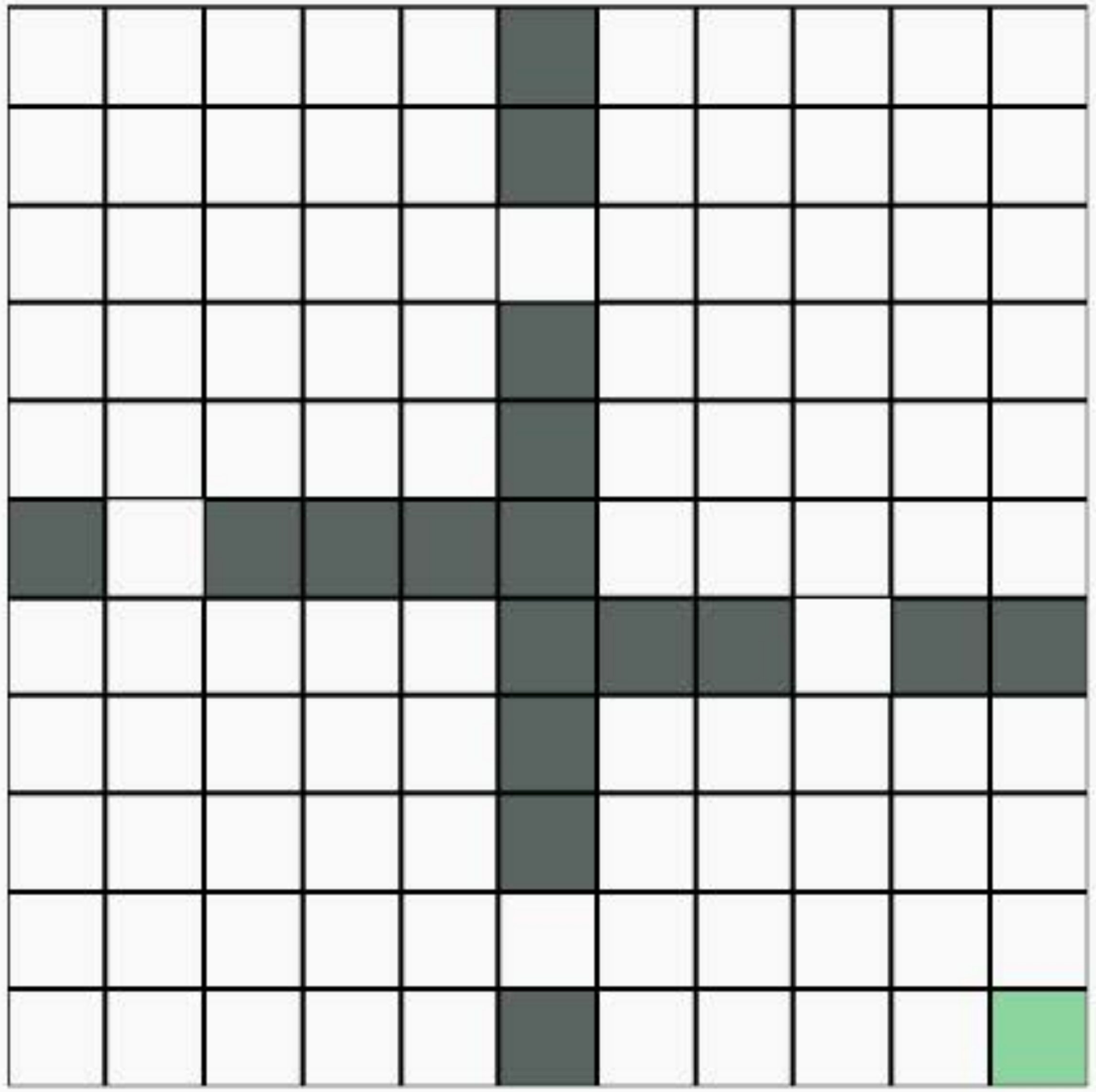


Approximation

Visuals: $K = 2$

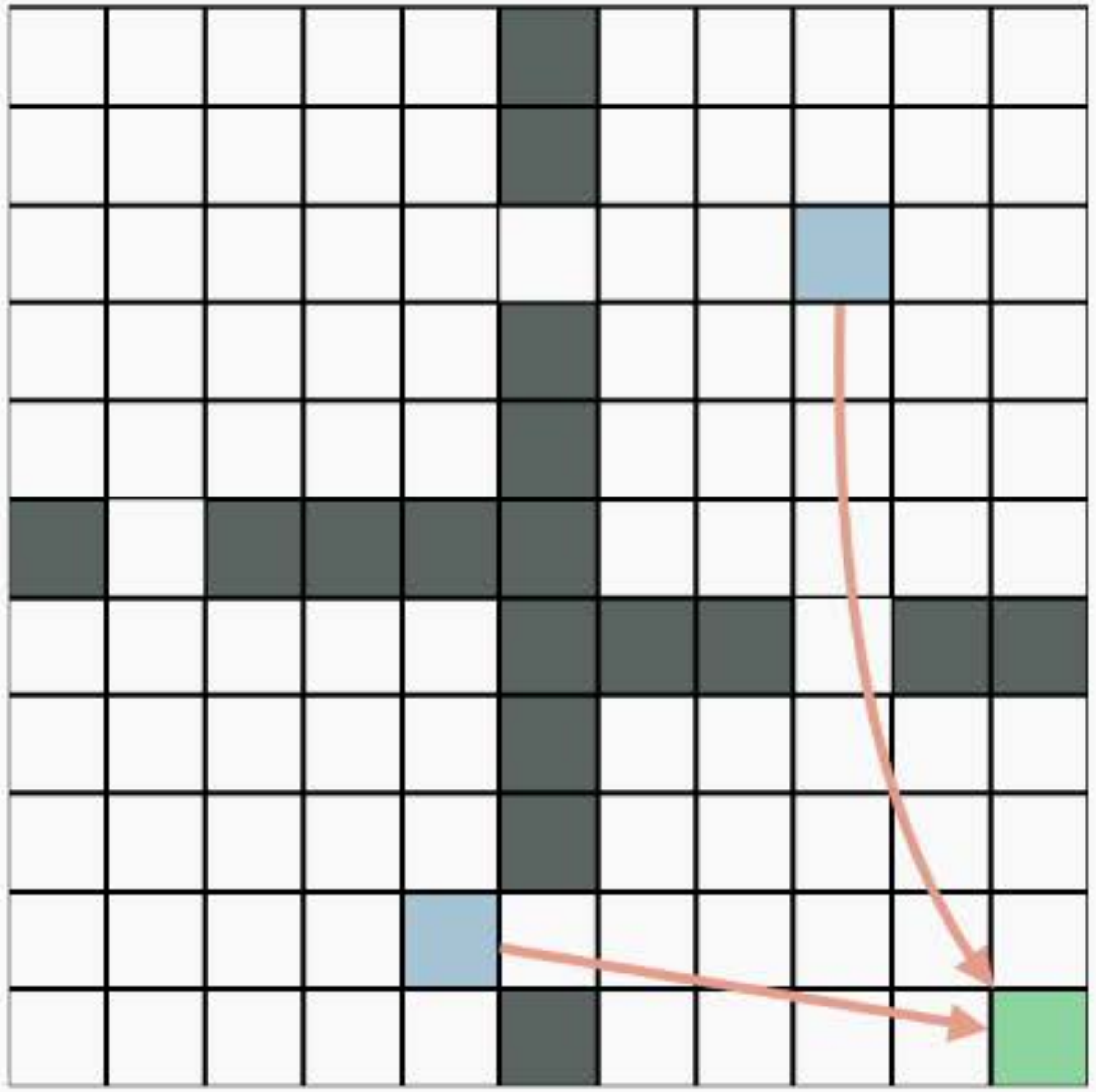


Optimal

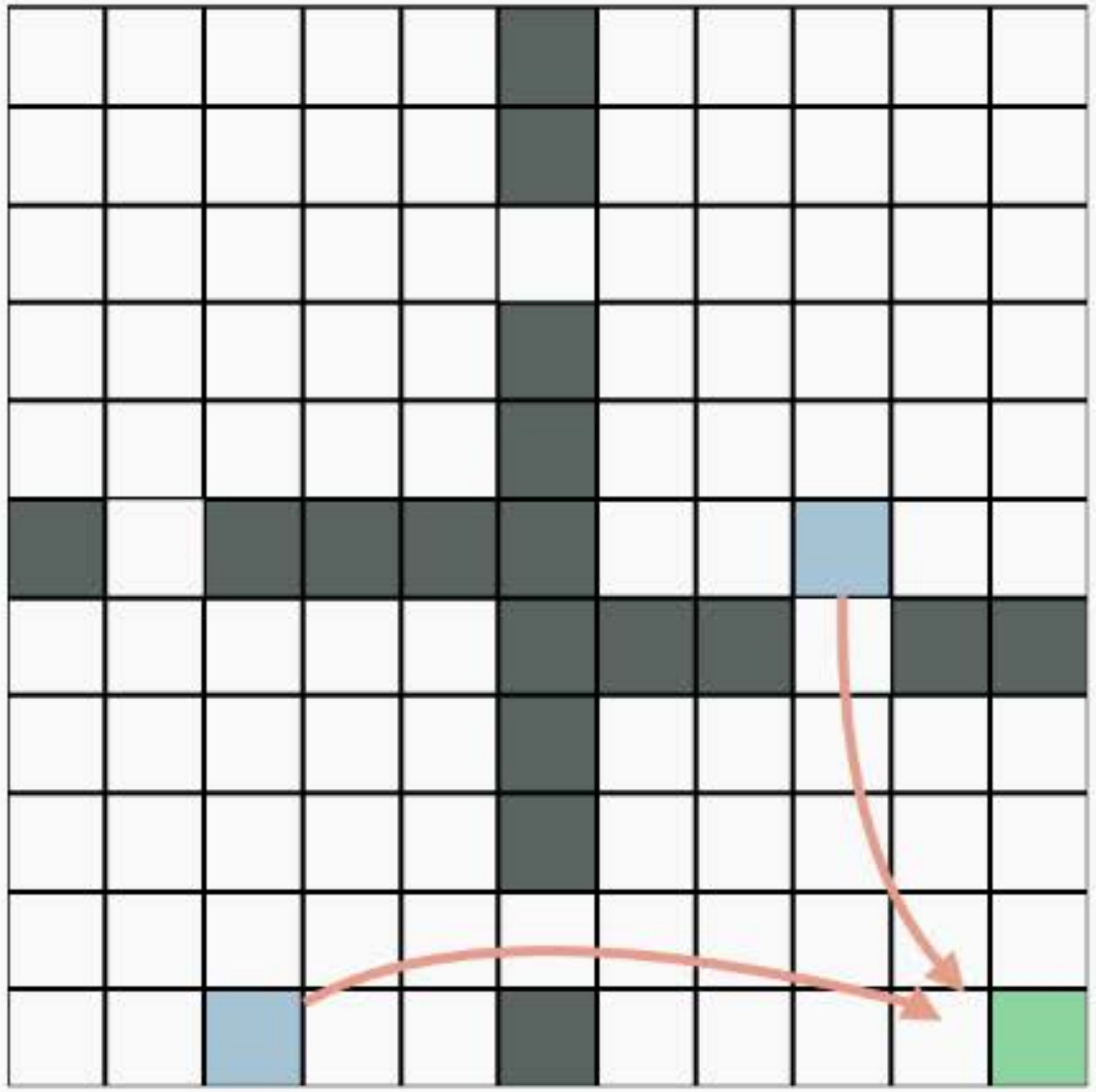


Approximation

Visuals: $K = 2$

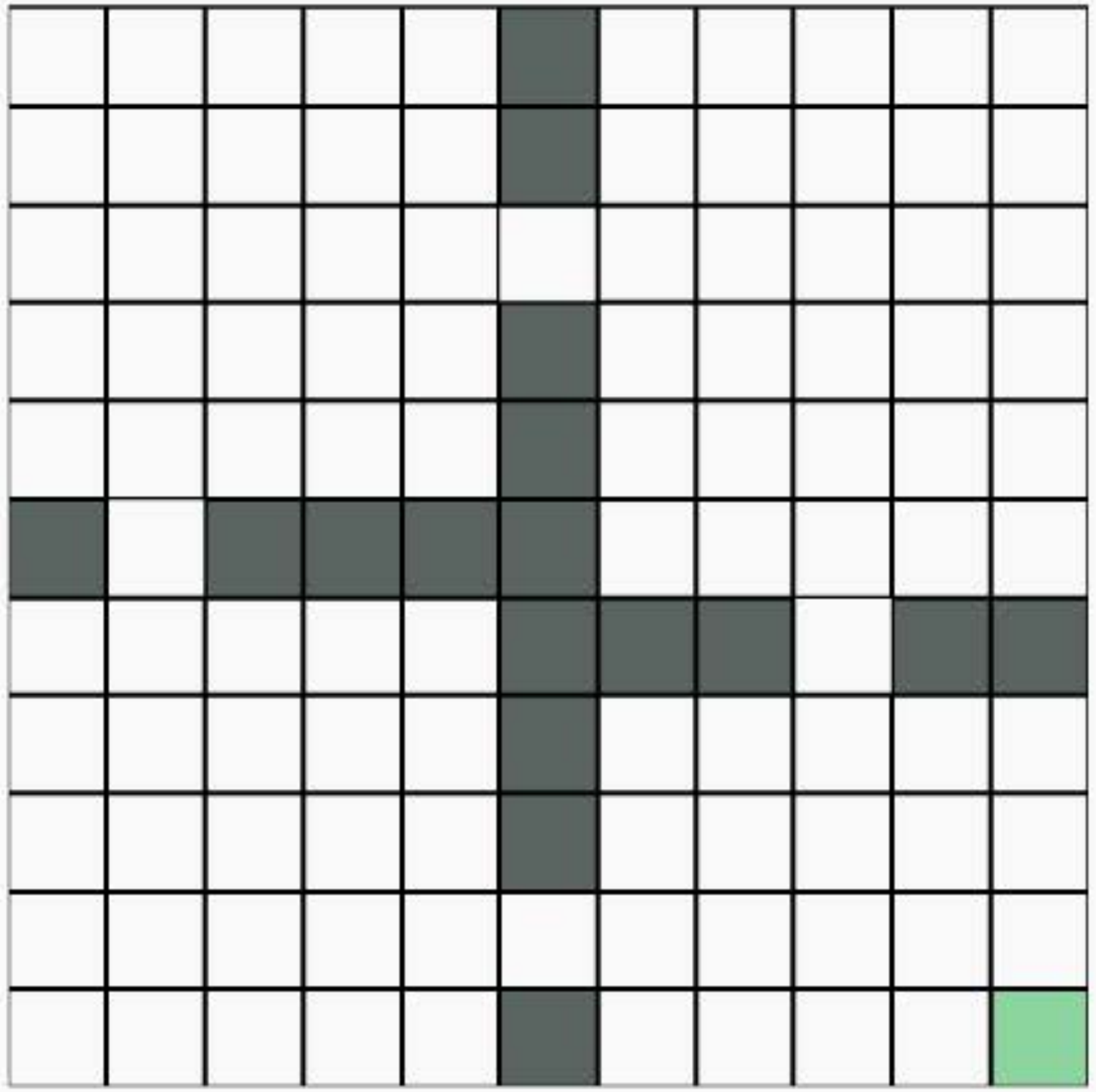


Optimal



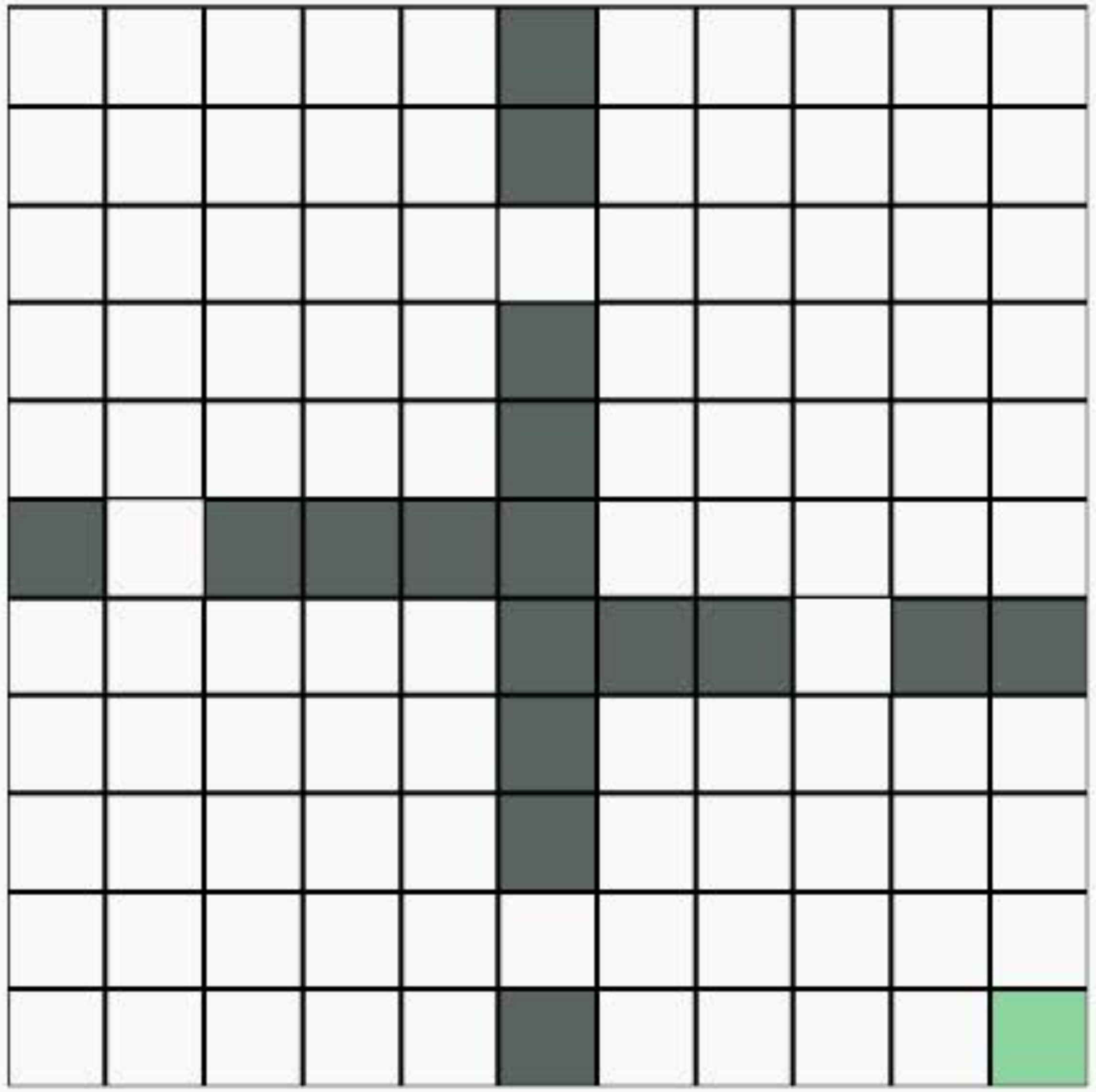
Approximation

Visuals: $K = 4$



Betweenness Options

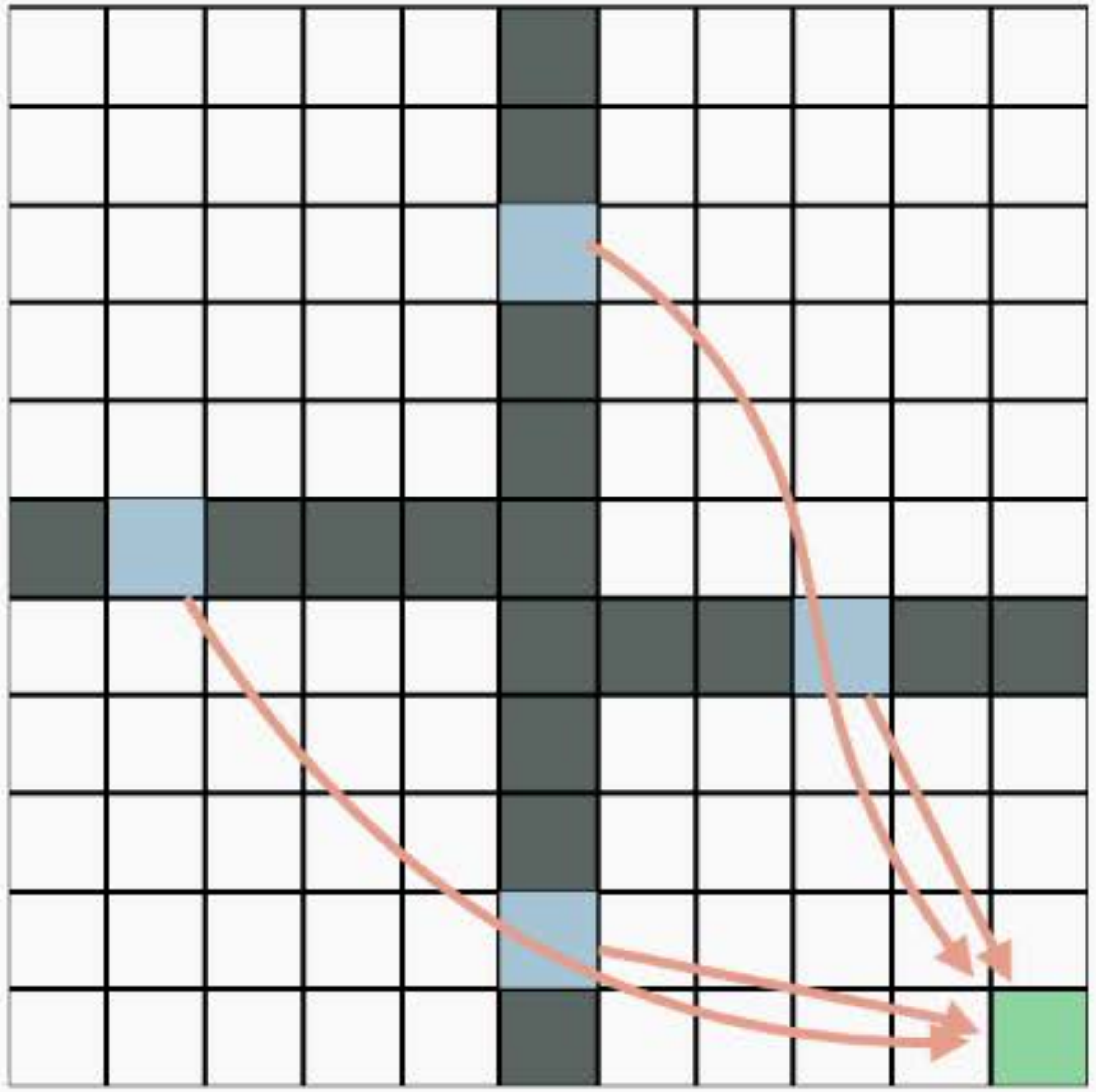
[Simsek and Barto 2005, 2008]



Eigen Options

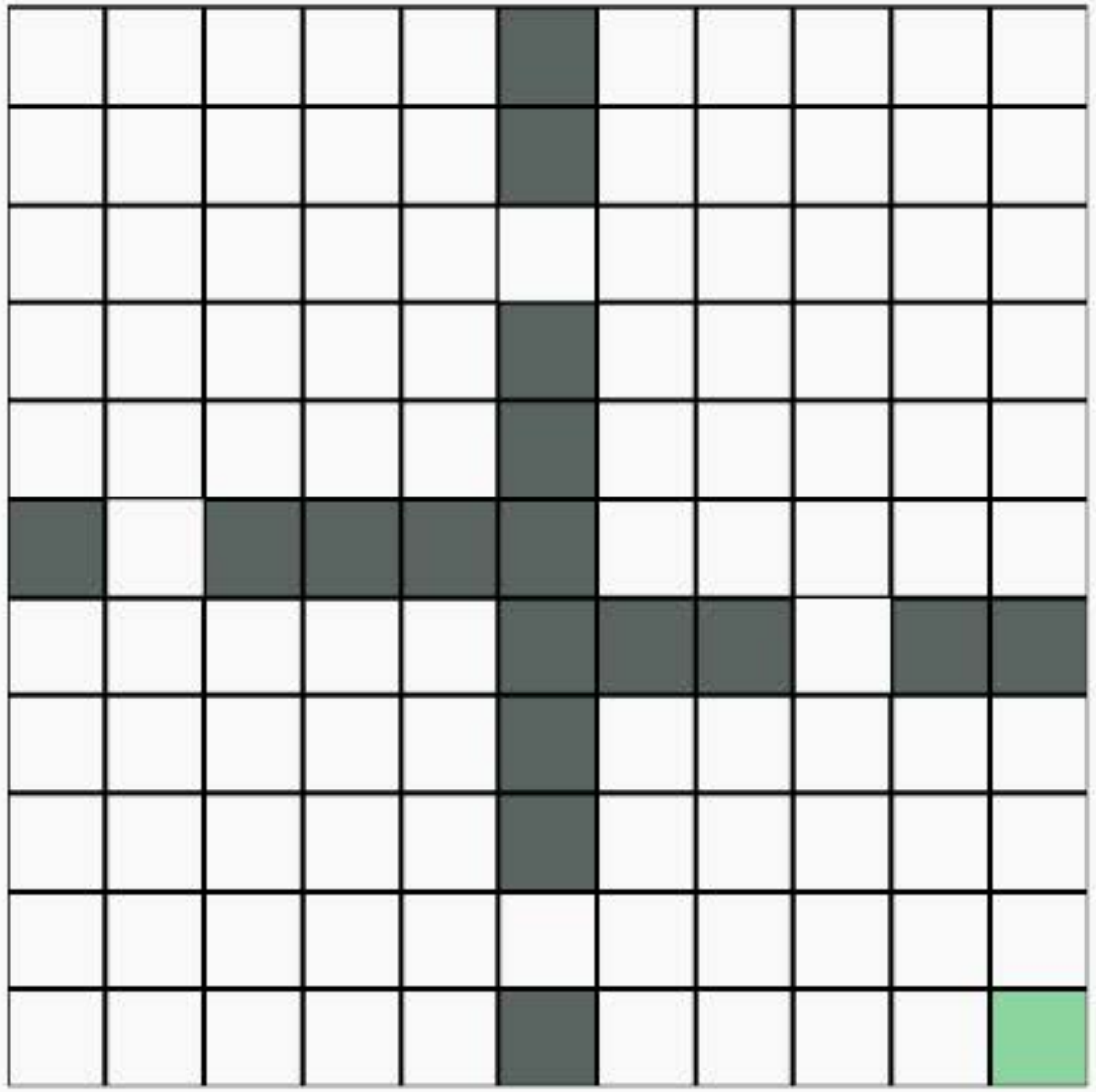
[Machado et al. 2017]

Visuals: $K = 4$



Betweenness Options

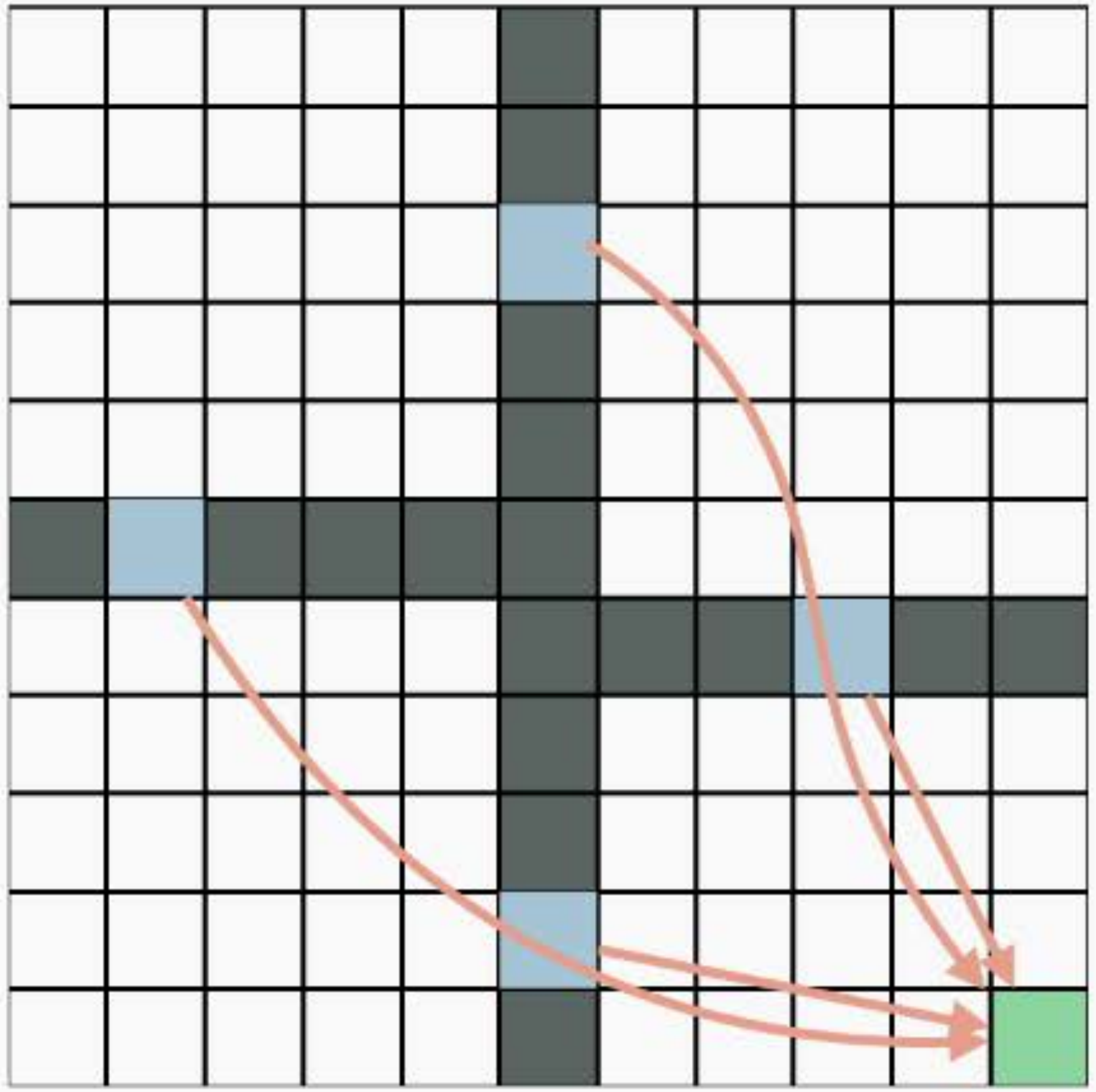
[Simsek and Barto 2005, 2008]



Eigen Options

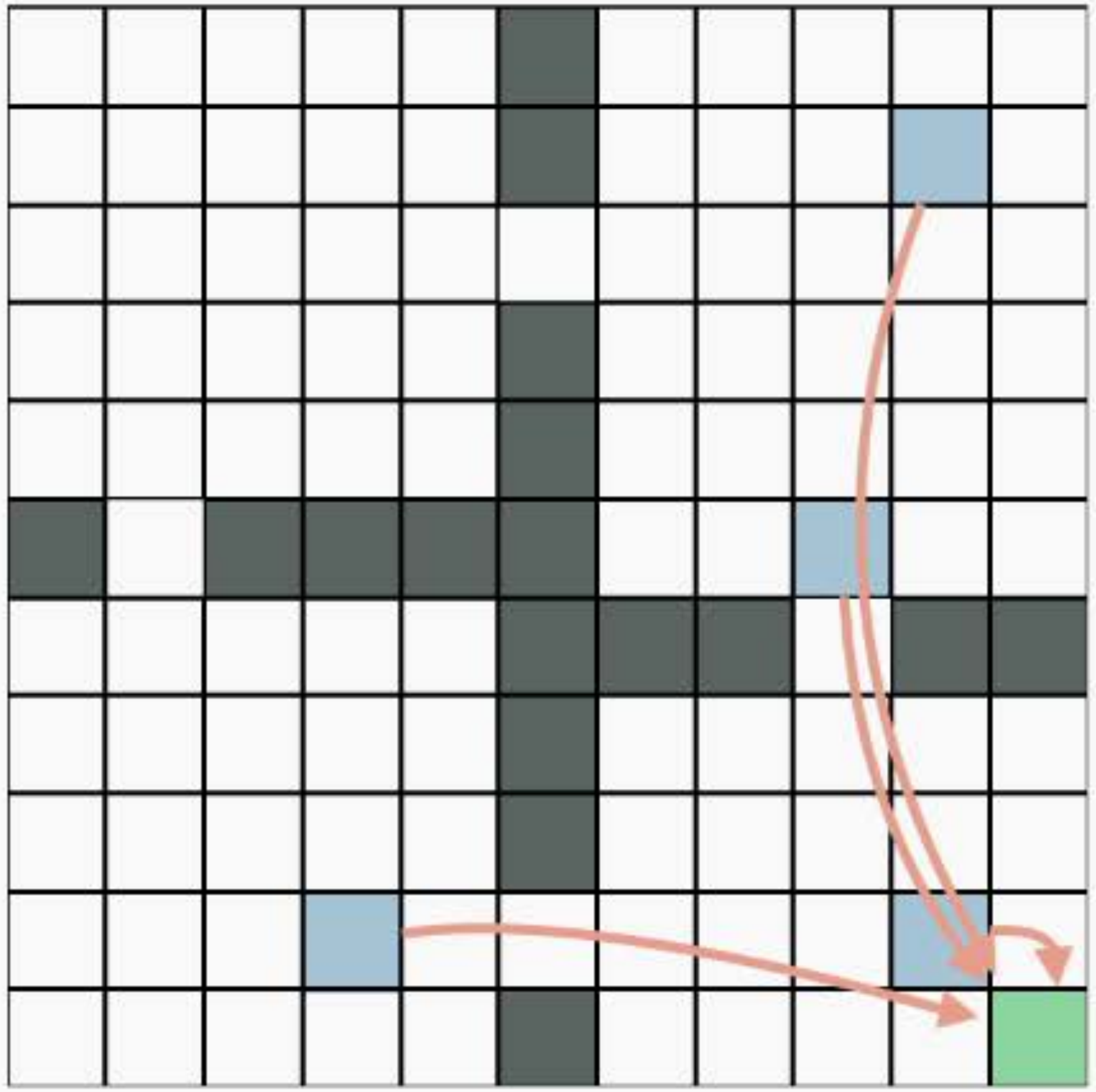
[Machado et al. 2017]

Visuals: $K = 4$



Betweenness Options

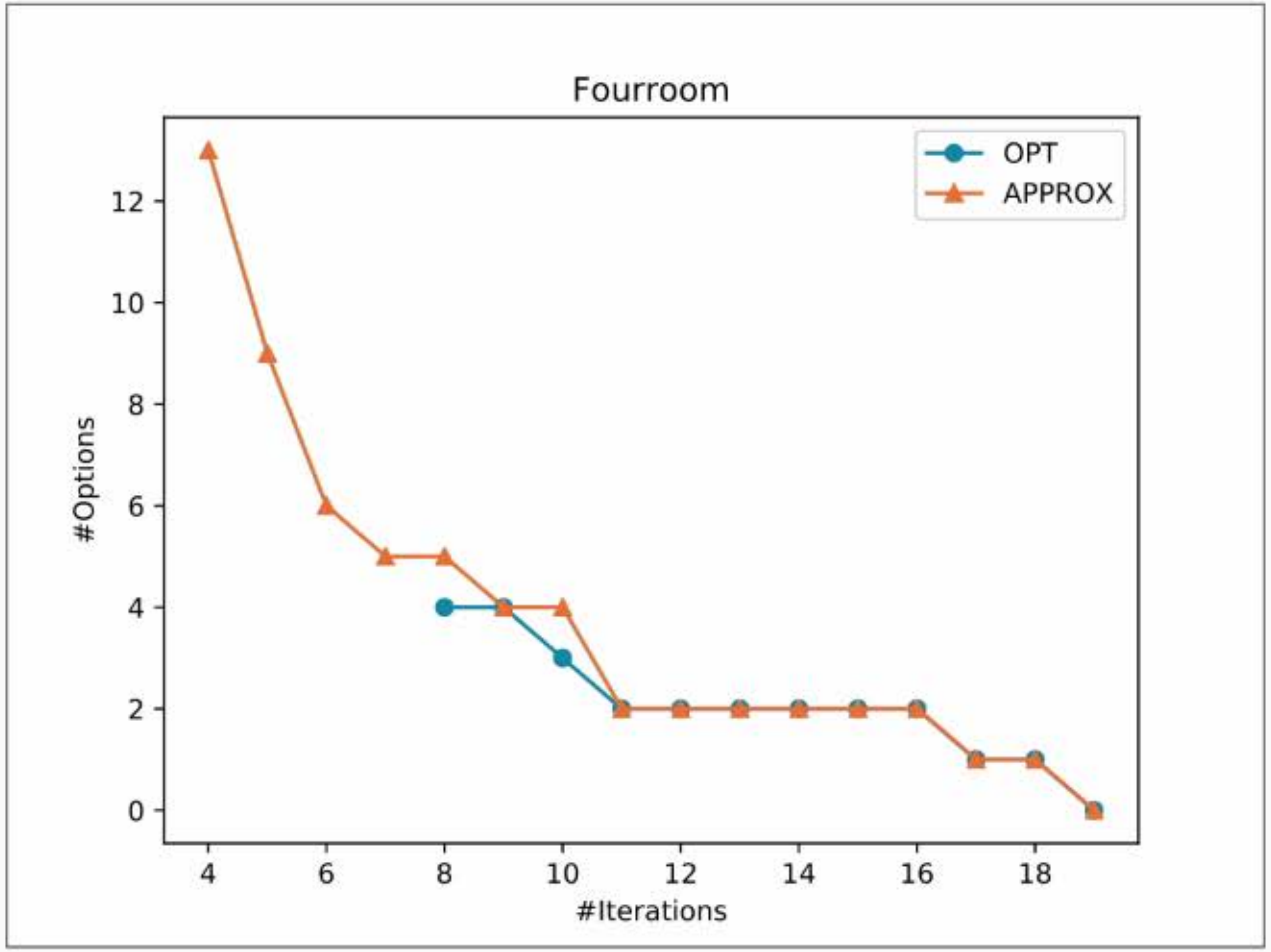
[Simsek and Barto 2005, 2008]



Eigen Options

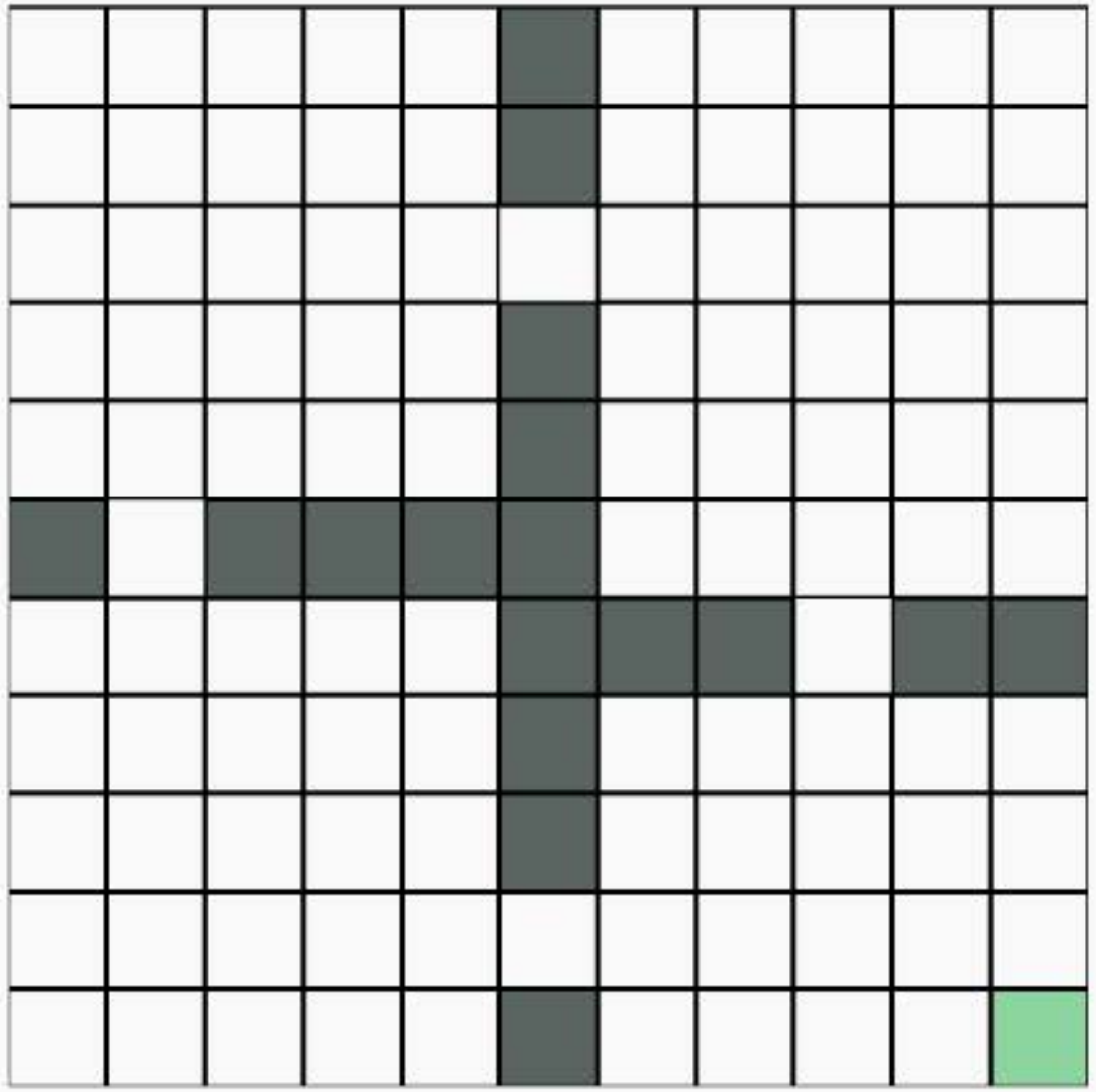
[Machado et al. 2017]

Evaluation



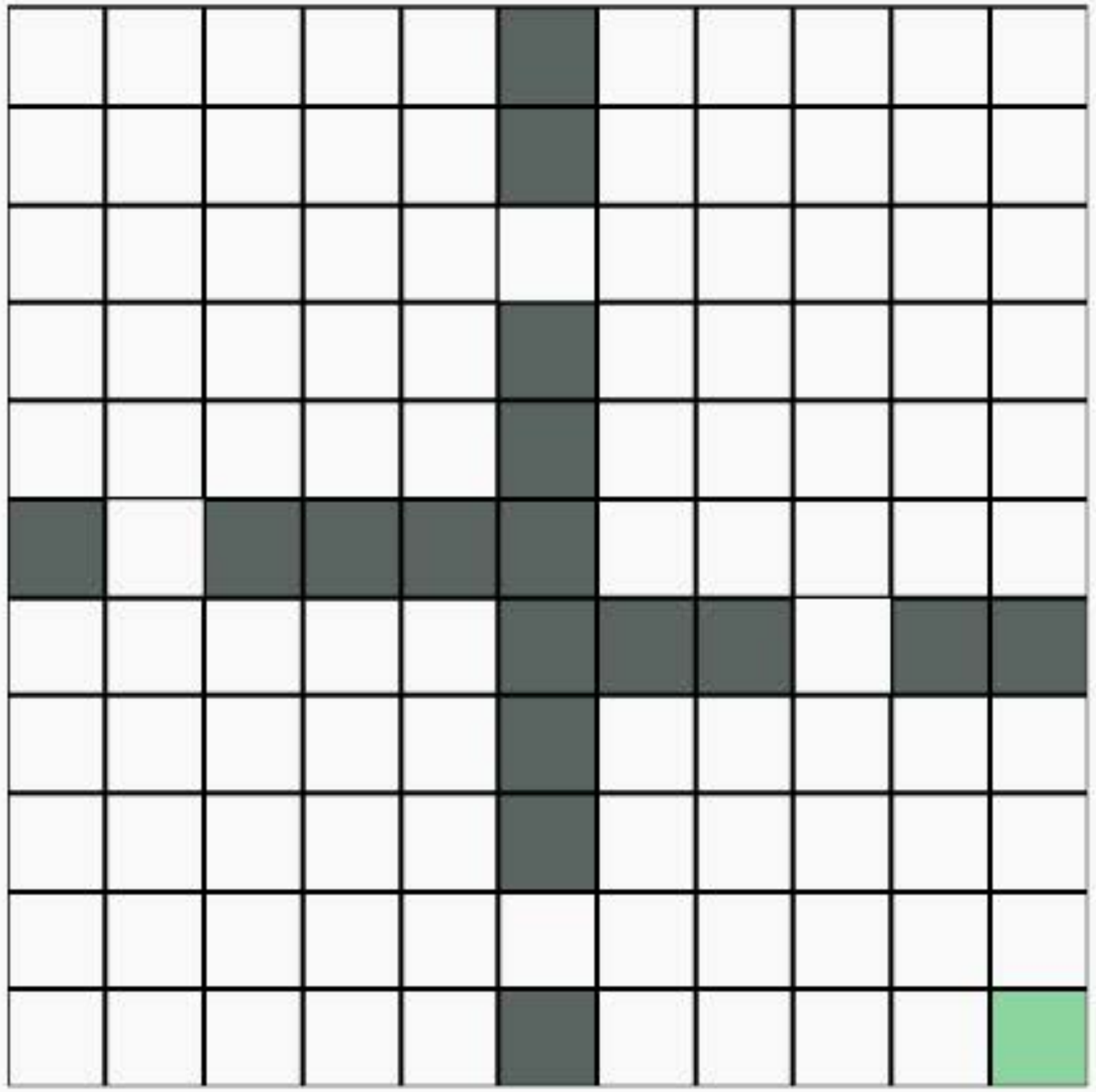
Fourroom
OPT
APPROX
#Options
#Iterations
0
2
4
6
8
10
12
4
6
8
10
12
14
16
18

Visuals: $K = 4$



Betweenness Options

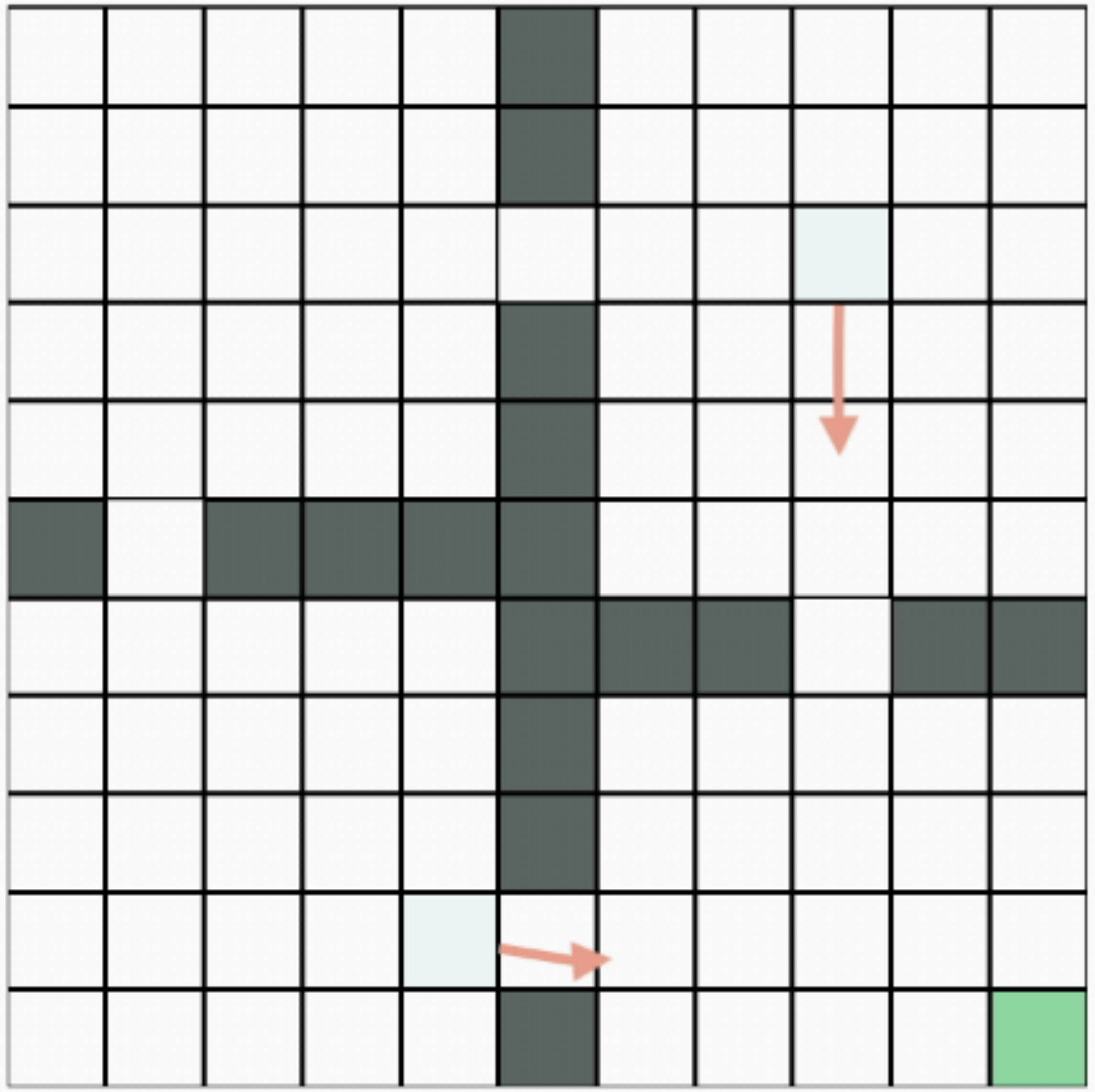
[Simsek and Barto 2005, 2008]



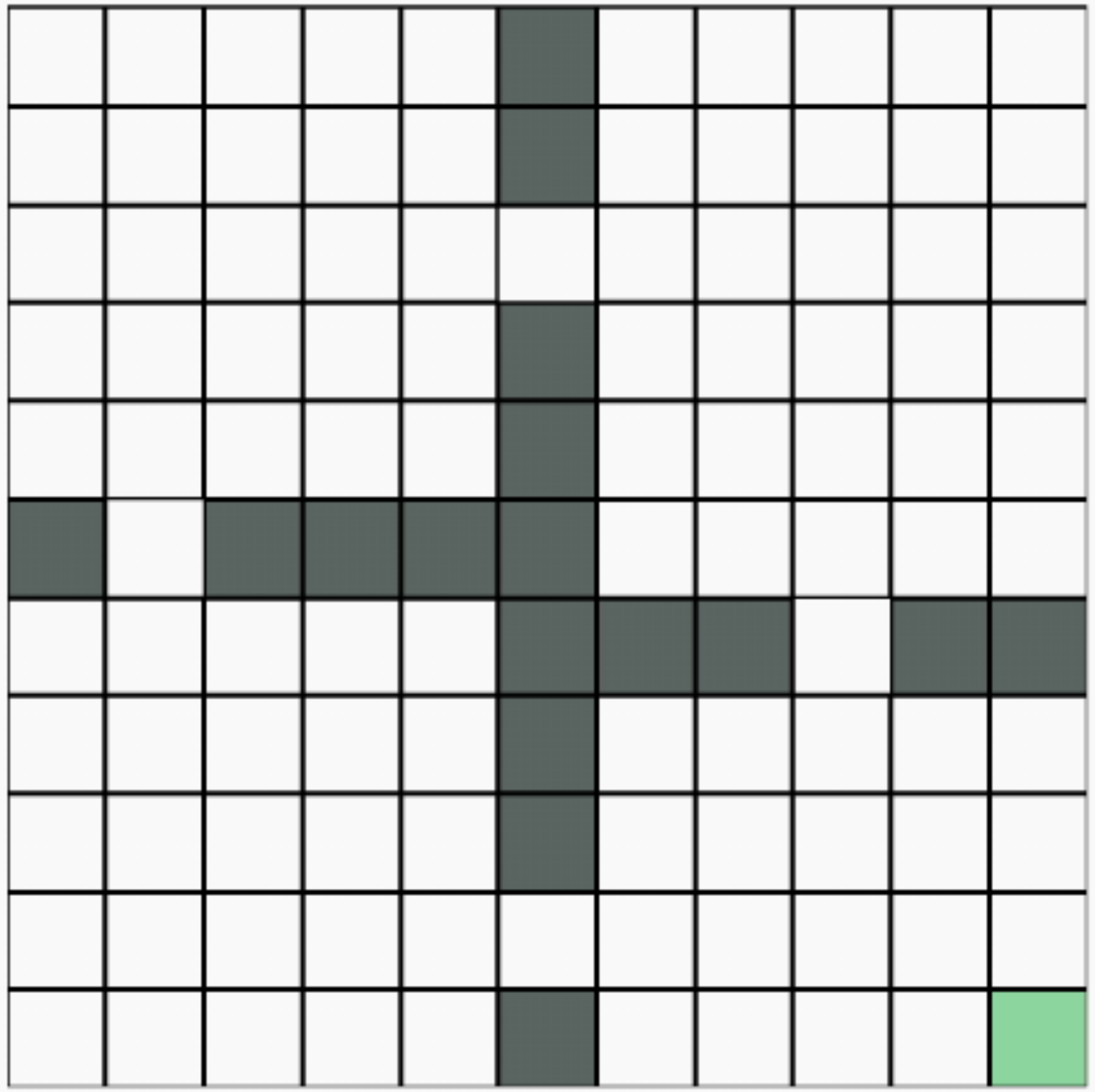
Eigen Options

[Machado et al. 2017]

Visuals: $K = 2$

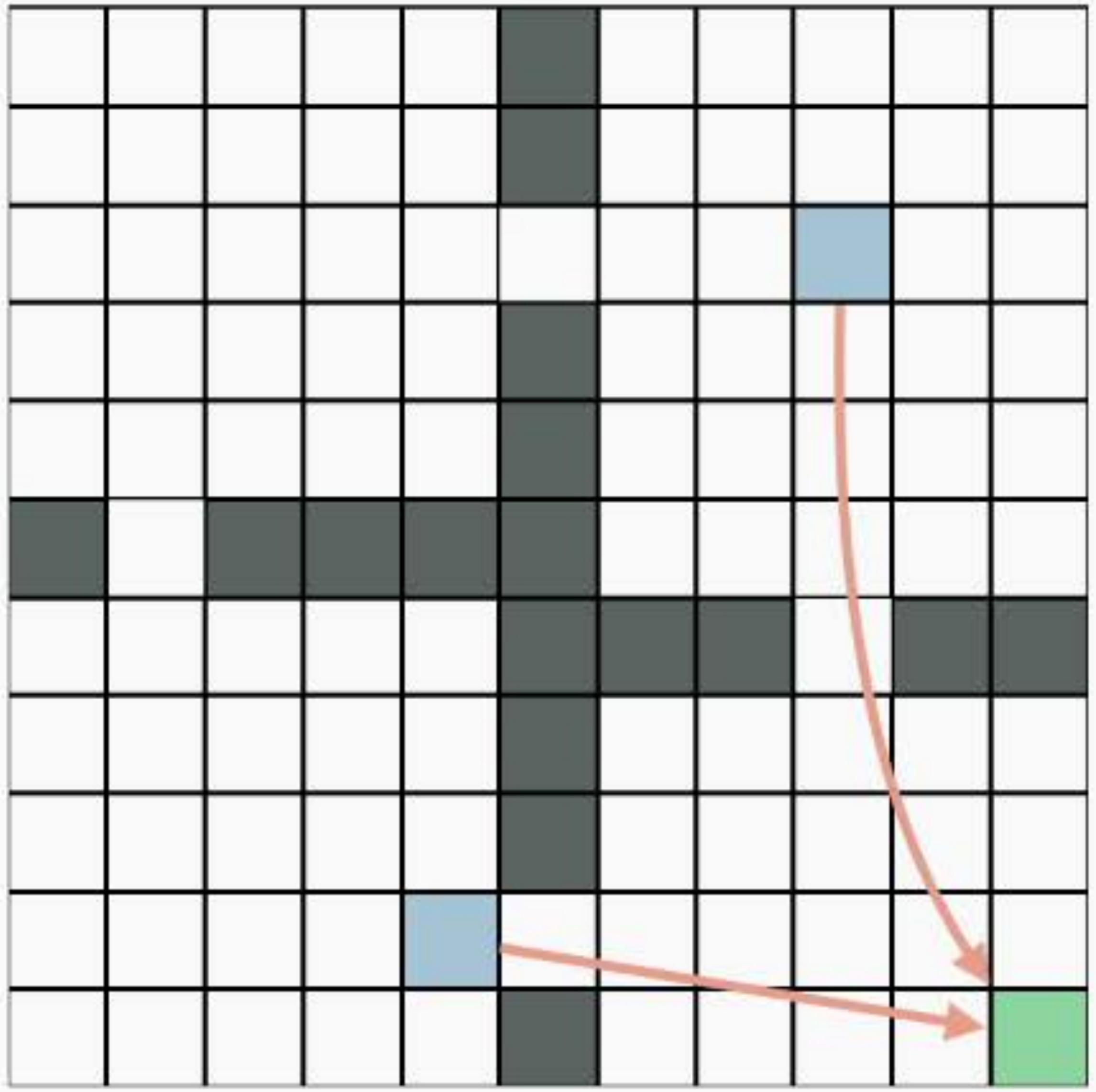


Optimal

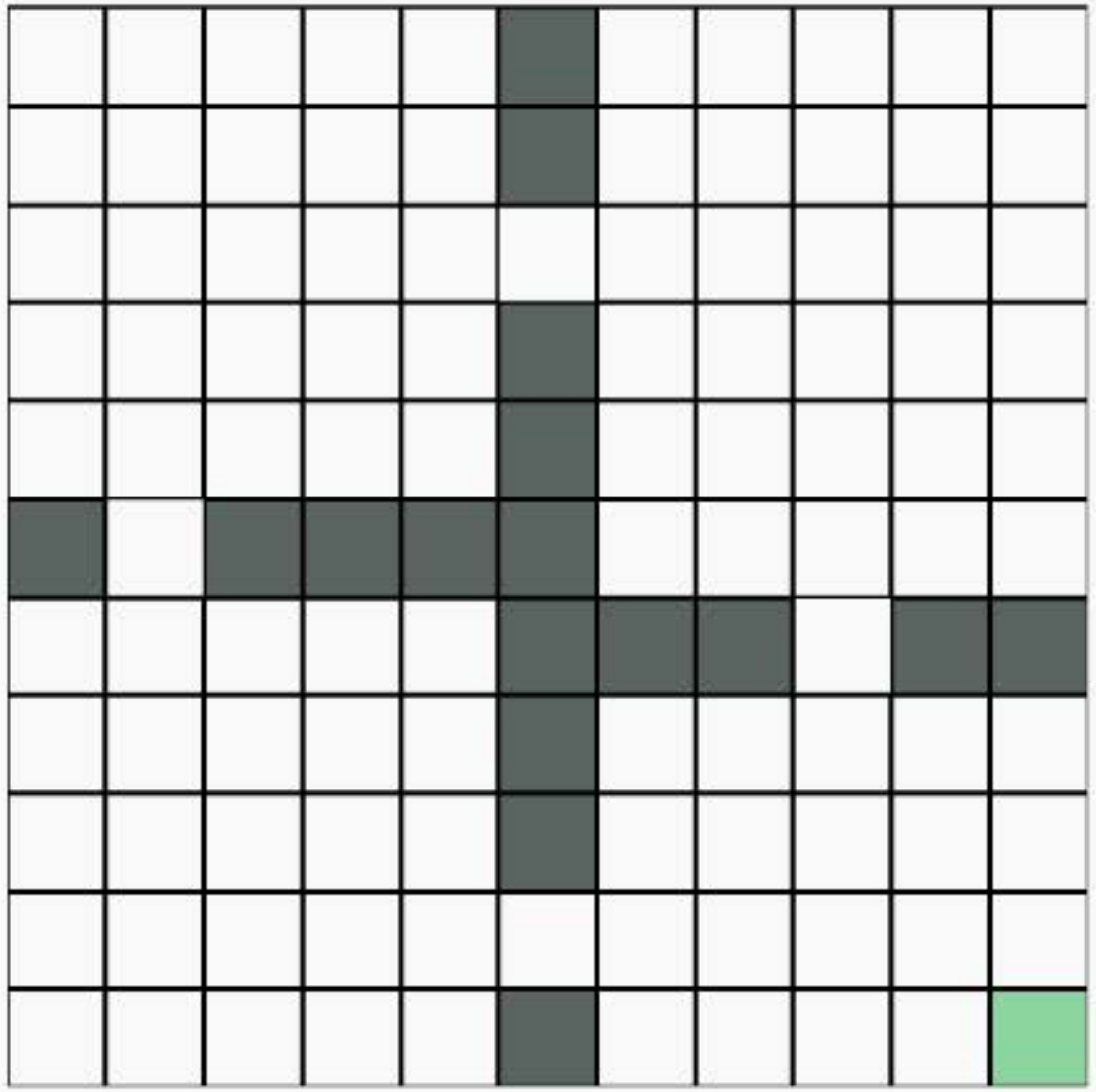


Approximation

Visuals: $K = 2$

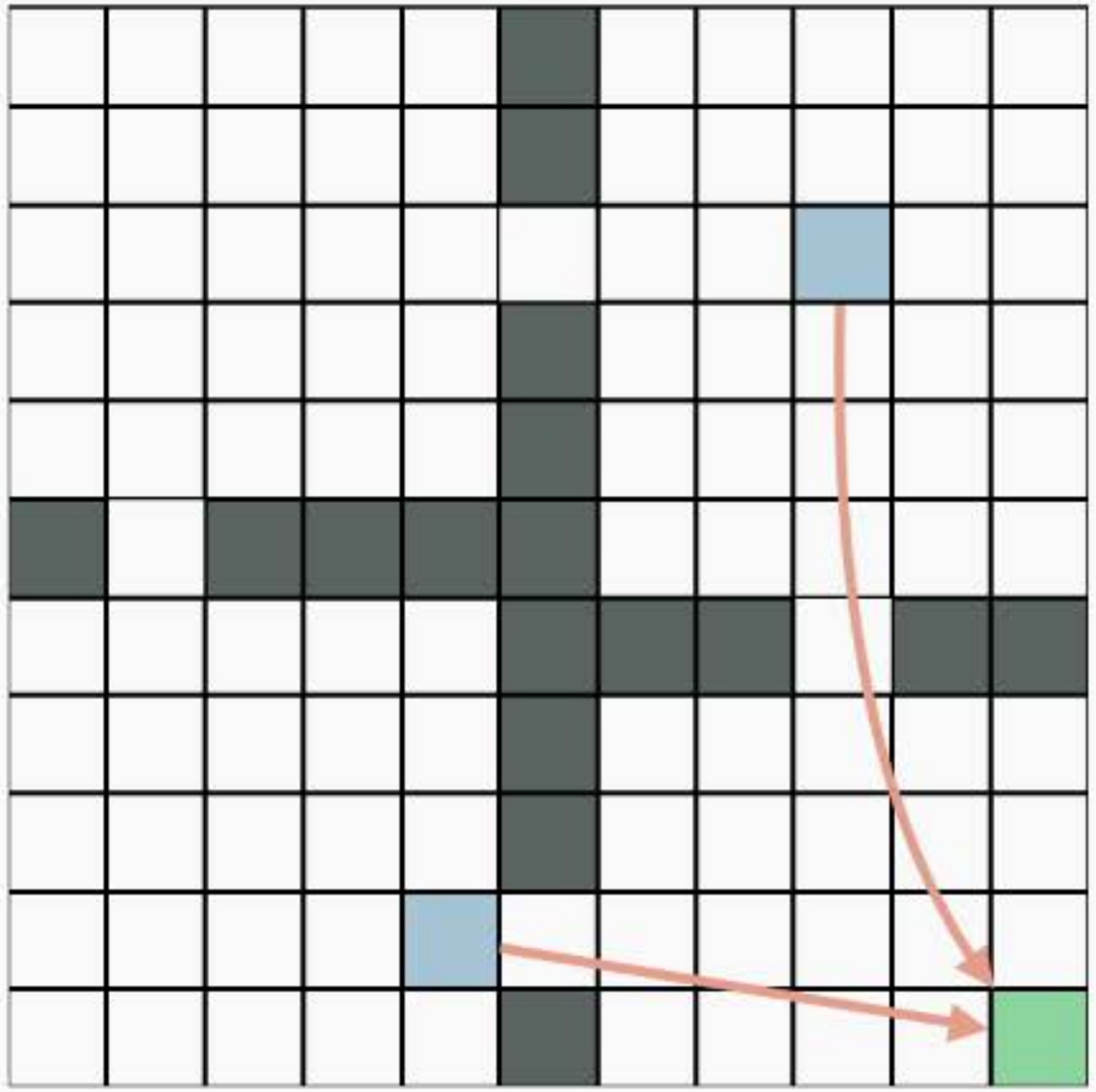


Optimal

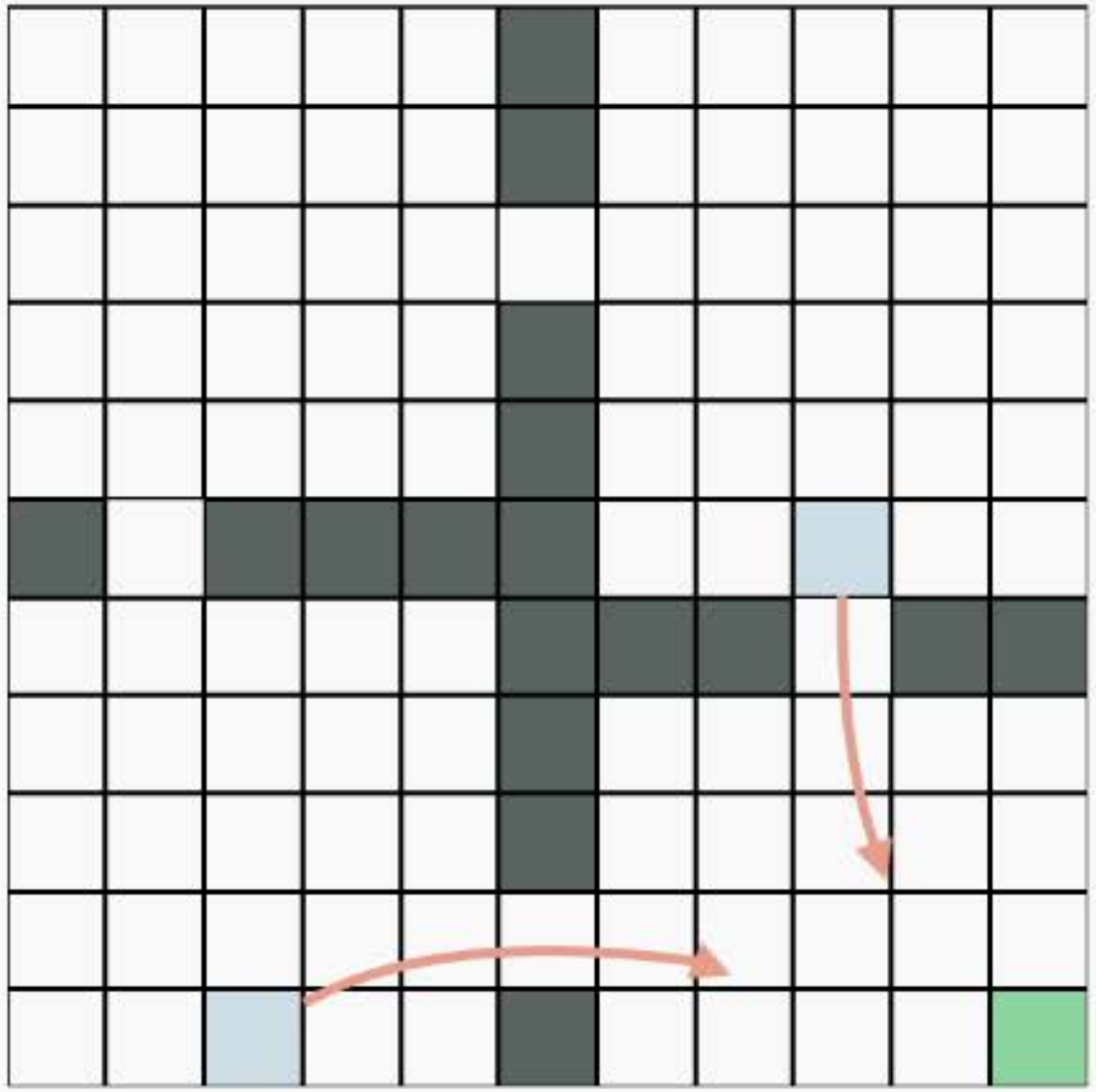


Approximation

Visuals: $K = 2$

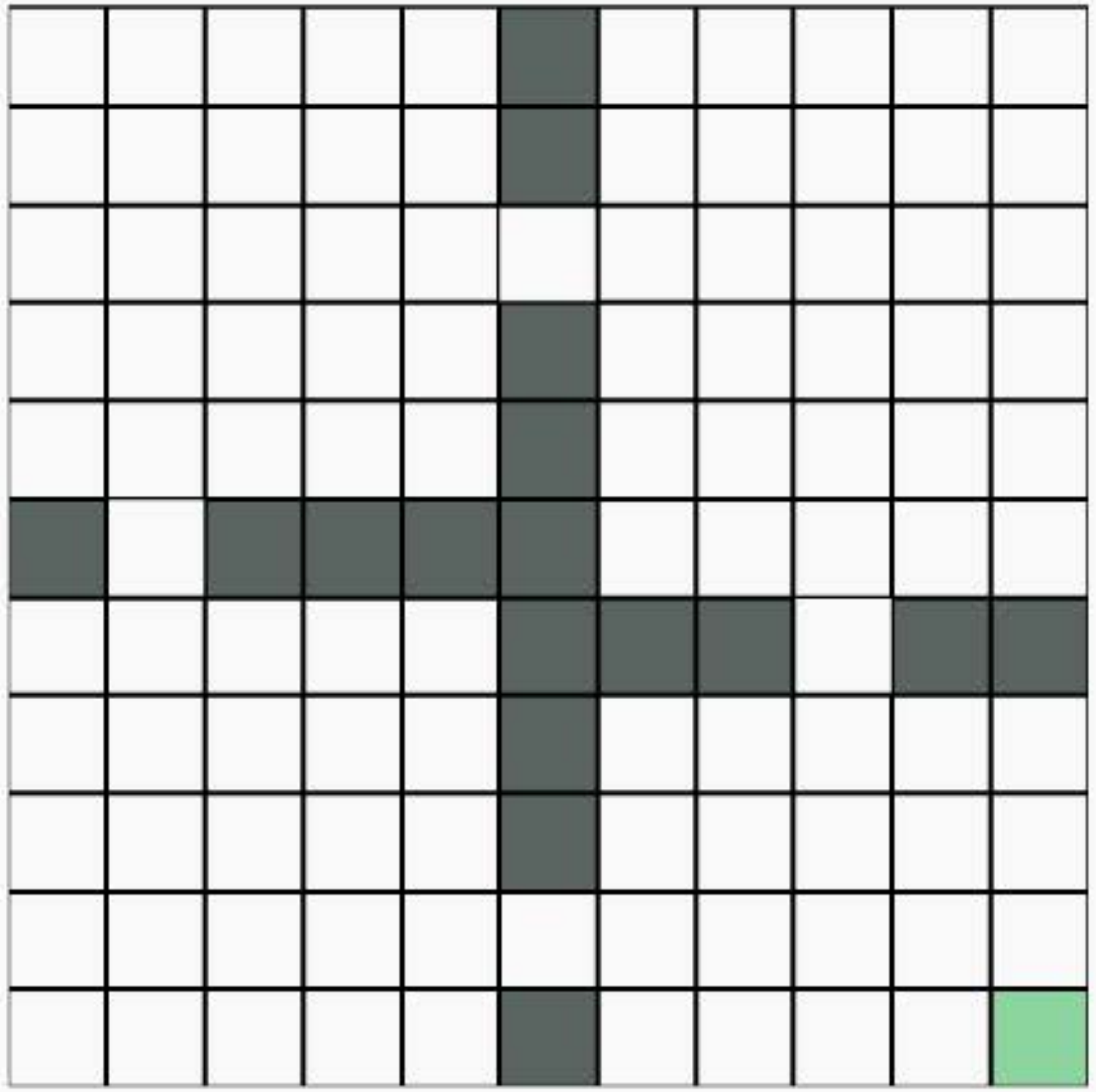


Optimal



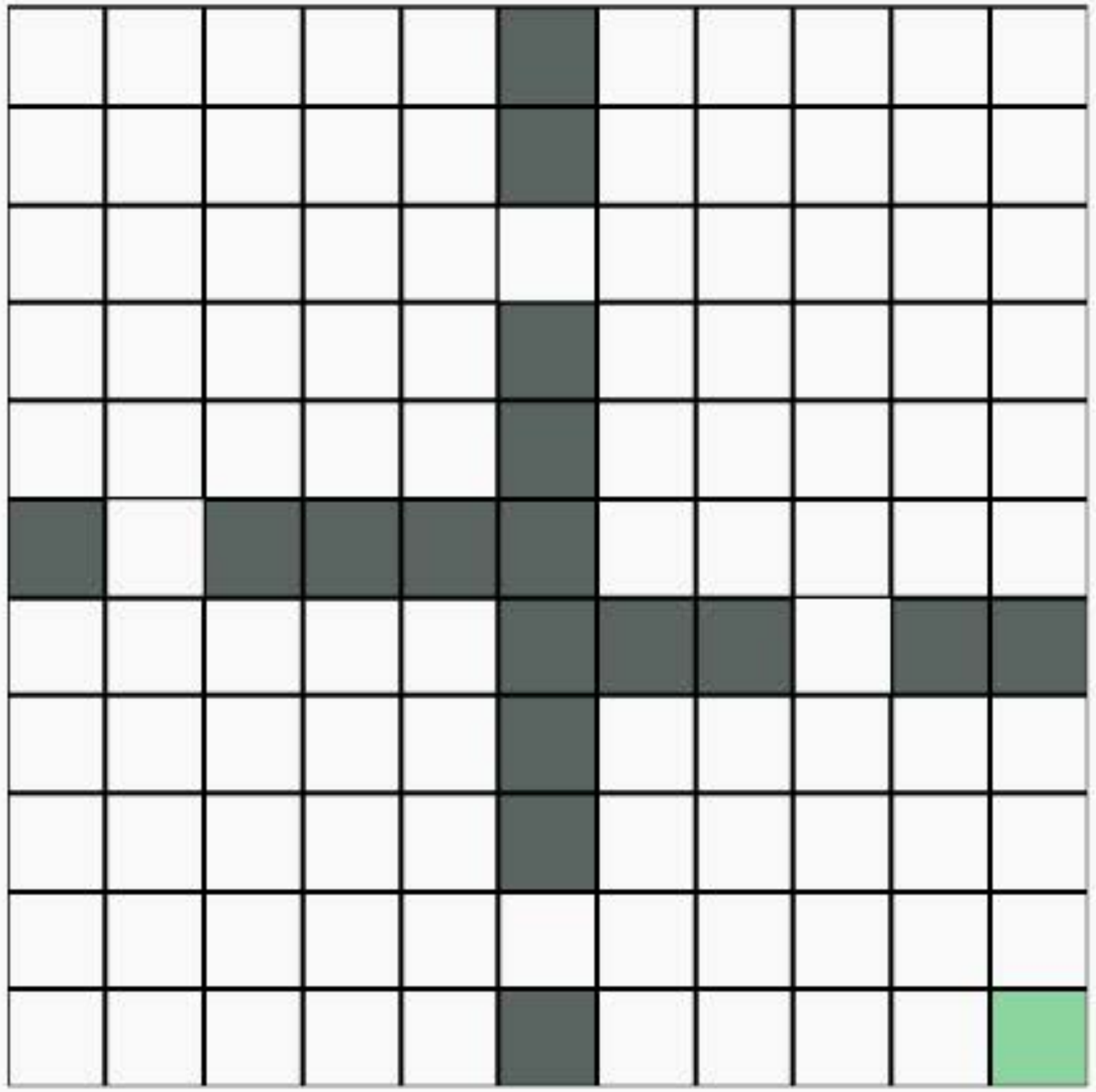
Approximation

Visuals: $K = 4$



Betweenness Options

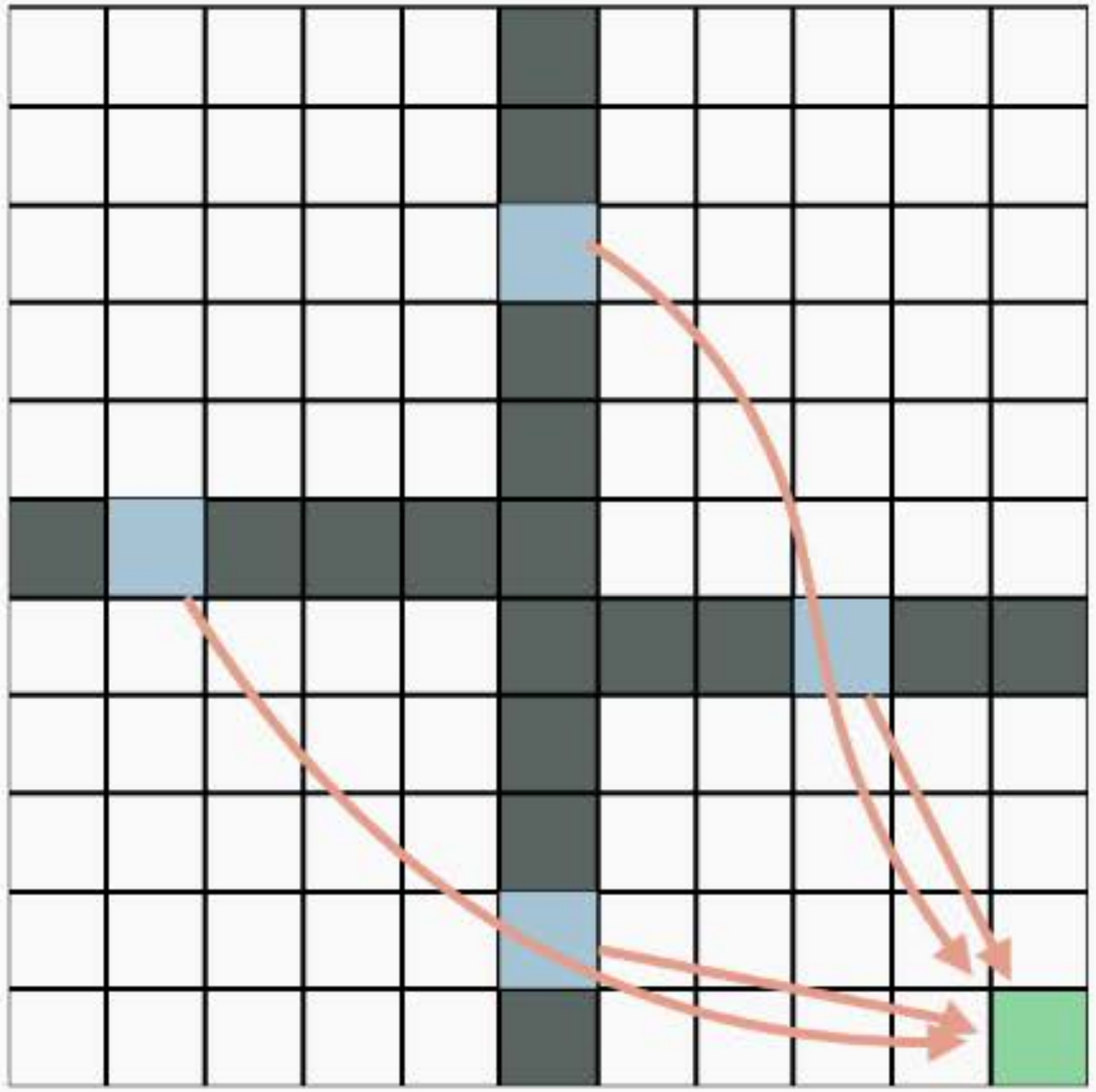
[Simsek and Barto 2005, 2008]



Eigen Options

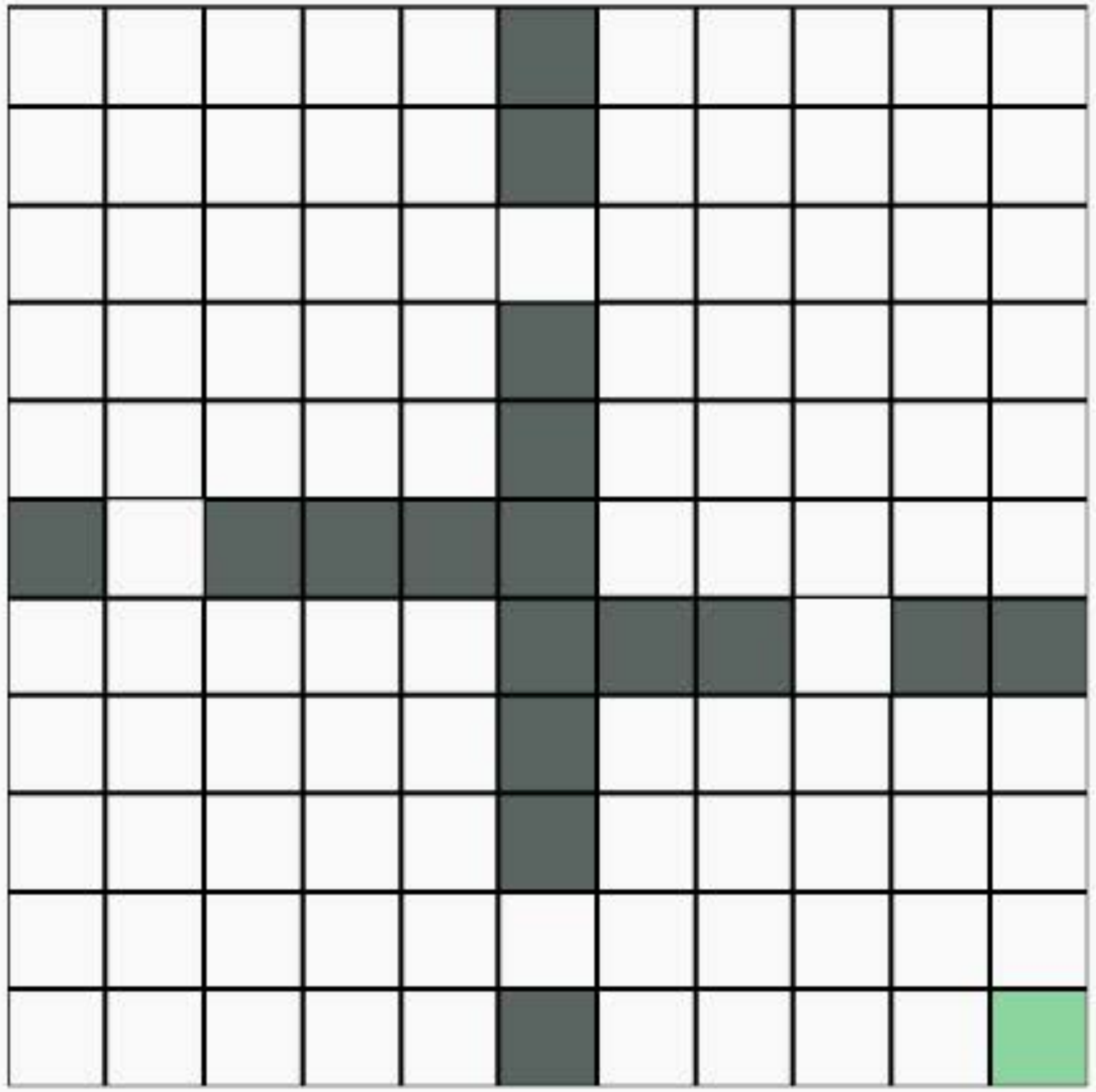
[Machado et al. 2017]

Visuals: $K = 4$



Betweenness Options

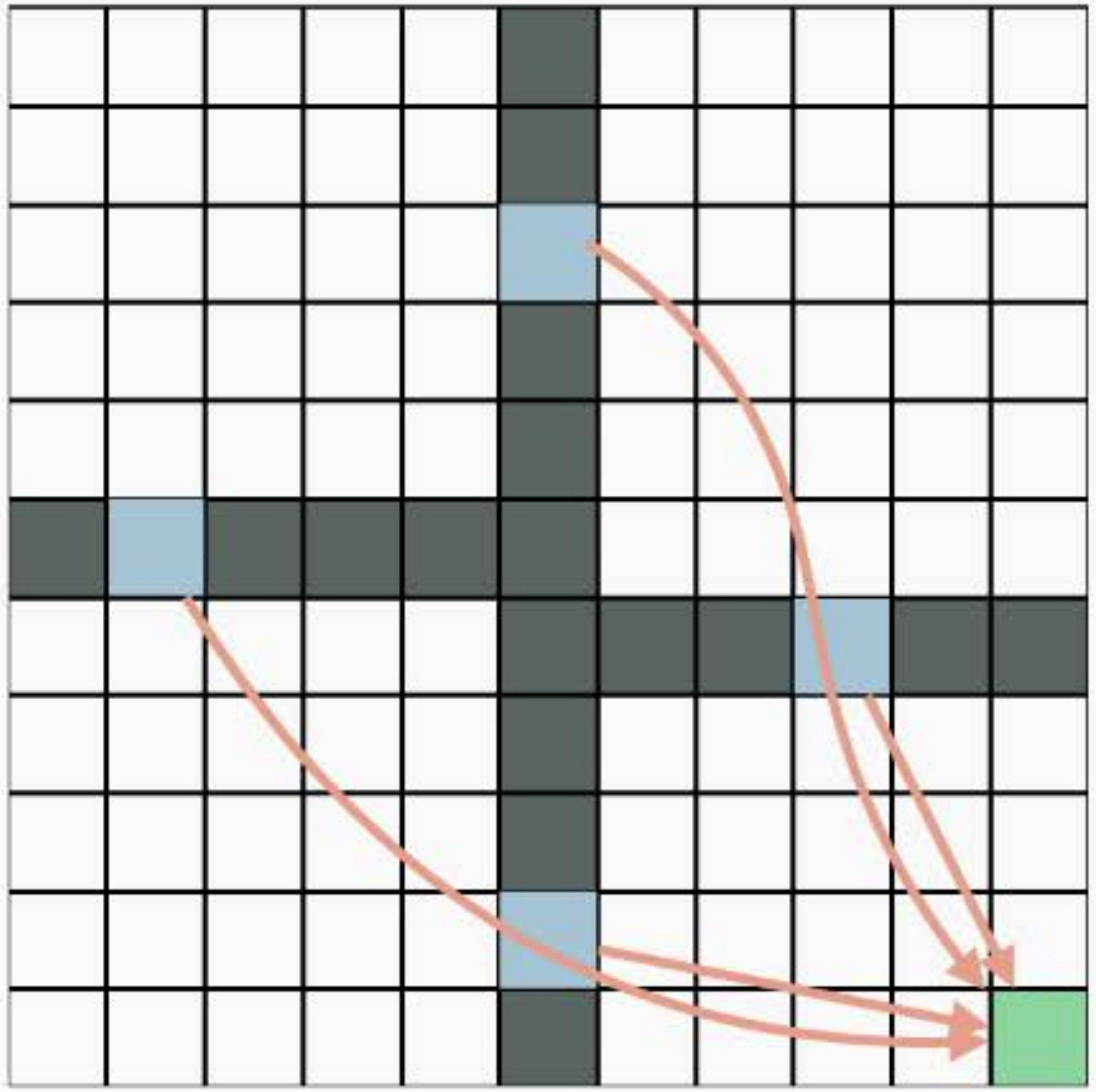
[Simsek and Barto 2005, 2008]



Eigen Options

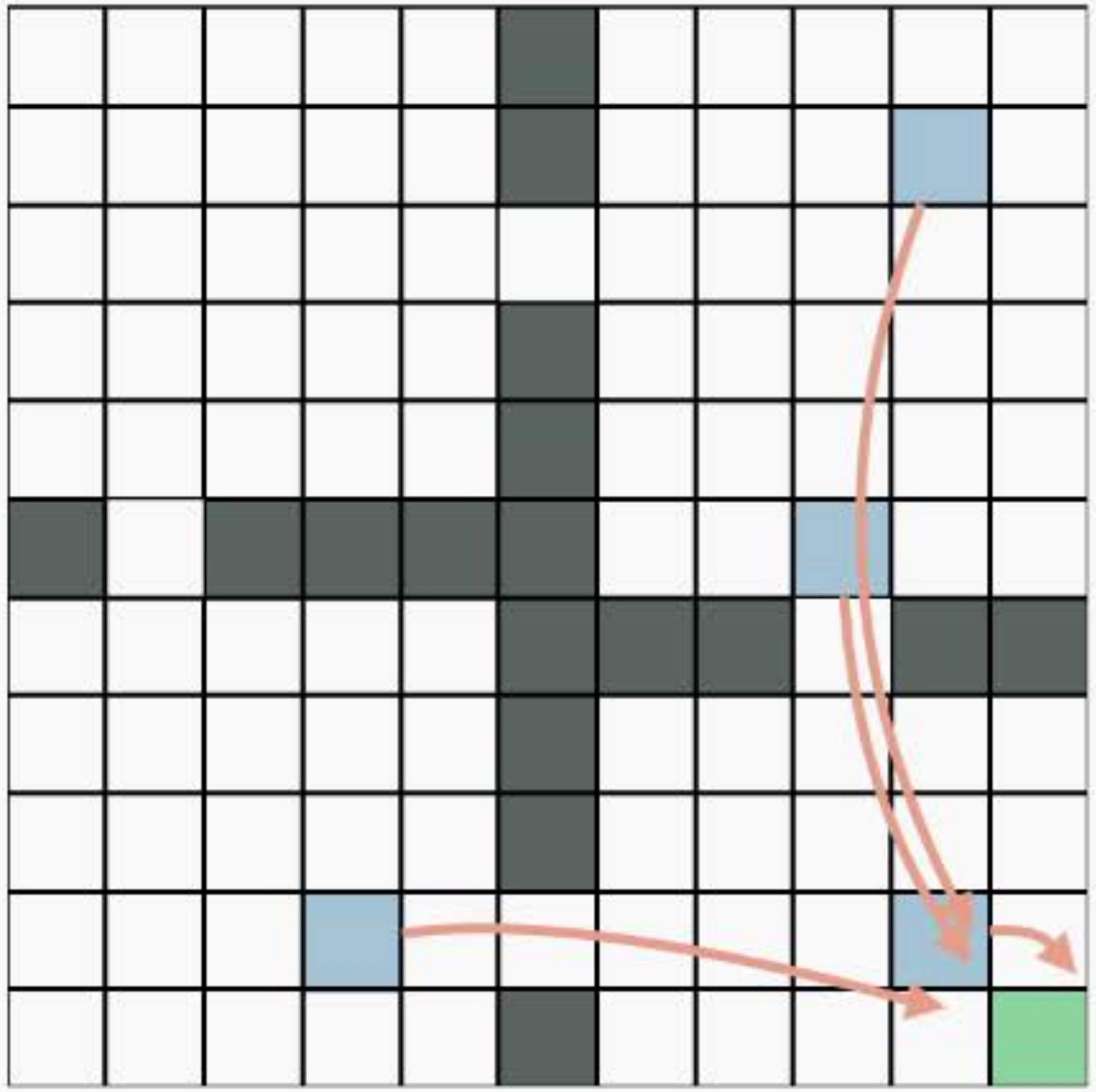
[Machado et al. 2017]

Visuals: $K = 4$



Betweenness Options

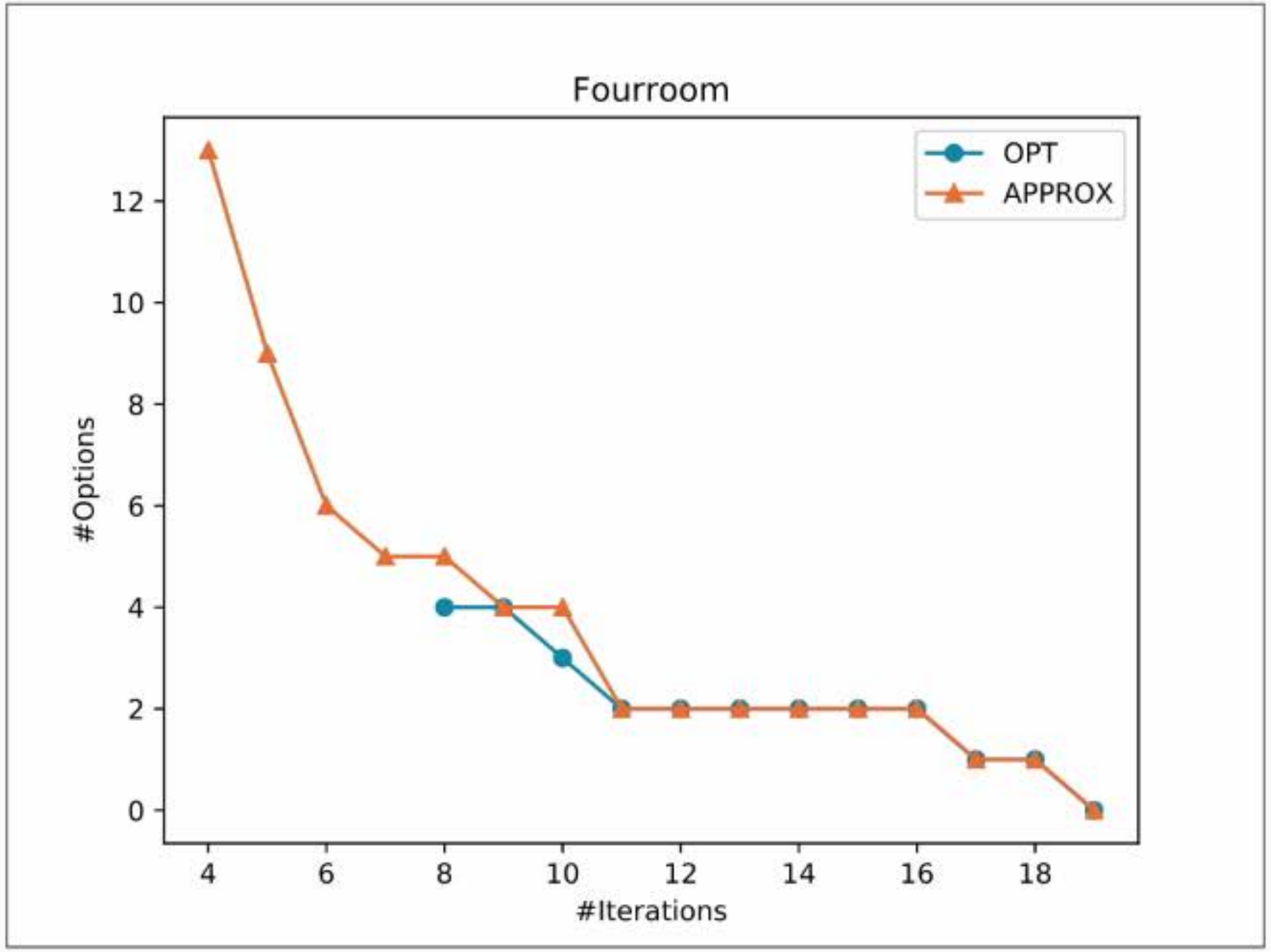
[Simsek and Barto 2005, 2008]



Eigen Options

[Machado et al. 2017]

Evaluation


Fourroom
OPT
APPROX
#Options
#Iterations
0
2
4
6
8
10
12
4
6
8
10
12
14
16
18

Results Summary

Question: *Can we find the set of options that minimize the number of iterations of VI?*

Results Summary

Question: *Can we find the set of options that minimize the number of iterations of VI?*

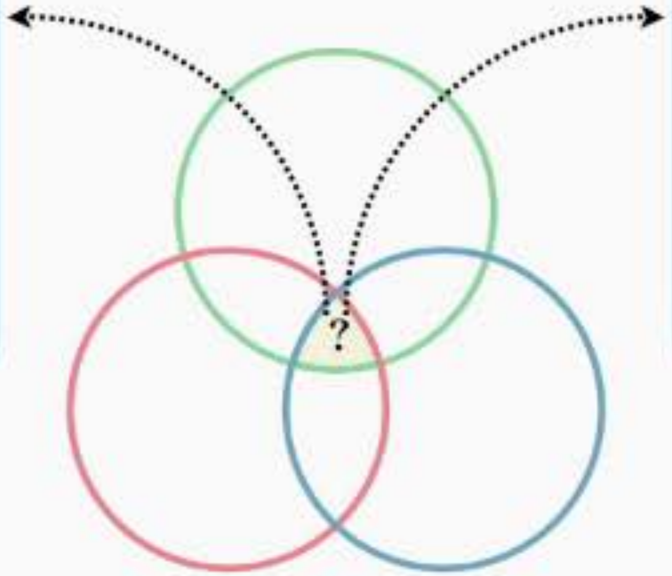
Answer: *Yes, but it takes serious computational work.*

Results Summary

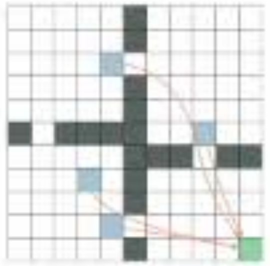
Easy To Construct

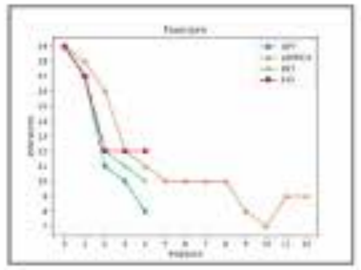
1 **State Abstraction**

*[**A**AAJLW AAAI '19]*



2 **Action Abstraction**





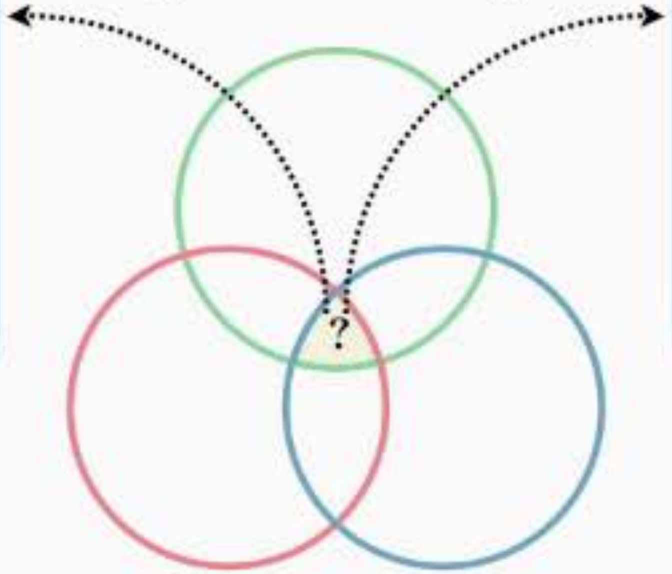
*[J**A**HLK ICML '19]*

Supports Efficient Reinforcement Learning

Preserves Solution Quality

Results Summary

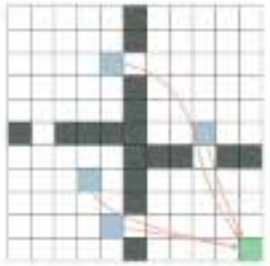
Easy To Construct

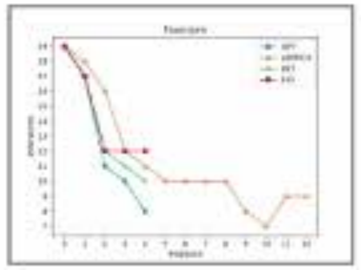


1 State Abstraction

[**A**AAJLW AAAI '19]

2 Action Abstraction



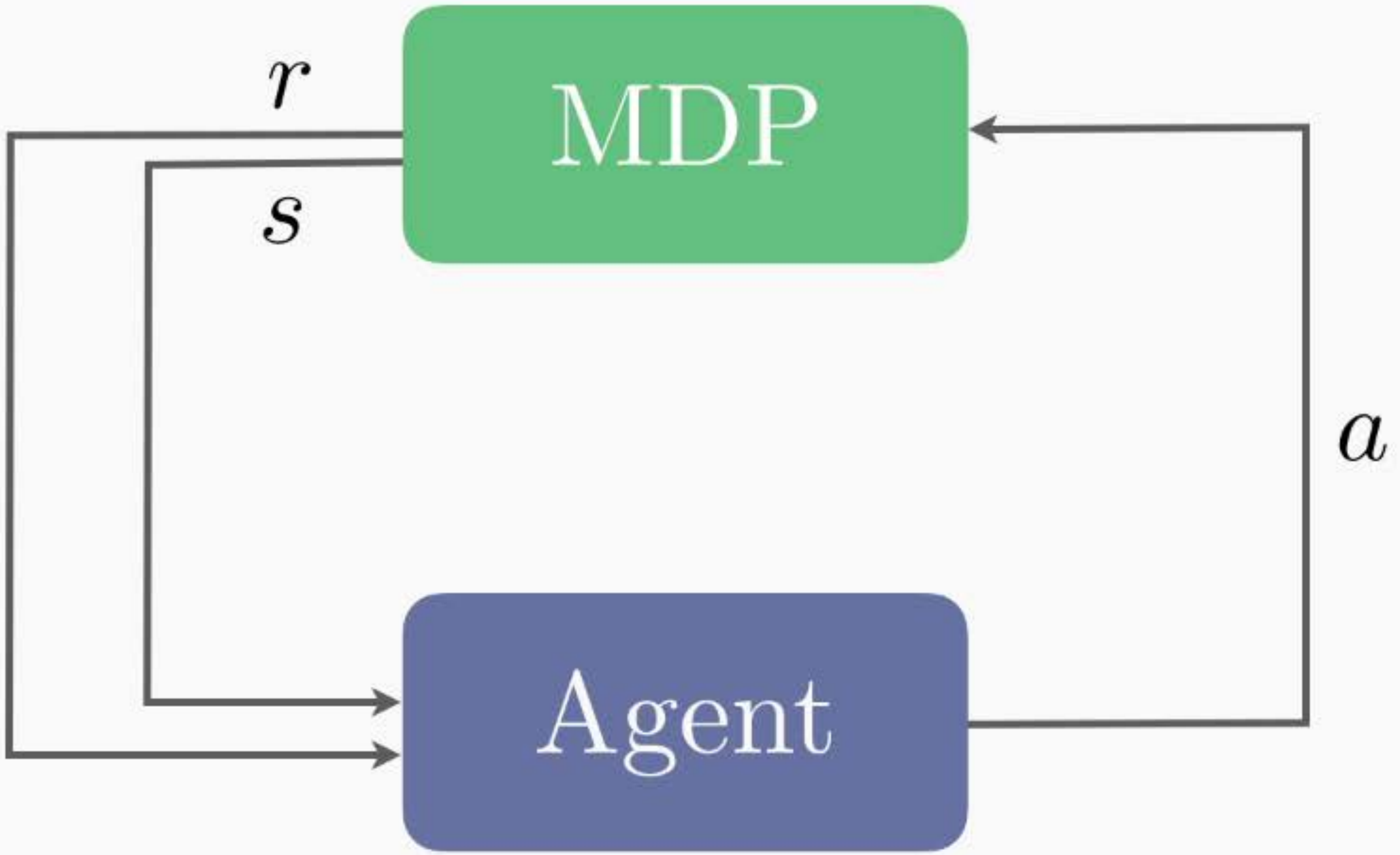


[J**A**HLK ICML '19]

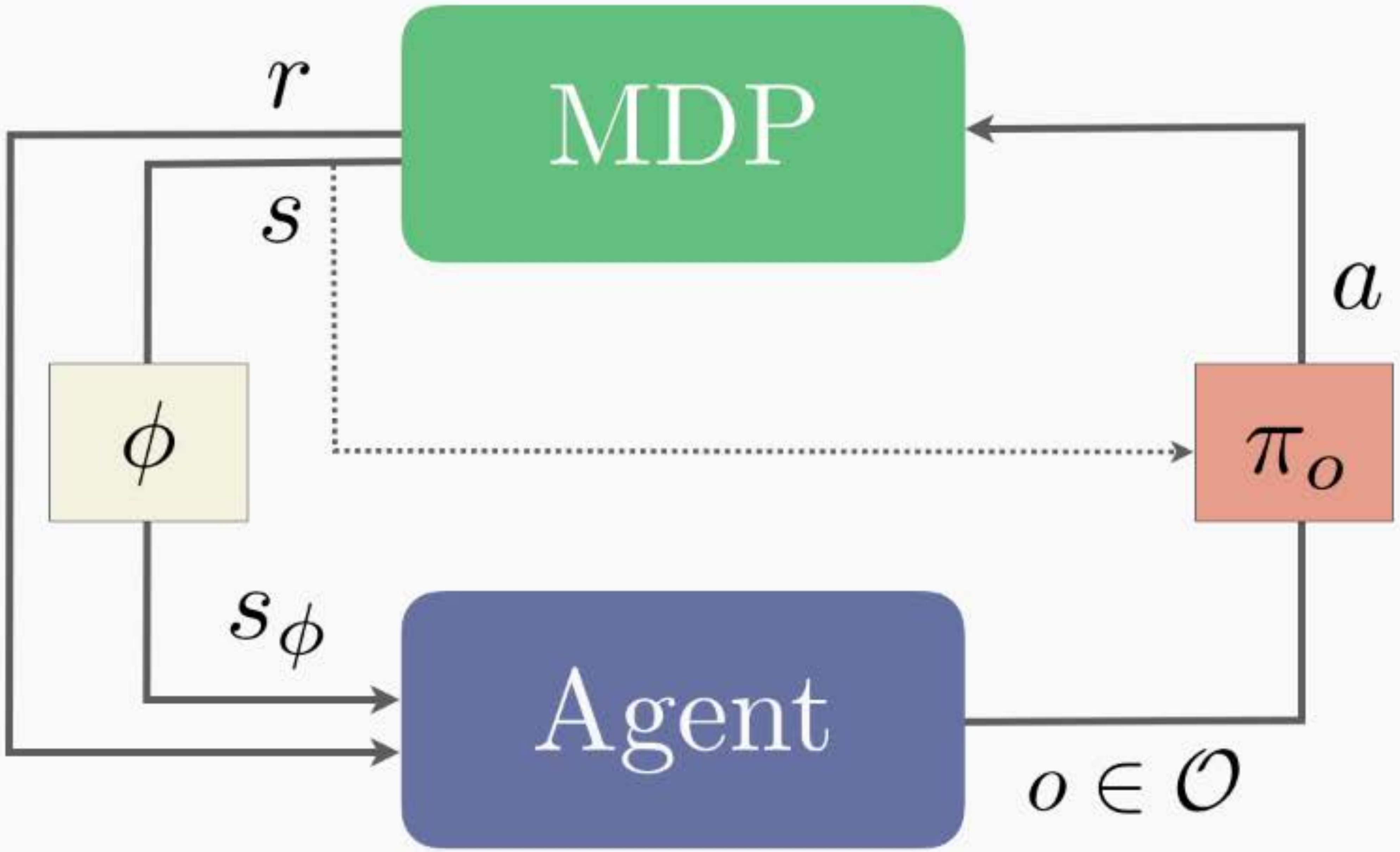
Supports Efficient Reinforcement Learning

Preserves Solution Quality

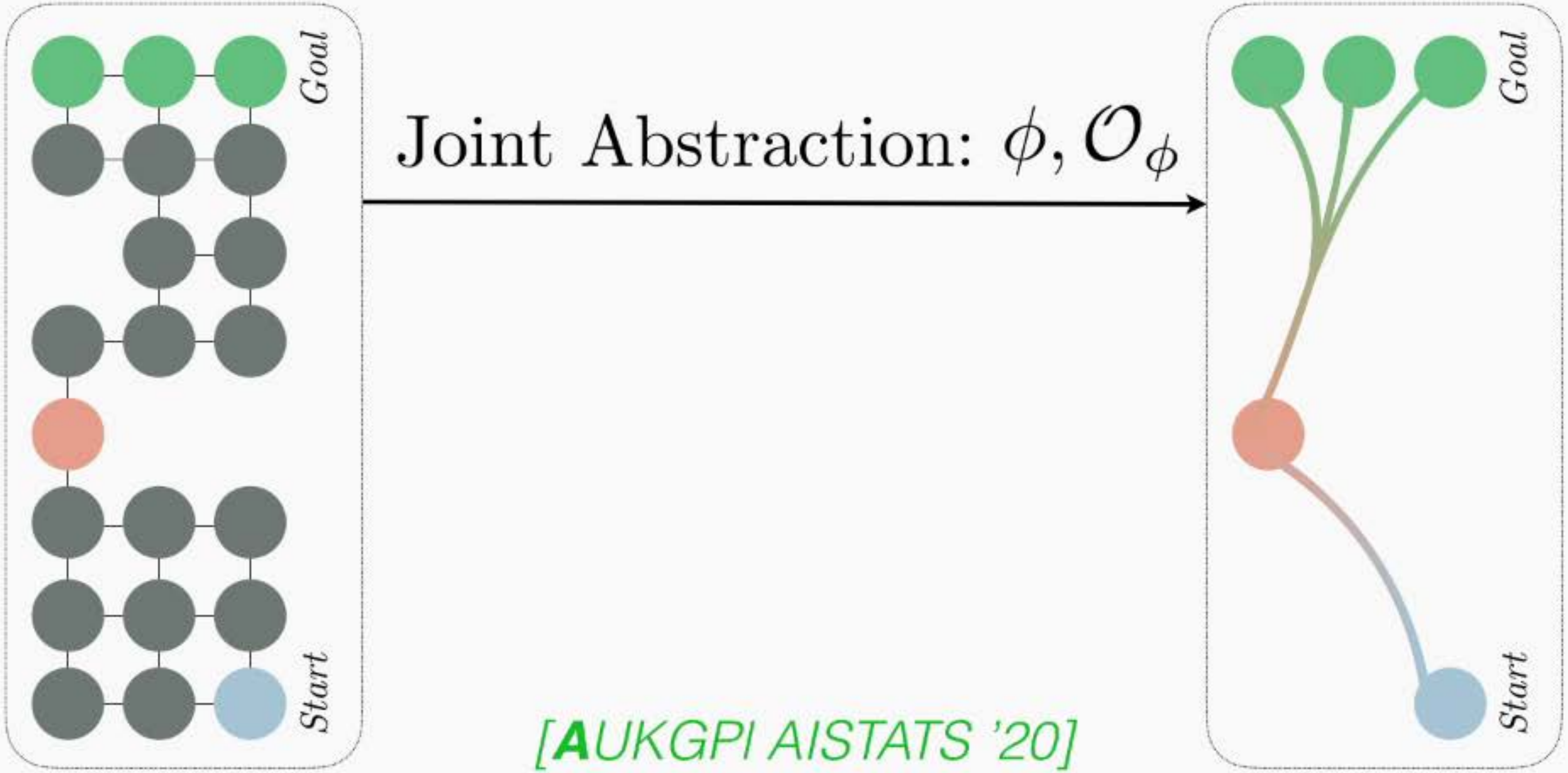
State-Action Abstraction



State-Action Abstraction

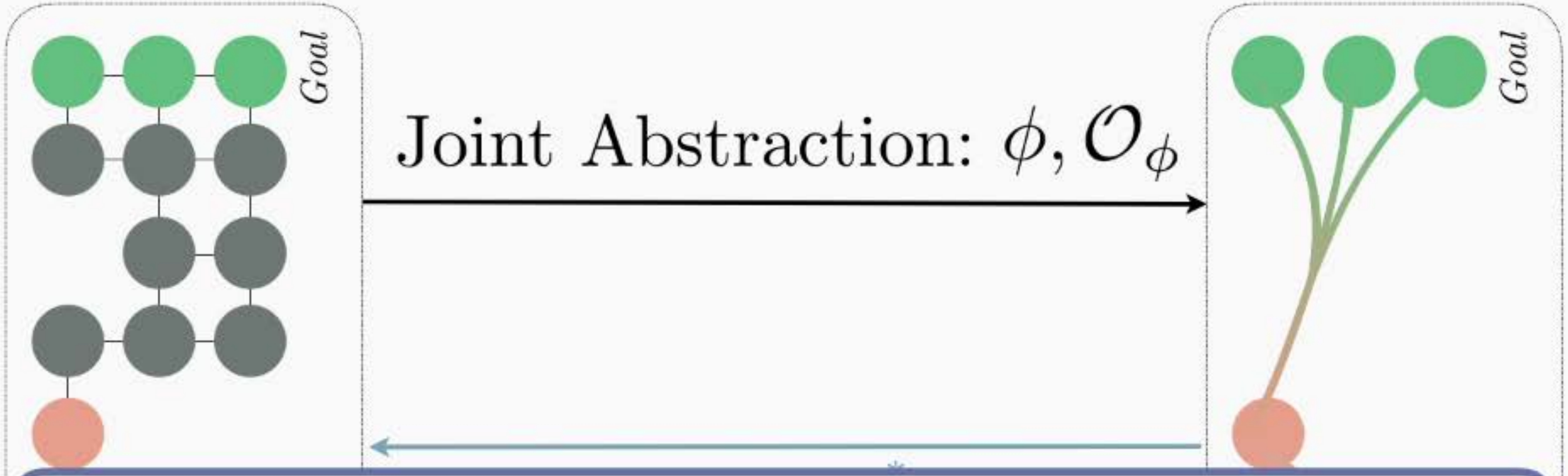


State-Action Abstraction



[**A**UKGPI AISTATS '20]

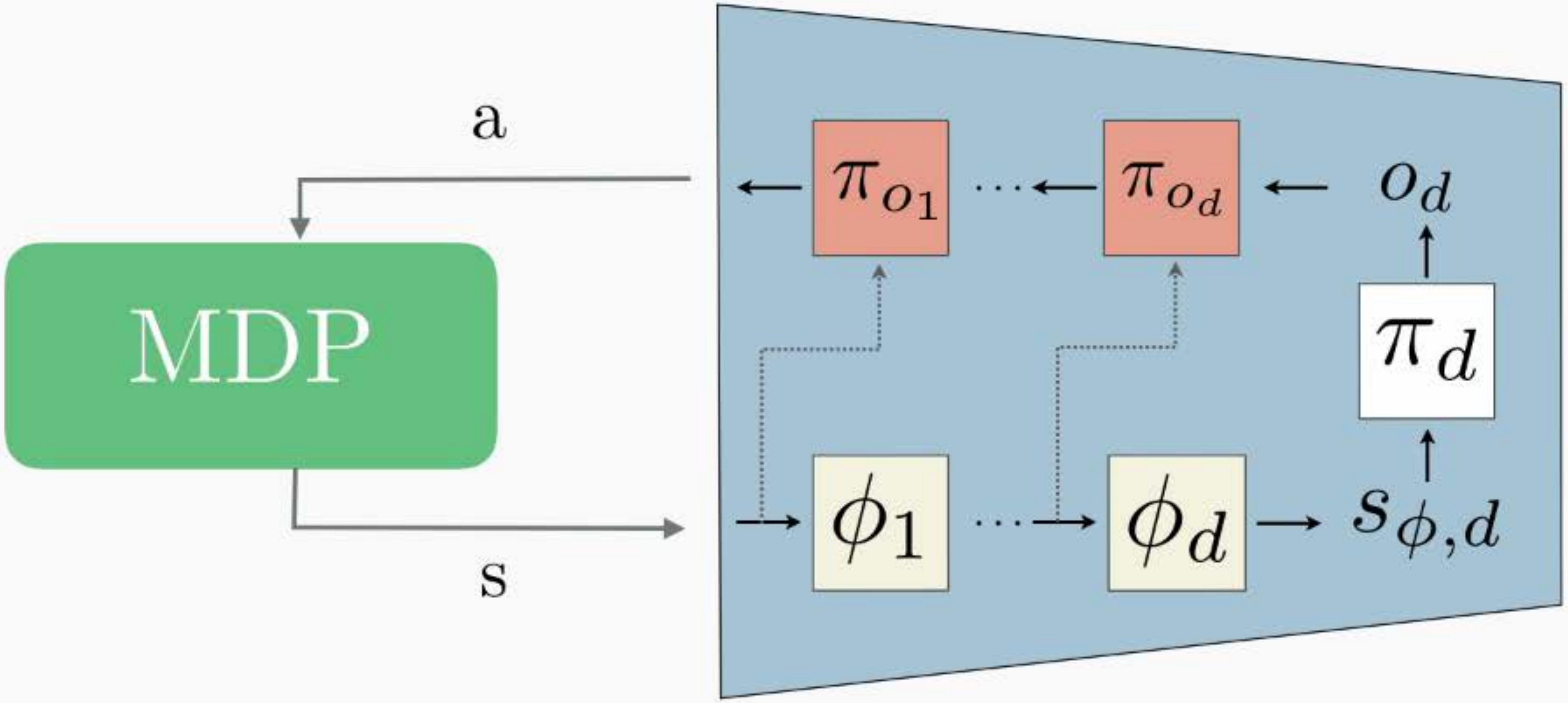
State-Action Abstraction



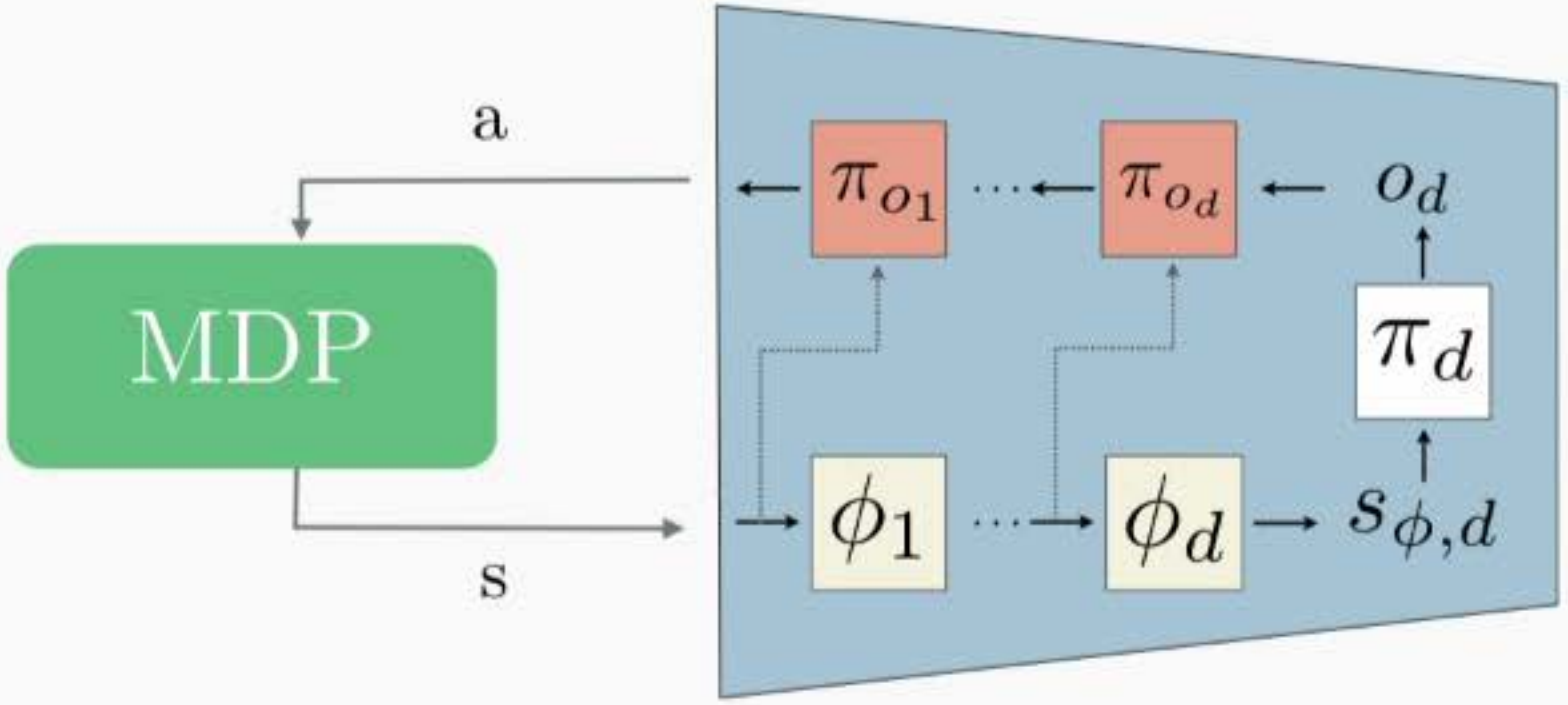
Theorem. Well chosen $\phi, \mathcal{O}_\phi$, preserve value:

$$|V^*(s_0) - V^{\pi^*_{\phi,\mathcal{O}_\phi}}(s_0)| \leq f(\varepsilon_{\mathcal{O}_\phi})$$

Hierarchical Abstraction



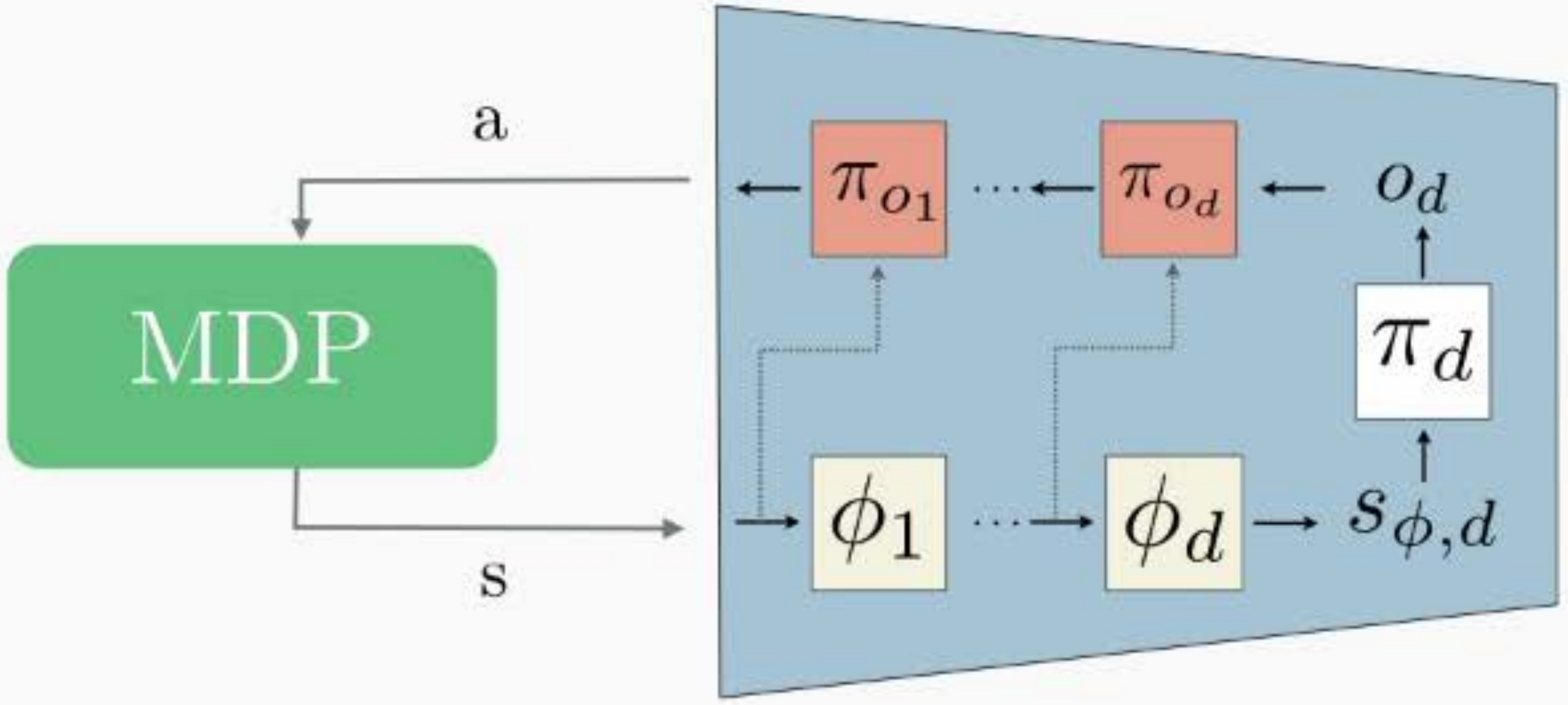
Hierarchical Abstraction



Theorem. Well chosen $\phi, \mathcal{O}_\phi$, preserve value:

$$|V^*(s_0) - V^{\pi_d^*}(s_0)| = O(d)$$

Hierarchical Abstraction



Theorem. Well chosen $\phi, \mathcal{O}_\phi$, preserve value:

$$|V^*(s_0) - V^{\pi_d^*}(s_0)| = O(d)$$

Work Overview

Abstraction in RL

[ICML 2019a]

[AAAI 2019]

[AISTATS 2020]

Work Overview

Abstraction in RL

[ICML 2016] *[ICML 2018a]*

[ICML 2019a] *[ICML 2019b]*

[AAAI 2019] *[IJCAI 2019]*

[Opinions in Behavioral Sci. 2019]

[AISTATS 2020]

Work Overview

Abstraction in RL

[ICML 2016] *[ICML 2018a]*

[ICML 2019a] *[ICML 2019b]*

[AAAI 2019] *[IJCAI 2019]*

[Opinions in Behavioral Sci. 2019]

[AISTATS 2020]

Human Cognition

[CCN 2019] *[AAAI 2020]*

Work Overview

Abstraction in RL

[ICML 2016] *[ICML 2018a]*

[ICML 2019a] *[ICML 2019b]*

[AAAI 2019] *[IJCAI 2019]*

[Opinions in Behavioral Sci. 2019]

[AISTATS 2020]

Multitask Learning

[ICAPS 2015] *[ICML 2018b]*

Human Cognition

[CCN 2019] *[AAAI 2020]*

Work Overview

Abstraction in RL

[ICML 2016] *[ICML 2018a]*
[ICML 2019a] *[ICML 2019b]*
[AAAI 2019] *[IJCAI 2019]*
[Opinions in Behavioral Sci. 2019]
[AISTATS 2020]

Human Cognition

[CCN 2019] *[AAAI 2020]*

Multitask Learning

[ICAPS 2015] *[ICML 2018b]*

Computational Sustainability

[EnviroInfo 2017]
[RLDM 2017]
[IAAI 2018]

Work Overview

Abstraction in RL

[ICML 2016] *[ICML 2018a]*
[ICML 2019a] *[ICML 2019b]*
[AAAI 2019] *[IJCAI 2019]*
[Opinions in Behavioral Sci. 2019]
[AISTATS 2020]

Human Cognition

[CCN 2019] *[AAAI 2020]*

Multitask Learning

[ICAPS 2015] *[ICML 2018b]*

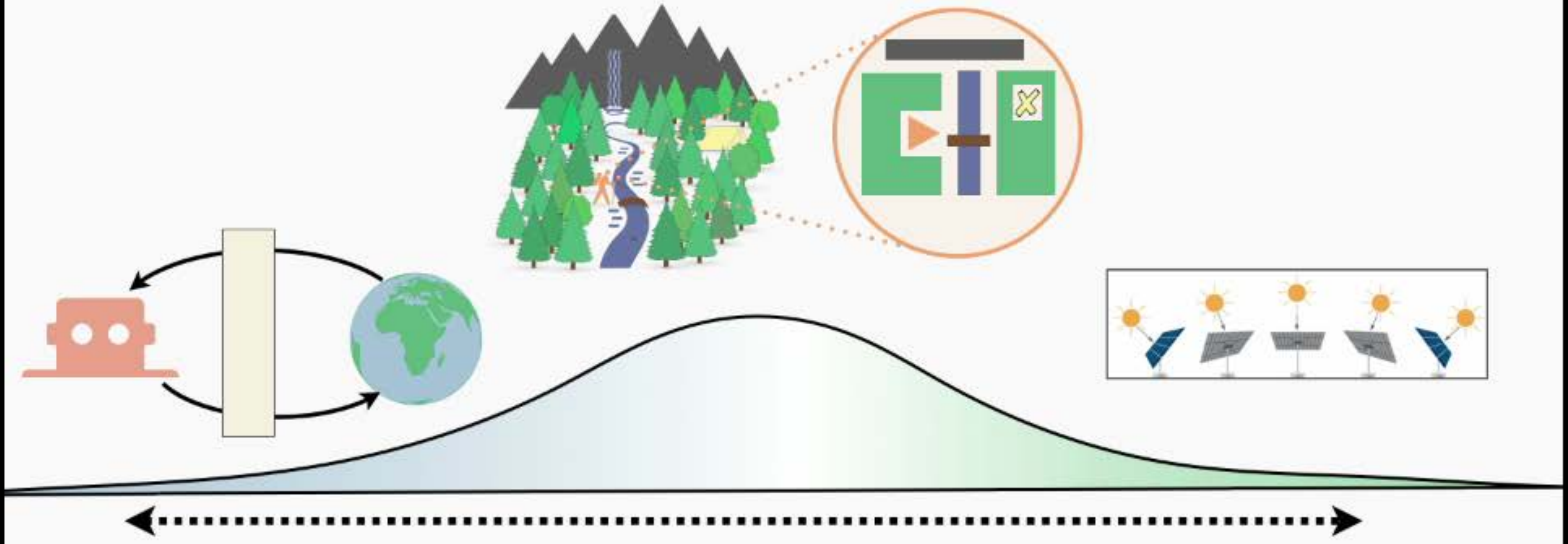
Computational Sustainability

[EnviroInfo 2017]
[RLDM 2017]
[IAAI 2018]

Philosophy of AI

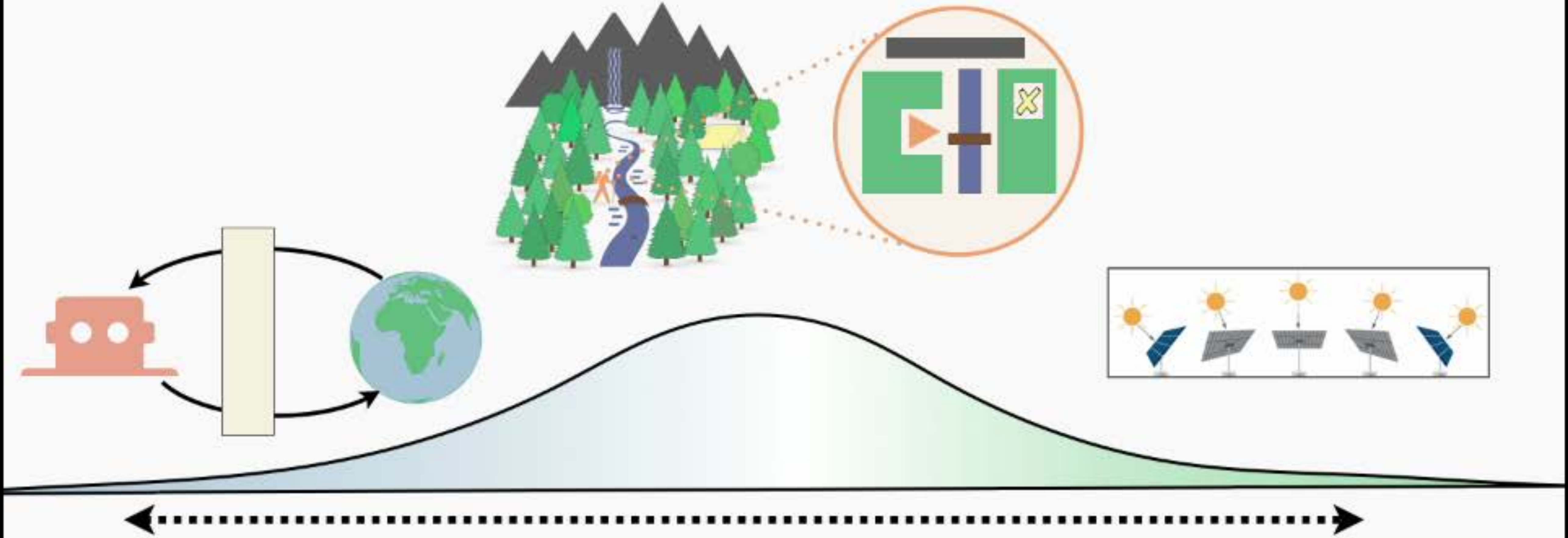
[AIES Workshop 2016]
[IACAP 2019]

Agenda



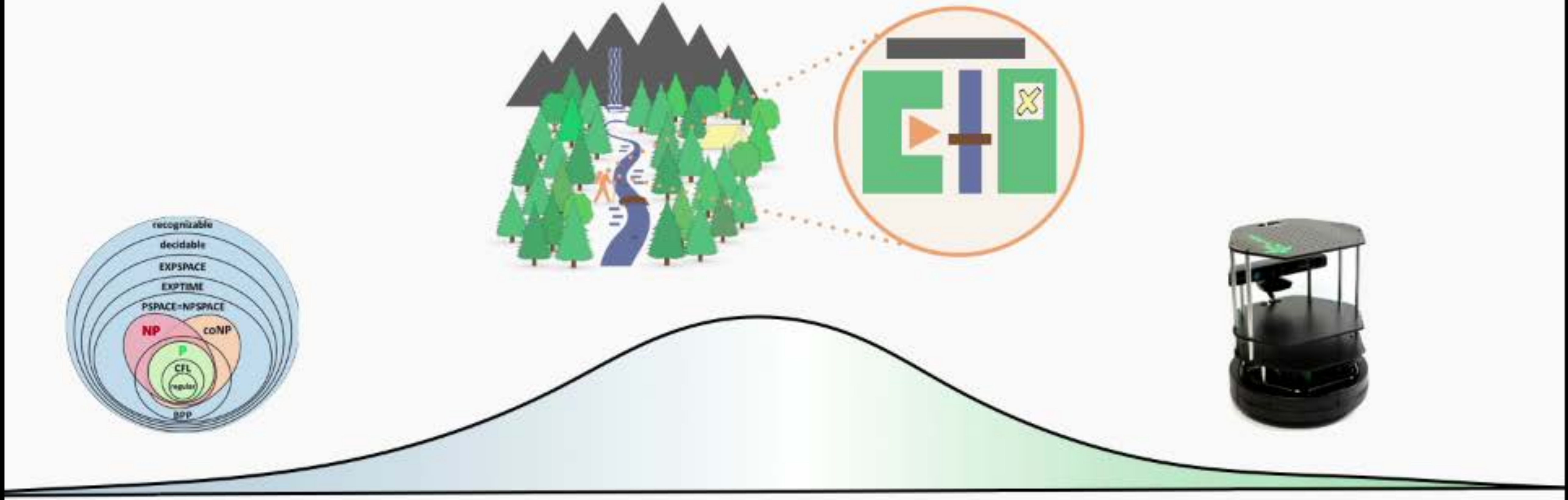
Theory Practice

Agenda



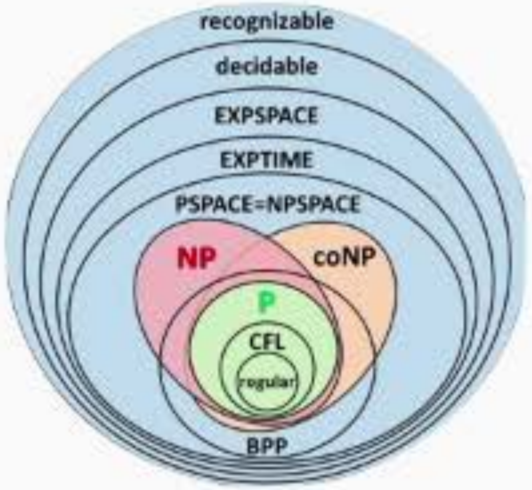
Guiding Question: *How can we ground computational problem solving to observation and action, rather than symbols?*

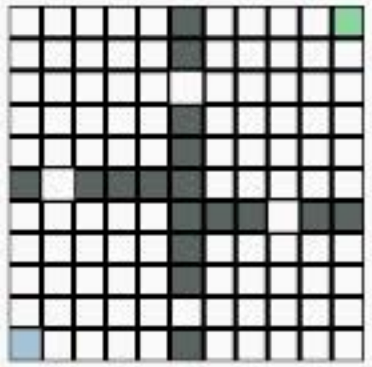
Future Agenda



Guiding Question: *How can we ground computational problem solving to observation and action, rather than symbols?*

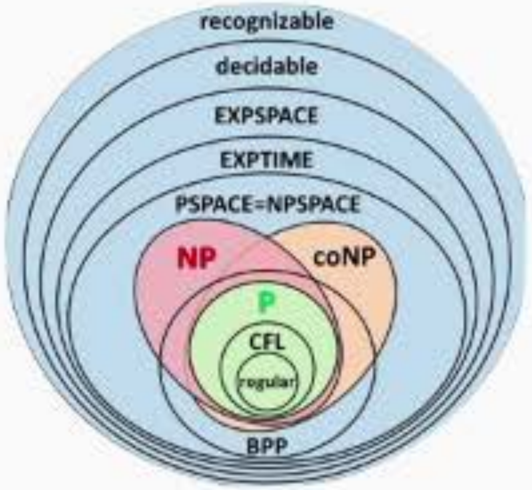
Future Agenda

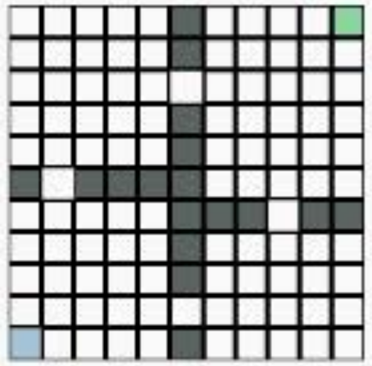




Guiding Question: *How can we ground computational problem solving to observation and action, rather than symbols?*

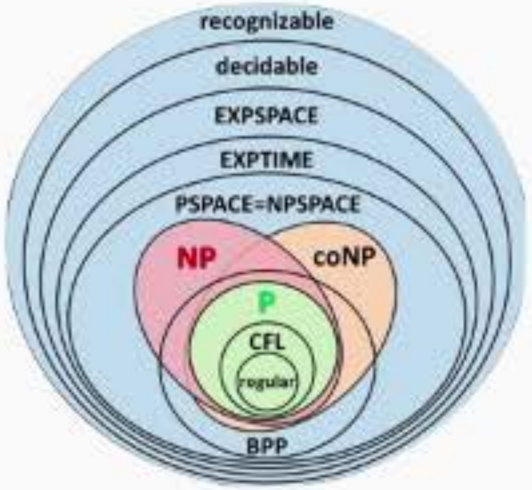
Future Agenda

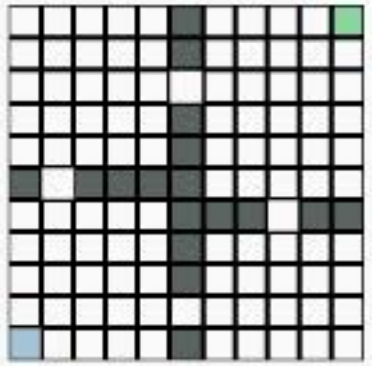




Guiding Question: *How can we ground computational problem solving to observation and action, rather than symbols?*

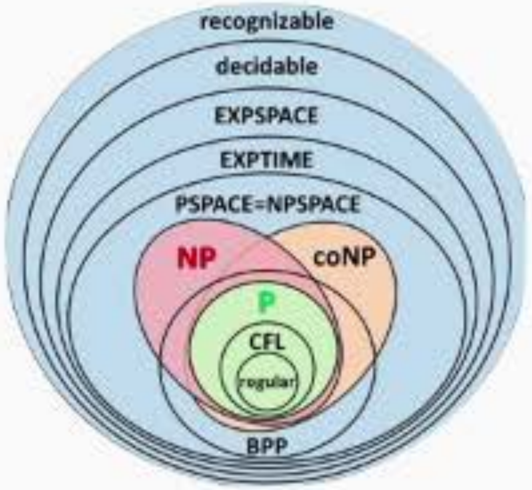
Future Agenda

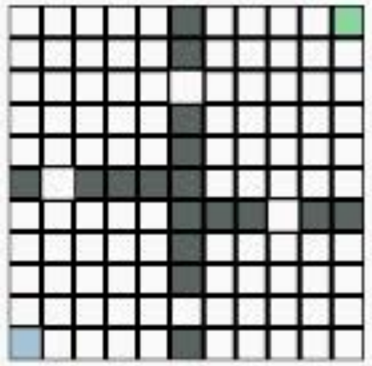




Guiding Question: *How can we ground computational problem solving to observation and action, rather than symbols?*

Future Agenda

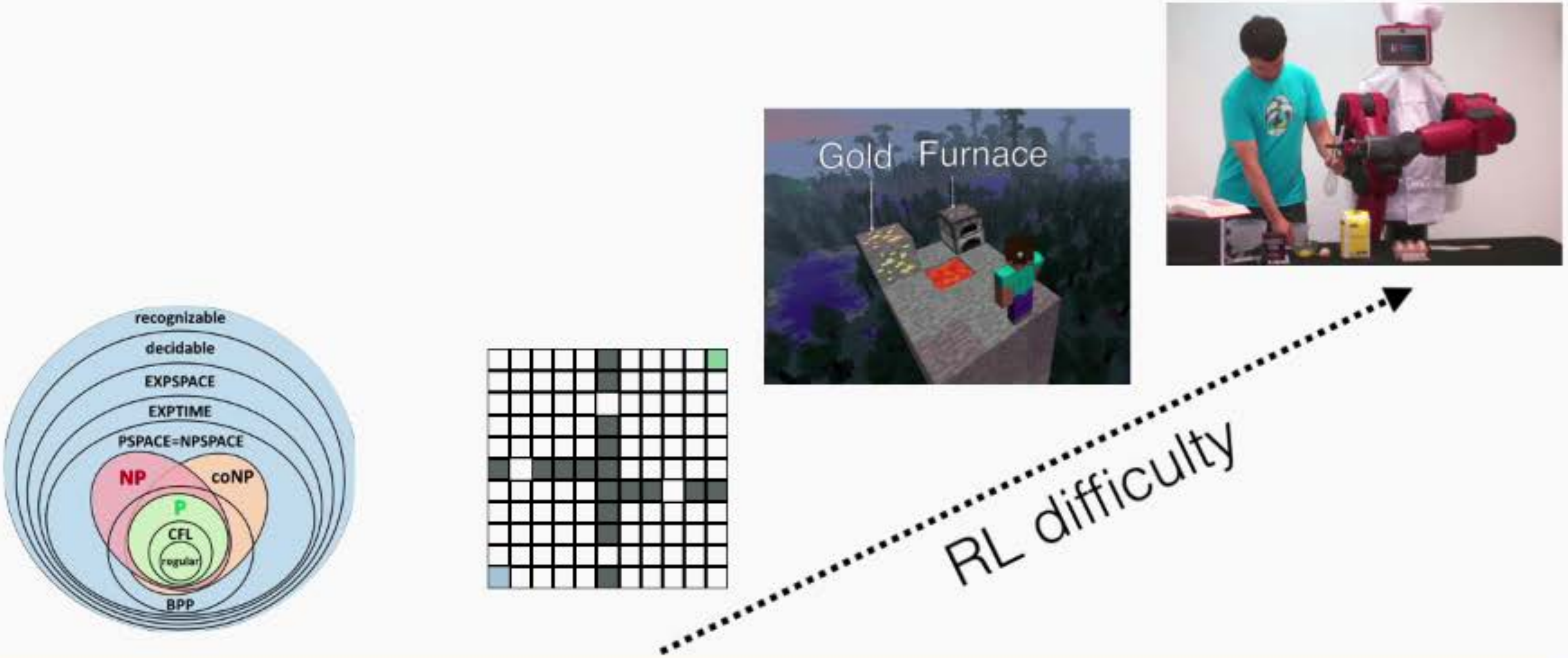




RL difficulty

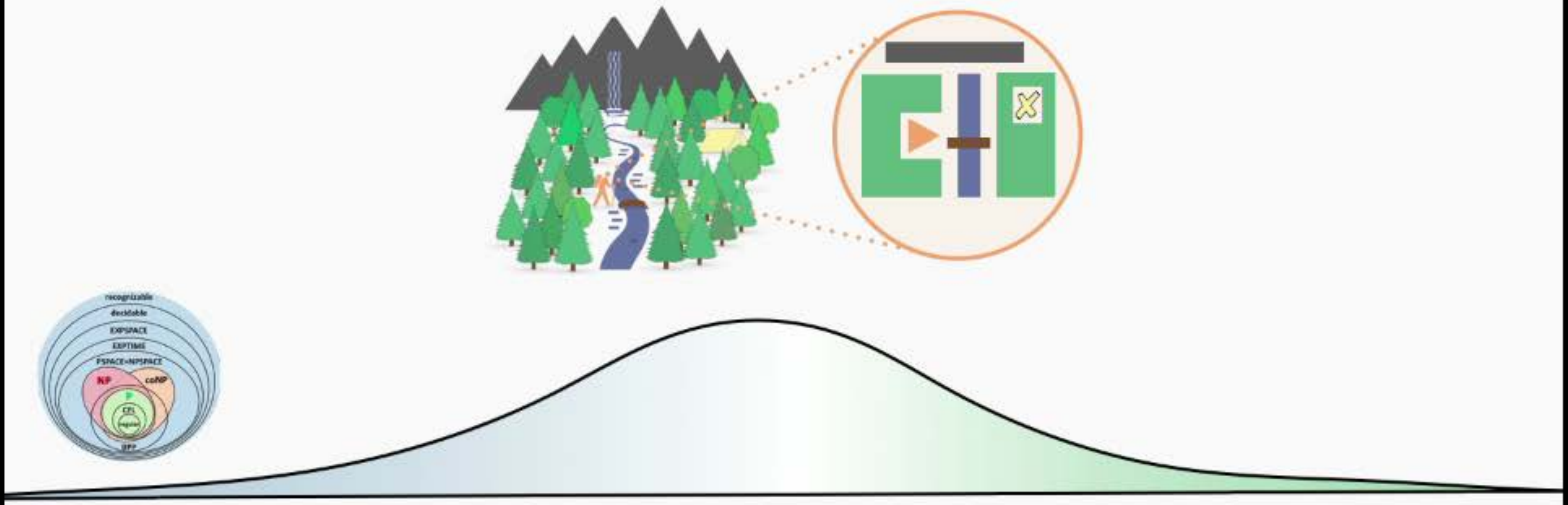
Guiding Question: *How can we ground computational problem solving to observation and action, rather than symbols?*

Future Agenda



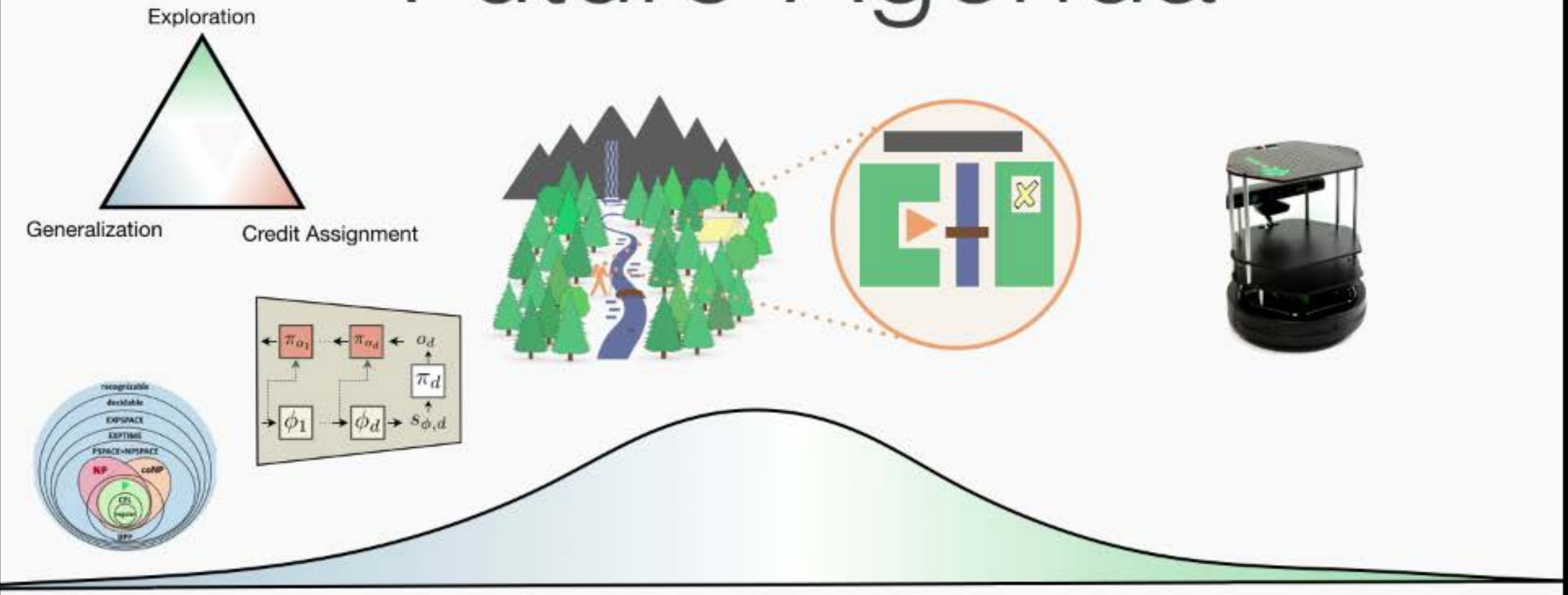
Guiding Question: *How can we ground computational problem solving to observation and action, rather than symbols?*

Future Agenda



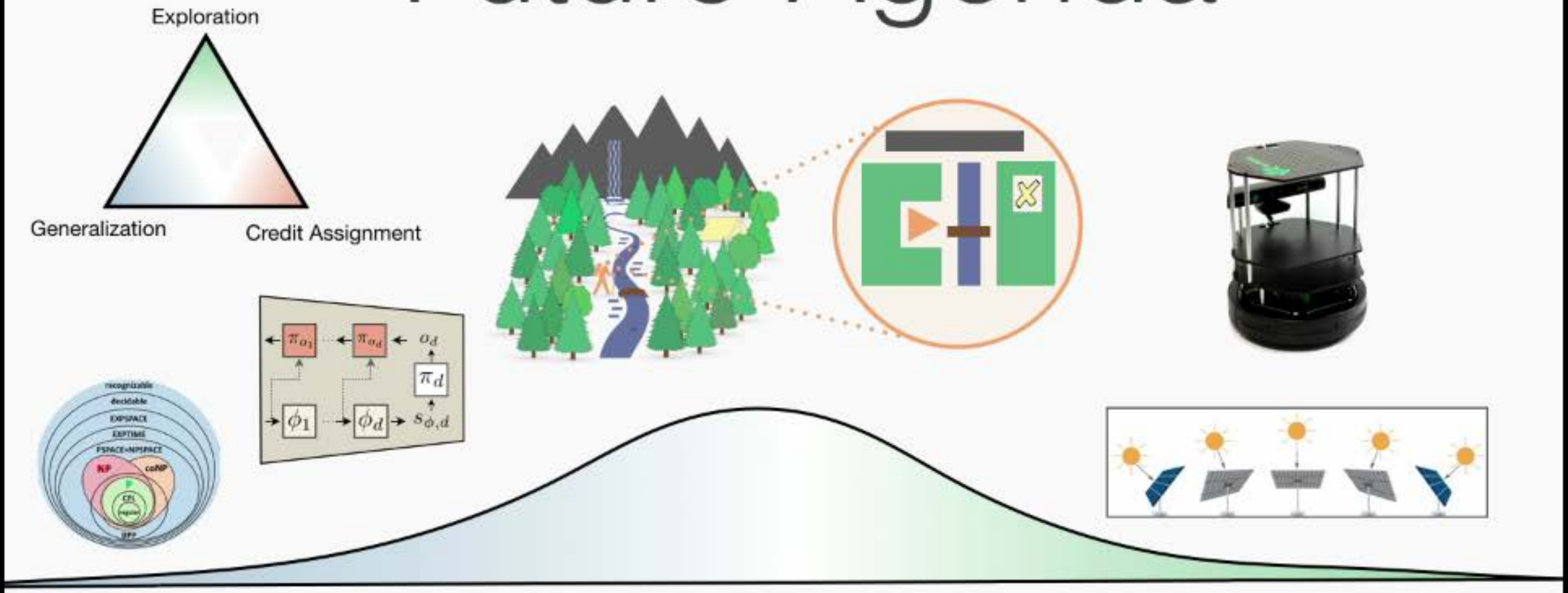
Guiding Question: *How can we ground computational problem solving to observation and action, rather than symbols?*

Future Agenda



Guiding Question: *How can we ground computational problem solving to observation and action, rather than symbols?*

Future Agenda



Guiding Question: *How can we ground computational problem solving to observation and action, rather than symbols?*

Thanks to Advisors!

Masters *PhD* *Undergrad*

Joshua Schechter

Stefanie Tellex

Michael L. Littman

David Liben-Nowell

Ana Moltchanova

George Konidaris

Peter Stone

Will Dabney

Fernando Diaz

Owain Evans

Committee *Internships*

Thanks to Collaborators!


Alekh Agarwal


Dilip Arumugam


Kavosh Asadi


Gabriel Barth-Maron


Stephen Brawner


Marie desJardins


Tom Griffiths


Yue Guo


D. Ellis Hershkowitz


Mark Ho


Yuu Jinnai


Khimya Khetarpal


Akshay Krishnamurthy


Lucas Lehnert


James MacGlashan


Doina Precup


Daniel Reichman


Emily Reif


John Salvatier


Robert Schapire


Andreas Stuhlmüeller


Nathan Umbanhower


Edward Williams


John Winder

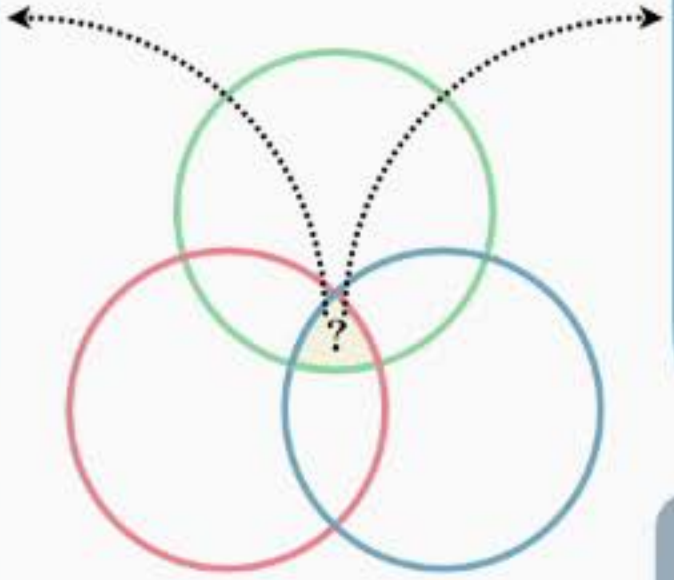

Lawson Wong

Summary

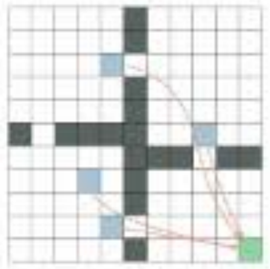
1 State Abstraction

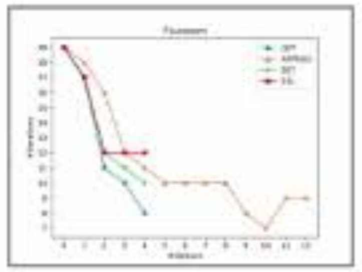
*[**A**AAJLW AAAI '19]*

Easy To Construct



2 Action Abstraction





*[J**A**HLK ICML '19]*

Supports Efficient Reinforcement Learning

Preserves Solution Quality

Contact: david_abel@brown.edu

Code: github.com/david_abel/simple_rl