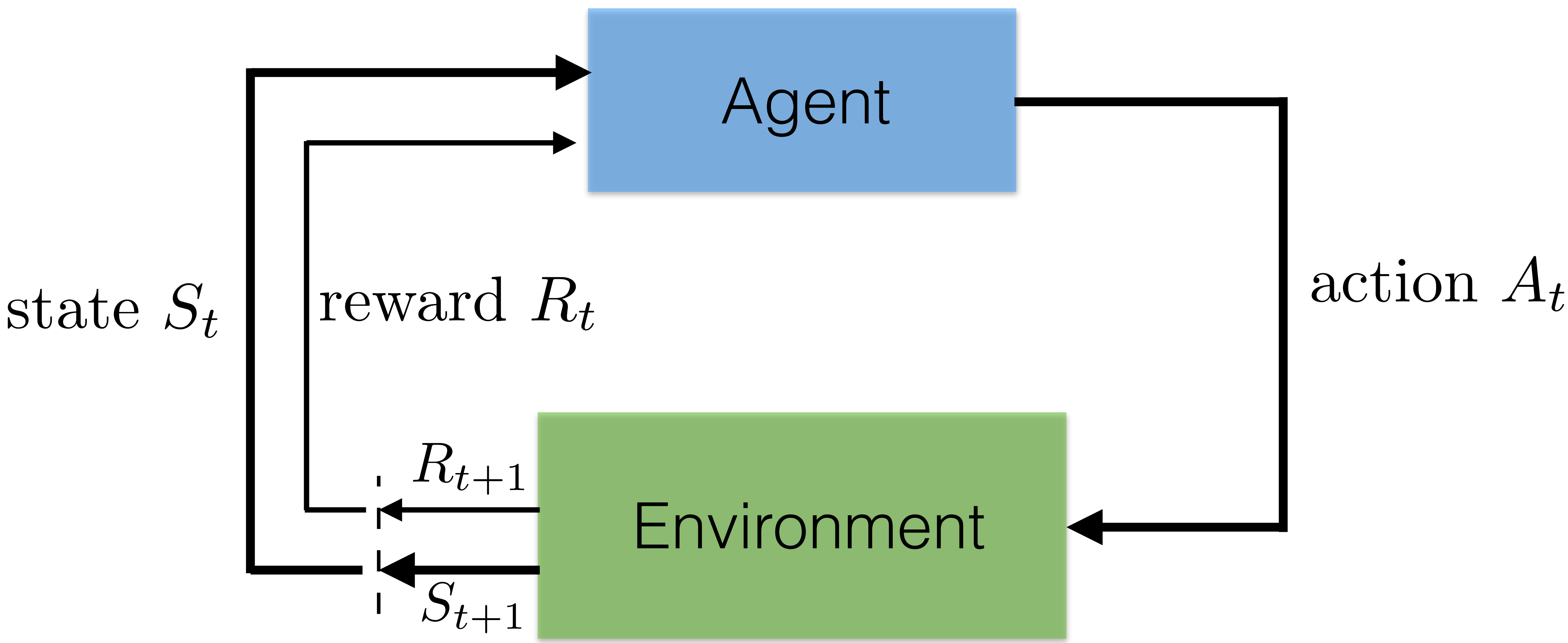
Unifying task specification for reinforcement learning

Martha White
Assistant Professor
University of Alberta

RL & AI

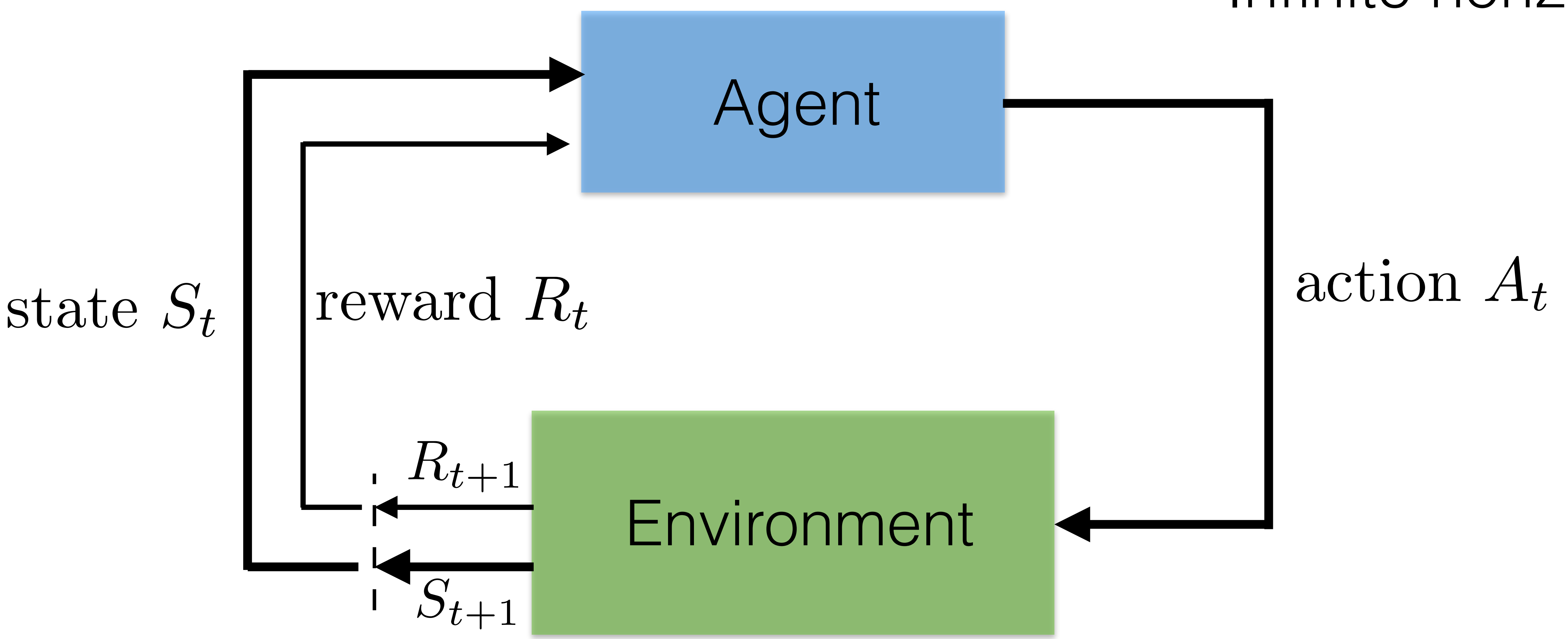
amii

Problem setting



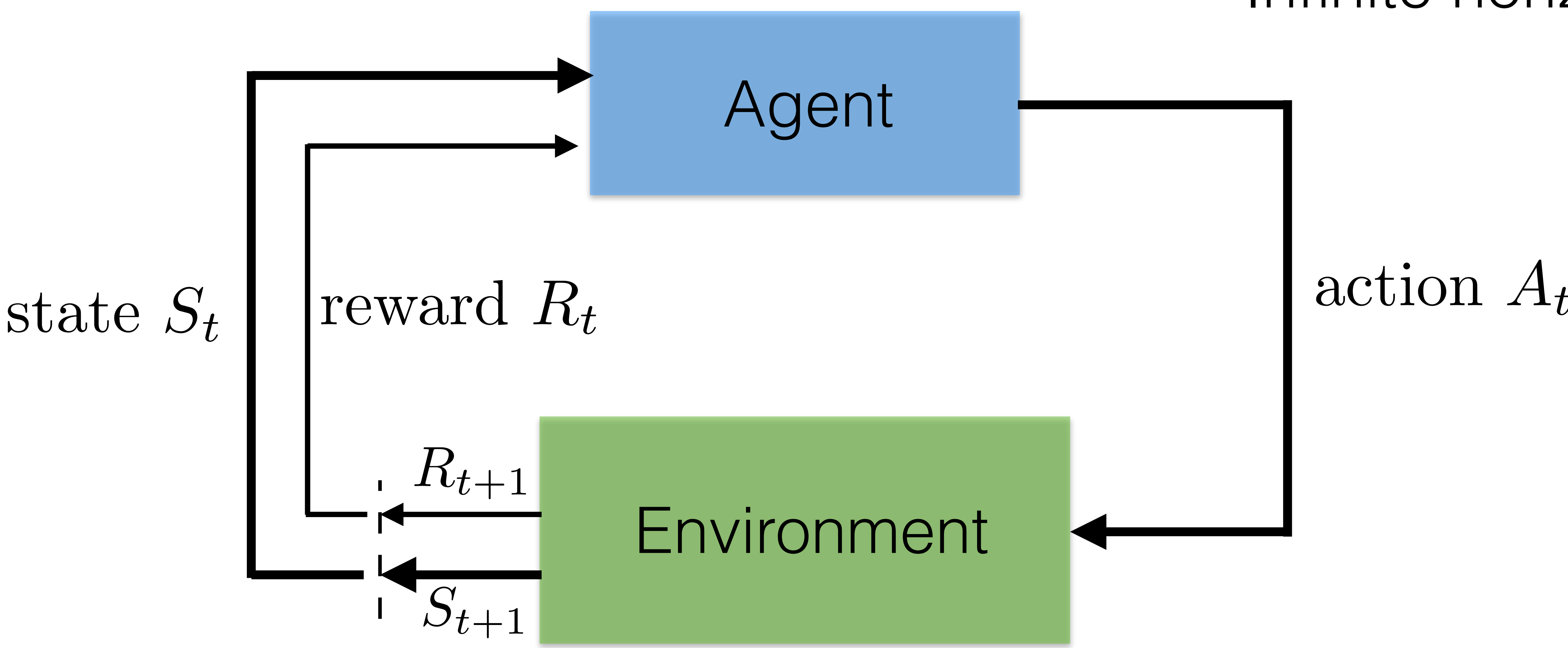
Problem setting

Infinite horizon



Problem setting

Infinite horizon



$\|$

Markov decision process: $(\mathcal{S}, \mathcal{A}, \mathrm{Pr}, \mathrm{R}, \gamma_c)$

What is this talk about?

$$\gamma_c \in [0, 1)$$

What is this talk about?

$$\gamma_c \in [0,1) \longrightarrow \gamma : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0,1]$$

Main take-away

Main take-away

- **Transition-based discounting** is useful for *you*

Main take-away

- **Transition-based discounting** is useful for *you*
    - to simplify algorithm development

Main take-away

- **Transition-based discounting** is useful for *you*
  - to simplify algorithm development
  - to unify theoretical characterizations

Main take-away

- **Transition-based discounting** is useful for *you*
  - to simplify algorithm development
  - to unify theoretical characterizations
  - to simplify implementation

Outline

- Generalization to transition-based discounting
- The theoretical and algorithmic implications
  - generalized Bellman operators
- Utility of the generalized problem formalism

Returns (continuing)

$$G_t = \sum_{i=0}^{\infty} \gamma_c^i R_{t+1+i} = R_{t+1} + \gamma_c G_{t+1}$$

$$V(s) = \mathbb{E}[G_t \mid S_t = s]$$

$$\gamma_c \in [0, 1)$$

Returns (continuing)

$$G_t = \sum_{i=0}^{\infty} \gamma_c^i R_{t+1+i} = R_{t+1} + \gamma_c G_{t+1}$$

$$V(s) = \mathbb{E}[G_t \mid S_t = s]$$

$$\gamma_c \in [0, 1)$$

$$s_0, a_0, r_1, s_1, a_1, r_2, s_2, a_2, r_3, s_3, a_3, \ldots$$

Returns (continuing)

$$G_t = \sum_{i=0}^{\infty} \gamma_c^i R_{t+1+i} = R_{t+1} + \gamma_c G_{t+1}$$

$$V(s) = \mathbb{E}[G_t \mid S_t = s]$$

$$\gamma_c \in [0, 1)$$

$$s_0, a_0, r_1, s_1, a_1, r_2, s_2, a_2, r_3, s_3, a_3, \ldots$$

$$\downarrow$$

$$r_1 + \gamma_c r_2 + \gamma_c^2 r_3 + \ldots$$

Returns (continuing)

$$G_t = \sum_{i=0}^{\infty} \gamma_c^i R_{t+1+i} = R_{t+1} + \gamma_c G_{t+1}$$

$$V(s) = \mathbb{E}[G_t \mid S_t = s]$$

$$\gamma_c \in [0, 1)$$

$$s_0, a_0, r_1, s_1, a_1, r_2, s_2, a_2, r_3, s_3, a_3, \ldots$$

$$\downarrow$$

$$r_2 + \gamma_c r_3 + \gamma_c^2 r_4 + \ldots$$

Returns (continuing)

$$G_t = \sum_{i=0}^{\infty} \gamma_c^i R_{t+1+i} = R_{t+1} + \gamma_c G_{t+1}$$

$$V(s) = \mathbb{E}[G_t \mid S_t = s]$$

$$\gamma_c \in [0, 1)$$

$$s_0, a_0, r_1, s_1, a_1, r_2, s_2, a_2, r_3, s_3, a_3, \ldots$$

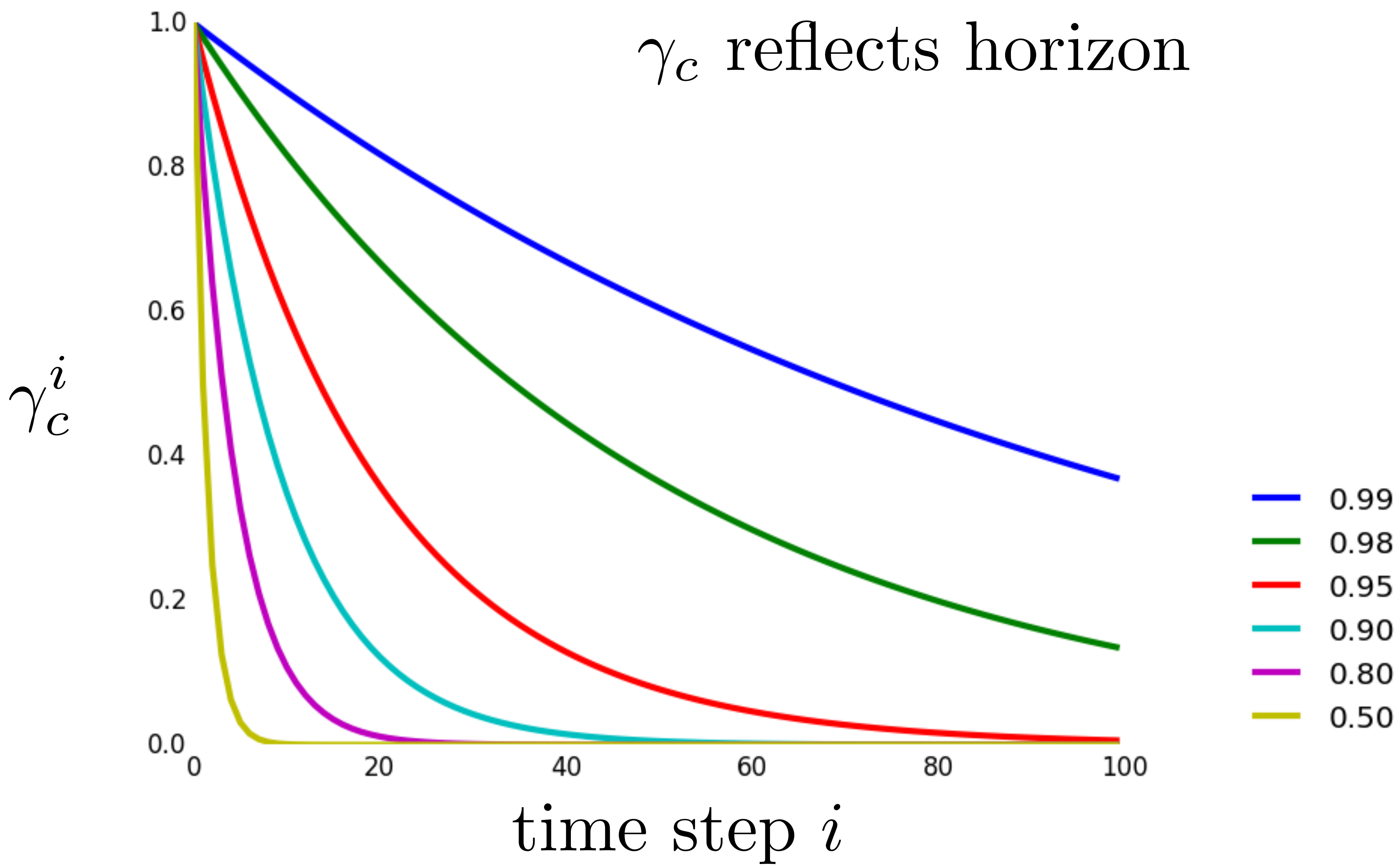
$$\downarrow$$

$$r_3 + \gamma_c r_4 + \gamma_c^2 r_5 + \ldots$$

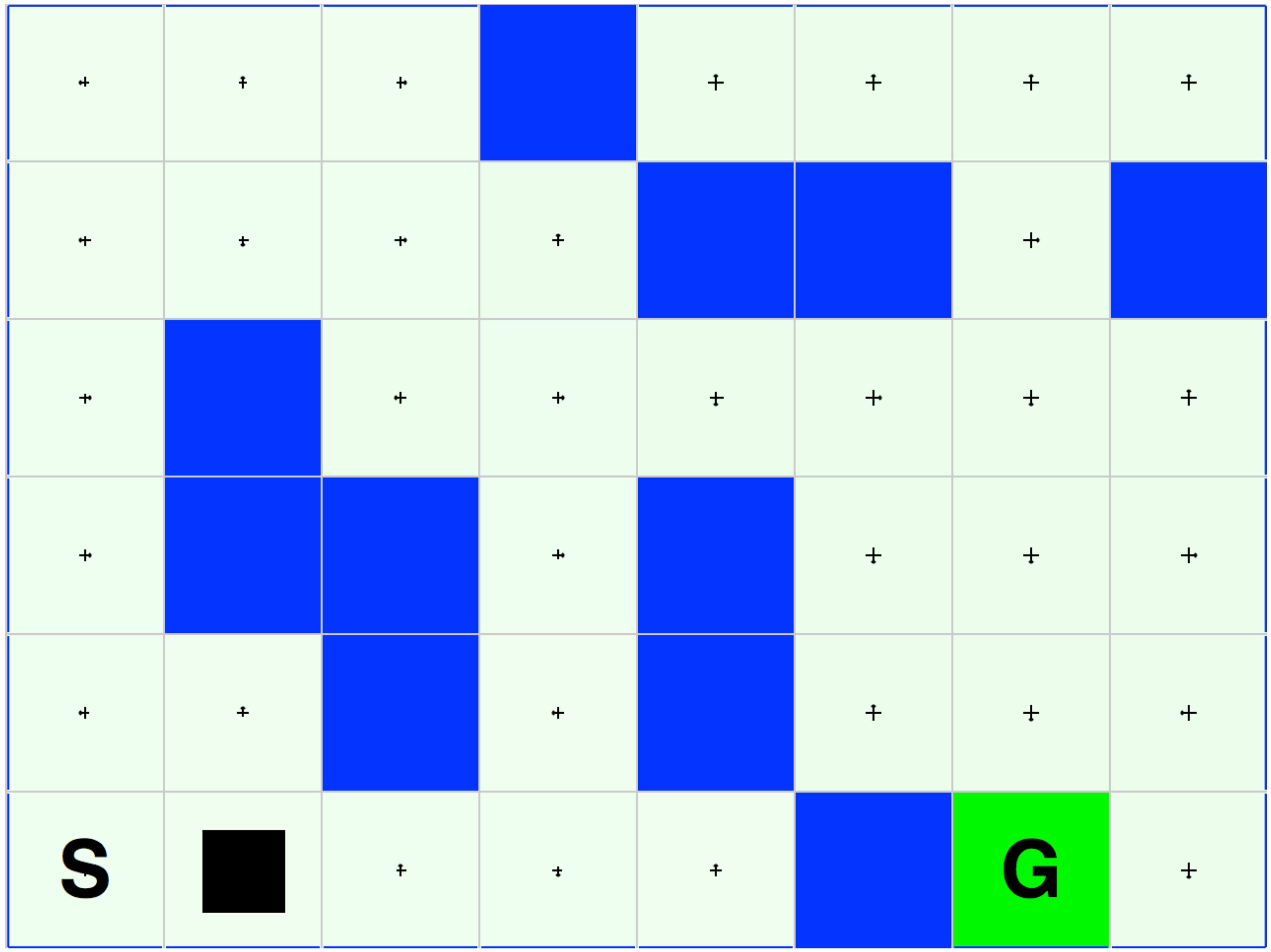
Selection of discount

$$G_t = \sum_{i=0}^{\infty} \gamma_c^i R_{t+1+i} \qquad\qquad \gamma_c \in [0, 1)$$

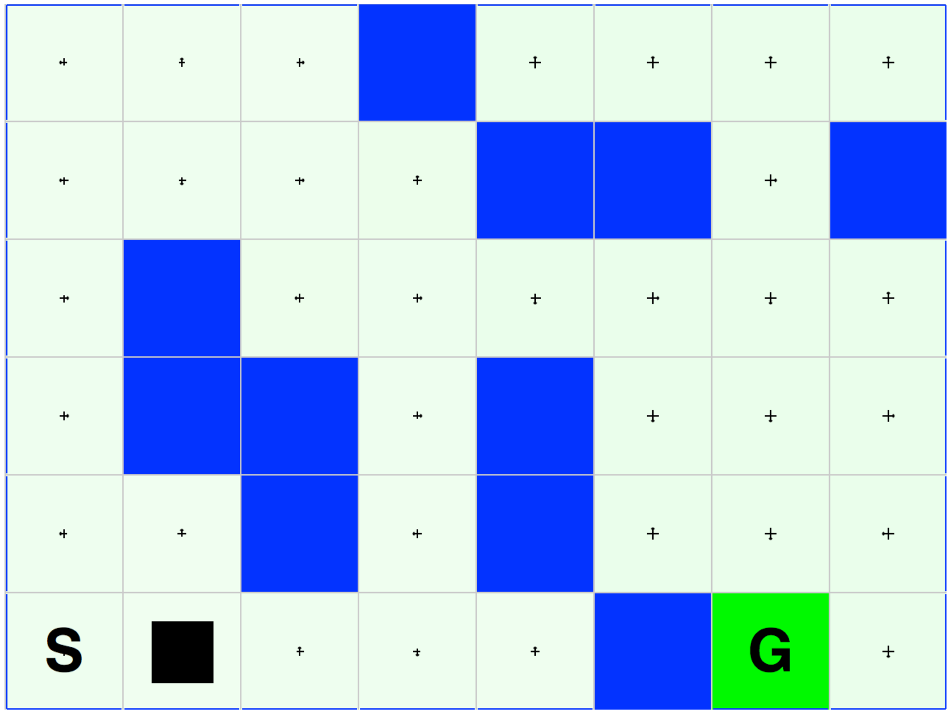
γ_c reflects horizon



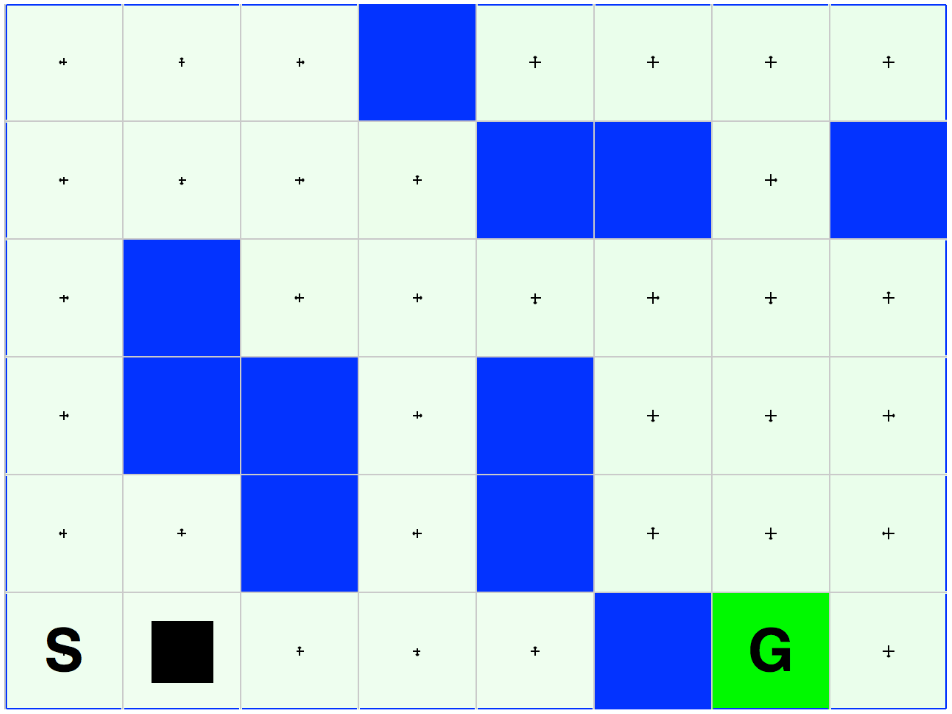
Returns (episodic)



Returns (episodic)



Returns (episodic)



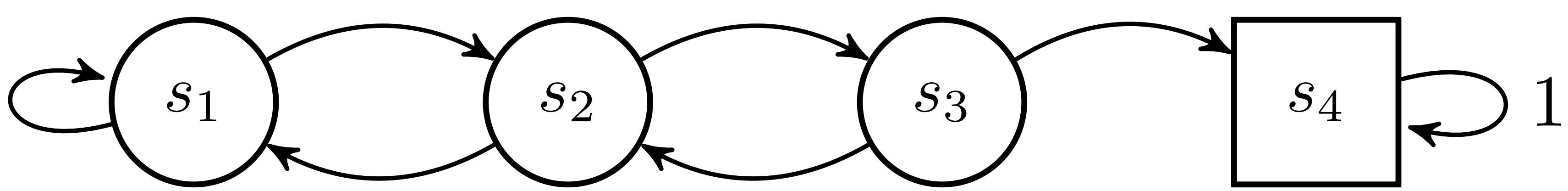
$$s_0 = (7,3), a_0 = \text{Sth}, r_1 = -1, s_1 = (7,2), a_1 = \text{Sth}, r_2 = -1, s_2 = (7,1)$$

$$\downarrow \qquad\qquad\qquad\qquad \downarrow \qquad\qquad\qquad\qquad \downarrow$$

$$r_1 + r_2 \qquad\qquad\qquad\qquad r_2 \qquad\qquad\qquad\qquad \text{Null}$$

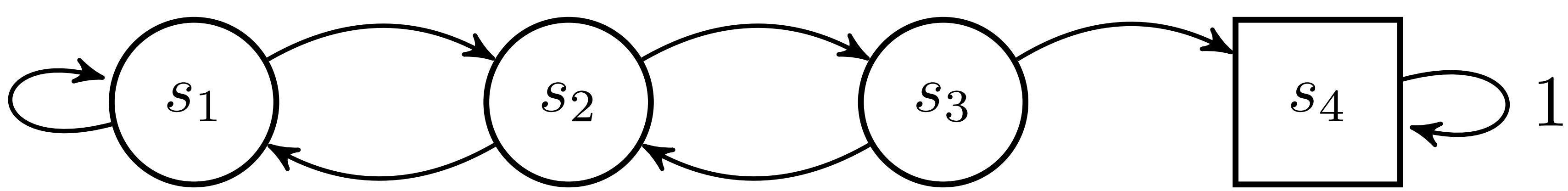
How do we unify the two?

- Algorithms and theory treat the two cases separately
- Absorbing state not a complete solution



How do we unify the two?

- Algorithms and theory treat the two cases separately
- Absorbing state not a complete solution



- Recent generalizations to state-based discount almost the complete solution

Unification using transition-based discounting

- Discount generalized to a function on (s,a,s')

$$\gamma : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,1]$$

- Can smoothly encode continuing or episodic
    - …and specify a whole new set of returns

Generalized return

$$\gamma : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$$

$$G_t = \sum_{i=0}^{\infty} \left(\prod_{j=0}^{i-1} \gamma(S_{t+j}, A_{t+j}, S_{t+j+1}) \right) R_{t+1+i}$$

$$= R_{t+1} + \gamma_{t+1} G_{t+1} \qquad \gamma_{t+1} = \gamma(S_t, A_t, S_{t+1})$$

Generalized return

$$\gamma : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$$

$$G_t = \sum_{i=0}^{\infty} \left(\prod_{j=0}^{i-1} \gamma(S_{t+j}, A_{t+j}, S_{t+j+1}) \right) R_{t+1+i}$$

$$= R_{t+1} + \gamma_{t+1} G_{t+1} \qquad \gamma_{t+1} = \gamma(S_t, A_t, S_{t+1})$$

If $\gamma(s, a, s') = \gamma_c$

$$\prod_{j=0}^{i-1} \gamma(S_{t+j}, A_{t+j}, S_{t+j+1}) = \gamma_c^{i-1}$$

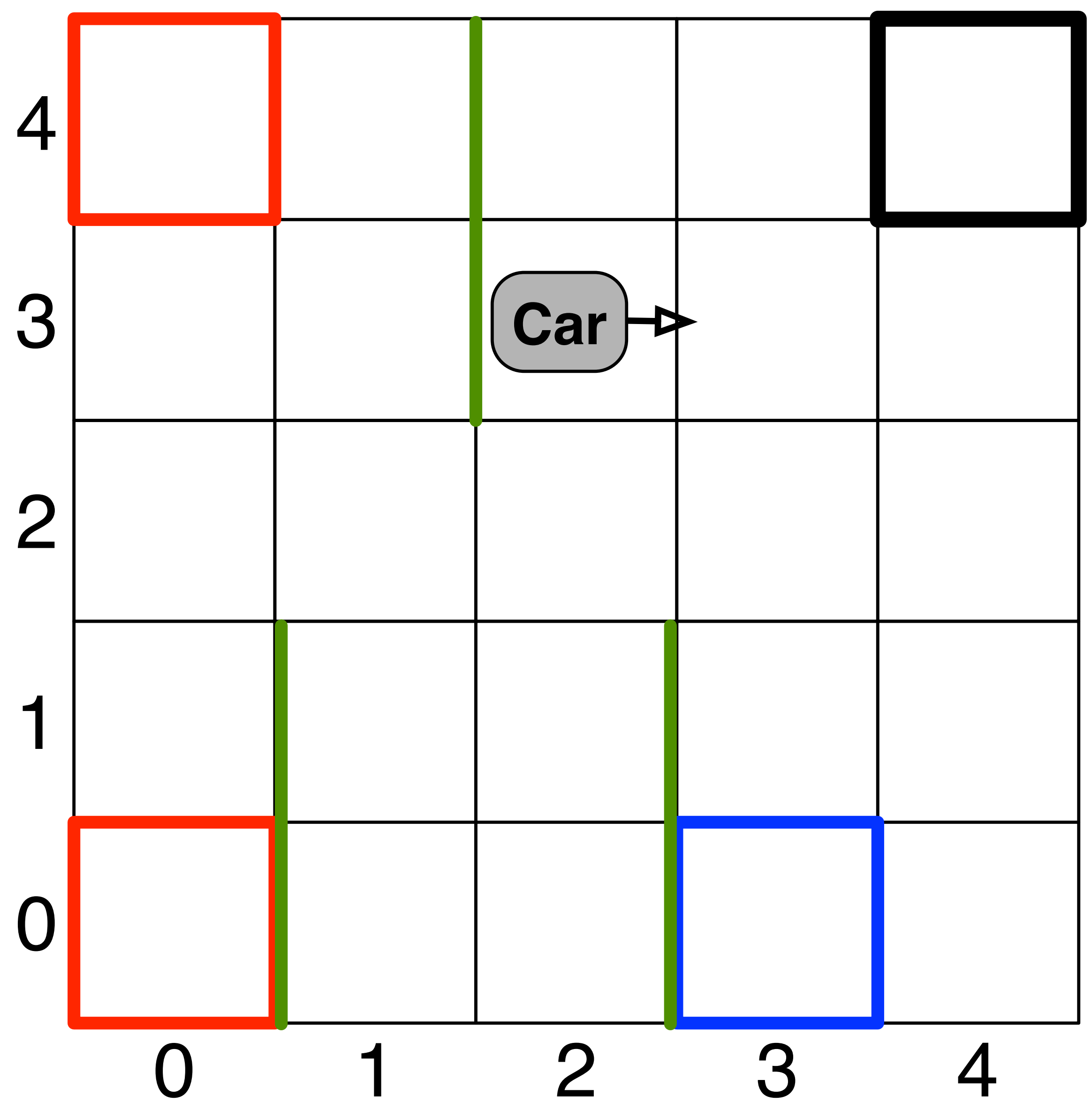
Encoding episodic tasks

- gamma(s,a,s') = 0 for a terminal transition
- s' is the start state for the next episode
- Return is truncated at termination by the product of discounts, with gamma(s,a,s') = 0

$$G_t = R_{t+1} + \gamma_{t+1} G_{t+1}$$

Example: taxi domain

Actions:
N, E, S, W,
Pickup,
Drop-off



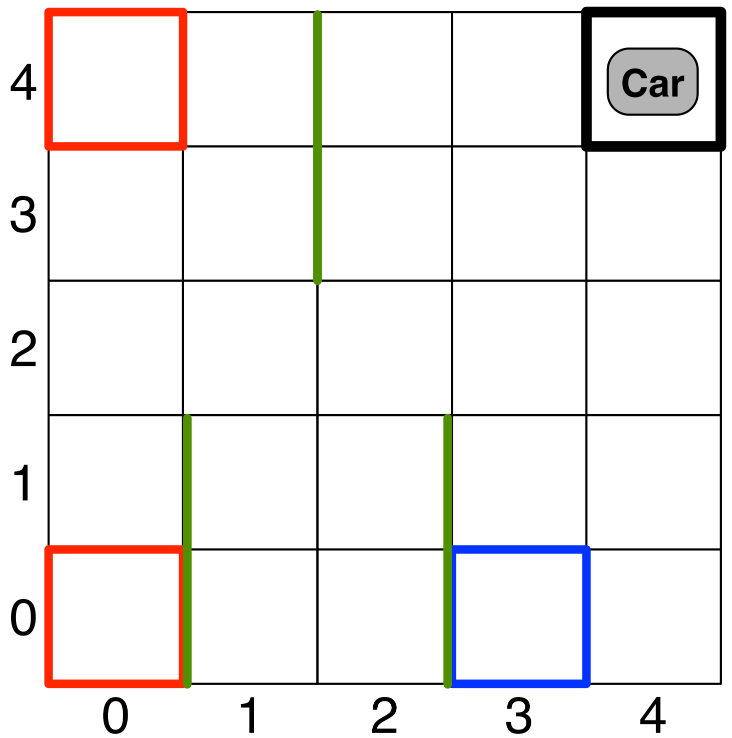
Passenger in black square

States:
(x, y, passenger location)

State: (2, 1, 3)

Location can be (0,1,2,3) or 4 for in taxi

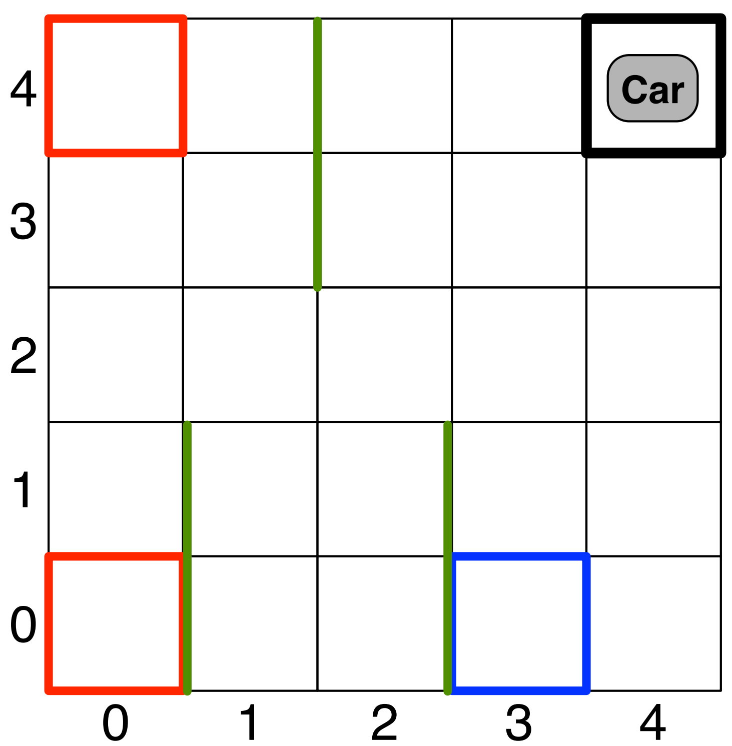
Example: taxi domain



$$G_t = R_{t+1} + \gamma_{t+1} G_{t+1}$$

Example: taxi domain

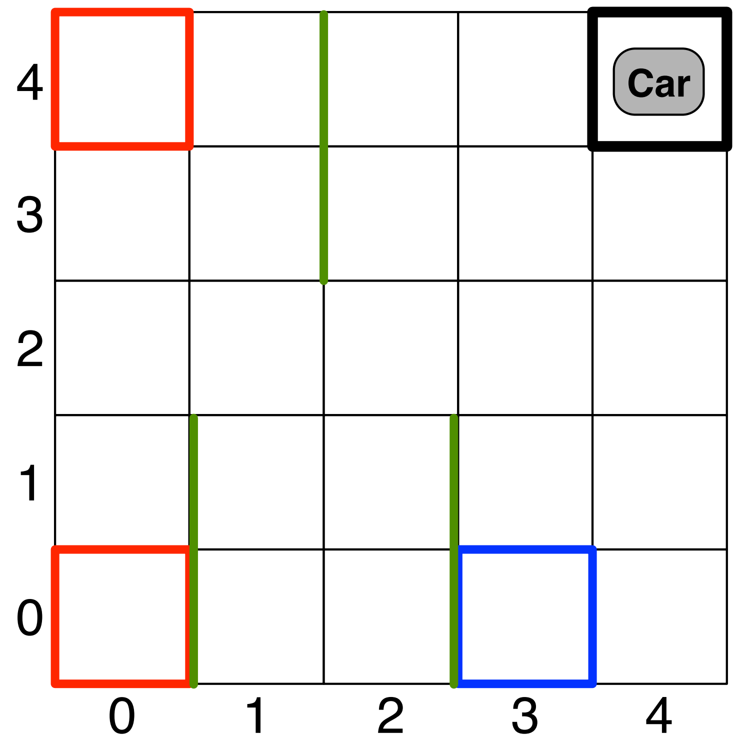
- What are the transition probabilities at (4,4,3)?



$$G_t = R_{t+1} + \gamma_{t+1} G_{t+1}$$

Example: taxi domain

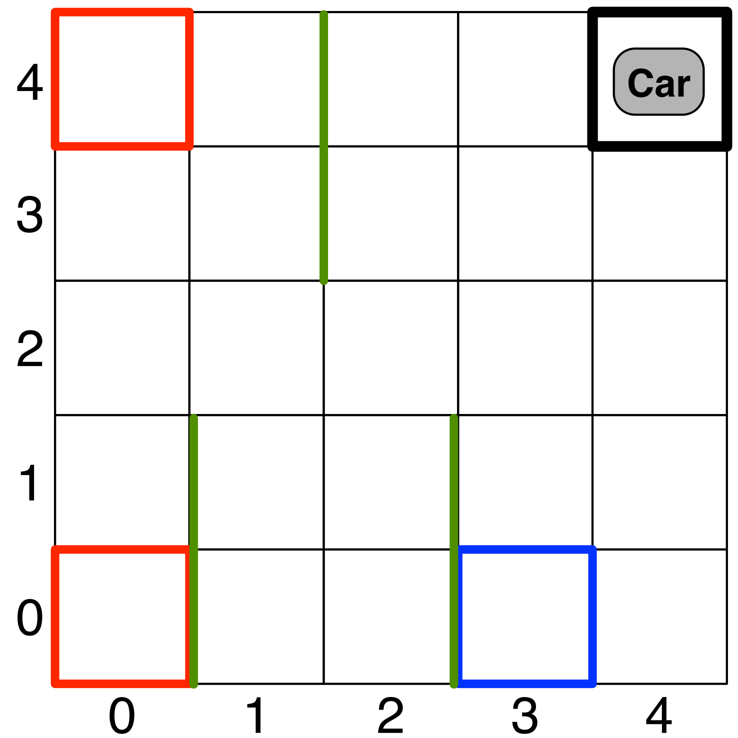
- What are the transition probabilities at (4,4,3)?
  - P((4,4,3), Pick-up, (4,4,4)) = 1.0



$$G_t = R_{t+1} + \gamma_{t+1} G_{t+1}$$

Example: taxi domain

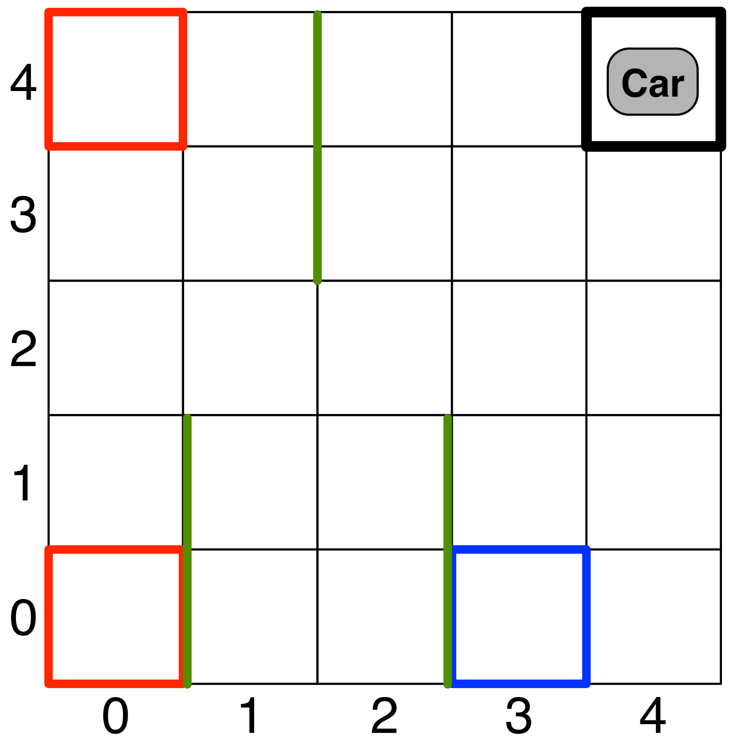
- What are the transition probabilities at (4,4,3)?
  - P((4,4,3), Pick-up, (4,4,4)) = 1.0
- What is the discount function?



$$G_t = R_{t+1} + \gamma_{t+1} G_{t+1}$$

Example: taxi domain

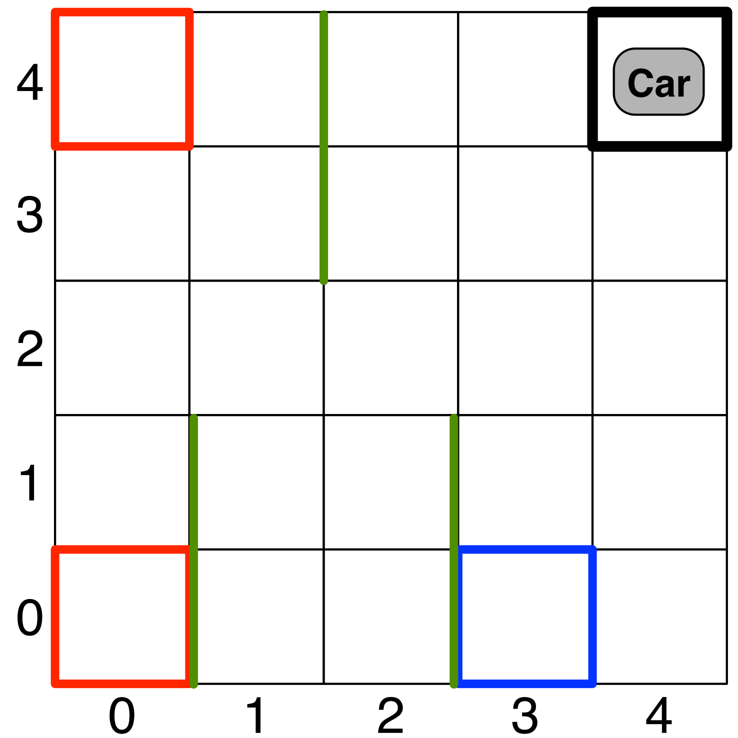
- What are the transition probabilities at (4,4,3)?
  - P((4,4,3), Pick-up, (4,4,4)) = 1.0
- What is the discount function?
  - $\boldsymbol{\gamma}$((4,4,3), Pick-up, (4,4,4)) = 0.0, else 1.0



$$G_t = R_{t+1} + \gamma_{t+1} G_{t+1}$$

Example: taxi domain

- What are the transition probabilities at (4,4,3)?
  - P((4,4,3), Pick-up, (4,4,4)) = 1.0
- What is the discount function?
  - $\boldsymbol{\gamma}$((4,4,3), Pick-up, (4,4,4)) = 0.0, else 1.0
- Why not $\boldsymbol{\gamma}_S$((4,4,4)) = 0.0?

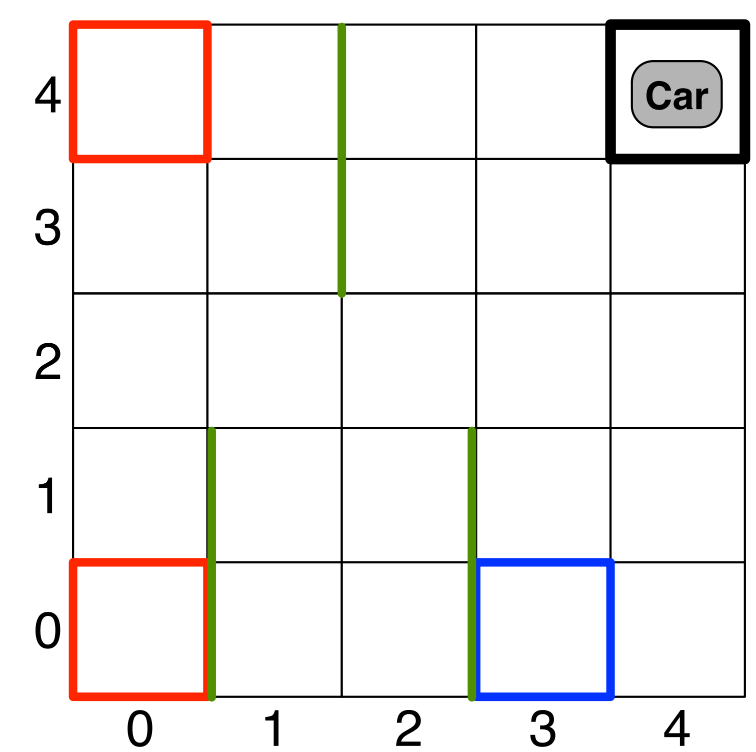


$$G_t = R_{t+1} + \gamma_{t+1} G_{t+1}$$

Example: taxi domain

- What are the transition probabilities at (4,4,3)?
  - P((4,4,3), Pick-up, (4,4,4)) = 1.0
- What is the discount function?
  - $\boldsymbol{\gamma}$((4,4,3), Pick-up, (4,4,4)) = 0.0, else 1.0
- Why not $\boldsymbol{\gamma}_S$((4,4,4)) = 0.0?
- Why not add a termination state?

$$G_t = R_{t+1} + \gamma_{t+1} G_{t+1}$$



What are the implications?

What are the implications?

- Unified analysis for episodic and continuing problems —> can extend previous results

What are the implications?

- Unified analysis for episodic and continuing problems —> can extend previous results
- How does this change the algorithms?

What are the implications?

- Unified analysis for episodic and continuing problems —> can extend previous results
- How does this change the algorithms?
    - very little
    - avoids two versions of an algorithm

Do all algorithms extend?

- Can define a generalized Bellman operator
  - recursive form for return, that is Markov

$$G_t = \sum_{i=0}^{\infty} \left(\prod_{j=1}^{i} \gamma_{t+j} \right) R_{t+1+i} = R_{t+1} + \gamma_{t+1} G_{t+1}$$

Do all algorithms extend?

- Can define a generalized Bellman operator
    - recursive form for return, that is Markov

$$G_t = \sum_{i=0}^{\infty} \left(\prod_{j=1}^{i} \gamma_{t+j} \right) R_{t+1+i} = R_{t+1} + \gamma_{t+1} G_{t+1}$$

- Replace γ_c with γ_{t+1}

Definitions for the operator

$$\begin{aligned}
\mathbf{v}_\pi(s) &= \mathbb{E}[G_t | S_t = s] \\
&= \mathbb{E}[R_{t+1} + \gamma_{t+1} G_{t+1} | S_t = s] \\
&= \mathbb{E}[R_{t+1} | S_t = s] + \mathbb{E}[\gamma_{t+1} \mathbf{v}^\pi (S_{t+1}) | S_t = s]
\end{aligned}$$

Definitions for the operator

$$\begin{aligned}
\mathbf{v}_\pi(s) &= \mathbb{E}[G_t | S_t = s] \\
&= \mathbb{E}[R_{t+1} + \gamma_{t+1} G_{t+1} | S_t = s] \\
&= \mathbb{E}[R_{t+1} | S_t = s] + \mathbb{E}[\gamma_{t+1} \mathbf{v}^\pi (S_{t+1}) | S_t = s] \\
&= \mathbf{r}_\pi(s) + \sum_{s'} \mathbf{P}_{\pi,\gamma}(s, s') \mathbf{v}_\pi(s')
\end{aligned}$$

$$\mathbf{P}_{\pi,\gamma}(s, s') = \sum_a \pi(s, a) \Pr(s, a, s') \gamma(s, a, s')$$

$$\mathbf{r}_\pi(s) = \sum_a \pi(s, a) \sum_{s'} \Pr(s, a, s') r(s, a, s')$$

Definitions for the operator

$$\mathbf{v}_\pi \in \mathbb{R}^{\text{number of states}}$$

$$\begin{aligned}
\mathbf{v}_\pi(s) &= \mathbb{E}[G_t | S_t = s] \\
&= \mathbb{E}[R_{t+1} + \gamma_{t+1} G_{t+1} | S_t = s] \\
&= \mathbb{E}[R_{t+1} | S_t = s] + \mathbb{E}[\gamma_{t+1} \mathbf{v}^\pi(S_{t+1}) | S_t = s] \\
&= \mathbf{r}_\pi(s) + \sum_{s'} \mathbf{P}_{\pi,\gamma}(s, s') \mathbf{v}_\pi(s')
\end{aligned}$$

$$\mathbf{P}_{\pi,\gamma}(s, s') = \sum_a \pi(s, a) \Pr(s, a, s') \gamma(s, a, s')$$

$$\mathbf{r}_\pi(s) = \sum_a \pi(s, a) \sum_{s'} \Pr(s, a, s') r(s, a, s')$$

Bellman operator

$$\mathbf{v}_\pi = \mathbf{r}_\pi + \sum_{s'} \mathbf{P}_{\pi,\gamma}(:, s')\mathbf{v}_\pi(s') = \mathbf{r}_\pi + \mathbf{P}_{\pi,\gamma}\mathbf{v}_\pi$$

Bellman operator

$$\mathbf{v}_\pi = \mathbf{r}_\pi + \sum_{s'} \mathbf{P}_{\pi,\gamma}(:, s')\mathbf{v}_\pi(s') = \mathbf{r}_\pi + \mathbf{P}_{\pi,\gamma}\mathbf{v}_\pi$$

$$\text{e.g., } \mathbf{r}_\pi + \gamma_c \mathbf{P}_\pi \mathbf{v}_\pi$$

Bellman operator

$$\mathbf{v}_\pi = \mathbf{r}_\pi + \sum_{s'} \mathbf{P}_{\pi,\gamma}(:, s')\mathbf{v}_\pi(s') = \mathbf{r}_\pi + \mathbf{P}_{\pi,\gamma}\mathbf{v}_\pi$$

$$\text{e.g., } \mathbf{r}_\pi + \gamma_c \mathbf{P}_\pi \mathbf{v}_\pi$$

Bellman operator

$$\mathbf{T}\mathbf{v} = \mathbf{r}_\pi + \mathbf{P}_{\pi,\gamma}\mathbf{v}$$

Reach solution (fixed point) when $\mathbf{T}\mathbf{v} = \mathbf{v}$

Given models, can use dynamic programming
Otherwise, stochastic approximations (e.g., TD)

Key property: contraction

- The operator $\mathbf{T}$ has to be a contraction
- If $\mathbf{T}$ is an expansion, then repeated application of $\mathbf{T}$ to $\mathbf{v}$ could expand to infinity

Key property: contraction

- The operator $\mathbf{T}$ has to be a contraction
- If $\mathbf{T}$ is an expansion, then repeated application of $\mathbf{T}$ to $\mathbf{v}$ could expand to infinity

$$\|\mathbf{T}\mathbf{v}_1 - \mathbf{T}\mathbf{v}_2\|_D = \|\mathbf{P}_{\pi,\gamma}\left(\mathbf{v}_1 - \mathbf{v}_2\right)\|_D \leq \|\mathbf{P}_{\pi,\gamma}\|_D \|\mathbf{v}_1 - \mathbf{v}_2\|_D$$

Contraction properties

$$s_{\mathbf{D}} = \|\mathbf{P}_{\pi,\gamma}\|_{\mathbf{D}}$$

- Smaller $s_{\mathbf{D}}$ corresponds to faster contraction
- Example: constant discount

$$\begin{aligned} s_{\mathbf{D}} &= \|\mathbf{P}_{\pi,\gamma}\|_{\mathbf{D}} \\ &= \gamma_c \|\mathbf{P}_{\pi}\|_{\mathbf{D}} \\ &= \gamma_c \end{aligned}$$

Extending previous results

Extending previous results

- LSTD convergence rates specifically derived for continuing case
  - extends to episodic with this generalization
- Unify seminal bias bounds for TD
  - with more explicit episodic bounds

Extending previous results

- LSTD convergence rates specifically derived for continuing case
    - extends to episodic with this generalization
- Unify seminal bias bounds for TD
    - with more explicit episodic bounds
- **New result:** convergence of ETD for transition-based trace, but not under on-policy weighting

Bias bounds

Continuing:

$$\|\mathbf{T}\mathbf{v}_1 - \mathbf{T}\mathbf{v}_2\|_D \leq \frac{\gamma_c(1-\lambda)}{1-\gamma_c\lambda}\|\mathbf{v}_1 - \mathbf{v}_2\|_D$$

SSP (episodic):

$$\text{Exists contraction constant } s < 1$$

Bias bounds

$$\|\mathbf{T}\mathbf{v}_1 - \mathbf{T}\mathbf{v}_2\|_{\mathbf{D}} \leq s_{\mathbf{D}}\|\mathbf{v}_1 - \mathbf{v}_2\|_{\mathbf{D}}$$

$$s_{\mathbf{D}} = \|\mathbf{P}_{\pi,\gamma}\|_{\mathbf{D}}$$

If policy reaches a transition where discount less than 1 guaranteed to have $s_{\mathbf{D}} < 1$

Bias bounds

$$\|\mathbf{T}\mathbf{v}_1 - \mathbf{T}\mathbf{v}_2\|_{\mathbf{D}} \leq s_{\mathbf{D}} \|\mathbf{v}_1 - \mathbf{v}_2\|_{\mathbf{D}}$$

$$s_{\mathbf{D}} = \|\mathbf{P}_{\pi,\gamma}\|_{\mathbf{D}}$$

If policy reaches a transition where discount less than 1 guaranteed to have $s_{\mathbf{D}} < 1$

EPISODIC TAXI	0.989
$\gamma_c = 0.99$	0.990
1% SINGLE PATH	0.989
10% SINGLE PATH	0.987
1% ALL PATHS	0.978
10% ALL PATHS	0.898

Bias bounds

$$\|\mathbf{T}\mathbf{v}_1 - \mathbf{T}\mathbf{v}_2\|_{\mathbf{D}} \leq s_{\mathbf{D}}\|\mathbf{v}_1 - \mathbf{v}_2\|_{\mathbf{D}}$$

$$s_{\mathbf{D}} = \|\mathbf{P}_{\pi,\gamma}\|_{\mathbf{D}}$$

If policy reaches a transition where discount less than 1 guaranteed to have $s_{\mathbf{D}} < 1$

λ_c	0.0	0.5	0.9	0.99	0.999
EPISODIC TAXI	0.989	0.979	0.903	0.483	0.086
$\gamma_c = 0.99$	0.990	0.980	0.908	0.497	0.090
1% SINGLE PATH	0.989	0.978	0.898	0.467	0.086
10% SINGLE PATH	0.987	0.975	0.887	0.439	0.086
1% ALL PATHS	0.978	0.956	0.813	0.304	0.042
10% ALL PATHS	0.898	0.815	0.468	0.081	0.009

Generalizing to probabilistic discounts

$$\Pr(r, \gamma | s, a, s')$$

Generalizing to probabilistic discounts

$$
\begin{aligned}
\mathbf{v}_\pi(s) &= \sum_{a,s'} \pi(s,a)\mathrm{Pr}(s,a,s')\mathrm{E}[r + \gamma \mathbf{v}_\pi(s') | s,a,s'] \\
&= \sum_{a,s'} \pi(s,a)\mathrm{Pr}(s,a,s')\mathrm{E}[r | s,a,s'] \\
&\quad + \sum_{a,s'} \pi(s,a)\mathrm{Pr}(s,a,s')\mathrm{E}[\gamma | s,a,s'] \mathbf{v}_\pi(s') \\
&= \mathbf{r}_\pi(s) + \sum_{s'} \mathbf{P}_{\pi,\gamma}(s,s')\mathbf{v}_\pi(s')
\end{aligned}
$$

for $\gamma(s,a,s') = \mathrm{E}[\gamma | s,a,s']$.

How are all these conclusions affected by function approximation?

- Assumed states; but likely partially observable
- May no longer be able to solve $\mathbf{TV} = \mathbf{V}$ but can
  - minimize Bellman residual $\|\mathbf{TV} - \mathbf{V}\|$
  - get projected fixed point (MSPBE): $\mathbf{\Pi TV} = \mathbf{V}$
  - …

Example: TD

initialize $\boldsymbol{\theta}$ arbitrarily

loop over episodes

initialize $\boldsymbol{e} = \mathbf{0}$

initialize S_0

repeat for each step in the episode

generate R_{t+1}, S_{t+1} for S_t

if terminal: $\delta \leftarrow R_{t+1} - \boldsymbol{\theta}^\top \phi(S_t)$

else: $\delta \leftarrow R_{t+1} + \gamma \boldsymbol{\theta}^\top \phi(S_{t+1}) - \boldsymbol{\theta}^\top \phi(S_t)$

$\mathbf{e} \leftarrow \mathbf{e} + \phi(S_t)$

$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \delta \mathbf{e}$

$\mathbf{e} \leftarrow \gamma \lambda \mathbf{e}$

Example: TD

initialize $\boldsymbol{\theta}$ arbitrarily

loop over episodes

  initialize $\boldsymbol{e} = \mathbf{0}$

  initialize S_0

repeat for each step in the episode

    generate R_{t+1}, S_{t+1} for S_t

    if terminal: $\delta \leftarrow R_{t+1} - \boldsymbol{\theta}^\top \phi(S_t)$

    else: $\delta \leftarrow R_{t+1} + \gamma \boldsymbol{\theta}^\top \phi(S_{t+1}) -$

    $\mathbf{e} \leftarrow \mathbf{e} + \phi(S_t)$

    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \delta \mathbf{e}$

    $\mathbf{e} \leftarrow \gamma \lambda \mathbf{e}$

Example: TD

initialize $\boldsymbol{\theta}$ arbitrarily

initialize $\boldsymbol{e} = \mathbf{0}$

initialize S_0

repeat until agent done interaction

  generate R_{t+1}, S_{t+1} for S_t

  $\delta \leftarrow R_{t+1} + \gamma_{t+1}\boldsymbol{\theta}^\top \phi(S_{t+1}) - \boldsymbol{\theta}^\top \phi(S_t)$

  $\mathbf{e} \leftarrow \mathbf{e} + \phi(S_t)$

  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha\delta\mathbf{e}$

  $\mathbf{e} \leftarrow \gamma_{t+1}\lambda\mathbf{e}$

initialize $\boldsymbol{\theta}$ arbitrarily

loop over episodes

  initialize $\boldsymbol{e} = \mathbf{0}$

  initialize S_0

repeat for each step in the episode

  generate R_{t+1}, S_{t+1} for S_t

  if terminal: $\delta \leftarrow R_{t+1} - \boldsymbol{\theta}^\top \phi(S_t)$

  else: $\delta \leftarrow R_{t+1} + \gamma\boldsymbol{\theta}^\top \phi(S_{t+1}) -$

  $\mathbf{e} \leftarrow \mathbf{e} + \phi(S_t)$

  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha\delta\mathbf{e}$

  $\mathbf{e} \leftarrow \gamma\lambda\mathbf{e}$

Outline

- ☑ Generalization to transition-based discounting
- ☑ The theoretical and algorithmic implications
- Utility of the generalized problem formalism

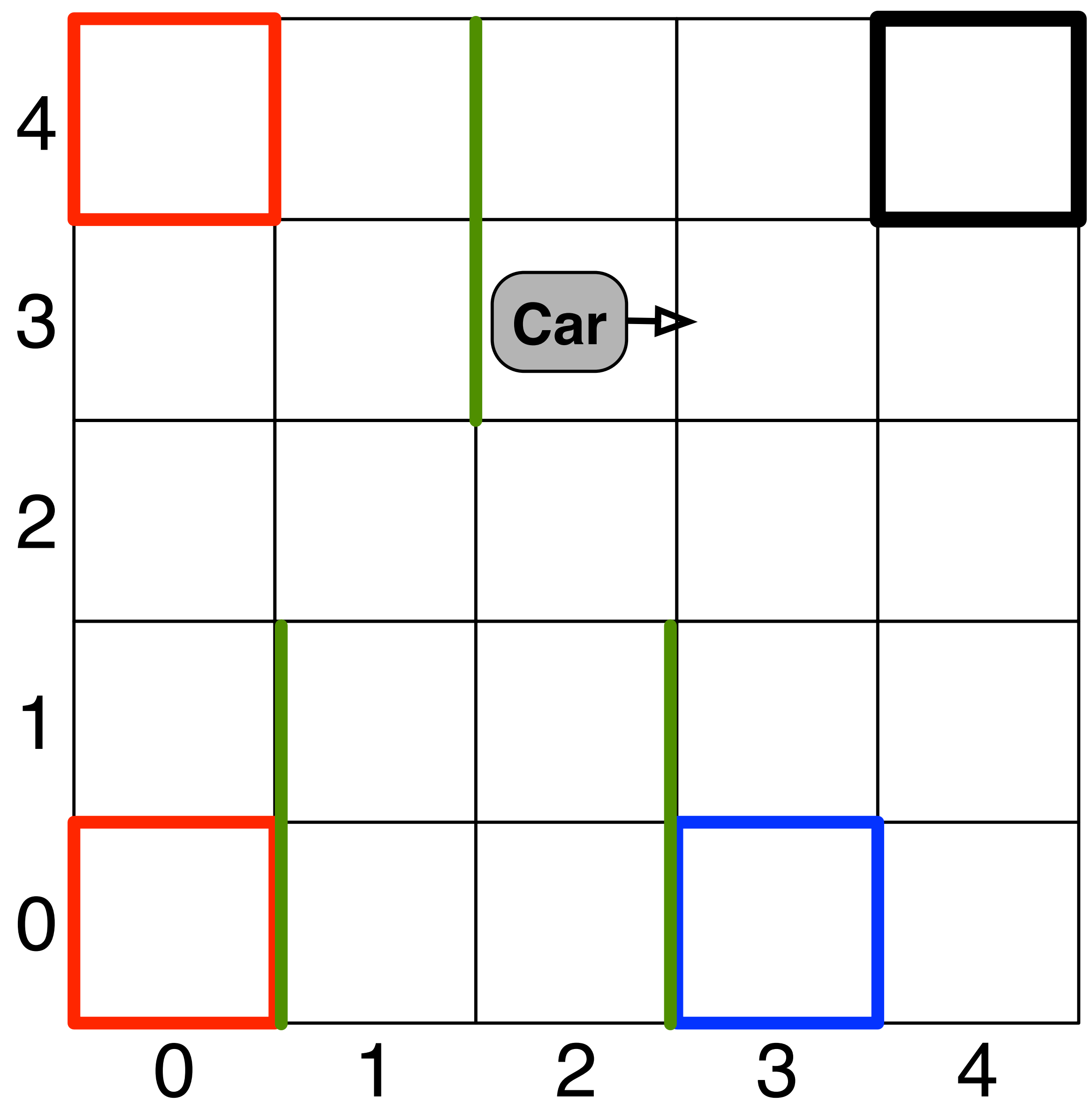
Additional utility

- **Control**
  - simplifies specification of subgoals
  - enables soft termination
- **Policy evaluation**
  - GVFs and predictive knowledge

Example: taxi domain

Actions:
N, E, S, W,
Pickup,
Drop-off

States:
(x, y,
passenger
location)



Passenger in black square

State: (2, 1, 3)

Location can be (0,1,2,3) or 4 for in taxi

Optimal policy

Can use average reward or continuing formulation

$$
\begin{aligned}
\mathbf{d}_\pi \mathbf{v}_\pi &= \mathbf{d}_\pi (\mathbf{r}_\pi + \mathbf{P}_{\pi,\gamma} \mathbf{v}_\pi) \\
&= \mathbf{d}_\pi \mathbf{r}_\pi + \gamma_c \mathbf{d}_\pi \mathbf{P}_\pi \mathbf{v}_\pi \\
&= \mathbf{d}_\pi \mathbf{r}_\pi + \gamma_c \mathbf{d}_\pi \mathbf{v}_\pi \\
\implies \mathbf{d}_\pi \mathbf{v}_\pi &= \frac{1}{1-\gamma_c} \mathbf{d}_\pi \mathbf{r}_\pi .
\end{aligned}
$$

Easy specification of subgoals

- Each pick-up and drop-off can be a subtask
    - numerically more stable that a constant discount
    - options easily encoded with this generalization

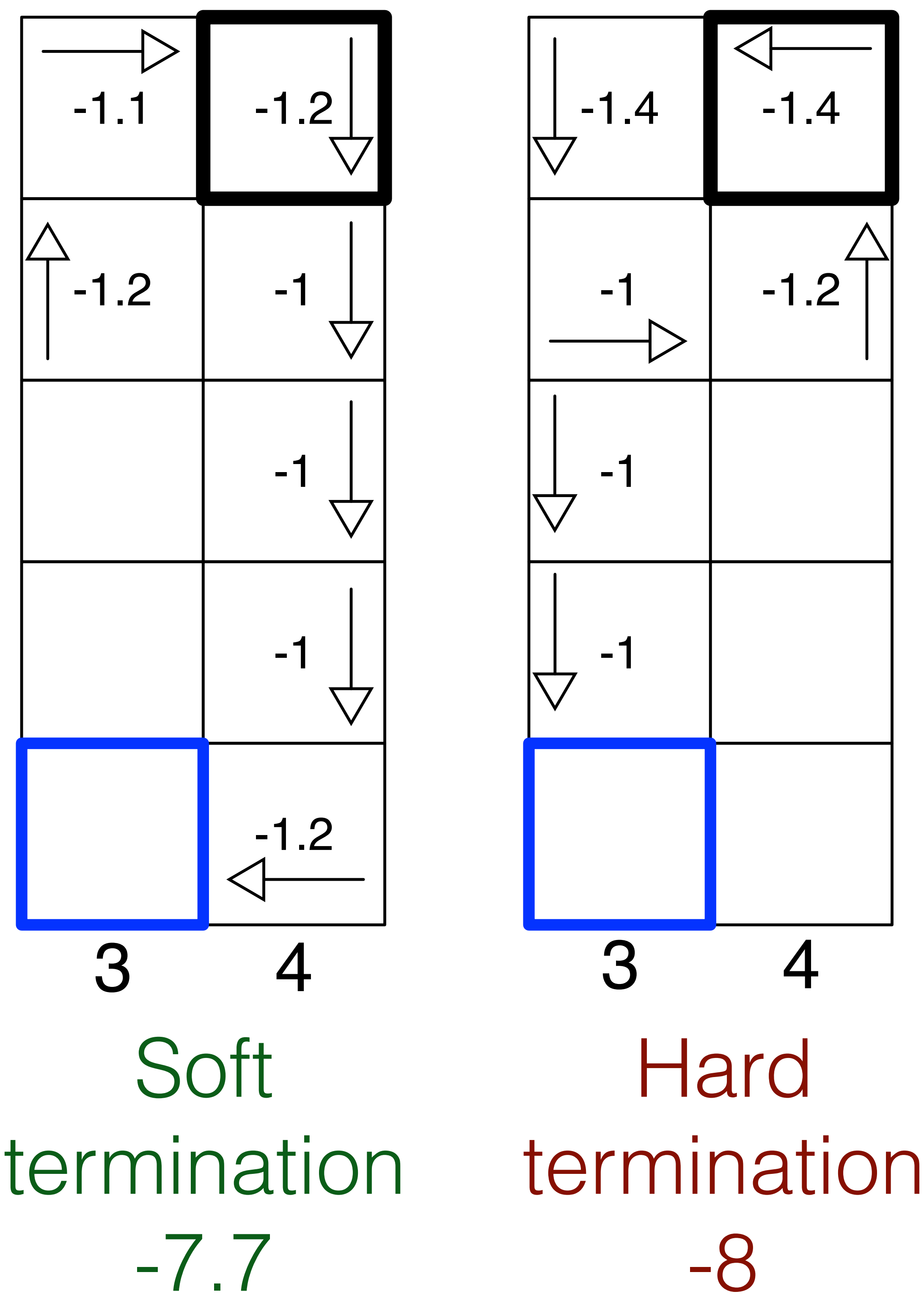
Easy specification of subgoals

- Each pick-up and drop-off can be a subtask
  - numerically more stable that a constant discount
  - options easily encoded with this generalization

	TOTAL PICKUP AND DROPOFF
TRANS-SOFT	7.74 ± 0.03
$\gamma_c = 0.1$	0.00 ± 0.00
$\gamma_c = 0.3$	0.02 ± 0.01
$\gamma_c = 0.5$	0.04 ± 0.01
$\gamma_c = 0.6$	0.03 ± 0.01
$\gamma_c = 0.7$	7.12 ± 0.03
$\gamma_c = 0.8$	7.34 ± 0.03
$\gamma_c = 0.9$	3.52 ± 0.06
$\gamma_c = 0.99$	0.01 ± 0.01

Benefits of soft termination

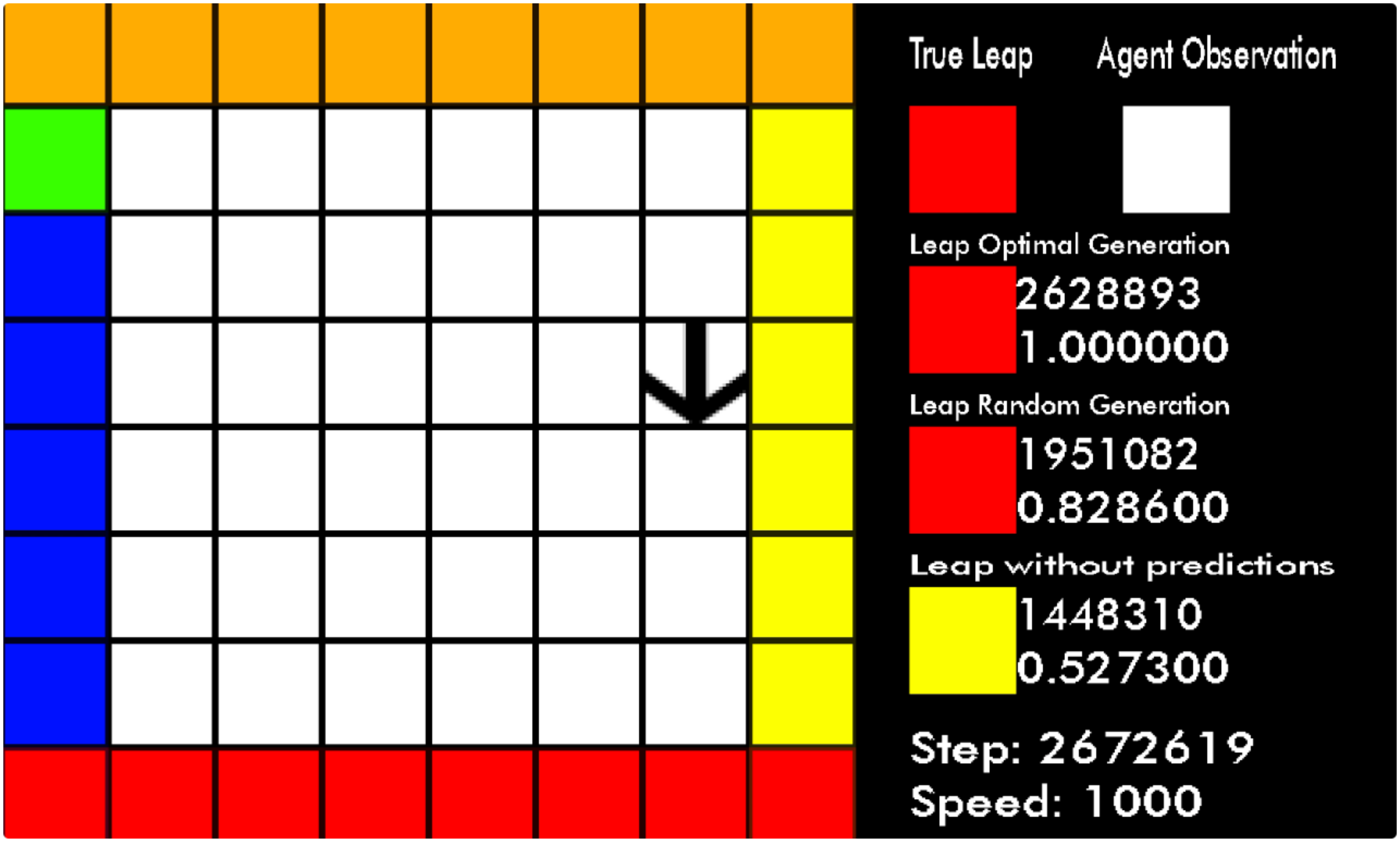
- **Soft termination**: gamma(s, a, s') = 0.1
- Some amount of the value after subgoal should be considered important



How do we use this generality?

- Do not need to use full generality
    - …we know at least two useful settings
- Particularly useful for policy evaluation and predictive knowledge
    - GVFs (Horde), Predictron
    - Predictive representations

Predictive knowledge

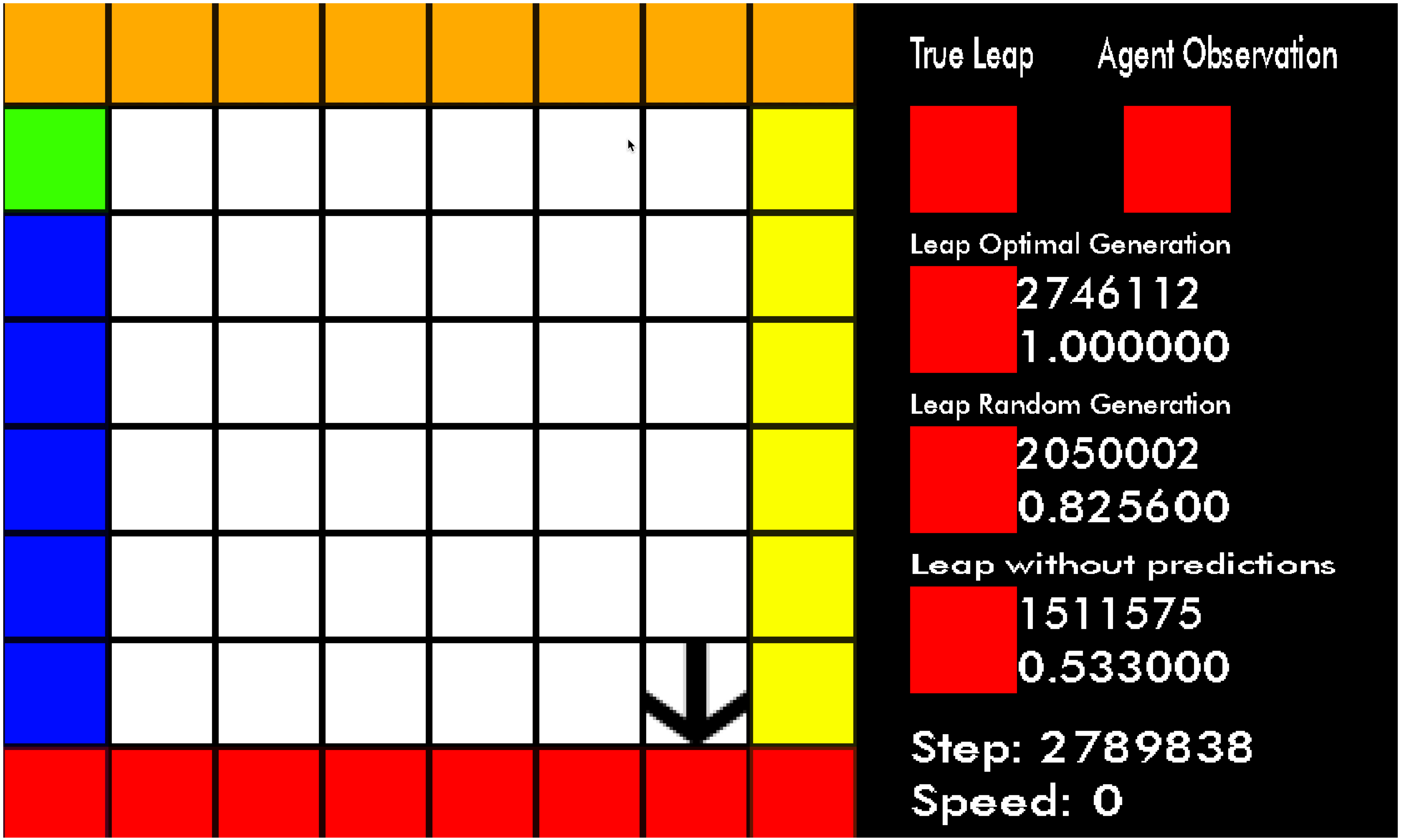


observations as cumulants, persistent policies, …

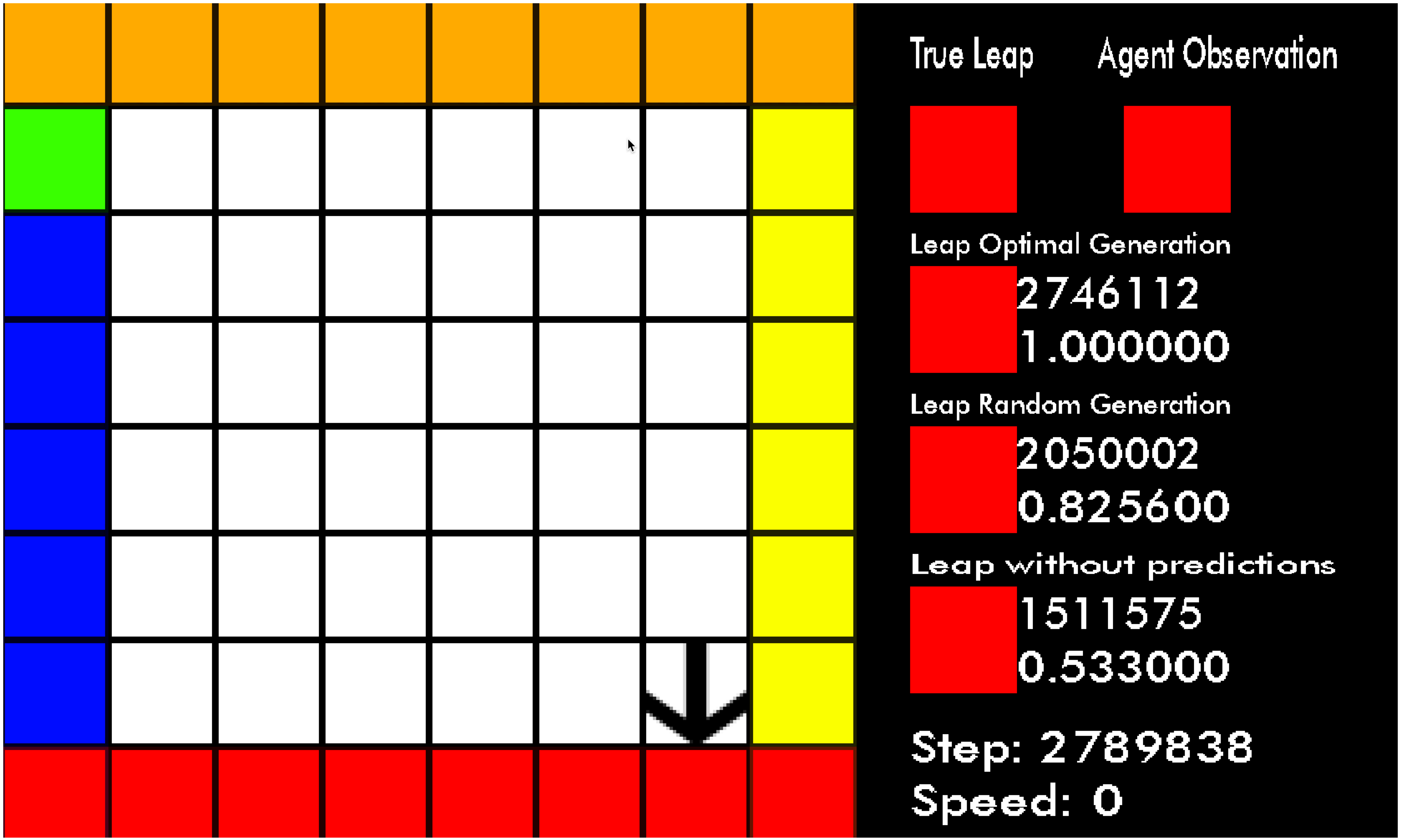
Suggestions for automatically setting the discount

- Parametrize the discount
  - similarly to option-critic
- Variety of constant discounts for different horizons
  - myopic gamma = 0 for one-step predictions
- Decrease discount based on stimuli
  - e.g., sudden drop in stimulus (light)

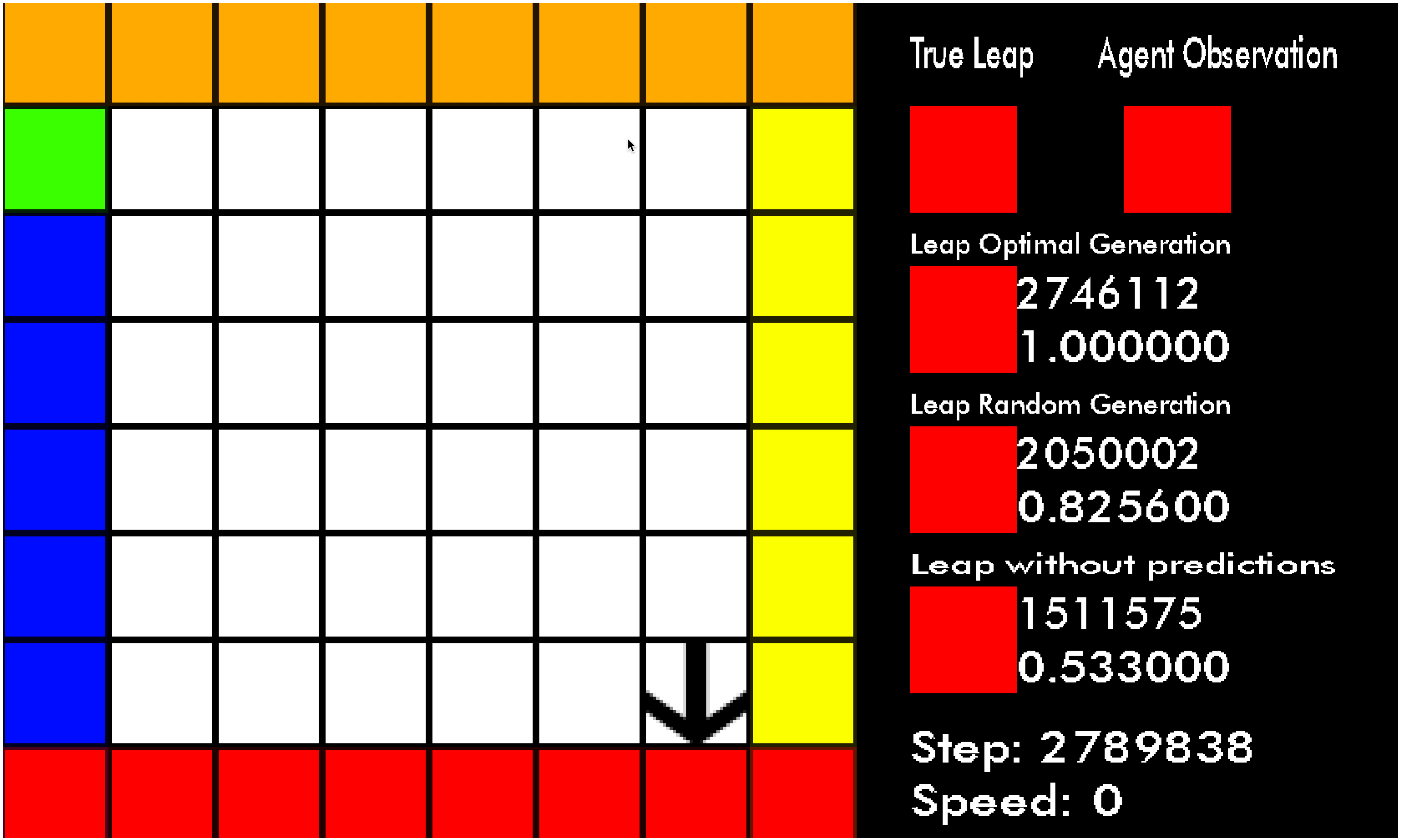
Learning in compass world



Learning in compass world



Learning in compass world



Is it really worth it?

Is it really worth it?

- If you’re considering state-based, may as well do transition-based (and avoid the addition of hypothetical states)

Is it really worth it?

- If you’re considering state-based, may as well do transition-based (and avoid the addition of hypothetical states)
- Algorithm implementation simpler for non-expert
  - **modular**: only need if statements in the discount function
  - **abstraction**: our A.I. algorithms should be as agnostic as possible to the problem settings
  - **simplicity**: e.g., computing stationary distributions from P

Is it really worth it?

- If you're considering state-based, may as well do transition-based (and avoid the addition of hypothetical states)

- Algorithm implementation simpler for non-expert
  - **modular**: only need if statements in the discount function
  - **abstraction**: our A.I. algorithms should be as agnostic as possible to the problem settings
  - **simplicity**: e.g., computing stationary distributions from P

- Our theoretical results should apply to both episodic and continuing problems

Is it really worth it?

- If you're considering state-based, may as well do transition-based (and avoid the addition of hypothetical states)
- Algorithm implementation simpler for non-expert
  - **modular**: only need if statements in the discount function
  - **abstraction**: our A.I. algorithms should be as agnostic as possible to the problem settings
  - **simplicity**: e.g., computing stationary distributions from P
- Our theoretical results should apply to both episodic and continuing problems
- Beneficial mindset shift to lifelong learning

Is it really worth it?

- If you’re considering state-based, may as well do transition-based (and avoid the addition of hypothetical states)
- Algorithm implementation simpler for non-expert
  - **modular**: only need if statements in the discount function
  - **abstraction**: our A.I. algorithms should be as agnostic as possible to the problem settings
  - **simplicity**: e.g., computing stationary distributions from P
- Our theoretical results should apply to both episodic and continuing problems
- Beneficial mindset shift to lifelong learning

Thank you!