Improvements:

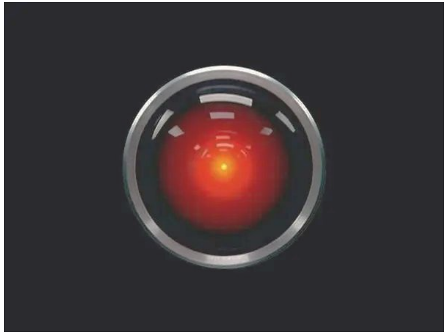
- Reduce content
- Provide example each of the preference elicitation interface for ChatGPT (or InstructGPT) and Christiano et al.

Models of human preference for learning in sequential tasks

W. Bradley Knox

The University of Texas at Austin

Aligned reward



2001: A Space Odyssey

Blues Brothers

Knox et al., *Reward (Mis)design for Autonomous Driving*
AIJ 2023

Research on aligned reward specification

Manual reward design (how it's usually done)

- Reward (mis)design for autonomous vehicles (AIJ 2023; arxiv 2021)
- The Perils of Trial-and-Error Reward Design: Misdesign through Overfitting and Invalid Task Specifications (AAAI 2023)

Reward inference

- The EMPATHIC framework for task learning from implicit human feedback (CoRL 2020)
- Models of human preference for learning reward functions (arxiv 2022)
- Learning Optimal Advantage from Preferences and Mistaking it for Reward (under review)

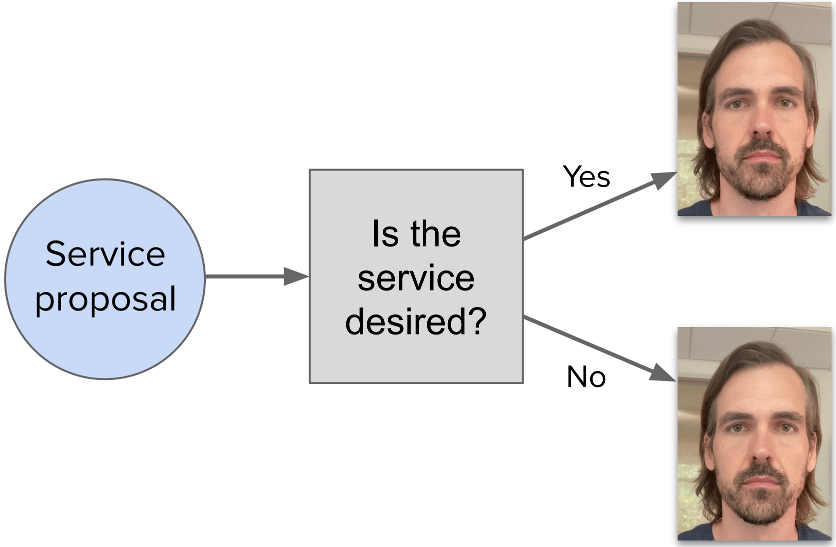
this talk

A model of human preference

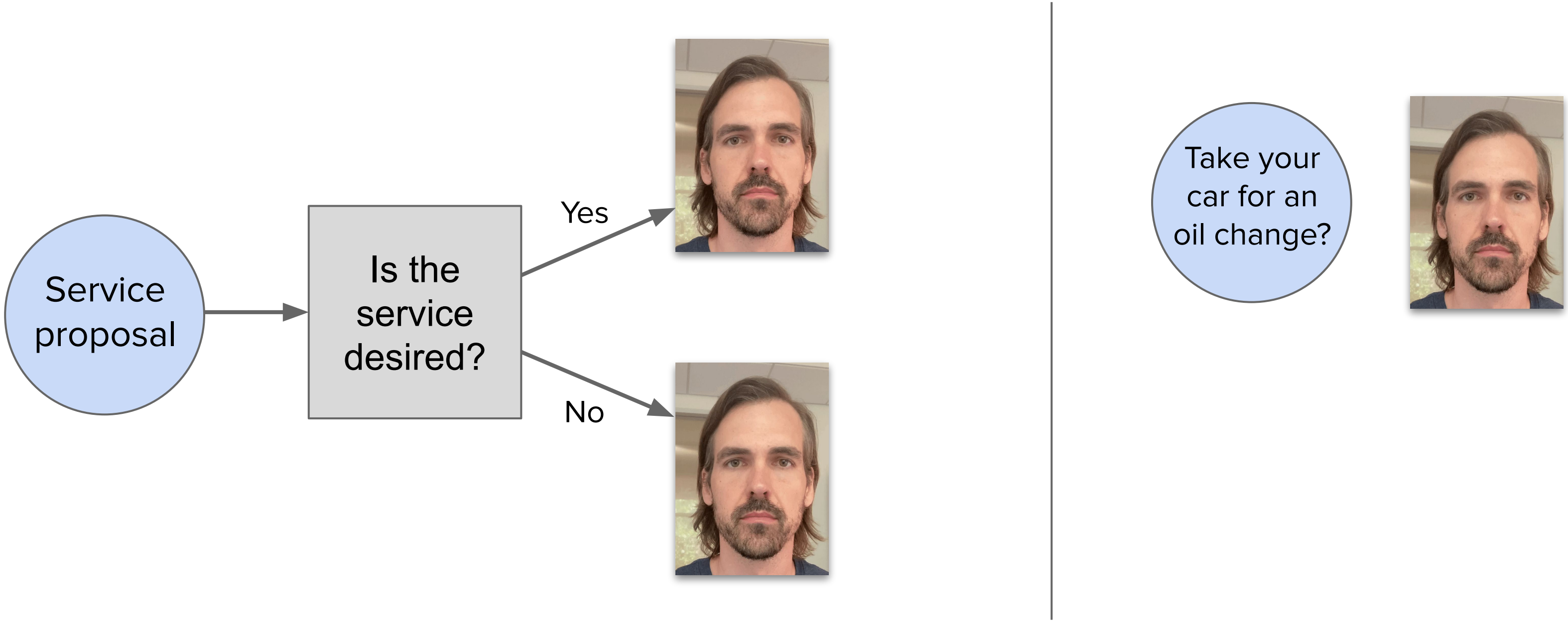
~~A model of human preference~~
A model of human head motion

~~A model of human preference~~
A model of human head motion

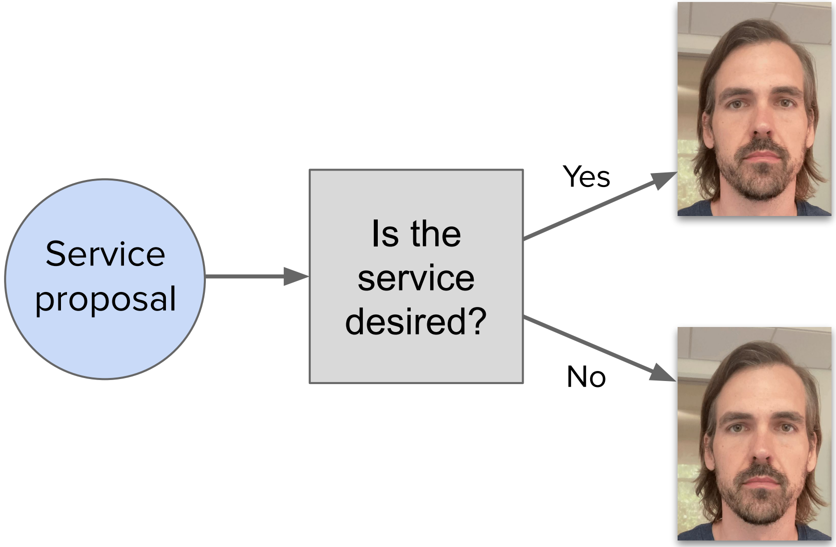
~~A model of human preference~~
A model of human head motion



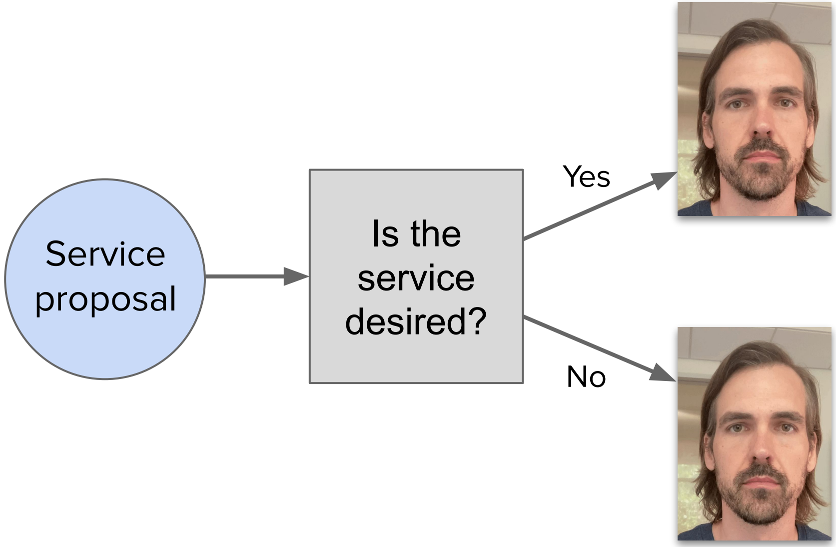
~~A model of human preference~~
A model of human head motion



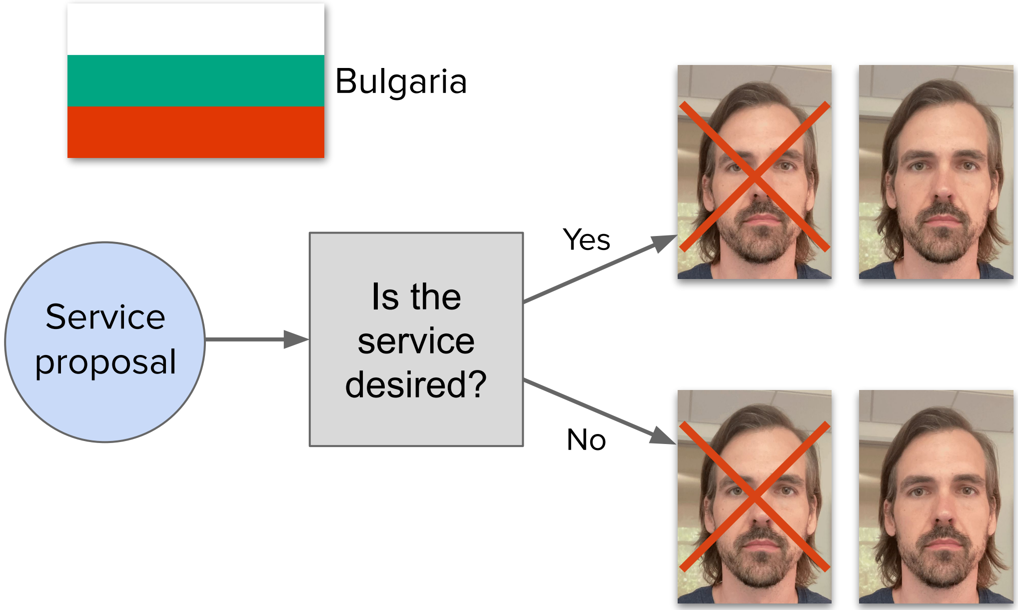
~~A model of human preference~~
A model of human head motion

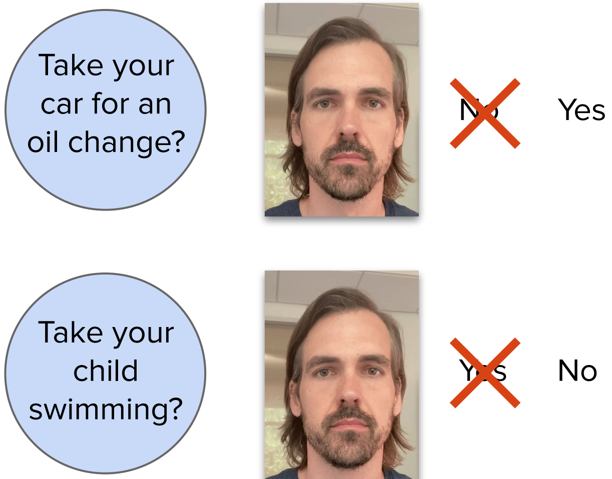


~~A model of human preference~~
A model of human head motion



~~A model of human preference~~
A model of human head motion





Takeaways

A key part of the current model for what drives human preferences in sequential tasks is unstudied and unvalidated.

Regret is an improved preference model that measures a segment's deviation from optimality.

The model of human preference is a critical piece for alignment.

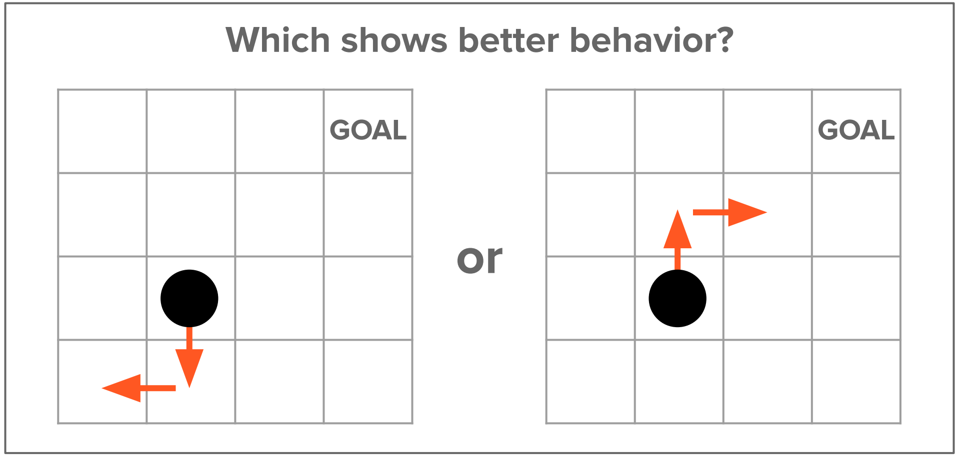
BACKGROUND ON REWARD

$$\underbrace{G(\tau)}_{} = \sum_{t=1}^{(T-1)} \underbrace{r\left(s_t, a_t, s_{t+1}\right)}_{}$$

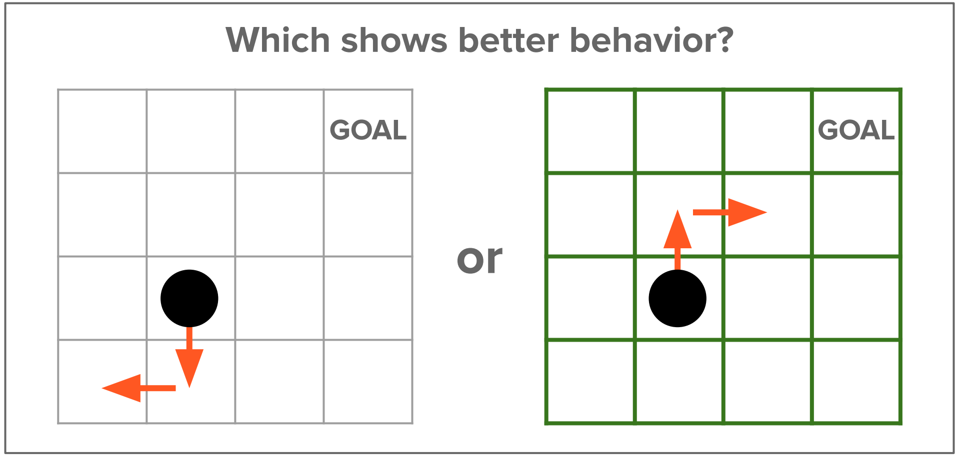
Field		
reinf. learning	return	reward
motion planning	-1 × cost	-1 × cost
control theory	-1 × cost	-1 × cost
evolutionary algs.	fitness	-
utility theory	utility	-
optimization	objective*	-
-	performance metric	-
-	score	-

* "Objective" more precisely refers to the goal of maximizing or minimizing the expectation of $G(\tau)$.

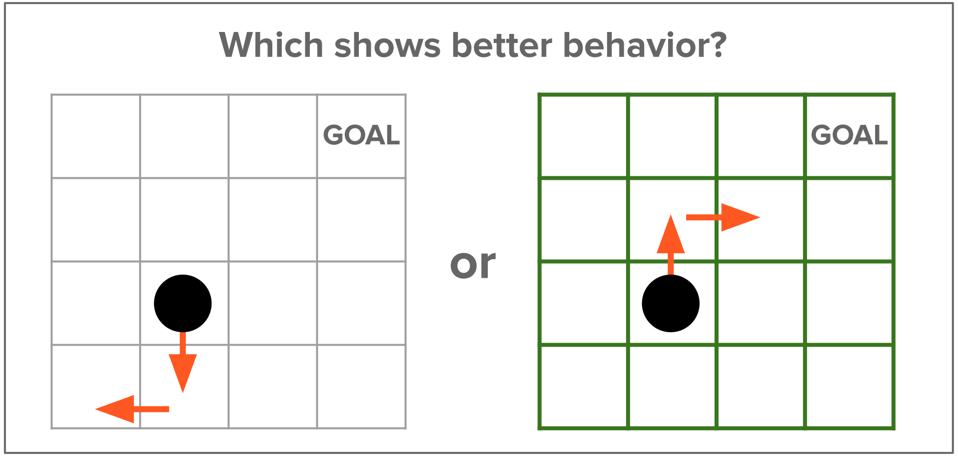
Preferences over segment pairs



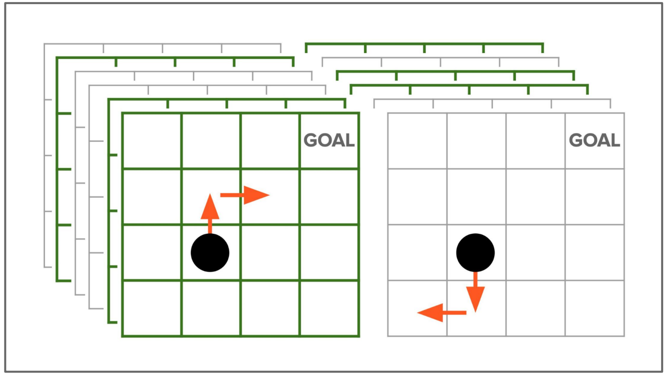
Preferences over segment pairs



Preferences over segment pairs

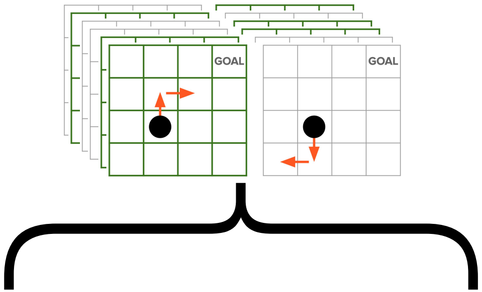


Preference elicitation
(or generation)



Preferences dataset

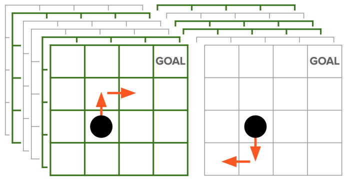
Learning a reward function from preferences



Given a preference model $P(\sigma_1 \succ \sigma_2 | \hat{r})$,

optimize $\hat{r}$ to maximize the likelihood of the *preferences dataset*.

Learning a reward function from preferences



Given a preference model $P(\sigma_1 \succ \sigma_2 | \hat{r})$,

optimize $\hat{r}$ to maximize the likelihood of the *preferences dataset*.

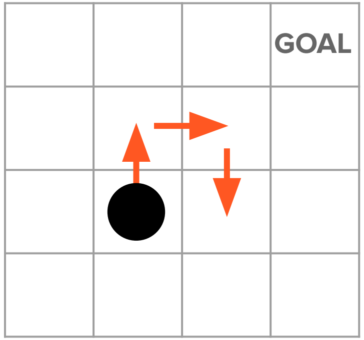
Likelihood as cross entropy loss

$$loss(\hat{r}, D_\succ) = -\sum_{(\sigma_1, \sigma_2, \mu) \in D_\succ} \mu_1 \log P(\sigma_1 \succ \sigma_2 | \hat{r}) + \mu_2 \log P(\sigma_1 \prec \sigma_2 | \hat{r})$$

Why preferences?

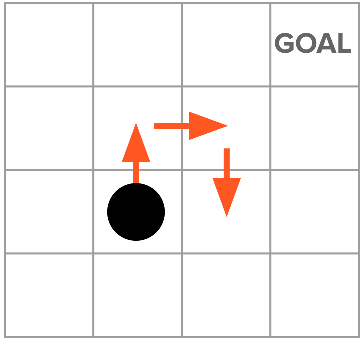
- Established technique in reward learning
- Intuitive for humans
- Judgment may be easier than control
- Connects to expected utility theory
- *In ideal settings, the reward function underlying the preferences can be recovered*

(Trajectory) segment notation



Segment σ

(Trajectory) segment notation

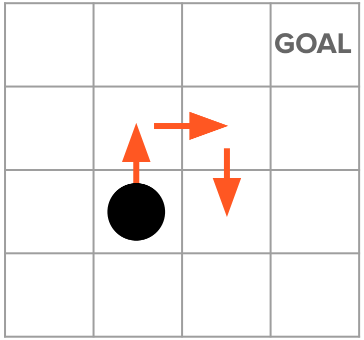


Segment σ

$|\sigma| = 3$

- The segment length
- The number of transitions in the segment

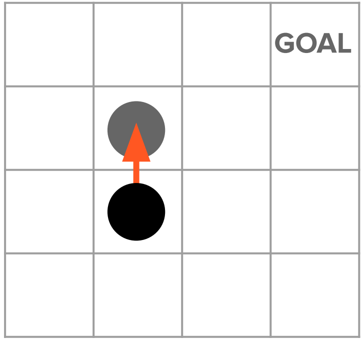
(Trajectory) segment notation



Segment σ

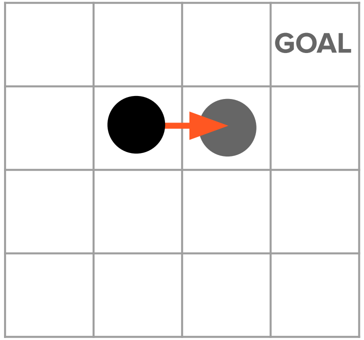
$|\sigma| = 3$

- The segment size
- The number of transitions in the segment

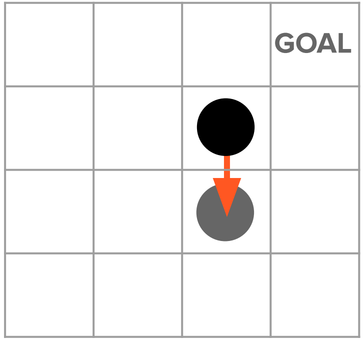


$$\underline{\sigma_t \triangleq (s_{\sigma,t}, a_{\sigma,t}, s_{\sigma,t+1})}$$

$$\sigma_0 = (s_{\sigma,0}, a_{\sigma,0}, s_{\sigma,1})$$

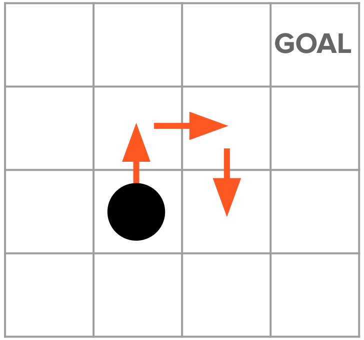


$$\sigma_1 = (s_{\sigma,1}, a_{\sigma,1}, s_{\sigma,2})$$



$$\sigma_2 = (s_{\sigma,2}, a_{\sigma,2}, s_{\sigma,3}) = (s_{\sigma,2}, a_{\sigma,2}, s_{\sigma,|\sigma|})$$

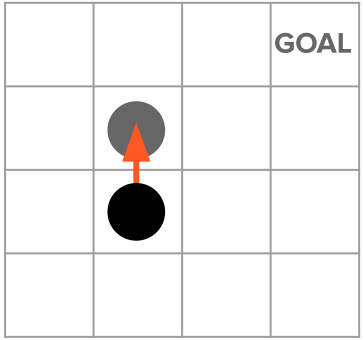
(Trajectory) segment notation



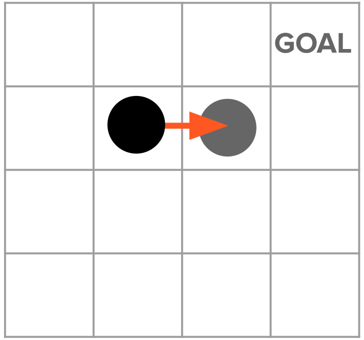
Segment σ

$|\sigma| = 3$

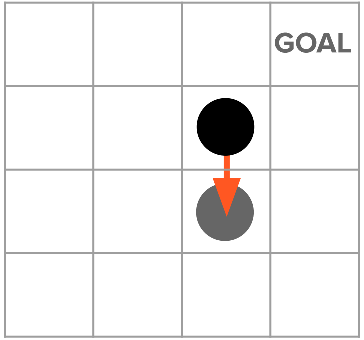
- The segment length
- The number of transitions in the segment



$$(s_{\sigma,0}, a_{\sigma,0}, s_{\sigma,1})$$



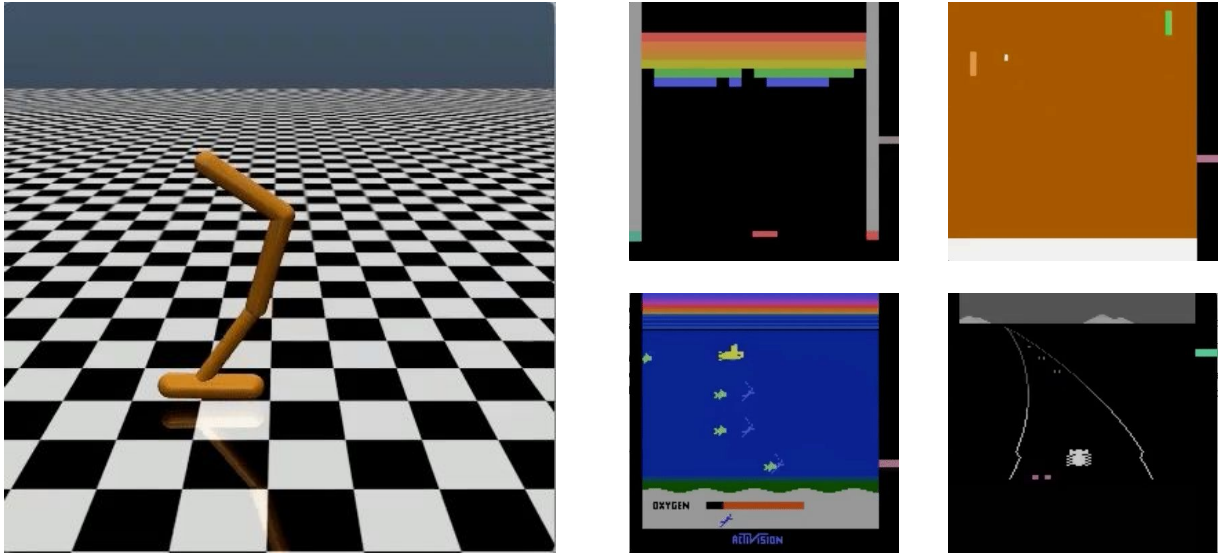
$$(s_{\sigma,1}, a_{\sigma,1}, s_{\sigma,2})$$



$$(s_{\sigma,2}, a_{\sigma,2}, s_{\sigma,3})$$

$$(s_{\sigma,2}, a_{\sigma,2}, s_{\sigma,|\sigma|})$$

Learning a reward function from preferences (related work)



Christiano et al., 2017 - deep reward function representations

Learning a reward function from preferences (related work)

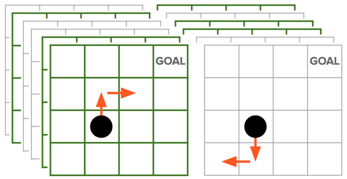
Fine-tuning large language models (LLMs)

ChatGPT

Ouyang et al., 2022

Learning a reward function from preferences (related work)

Sadigh et al., 2017 - active learning



Given a preference model $P(\sigma_1 \succ \sigma_2 | \hat{r})$,

optimize $\hat{r}$ to maximize the likelihood of the *preferences dataset*.

Christiano et al., 2017 - deep reward

Ibarz et al., 2018 - add demonstrations
Bıyık et al. 2021

Lee et al. 2021 - benchmark for learning from preferences
Wang et al. 2022 - extracting skills too from preferences
Lee et al. 2022 - pre-training and reward-relabeled replay

Models of human preference for learning reward functions


W. Bradley Knox[†,1,2]


Stephane Hatgis-Kessell[1,2]


Serena Booth[1,3]


Scott Niekum[‡,2]


Peter Stone[2,4]


Alessandro Allievi[1,2]

[1]Bosch [2]The University of Texas at Austin [3]MIT CSAIL [4]Sony AI

* First two authors contributed equally

[†]Now affiliated with University of Texas at Austin

[‡]Now affiliated with University of Massachusetts Amherst

Outline:

- Preference models
- Identifiability theory of preference models
- Performance with each preference model

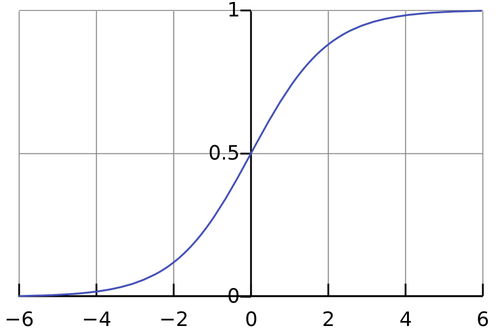
2nd half: When our proposed model drives preferences but the dominant model is assumed

Models of human preference

The missing piece: the model of preference

$$P(\sigma_1 \succ \sigma_2) = \frac{\exp\left[f(\sigma_1)\right]}{\exp\left[f(\sigma_1)\right] + \exp\left[f(\sigma_2)\right]}$$

$$= logistic(f(\sigma_1) - f(\sigma_2))$$



$f(\sigma_1) - f(\sigma_2)$

(Shorthand notation above leaves out from P and f an implied reward function as input.)

The missing piece: the model of preference

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

The missing piece: the model of preference

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Current dominant model:

Partial return

$$f(\sigma) = \text{discounted sum of reward in } \sigma, \quad \boxed{\sum_{t=0}^{|\sigma|-1} \gamma^t r(s_t, a_t)}$$

Partial return is assumed by all related work I covered.

The missing piece: the model of preference

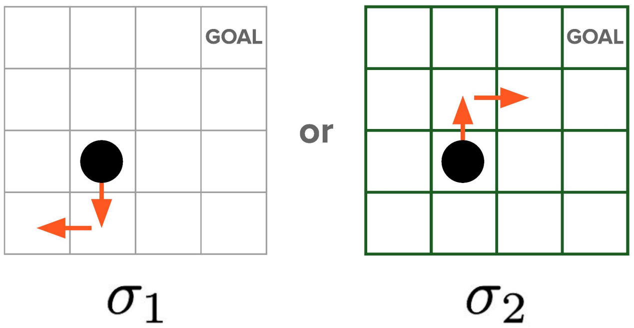
$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Partial return: $f(\sigma) = \text{discounted sum of reward in } \sigma$

Which shows better behavior?

Assume -1 reward per step
and no discounting.

Partial return is indifferent!



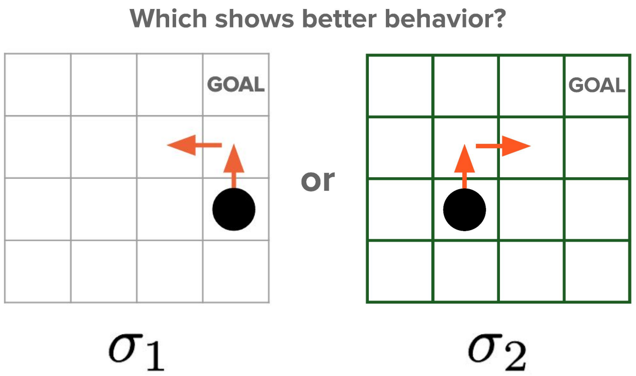
The missing piece: the model of preference

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Partial return: $f(\sigma) = \text{discounted sum of reward in } \sigma$

Assume -1 reward per step and no discounting.

Partial return is indifferent!



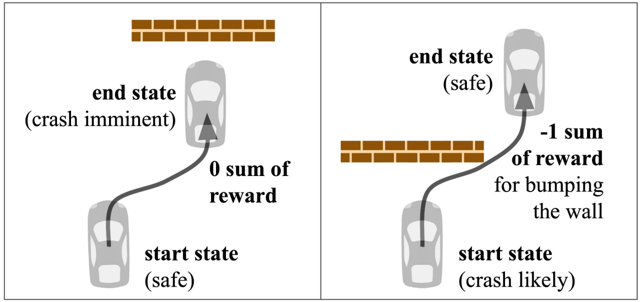
The missing piece: the model of preference

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Partial return: $f(\sigma) = \text{discounted sum of reward in } \sigma$

Partial return prefers the left segment!

Which shows better behavior?

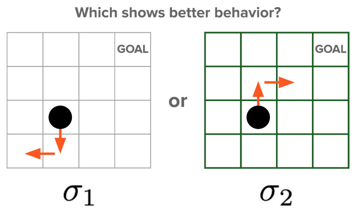


σ_1 σ_2

Problems with the partial return preference model

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Partial return: $f(\sigma) = \text{discounted sum of reward in } \sigma$



Issue:

Humans intuitively appear to consider state value and decision quality. The partial return preference model does not.

Let's address these concerns.

The missing piece: the model of preference

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Proposed preference model: Regret

$$f(\sigma) = -regret(\sigma)$$

The **regret** of a segment is a measure of its **deviation from optimal decision-making.**

The missing piece: the model of preference

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Proposed preference model: Regret

$$f(\sigma) = -regret(\sigma)$$

when all transitions are deterministic

$$regret_d(\sigma|\tilde{r}) \triangleq \sum_{t=0}^{|\sigma|-1} regret_d(\sigma_t|\tilde{r}) = V^*_{\tilde{r}}(s_{\sigma,0}) - (\Sigma_\sigma \tilde{r} + V^*_{\tilde{r}}(s_{\sigma,|\sigma|}))$$

The missing piece: the model of preference

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Proposed preference model: Regret

$$f(\sigma) = -regret(\sigma)$$

when all transitions are deterministic

$$regret_d(\sigma|\tilde{r}) \triangleq \sum_{t=0}^{|\sigma|-1} regret_d(\sigma_t|\tilde{r}) = V^*_{\tilde{r}}(s_{\sigma,0}) - (\boxed{\Sigma_\sigma \tilde{r}} + V^*_{\tilde{r}}(s_{\sigma,|\sigma|}))$$

Partial return

The missing piece: the model of preference

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Proposed preference model: Regret

$$f(\sigma) = -regret(\sigma)$$

when all transitions are deterministic →

$$regret_d(\sigma|\tilde{r}) \triangleq \sum_{t=0}^{|\sigma|-1} regret_d(\sigma_t|\tilde{r}) = V^*_{\tilde{r}}(s_{\sigma,0}) - \Big(\boxed{\Sigma_\sigma \tilde{r}} + \boxed{V^*_{\tilde{r}}(s_{\sigma,|\sigma|})}\Big)$$

Partial return

Best possible expected return from the *end* state (i.e., by optimal policy)

The missing piece: the model of preference

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Proposed preference model: Regret

$$f(\sigma) = -regret(\sigma)$$

Best possible expected return from the *start* state given the segment σ

when all transitions are deterministic

$$regret_d(\sigma|\tilde{r}) \triangleq \sum_{t=0}^{|\sigma|-1} regret_d(\sigma_t|\tilde{r}) = V^*_{\tilde{r}}(s_{\sigma,0}) - \left(\boxed{\Sigma_\sigma \tilde{r}} + \boxed{V^*_{\tilde{r}}(s_{\sigma,|\sigma|})}\right)$$

Partial return

Best possible expected return from the *end* state (i.e., by optimal policy)

The missing piece: the model of preference

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Proposed preference model: Regret

$$f(\sigma) = -regret(\sigma)$$

when all transitions are deterministic

$$regret_d(\sigma|\tilde{r}) \triangleq \sum_{t=0}^{|\sigma|-1} regret_d(\sigma_t|\tilde{r}) = \boxed{V^*_{\tilde{r}}(s_{\sigma,0})} - \Big(\overbrace{\boxed{\Sigma_\sigma \tilde{r}} + \boxed{V^*_{\tilde{r}}(s_{\sigma,|\sigma|})}}^{\text{Best possible expected return from the } \textit{start} \text{ state given the segment } \sigma}\Big)$$

Best possible expected return from the *start* state (i.e., by optimal policy)

Partial return

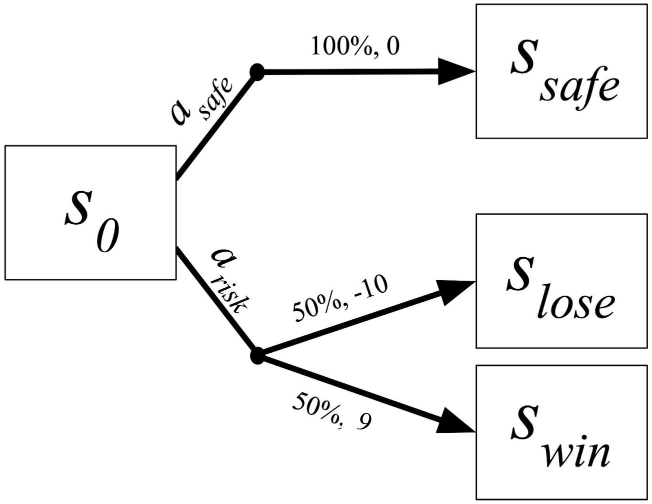
Best possible expected return from the *end* state (i.e., by optimal policy)

What if transitions can be stochastic?

when all transitions are deterministic →

$$regret_d(\sigma|\tilde{r}) \triangleq \sum_{t=0}^{|\sigma|-1} regret_d(\sigma_t|\tilde{r}) = V^*_{\tilde{r}}(s_{\sigma,0}) - (\Sigma_\sigma \tilde{r} + V^*_{\tilde{r}}(s_{\sigma,|\sigma|}))$$

The lottery:



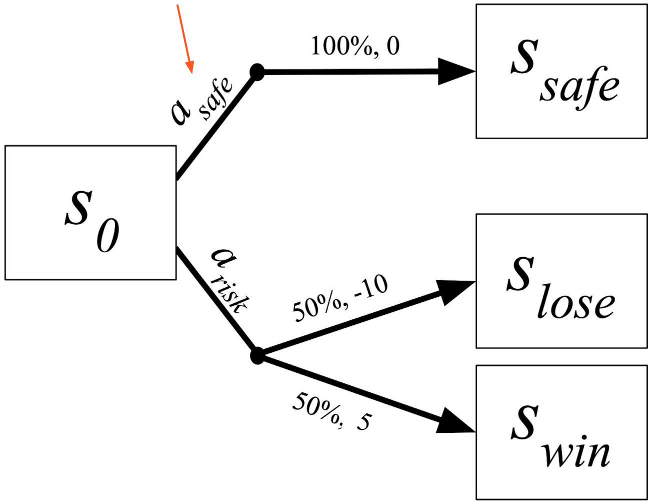
What if transitions can be stochastic?

when all transitions are deterministic

$$regret_d(\sigma|\tilde{r}) \triangleq \sum_{t=0}^{|\sigma|-1} regret_d(\sigma_t|\tilde{r}) = V^*_{\tilde{r}}\cancel{(s_{\sigma,0})} - (\Sigma_\sigma \tilde{r} + V^*_{\tilde{r}}\cancel{(s_{\sigma,|\sigma|})})$$

The lottery:

Best decision



Best outcome

The missing piece: the model of preference

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Proposed preference model: Regret

$$f(\sigma) = -regret(\sigma)$$

$$= \text{sum of } A^*(s,a) \text{ for each } (s,a) \text{ in } \sigma$$

when all transitions are deterministic

$$regret_d(\sigma|\tilde{r}) \triangleq \sum_{t=0}^{|\sigma|-1} regret_d(\sigma_t|\tilde{r}) = V^*_{\tilde{r}}(s_{\sigma,0}) - (\Sigma_\sigma \tilde{r} + V^*_{\tilde{r}}(s_{\sigma,|\sigma|}))$$

$$regret(\sigma|\tilde{r}) = \sum_{t=0}^{|\sigma|-1} regret(\sigma_t|\tilde{r}) = \sum_{t=0}^{|\sigma|-1} \Big[V^*_{\tilde{r}}(s_{\sigma,t}) - Q^*_{\tilde{r}}(s_{\sigma,t}, a_{\sigma,t})\Big] = \sum_{t=0}^{|\sigma|-1} -A^*_{\tilde{r}}(s_{\sigma,t}, a_{\sigma,t})$$

The missing piece: the model of preference

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Proposed preference model: Regret

$$f(\sigma) = -regret(\sigma)$$
$$= \text{ discounted sum of } A^*(s,a) \text{ for each } (s,a) \text{ in } \sigma$$

Note: $A^*(s,a) \triangleq Q^*(s,a) - V^*(s)$ and $max_a A_r^*(s,a) = 0 \text{ for all } s$

The missing piece: the model of preference

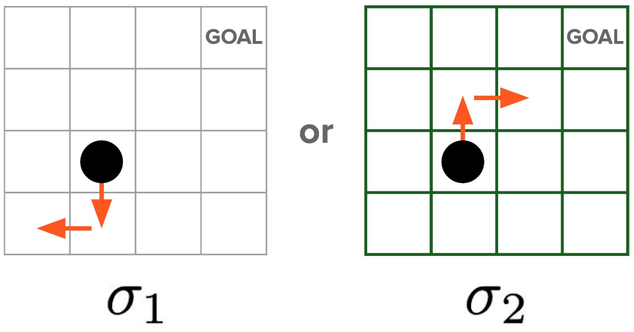
$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Regret: $f(\sigma) =$ discounted sum of $A^*(s,a)$ for each (s,a) in σ

Which shows better behavior?

Assume -1 reward per step
and no discounting.

Regret prefers σ_2.



The missing piece: the model of preference

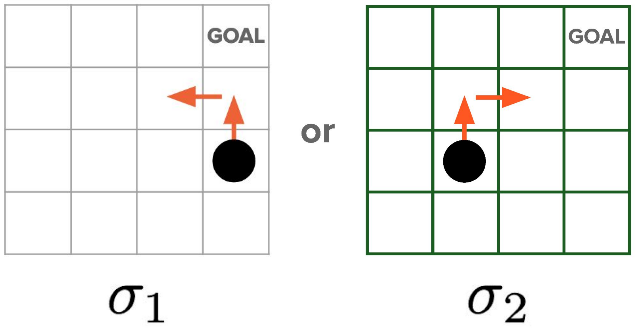
$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Regret: $f(\sigma) =$ discounted sum of $A^*(s,a)$ for each (s,a) in σ

Assume -1 reward per step and no discounting.

Regret prefers σ_2.

Which shows better behavior?



The missing piece: the model of preference

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Regret: $f(\sigma) =$ discounted sum of $A^*(s, a)$ for each (s, a) in σ

Precedent: in IRL, demonstrations are often assumed to noisily optimal (Boltzmann rational with respect to the Q* function).

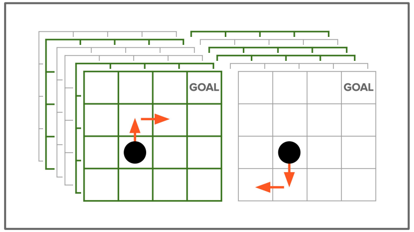
Main downside: Like IRL, learning *reward* with regret appears to require solving an MDP in the inner loop of learning or an approximation of doing so.

Theoretical properties

Reward identifiability

Visual definition:

Ground truth r → $P(\sigma_1 \succ \sigma_2 | r)$ →



Infinitely exhaustive preferences dataset

$\exists$ an algorithm that guarantees …

Equivalent
(same set of optimal policies)

Learned $\hat{r}$

Reward identifiability

Reward is identifiable with **regret**-based preferences for any MDP.

Reward identifiability

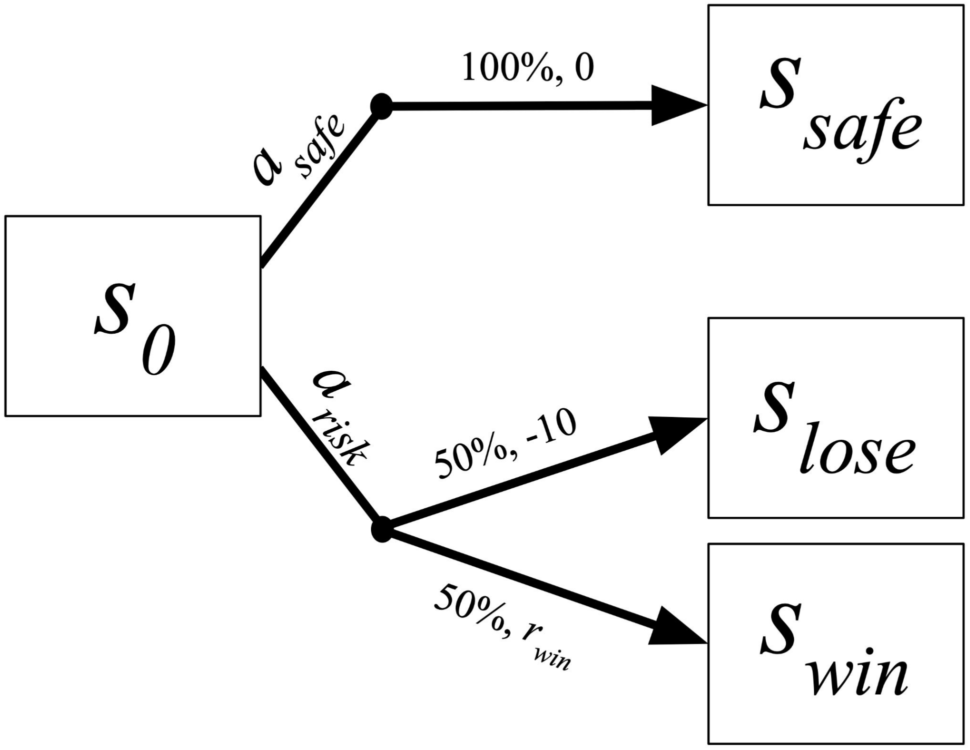
Reward is not identifiable with preferences by **partial return**, in multiple contexts:

- **In variable-horizon tasks**, based upon the model's invariance to a constant shift in the reward function*
- **With segment lengths of 1**, based upon discount factor (γ) ambiguity
- **Without Boltzmann noise in preference labeling**, based upon lotteries requiring preferences over outcome distributions

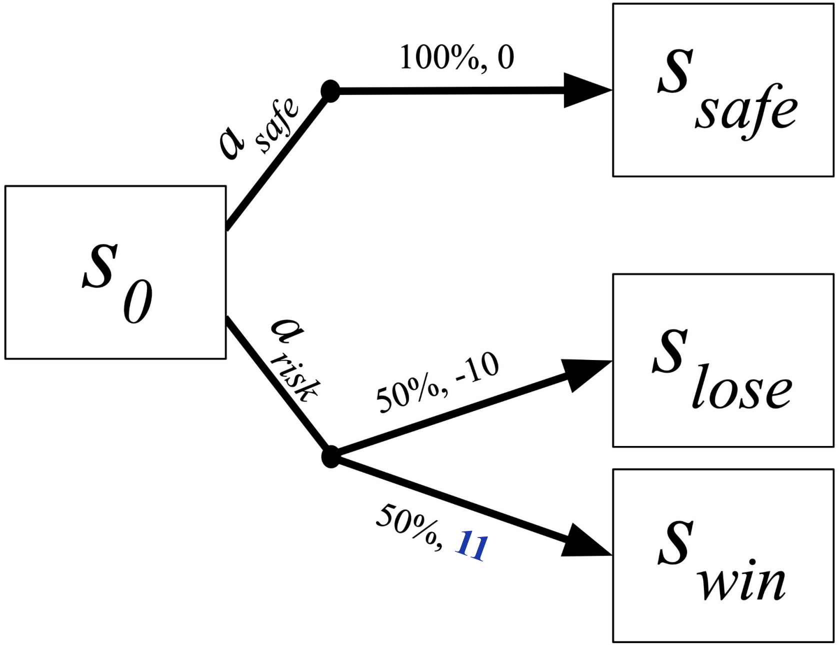
*Under typical settings of each segment in a labeled pair having the same length and not including transitions from absorbing state (which removes the variable horizon attribute).

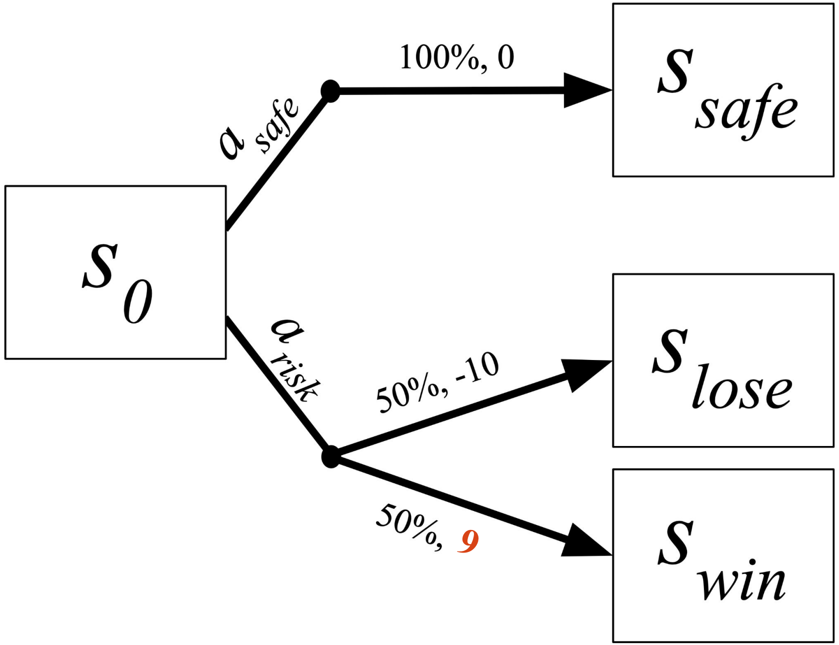
Reward identifiability

With partial return, reward is not generally identifiable without preference noise that reveals rewards' relative proportions.



Reward identifiability



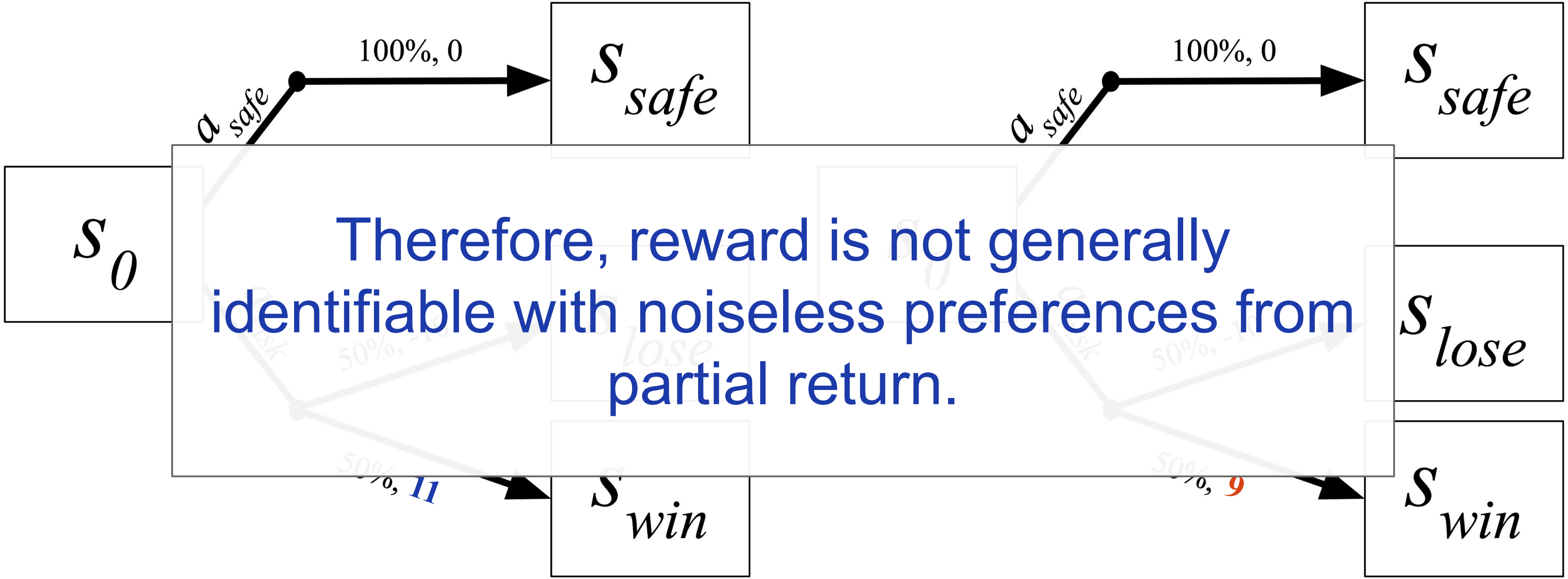


If $r_{win} = 11, a_{risk}$ is optimal.

If $r_{win} = 9$, a_{safe} is optimal.

Yet both create the same (noiseless) preferences!!

Reward identifiability



Therefore, reward is not generally identifiable with noiseless preferences from partial return.

If $r_{win} = 11, a_{risk}$ is optimal. **If $r_{win} = 9$, a_{safe} is optimal.**

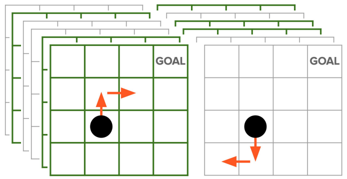
Yet both create the same (noiseless) preferences!!

Reward identifiability

Similarly, reward is **not generally identifiable for inverse reinforcement learning** from (noiseless) demonstrations of optimal behavior.

An algorithm for reward learning with *estimated* regret

Learning a reward function from preferences



Given a preference model $P(\sigma_1 \succ \sigma_2 | \hat{r})$,

optimize $\hat{r}$ to maximize the likelihood of the *preferences dataset*.

Efficiently estimating value functions

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Regret preference model

$$f(\sigma) = -regret(\sigma)$$

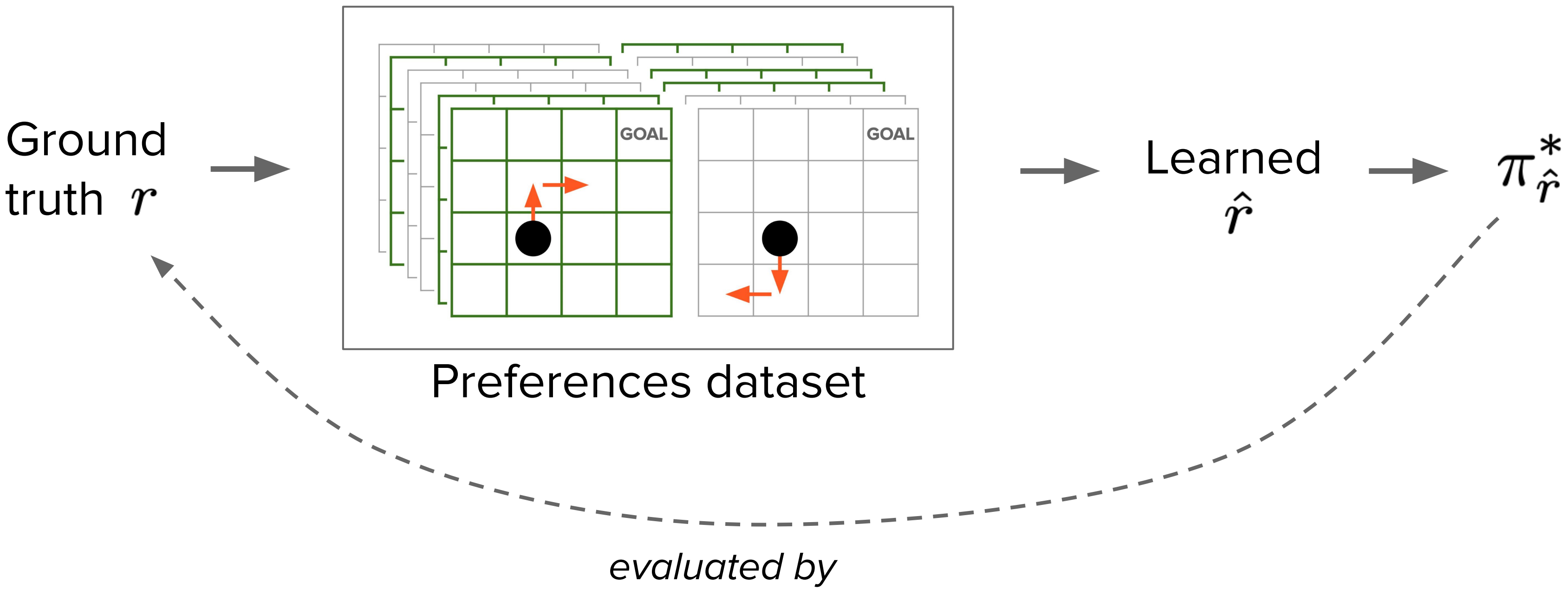
$$= \text{ discounted sum of } A^*(s,a) \text{ for each } (s,a) \text{ in } \sigma$$

$$\boxed{regret(\sigma|\tilde{r}) = \sum_{t=0}^{|\sigma|-1} regret(\sigma_t|\tilde{r}) = \sum_{t=0}^{|\sigma|-1} \left[V^*_{\tilde{r}}(s_{\sigma,t}) - Q^*_{\tilde{r}}(s_{\sigma,t}, a_{\sigma,t})\right] = \sum_{t=0}^{|\sigma|-1} -A^*_{\tilde{r}}(s_{\sigma,t}, a_{\sigma,t})}$$

We assume linear reward functions and use successor features to quickly estimate Q* and V* for new reward parameters.

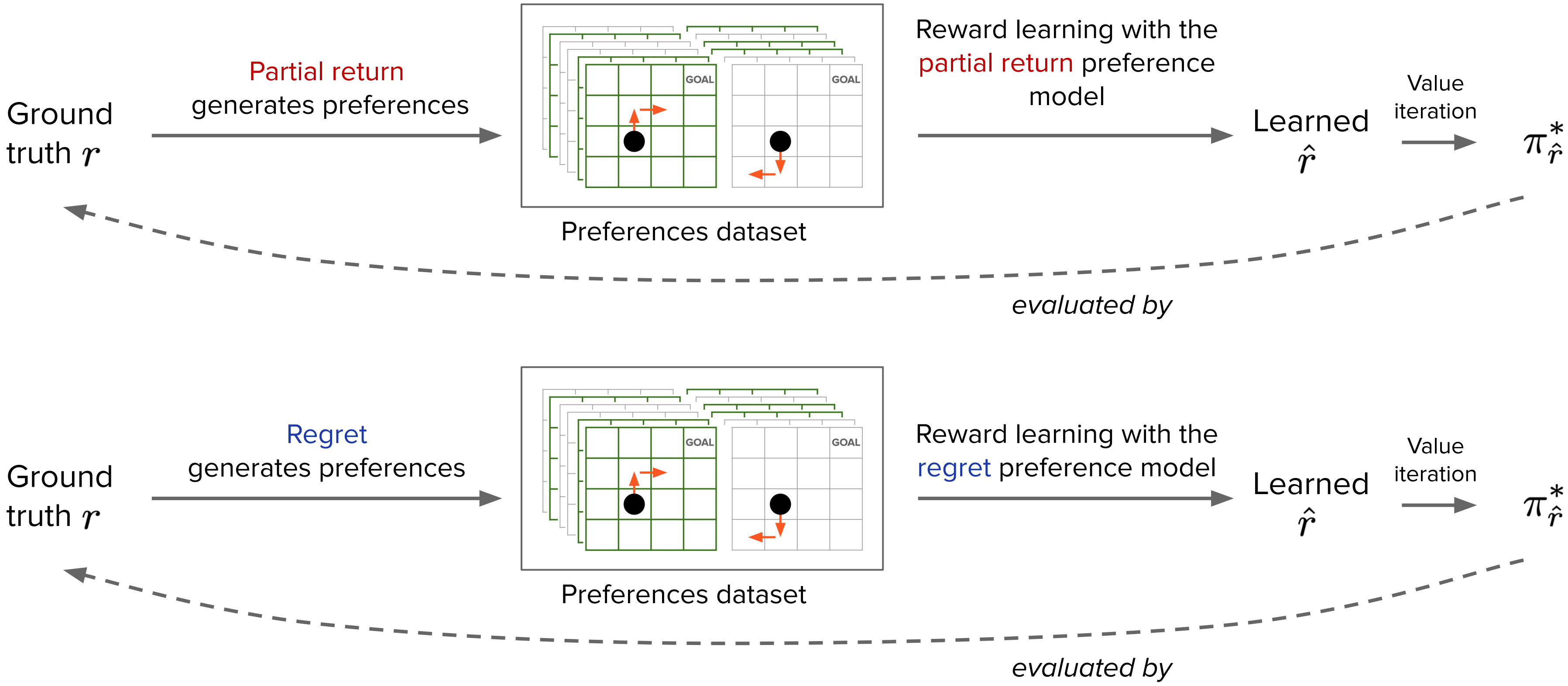
Learning reward functions

Evaluating a learned reward function

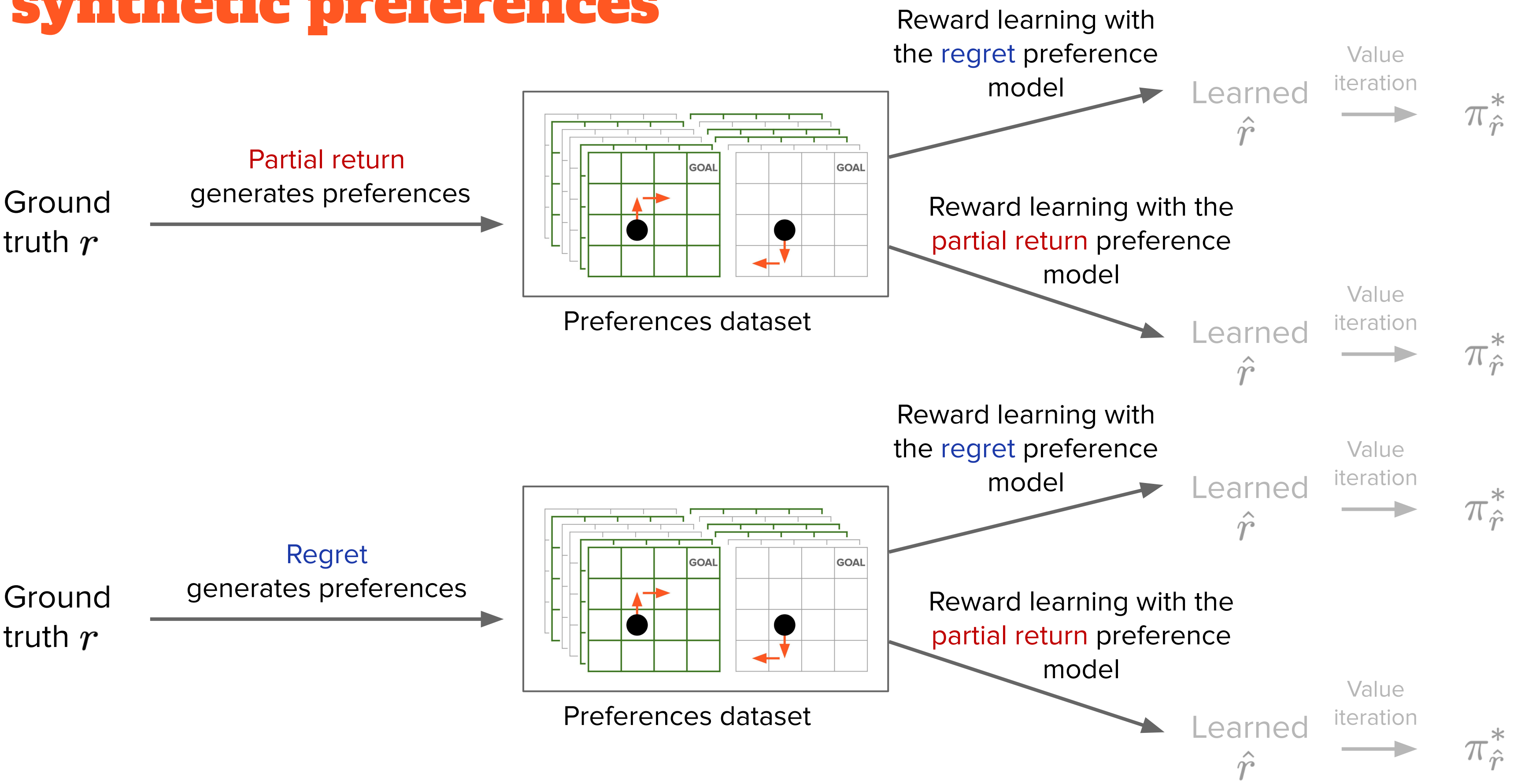


Results, Pt. I: Learning reward functions with synthetic preferences

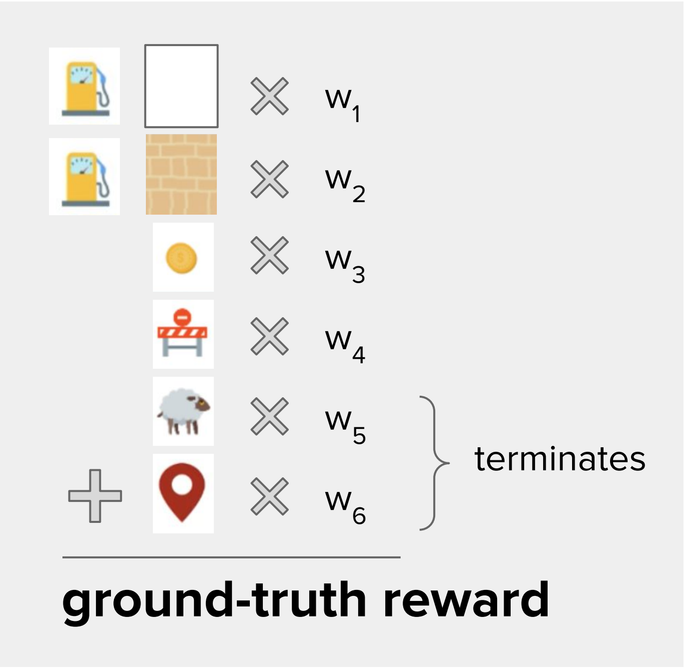
Evaluating a reward function learned from synthetic preferences



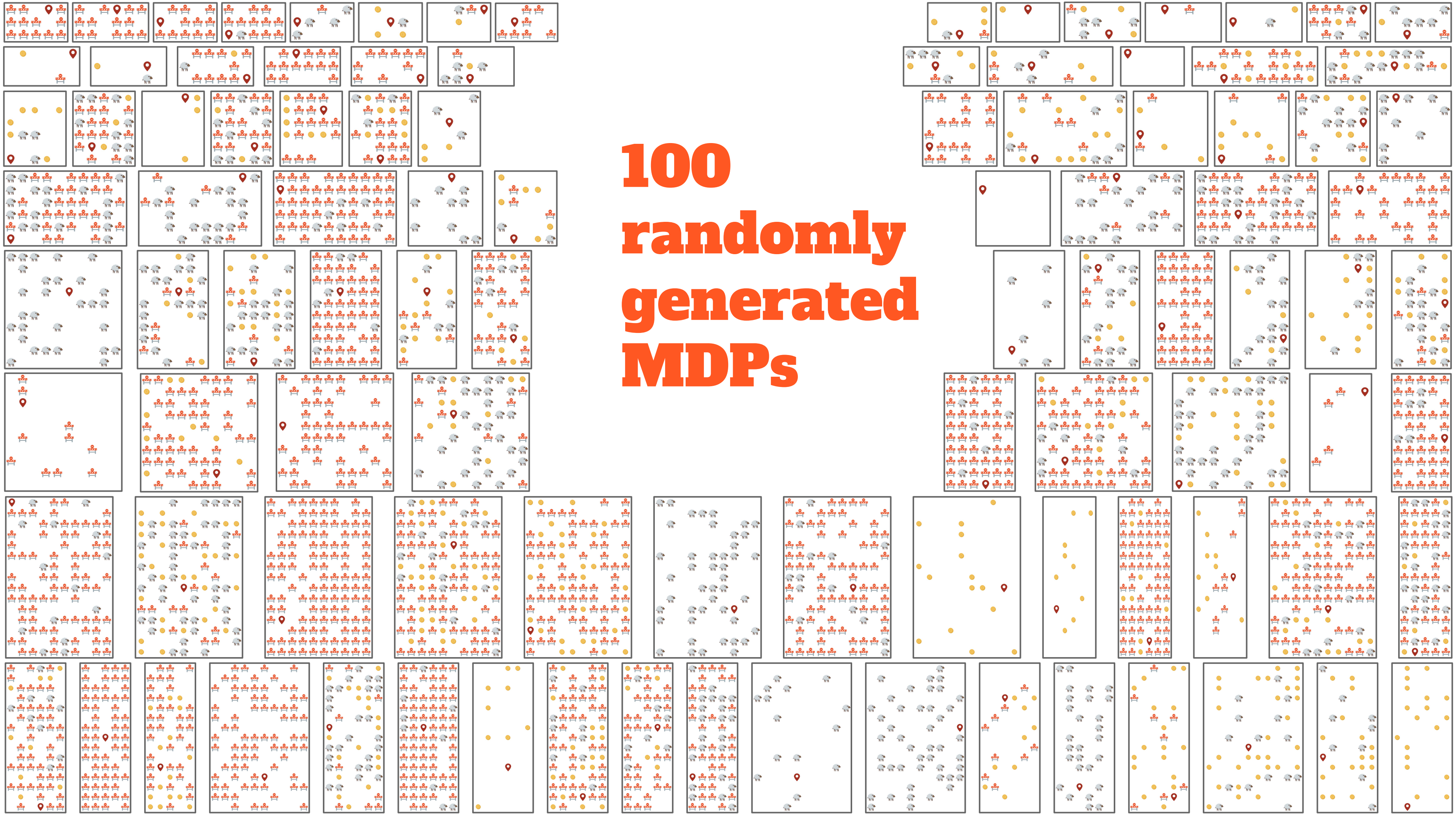
Evaluating a reward function learned from synthetic preferences



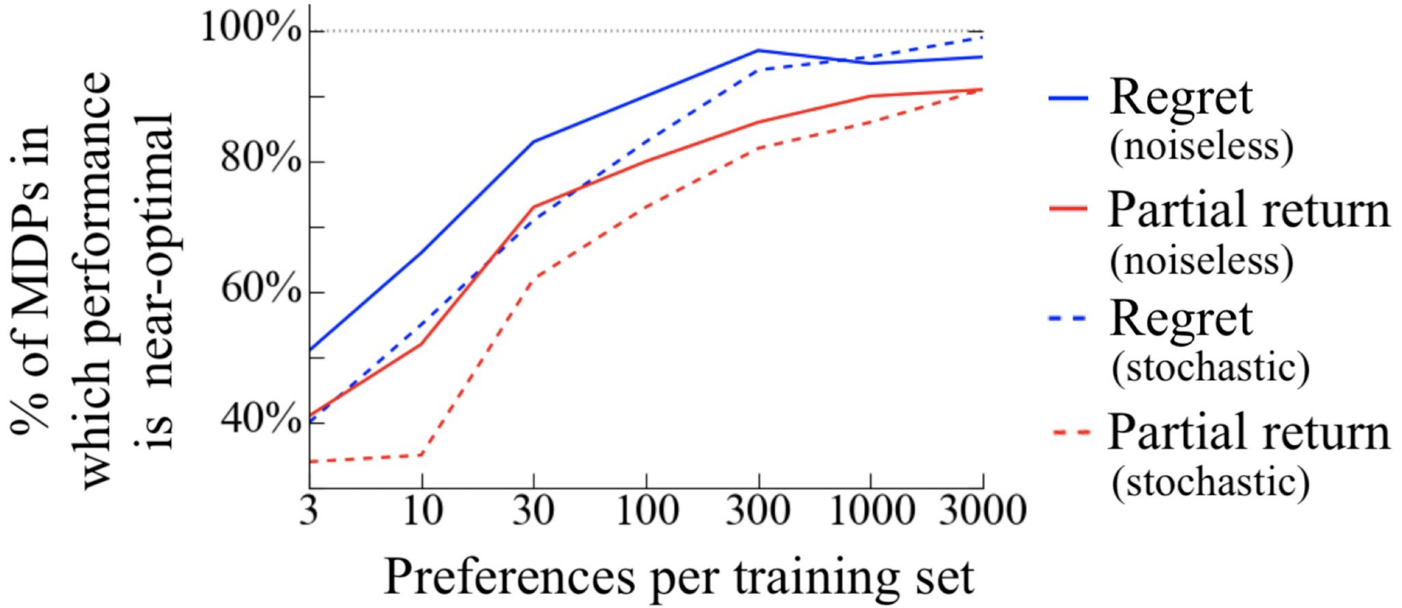
The delivery domain

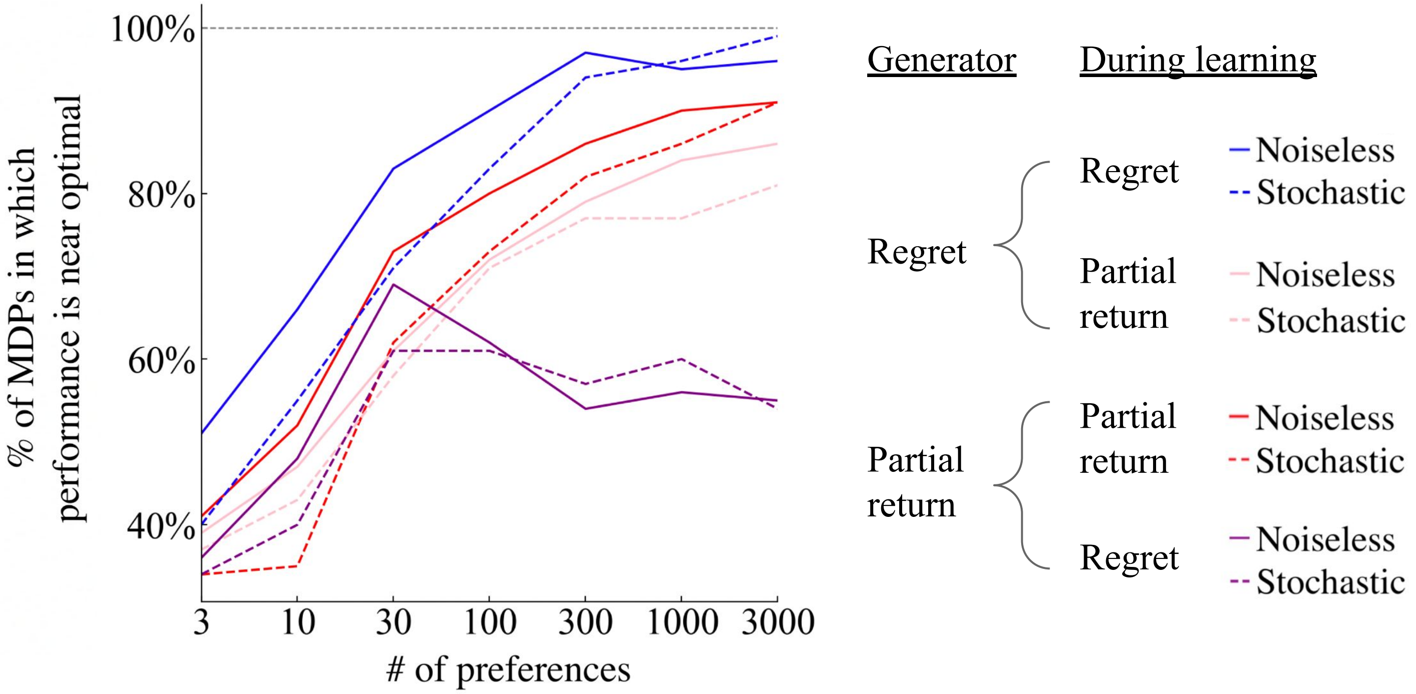


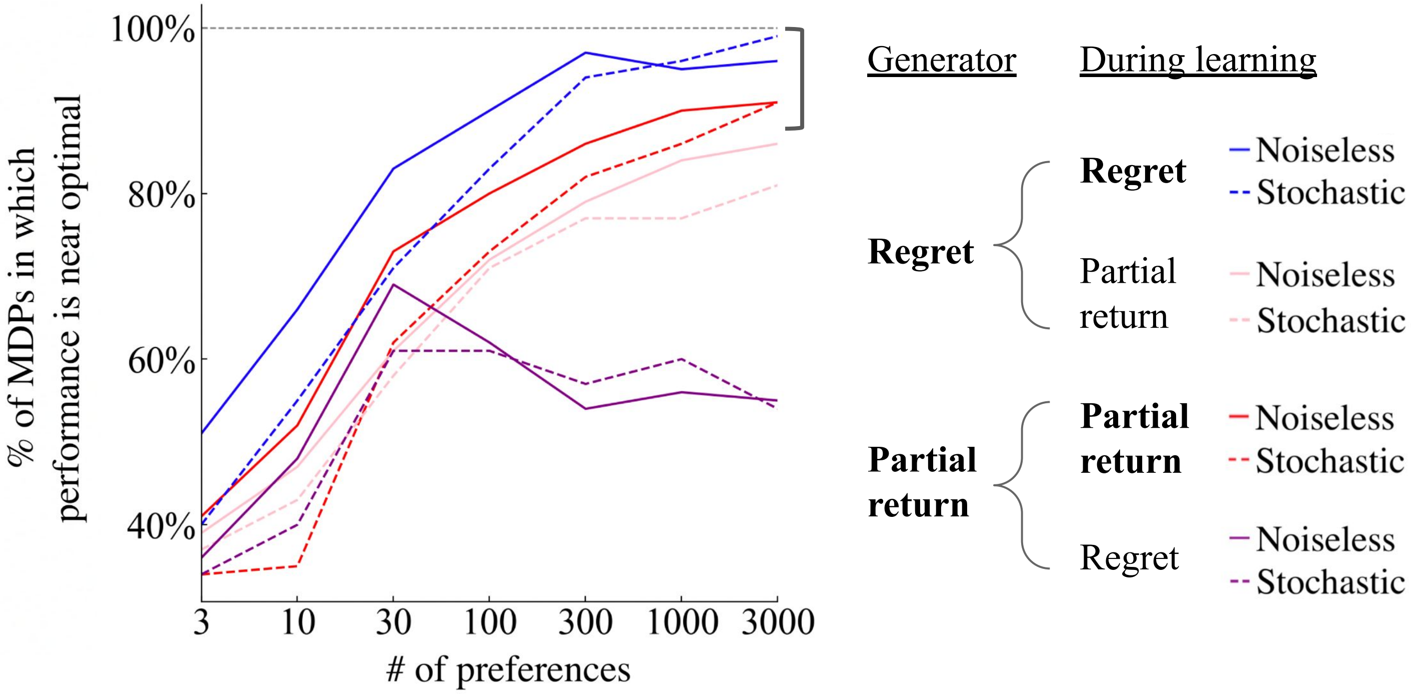
100 randomly generated MDPs



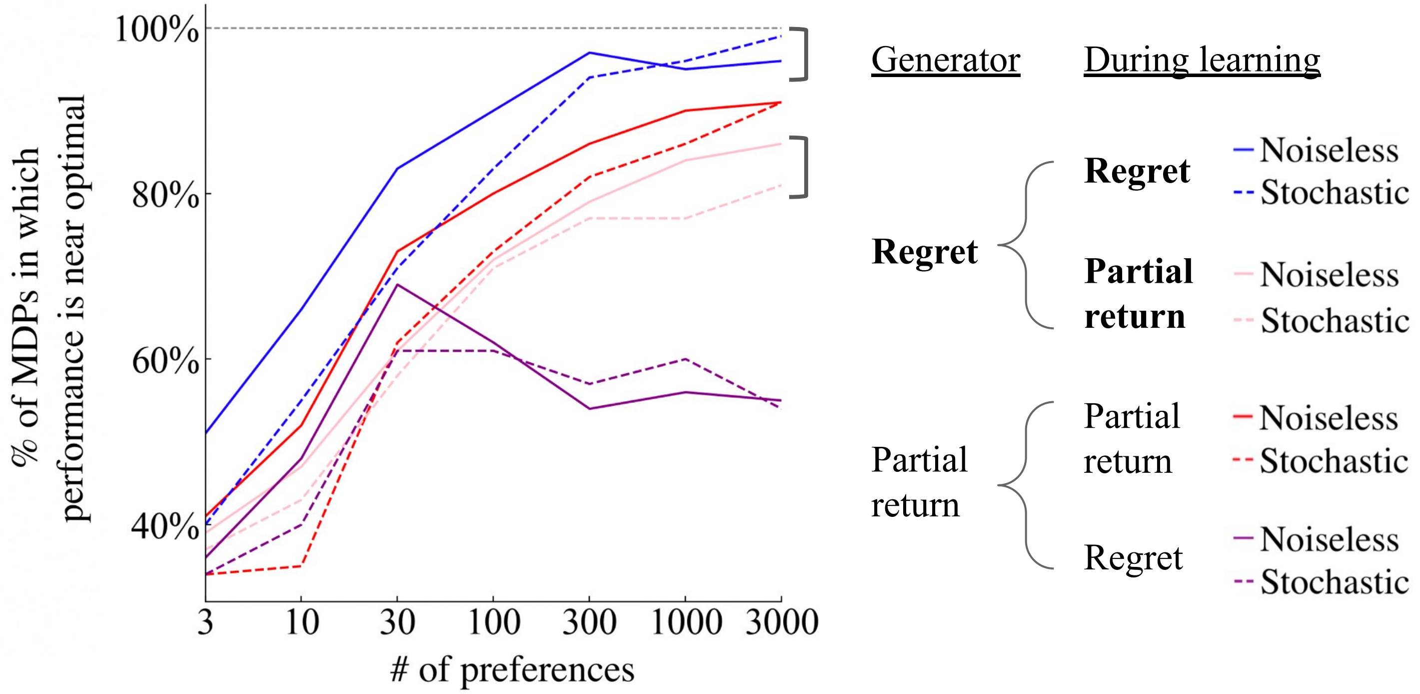
When each model is perfect, because it creates its own preference dataset



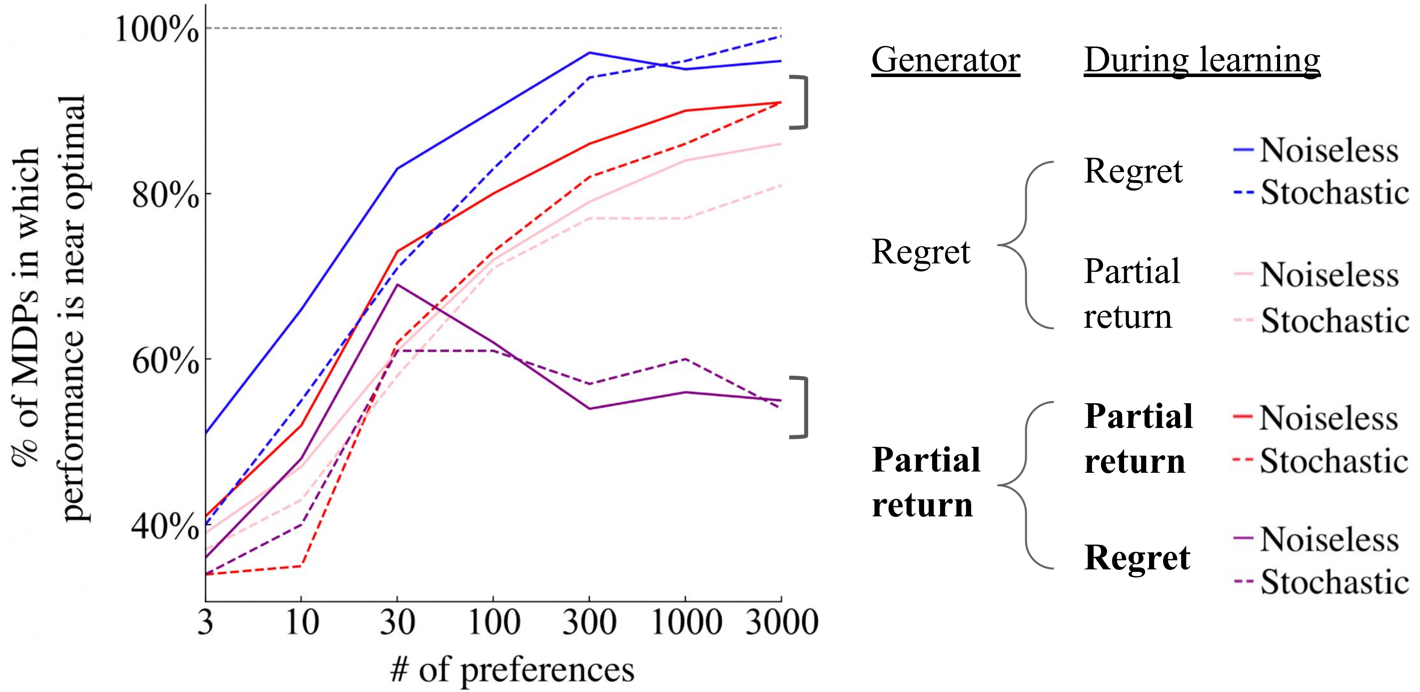

100%
80%
60%
40%
% of MDPs in which performance is near optimal
3
10
30
100
300
1000
3000
# of preferences
Generator
During learning
Regret
Regret
Noiseless
Stochastic
Partial return
Noiseless
Stochastic
Partial return
Partial return
Noiseless
Stochastic
Regret
Noiseless
Stochastic



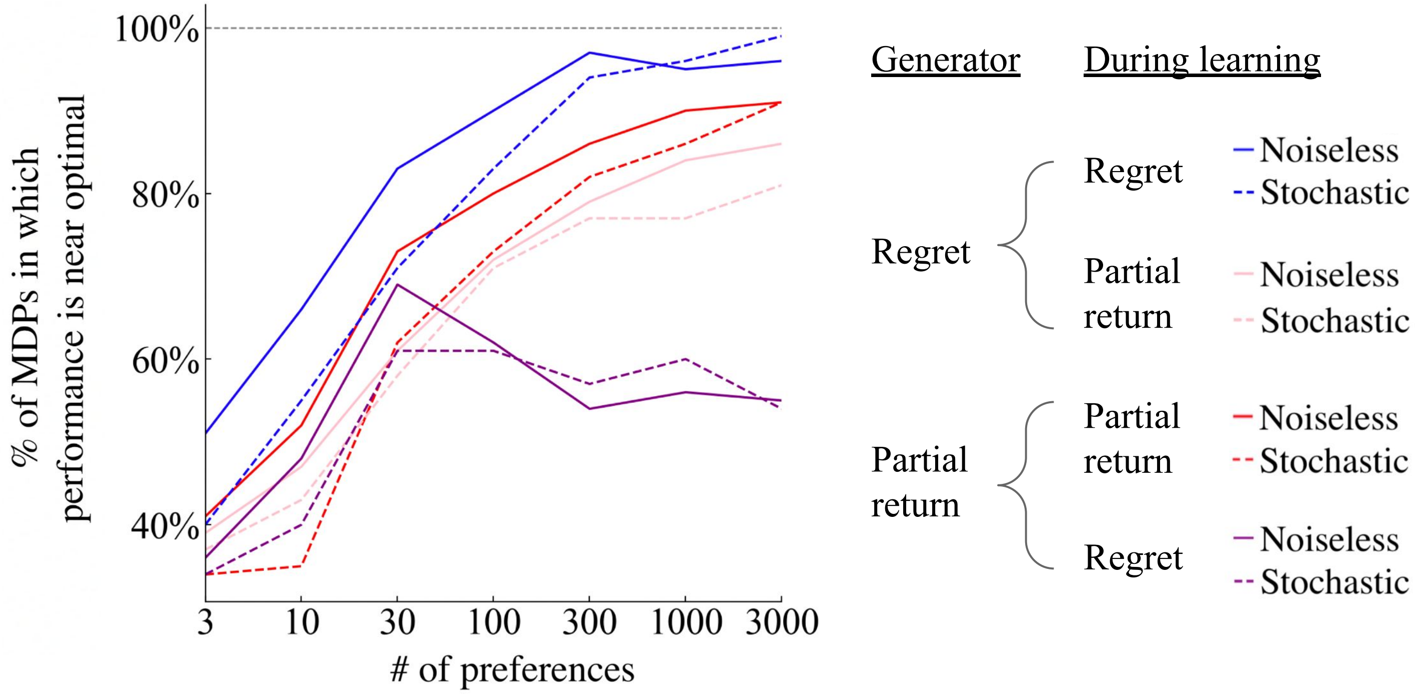
***When each preference model is applied on data it created*, the regret preference model outperforms the partial return model**



100%
80%
60%
40%
% of MDPs in which performance is near optimal
3
10
30
100
300
1000
3000
# of preferences
Generator
During learning
Regret
Regret
Noiseless
Stochastic
Partial return
Noiseless
Stochastic
Partial return
Partial return
Noiseless
Stochastic
Regret
Noiseless
Stochastic



100%
80%
60%
40%
% of MDPs in which performance is near optimal
3
10
30
100
300
1000
3000
# of preferences
Generator
During learning
Regret
Regret
Noiseless
Stochastic
Partial return
Noiseless
Stochastic
Partial return
Partial return
Noiseless
Stochastic
Regret
Noiseless
Stochastic



When a preference model generates a dataset,
***that same model* produces the most aligned reward functions.**

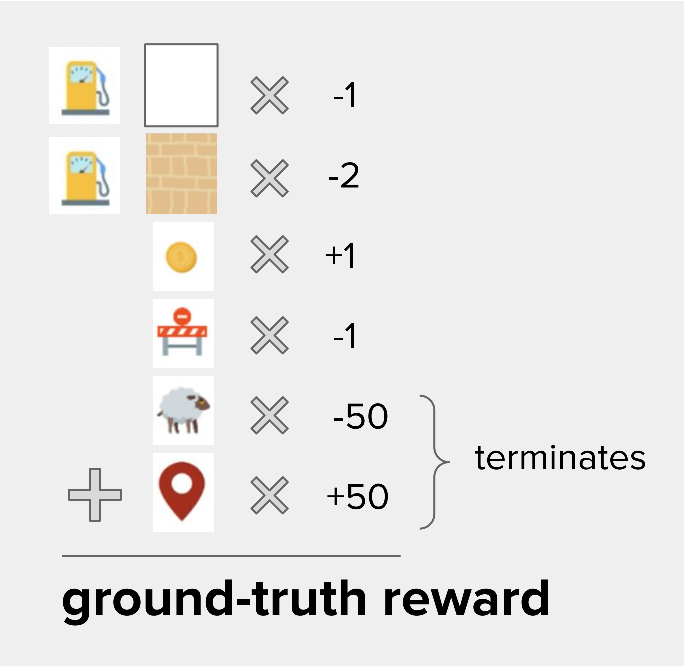
The correctness of the preference model affects alignment!

Results, Pt. II:

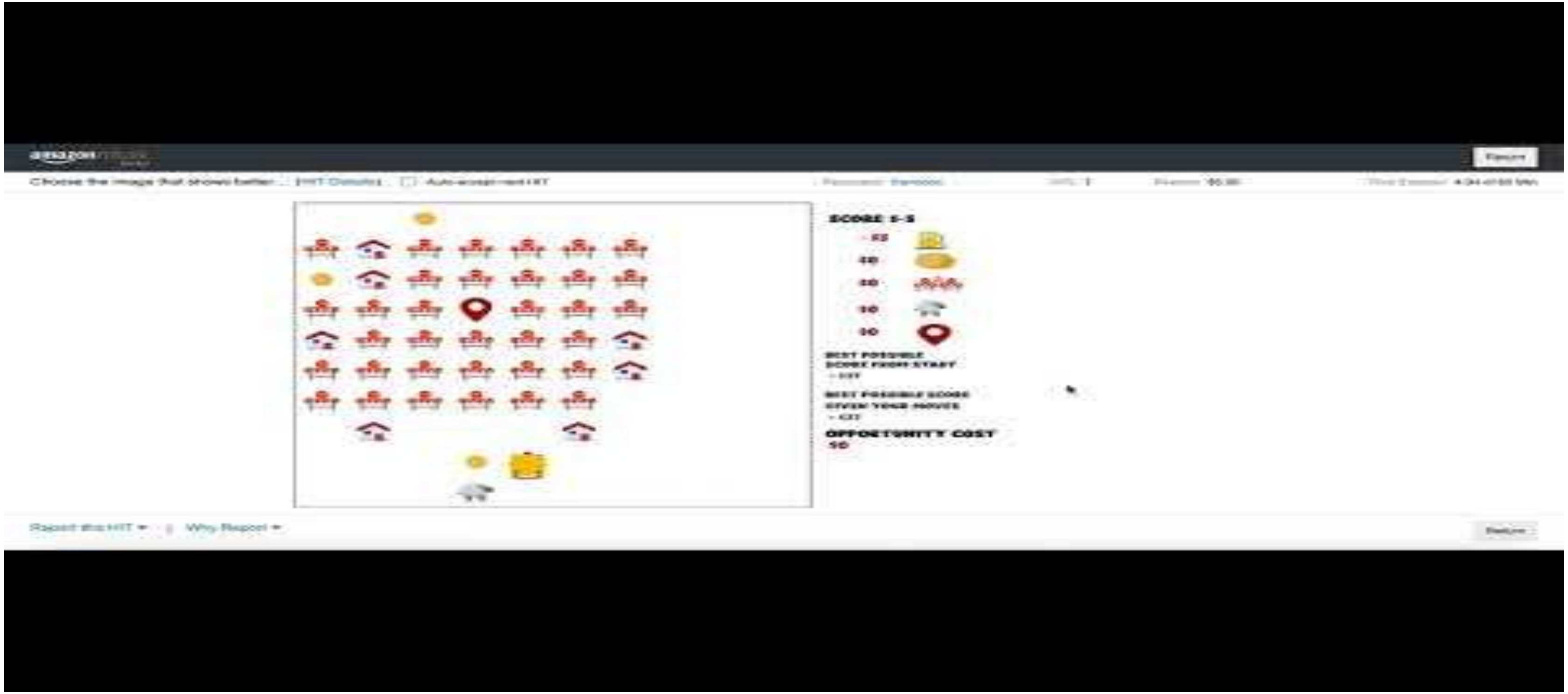
human-generated preferences

A dataset of human preferences

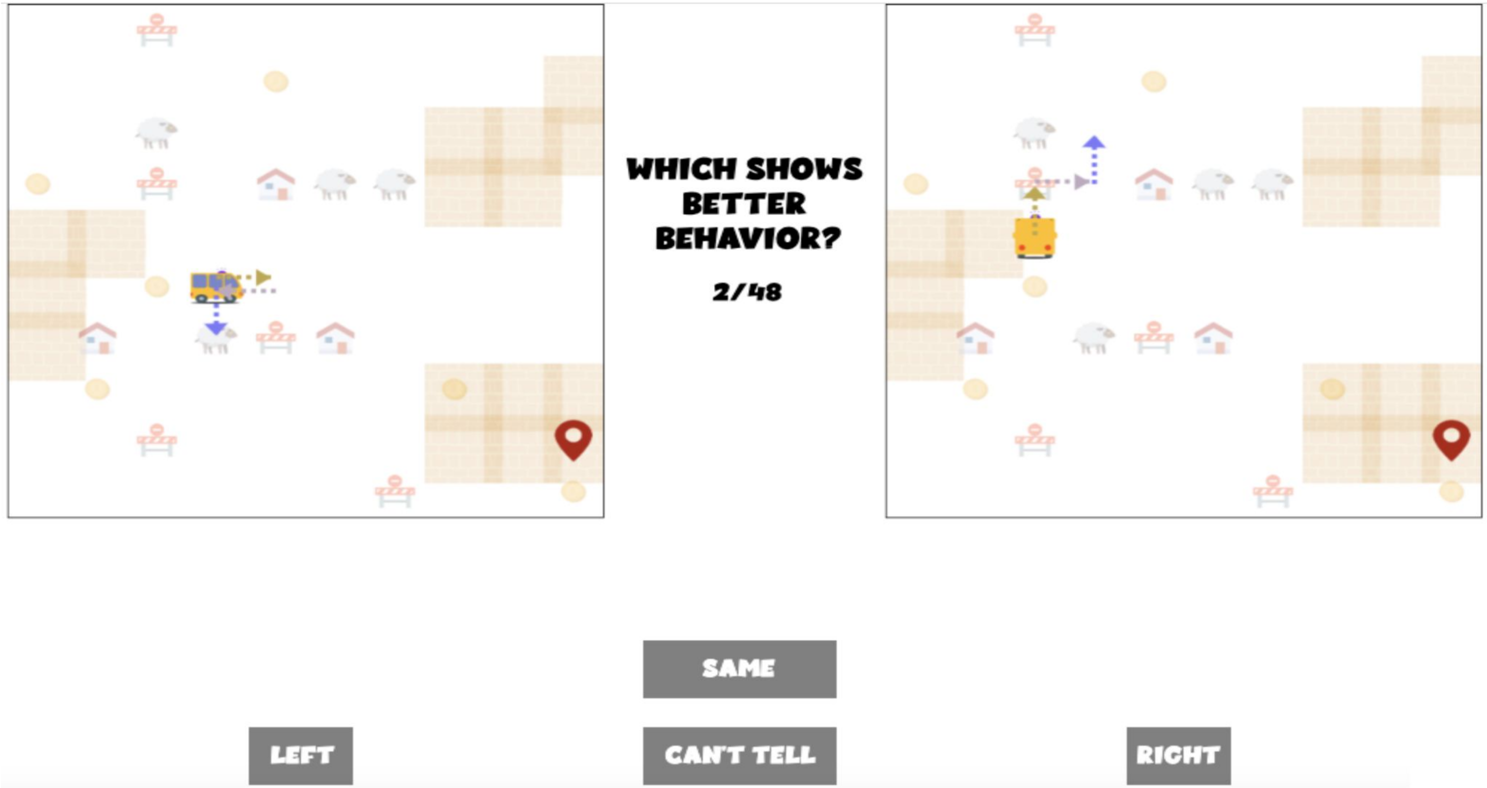
The delivery task



The delivery task



Preference elicitation



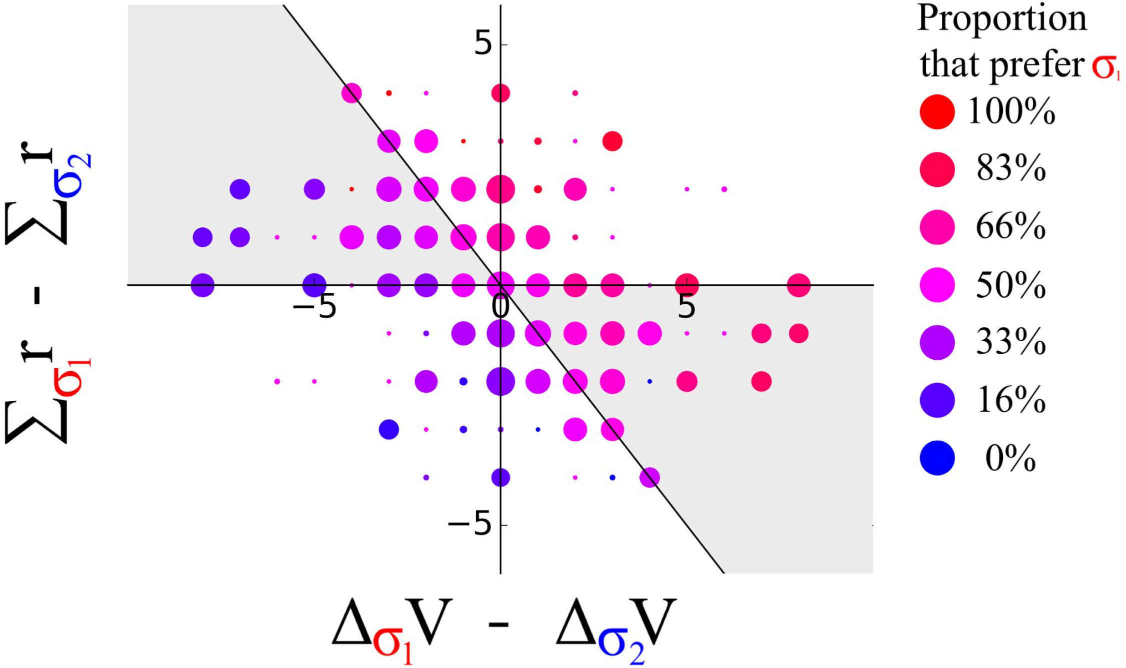
Human preferences visualized

Recall $regret_d(\sigma|\tilde{r}) \triangleq \sum_{t=0}^{|\sigma|-1} regret_d(\sigma_t|\tilde{r}) = \boxed{V^*_{\tilde{r}}(s_{\sigma,0})} - (\boxed{\Sigma_\sigma \tilde{r}} + \boxed{V^*_{\tilde{r}}(s_{\sigma,|\sigma|})})$

Best possible expected return from the *start* state (i.e., by optimal policy)

Partial return

Best possible expected return from the *end* state (i.e., by optimal policy)



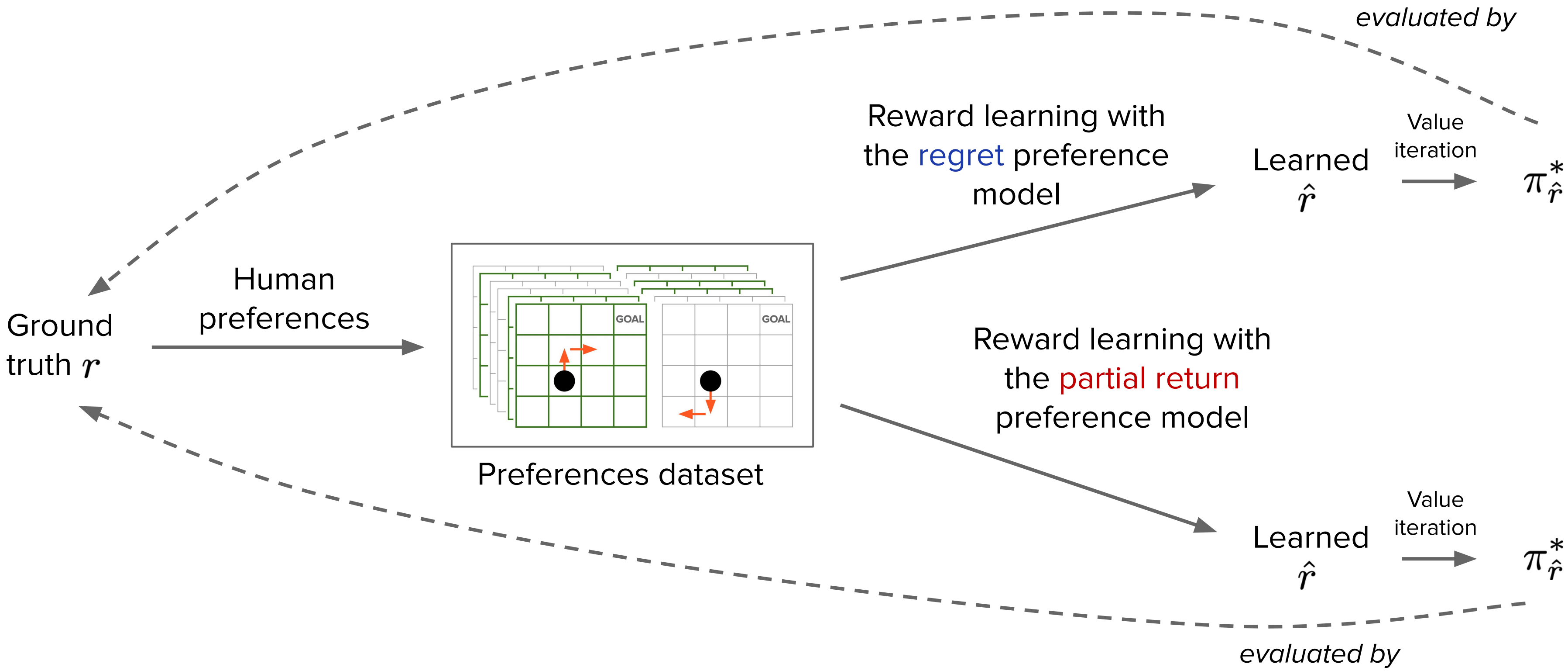
Explaining human preferences with different preference models

Preference model	Loss
$P(\cdot) = 0.5$ (uninformed)	0.69
P_{Σ_r} (partial return)	0.62
P_{regret}	**0.57**

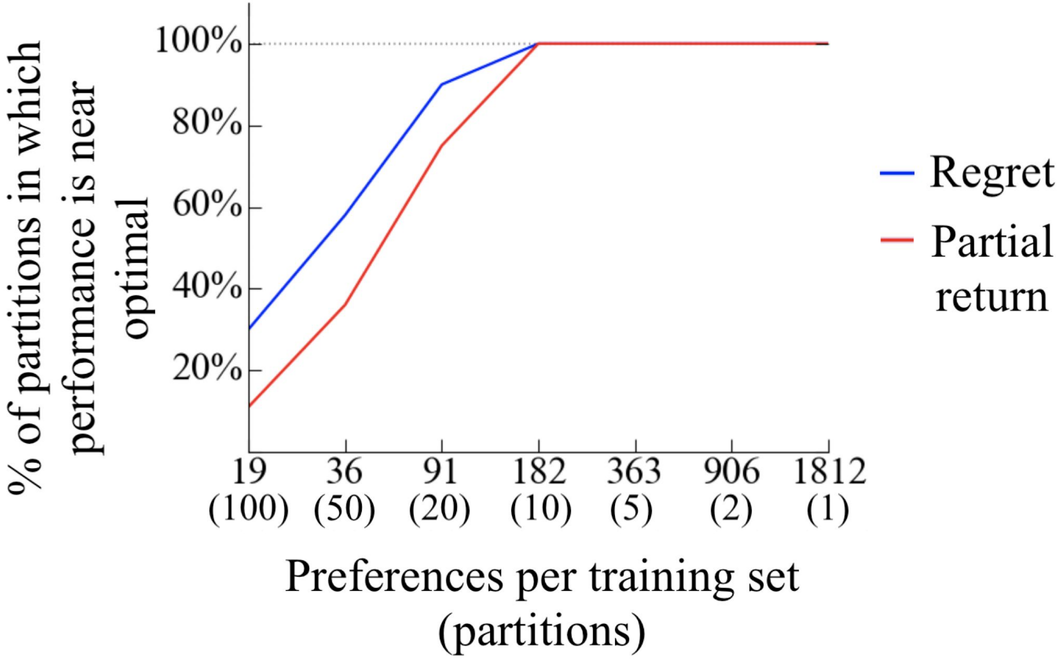
Mean cross-entropy test loss over 10-fold cross validation (n=1812) from predicting human preferences. Lower is better.

Learning reward functions with human preferences

Evaluating a reward function learned from human preferences



Performance with random partitions of human preferences dataset



Conclusion

Benefits of the regret preference model
(over the partial return model)

1. Humans intuitively appear to consider state value. The regret preference model also considers state value (in expectation).

2. Always prefers optimal segments over suboptimal segments, making it reward identifiable with noiseless preferences or stochastic preferences.

3. More sample efficient
    - when learning from its own preferences.
    - when learning from human preferences.

4. When $|\sigma| = 1$, the discount factor is considered, which is critical because the discount factor and the reward function *interact* to determine the set of optimal policies.

Results from past work

The regret preference model was superior by:

- **Intuition / self-reflection**
- **Theory** - reward identifiability
- **Descriptive** - gave a higher likelihood to our human preference dataset
- **Performance of learned reward functions** - both with human preferences and when each model generates its own training set

Summary

- Critique partial return as a poor model of human preference
- A **new preference model with *regret(σ)*** as the segment statistic
- Found that the regret preference model is superior by:
  - **Intuition / self-reflection**
  - **Theory** - reward identifiability
  - **Descriptive** - gave a higher likelihood to our human preference dataset
  - **Performance of learned reward functions** - both with human preferences and when each model generates its own training set
- We show that **the choice of preference model impacts the performance** of learned reward functions.

Limitations and future work

- **Efficient estimation of regret** for complex tasks (including deep learning settings).
- Develop **prescriptive methods to nudge humans** to conform more to normatively appealing preference models.
- Usage of the **partial return preference model *has* had considerable success**. **Why?**

Learning Optimal Advantage from Preferences and Mistaking it for Reward


W. Bradley Knox[1,4]


Stephane Hatgis-Kessell[1]


Sigurdur Orn Adalgeirsson[4]


Serena Booth[2]


Anca Dragan[5]


Scott Niekum[6]


Peter Stone[1,3]

[1]UT Austin [2]MIT CSAIL [3]Sony AI
[4]Google Research [5]UC Berkeley
[6]UMass Amherst

If the partial return preference model is so bad…

…why has using it performed so well in practice?

When regret drives preferences but the dominant model is assumed

(i.e., using A_r^* as r)

Outline:

- When A_r^* is known exactly
- When A_r^* is approximated
- Reframing RLHF for LLMs

Assuming the partial return preference model when regret is correct

(Learning A^*_r and using it as r)

A unified representation of the preference models

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Partial return: $f(\sigma) = \text{discounted sum of } r(s,a) \text{ for each } (s,a) \text{ in } \sigma$

Regret: $f(\sigma) = \text{ discounted sum of } A^*(s,a) \text{ for each } (s,a) \text{ in } \sigma$

Unification: $f(\sigma) = \text{discounted sum of } g(s,a) \text{ for each } (s,a) \text{ in } \sigma$

If you assume partial return but preferences are by regret, then **you are using (an approximation of) A* as a reward function**.

A unified representation of the preference models

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

$$= logistic\left(\sum_{t=0}^{|\sigma_1|-1} \tilde{r}(s_t^\sigma, a_t^\sigma) - \sum_{t=0}^{|\sigma_2|-1} \tilde{r}(s_t^\sigma, a_t^\sigma)\right)$$ Partial return

$$= logistic\left(\sum_{t=0}^{|\sigma_1|-1} A_{\tilde{r}}^*(s_t^\sigma, a_t^\sigma) - \sum_{t=0}^{|\sigma_2|-1} A_{\tilde{r}}^*(s_t^\sigma, a_t^\sigma)\right)$$ Regret

$$= logistic\left(\sum_{t=0}^{|\sigma_1|-1} g(s_t^\sigma, a_t^\sigma) - \sum_{t=0}^{|\sigma_2|-1} g(s_t^\sigma, a_t^\sigma)\right)$$ Unification

If you assume partial return but preferences are by regret, then **you are using (an approximation of) A* as a reward function**.

A unified representation of the preference models

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

$$= logistic\Big(\sum_{t=0}^{|\sigma_1|-1} \tilde{r}(s_t^\sigma, a_t^\sigma) - \sum_{t=0}^{|\sigma_2|-1} \tilde{r}(s_t^\sigma, a_t^\sigma)\Big)$$ Partial return

$$= logistic\Big(\sum_{t=0}^{|\sigma_1|-1} A_{\tilde{r}}^*(s_t^\sigma, a_t^\sigma) - \sum_{t=0}^{|\sigma_2|-1} A_{\tilde{r}}^*(s_t^\sigma, a_t^\sigma)\Big)$$ Regret

$$= logistic\Big(\sum_{t=0}^{|\sigma_1|-1} g(s_t^\sigma, a_t^\sigma) - \sum_{t=0}^{|\sigma_2|-1} g(s_t^\sigma, a_t^\sigma)\Big)$$ Unification

3 algorithms

Dataset created by reward function r and	Algorithm for learning from preferences	Output of learning from preferences	Additional step to create policy (other than greedy action selection)	
partial return preference model	learning g	$\hat{r}$	policy improvement	$\hat{\pi}^*_r$
regret preference model	learning by regret algorithm	$\hat{r}$	policy improvement	$\hat{\pi}^*_r$
regret preference model	learning g	$\hat{A}^*_r$	*nothing*	$\hat{\pi}^*_r$

3 algorithms

Dataset created by reward function r and	Algorithm for learning from preferences	Output of learning from preferences	Additional step to create policy (other than greedy action selection)	
partial return preference model	learning *g* →	$\hat{r}$	policy improvement →	$\hat{\pi}_r^*$
regret preference model	learning by regret algorithm →	$\hat{r}$	policy improvement →	$\hat{\pi}_r^*$
regret preference model	learning *g* →	$\hat{A}_r^*$	*nothing* →	$\hat{\pi}_r^*$

4 algorithms

Dataset created by reward function r and	Algorithm for learning from preferences	Assumed Output of learning from preferences	Additional step to create policy (other than greedy action selection)	
regret ~~partial return~~ preference model	learning g	$\hat{r}$	policy improvement	$\hat{\pi}^*_r$
regret preference model	learning by regret algorithm	$\hat{r}$	policy improvement	$\hat{\pi}^*_r$
regret preference model	learning g	$\hat{A}^*_r$	*nothing*	$\hat{\pi}^*_r$

4 algorithms

Dataset created by reward function r and	**Algorithm for learning from preferences**	**Assumed Output of learning from preferences**	**Additional step to create policy** (other than greedy action selection)	
regret ~~partial return~~ preference model	learning *g*	$\hat{r}$	policy improvement	$\hat{\pi}_r^*$
regret preference model	learning by regret algorithm	$\hat{r}$	policy improvement	$\hat{\pi}_r^*$
regret preference model	learning *g*	$\hat{A}_r^*$	*nothing*	$\hat{\pi}_r^*$

4 algorithms

Dataset created by reward function r and	Algorithm for learning from preferences	Assumed Output of learning from preferences	Additional step to create policy (other than greedy action selection)	
regret ~~partial return~~ preference model	learning g	$\hat{r}$	policy improvement	$\hat{\pi}_r^*$
regret preference model	learning by regret algorithm	$\hat{r}$	policy improvement	$\hat{\pi}_r^*$
regret preference model	learning g	$\hat{A}_r^*$	*nothing*	$\hat{\pi}_r^*$

$greedy\ \widehat{A_r^*}$

4 algorithms

$greedy\ Q^*_{r_{\widehat{A^*_r}}}$

Dataset created by reward function r and	**Algorithm for learning from preferences**	**Assumed Output of learning from preferences**	**Additional step to create policy** (other than greedy action selection)	
regret ~~partial return~~ preference model	learning *g*	$\hat{r}$	policy improvement	$\hat{\pi}^*_r$
regret preference model	learning by regret algorithm	$\hat{r}$	policy improvement	$\hat{\pi}^*_r$
regret preference model	learning *g*	$\hat{A}^*_r$	*nothing*	$\hat{\pi}^*_r$

$greedy\ \widehat{A^*_r}$

Using A_r^* as reward

Optimal policies are preserved.

The set of optimal policies under r and $r_{A_r^*} \triangleq A_r^*$ is the same, regardless of the discount factor used with $r_{A_r^*}$.

Intuition:

$$A_r^*(s,a) = 0 \iff (s,a) \text{ is optimal w.r.t. } r$$
$$A_r^*(s,a) < 0 \iff (s,a) \text{ is suboptimal w.r.t. } r$$

so:

$$\text{trajectory } \tau \text{ has } return = 0 \text{ under } r' \iff \text{ all } (s,a) \text{ in } \tau \text{ are optimal w.r.t. } r$$
$$\text{trajectory } \tau \text{ has } return < 0 \text{ under } r' \iff \text{ some } (s,a) \text{ in } \tau \text{ is suboptimal w.r.t. } r$$

Therefore a trajectory gets maximal return under r' iff that trajectory is optimal w.r.t. r.

Reward is highly shaped.

From Ng, Harada, and Russell's 1999 paper on potential-based shaping:

> about the domain. As to how one may do this, Corollary 2 suggests a particularly nice form for Φ, if we know enough about the domain to try choosing it as such. We see that if $\Phi(s) = V^*_M(s)$ (with $\Phi(s_0) = 0$ in the undiscounted case), then Equation (4) tells us that the value function in M' is $V^*_{M'}(s) \equiv 0$ — and

Set $\Phi \triangleq V^*_r$.

With some algebra, we find that this definition of the potential function makes Ng et al.'s shaped reward function $r_{A^*_r} \triangleq A^*_r$, the optimal advantage function with respect to r!

An underspecification issue is resolved.

When segment lengths |σ| are 1: $\sum_{t=0}^{|\sigma|-1} \gamma^t r(s_t, a_t) = \gamma^0 r(s_0, a_0) = r(s_0, a_0)$

	Affected by the γ in the human's mind?
Preferences training set generated via partial return	*No*
Reward function learned via partial return	*No*
The set of optimal policies	*Yes*
The choice of γ during policy optimization	*Not without dataset augmentation*

However, for $r_{A_r^*} \triangleq A_r^*$,

$$\text{a trajectory is optimal} \iff \text{its discounted sum of } A_r^*(s,a) \text{ values is } 0$$

so γ has no impact on the set of optimal policies.

Policy improvement wastes computation and environment sampling.

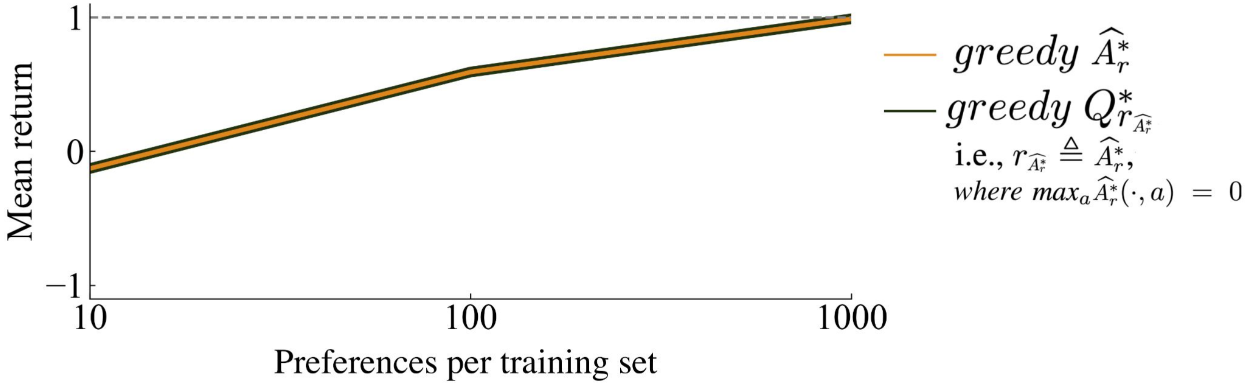
If we have A^*_r, then why do policy improvement to get the same policy as $\pi^*_r(s) = argmax_a A^*_r(s, a)$?

Using $\widehat{A_r^*}$, an approximation of A_r^*, as reward

If the max of $\widehat{A_r^*}$ in every state is 0, behavior is identical between $greedy\ \widehat{A_r^*}$ and $greedy\ Q^*_{r_{\widehat{A_r^*}}}$.

Proof is in the paper. Empirical validation:

Across 90 small gridworld tasks



I.e., while $\widehat{A_r^*}$ might not be optimal, treating $\widehat{A_r^*}$ as a reward function does not worsen (or improve) performance *if* the condition above is met.

But the max of $\widehat{A_r^*}$ in every state is not generally 0.

Let $g'(s,a) = g(s,a) + constant.$

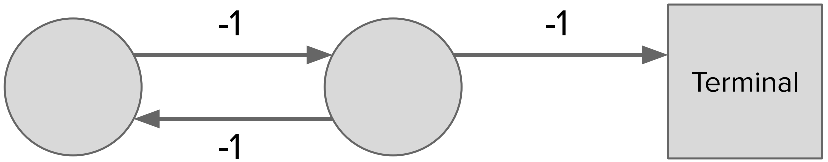
Then $logistic\Big(\sum_{t=0}^{|\sigma_1|-1} g(s_t^\sigma, a_t^\sigma) - \sum_{t=0}^{|\sigma_2|-1} g(s_t^\sigma, a_t^\sigma)\Big) = logistic\Big(\sum_{t=0}^{|\sigma_1|-1} g'(s_t^\sigma, a_t^\sigma) - \sum_{t=0}^{|\sigma_2|-1} g'(s_t^\sigma, a_t^\sigma)\Big).$

The likelihood is not affected by arbitrary shifts, so we should generally expect that $max_a \widehat{A_r^*}(s,a) \neq 0$.

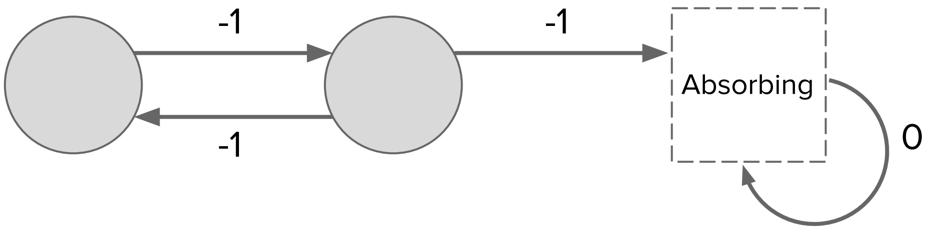
More generally, in variable horizon tasks, such constant shifts to reward can create catastrophic changes to the set of optimal policies. How can we reduce this issue?

An ameliorative tactic: include segments with transitions from absorbing state

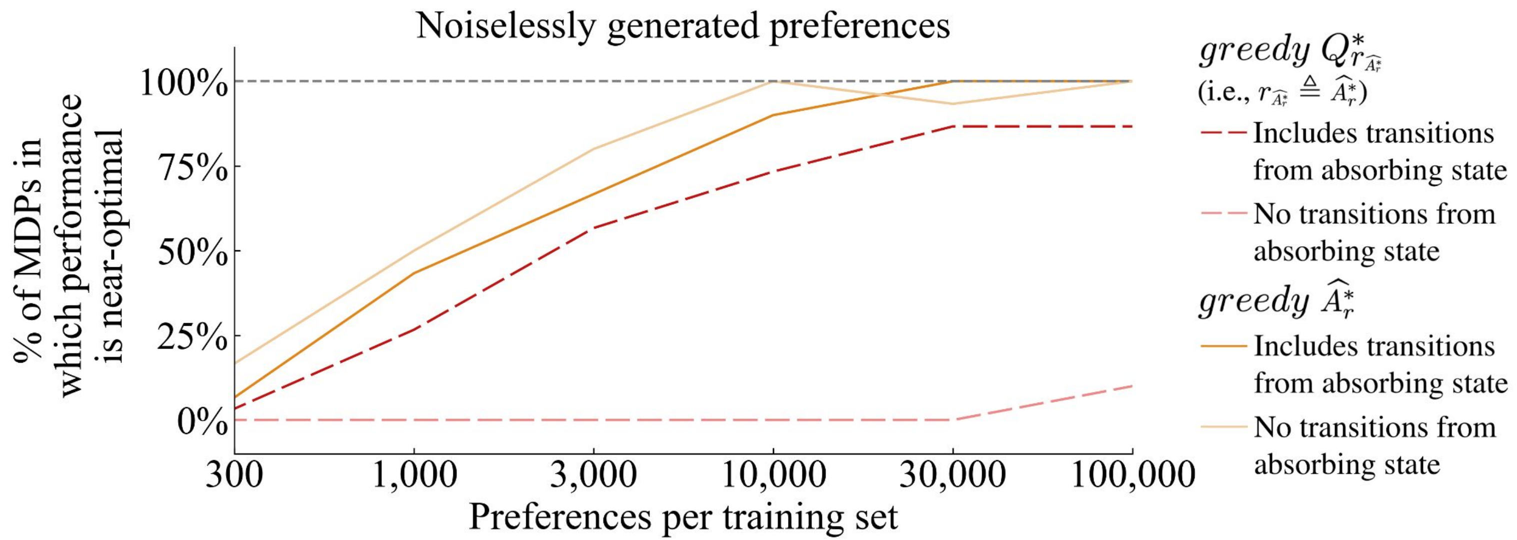
A simple episodic MDP



Absorbing state - turns episodic tasks into continuing (infinite) ones



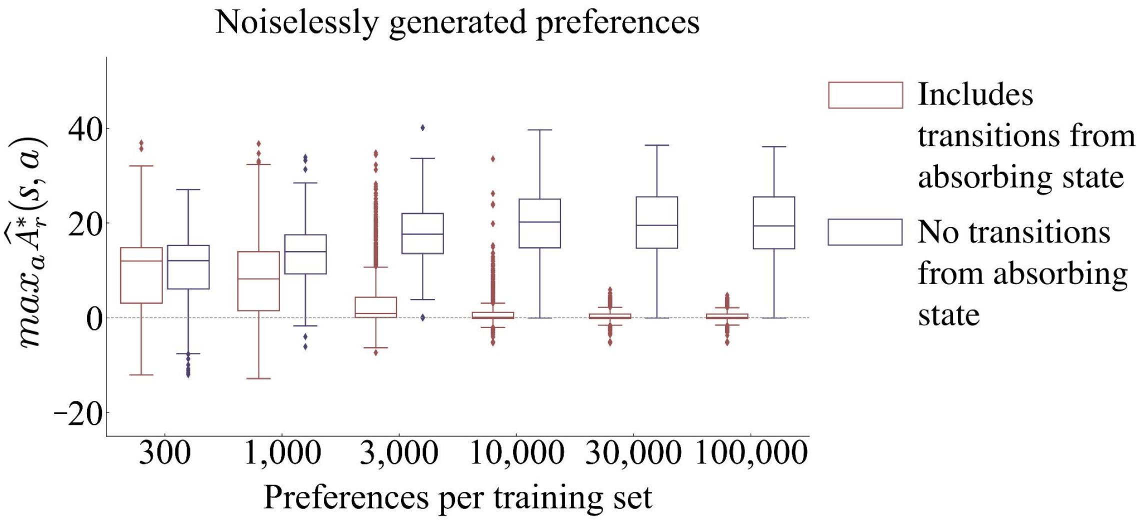
An ameliorative tactic: include segments with transitions from absorbing state



Results from 30 gridworld MDPs

An ameliorative tactic: include segments with transitions from absorbing state

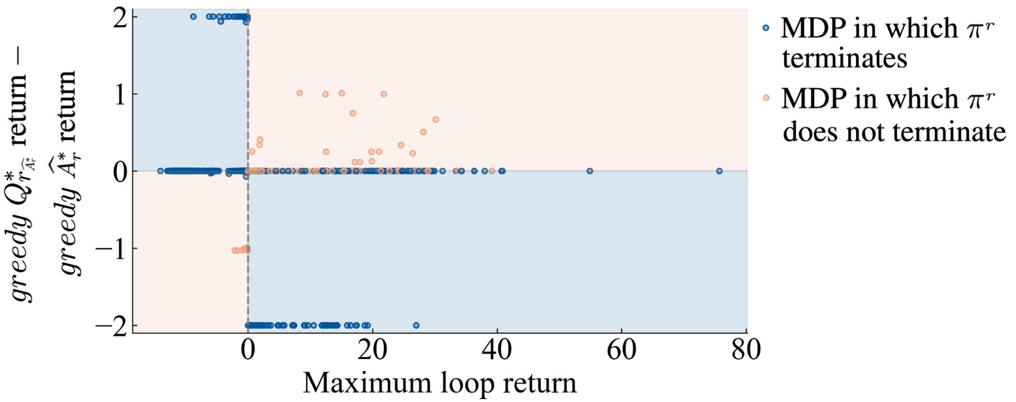
Transitions from absorbing state push the maximum per state towards 0.



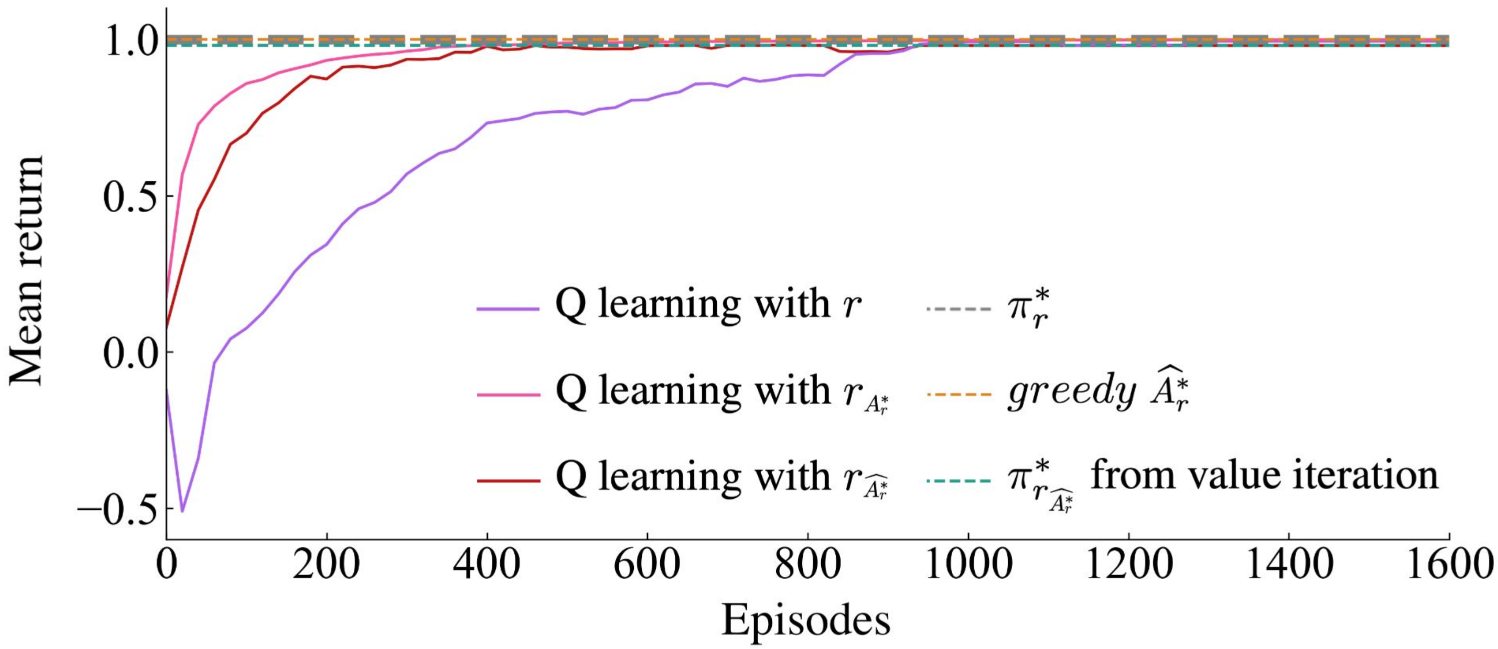
Results from the same 30 gridworld MDPs

Table 1: Hypothesis regarding which algorithm performs as well or better than the other, given 2 conditions.

Condition	π^*_r **terminates**	π^*_r **does not terminate**
Max loop partial return > 0	$greedy\ Q^*_{r_{\widehat{A^*_r}}}$	$greedy\ \widehat{A^*_r}$
Max loop partial return < 0	$greedy\ \widehat{A^*_r}$	$greedy\ Q^*_{r_{\widehat{A^*_r}}}$



Reward is also highly shaped with approximation error



For 100 MDPs, each $\widehat{A_r^*}$ learned with 100K noiselessly generated preferences

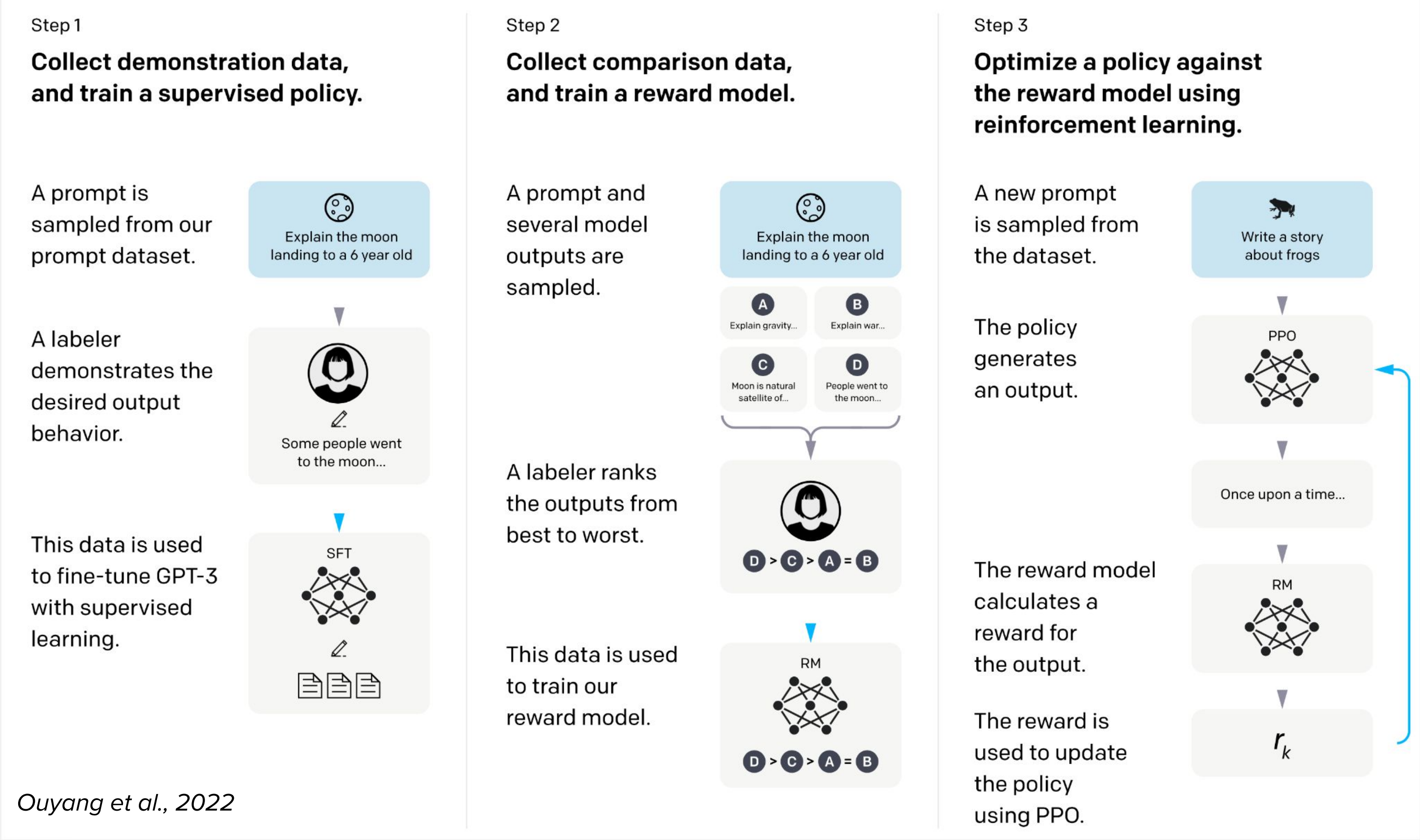
Is using $\widehat{A_r^*}$ as reward advised?

No!

But it's not as bad as we would have expected (if a pitfall is addressed).

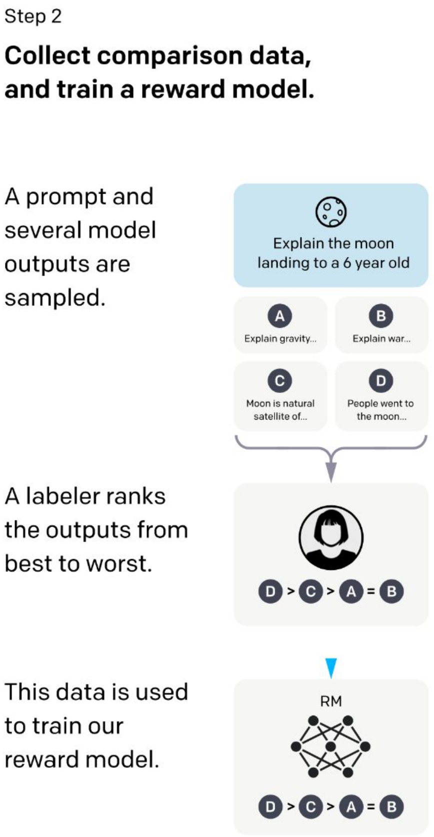
A better framing of fine-tuning LLMs with RLHF

Fine-tuning InstructGPT (and ChatGPT)



Ouyang et al., 2022

Fine-tuning InstructGPT (and ChatGPT)



Ouyang et al., 2022

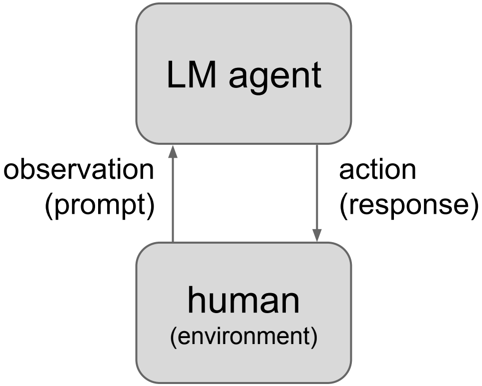
Mapping this to the previous content

- Their "reward model" is our $\hat{r}$.
- They assume the partial return preference model.
- Segment length is 1.
- State is the full observation history.
- The next state is not in the segment and not an input to $\hat{r}$.
- A ranking of n responses is turned into many preferences (precisely $(n^2-n)/2$ preferences).

The same approach is used for DeepMind's Sparrow (Glaese et al., 2022), Llama 2 (Touvron, 2023), and other influential work (Ziegler et al., 2019 and Bai et al.; 2022).

The multi-turn language problem

LM framing:	human's prompt	LM's response	human's prompt	LM's response	human's prompt	LM's response	…
RL framing:	*observation*	*action*	*observation*	*action*	*observation*	*action*	
R(s,a):	r_0		r_1		r_2		
	return = $r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots$		**return = $r_1 + \gamma r_2 + \ldots$**		**return = $r_2 + \ldots$**		



Arbitrary and counterintuitive discounting of reward

LM framing:	human's prompt	LM's response	human's prompt	LM's response	human's prompt	LM's response	…
RL framing:	*observation*	*action*	*observation*	*action*	*observation*	*action*	
R(s,a):	r_0		r_1		r_2		
	return = $\boldsymbol{r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots}$		**return =** $\boldsymbol{r_1 + \gamma r_2 + \ldots}$		**return =** $\boldsymbol{r_2 + \ldots}$		

When fine-tuning LLMs with RLHF, reward is used in a "bandit environment".
But the multi-turn problem not a bandit problem!

Treating this sequential problem as a bandit problem is **equivalent to setting γ=0.**

This bandit usage of a reward function **is counterintuitive, is unexplained, and confuses many people.**

Arbitrary and counterintuitive discounting of reward

LM framing:	human's prompt	LM's response	human's prompt	LM's response	human's prompt	LM's response	…
RL framing:	*observation*	*action*	*observation*	*action*	*observation*	*action*	
R(s,a):	r_0		r_1		r_2		
	return = $r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots$		**return =** $r_1 + \gamma r_2 + \ldots$		**return =** $r_2 + \ldots$		

Setting γ=0 isn't necessarily wrong because the choice of γ is arbitrary when assuming the partial return model. But it's counterintuitive, is unexplained, and confuses many people.

Does RLHF fine-tuning for multi-turn language tasks unknowingly assume a regret preference model?

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Unification: $f(\sigma) = \text{discounted sum of } g(s,a) \text{ for each } (s,a) \text{ in } \sigma$

Partial return

$$g = r$$

Regret

$$g = A_r^*$$

Does RLHF fine-tuning for multi-turn language tasks unknowingly assume a regret preference model?

Partial return

Assume learned *g* approximates *r*.
Assume γ=0.

$$\begin{aligned}\pi_r^*(s) &= argmax_a\ Q_r^*(s,a)\\ &= argmax_a(r(s,a) + \gamma E_{s'}[V_r^*(s')])\\ &= argmax_a\ r(s,a)\\ &= argmax_a\ g(s,a)\end{aligned}$$

Regret

Assume learned *g* approximates *A**.
No γ hyperparameter.

$$\begin{aligned}\pi_r^*(s) &= argmax_a\ A_r^*(s,a)\\ &= argmax_a\ g(s,a)\end{aligned}$$

The current assumption of the partial return preference model and the arbitrary assumption of γ=0 together give **the same result as simply assuming our regret preference model**!

Does RLHF fine-tuning for multi-turn language tasks unknowingly assume a regret preference model?

Partial return

Assume learned *g* approximates *r*.
Assume *γ=0*.

$$\begin{aligned}\pi_r^*(s) &= argmax_a\ Q_r^*(s,a) \\ &= argmax_a(r(s,a) + \gamma E_{s'}[V_r^*(s')]) \\ &= argmax_a\ r(s,a) \\ &= argmax_a\ g(s,a)\end{aligned}$$

Regret

Assume learned *g* approximates *A**.
No γ hyperparameter.

$$\begin{aligned}\pi_r^*(s) &= argmax_a\ A_r^*(s,a) \\ &= argmax_a\ g(s,a)\end{aligned}$$

The current assumption of the partial return preference model and the arbitrary assumption of γ=0 together give **the same result as simply assuming our regret preference model**.

What is learned during RLHF for LLMs is better thought of as an approximation of A^*, not of r.

Benefits of assuming that learning from preferences produces an A^*

- uses the more supported regret preference model
- explains the previously hard to justify treatment of a sequential task as a bandit problem
    - because that's how to force r to act like A^* (or Q^*)
- removes underspecification regarding γ
- if you want a reward function that will be added over multiple turns of interaction, suggests a different algorithm

Summary

Using A* as a reward function is less harmful than expected.

It's still not advised.

A new framing of RLHF for LLMs: **optimizing to an approximation of A*.**

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Partial return: $f(\sigma) = \text{sum of reward in } \sigma$

Regret: $f(\sigma) = \text{sum of } A^*(s,a) \text{ for each } (s,a) \text{ in } \sigma$

Papers

Regret preference model

Mistaking A* for reward

Conclusion

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Partial return: $f(\sigma) = \text{sum of reward in } \sigma$

Regret: $f(\sigma) = \text{sum of } A^*(s,a) \text{ for each } (s,a) \text{ in } \sigma$

Summary ~~Takeaways~~

Paper, **human preferences dataset**, and code

A key part of the current model for what drives human preferences in sequential tasks is unstudied and unvalidated.

The sum of reward in each trajectory segment does not explain well how humans give preferences.

You wouldn't want them to, based on theoretical properties.

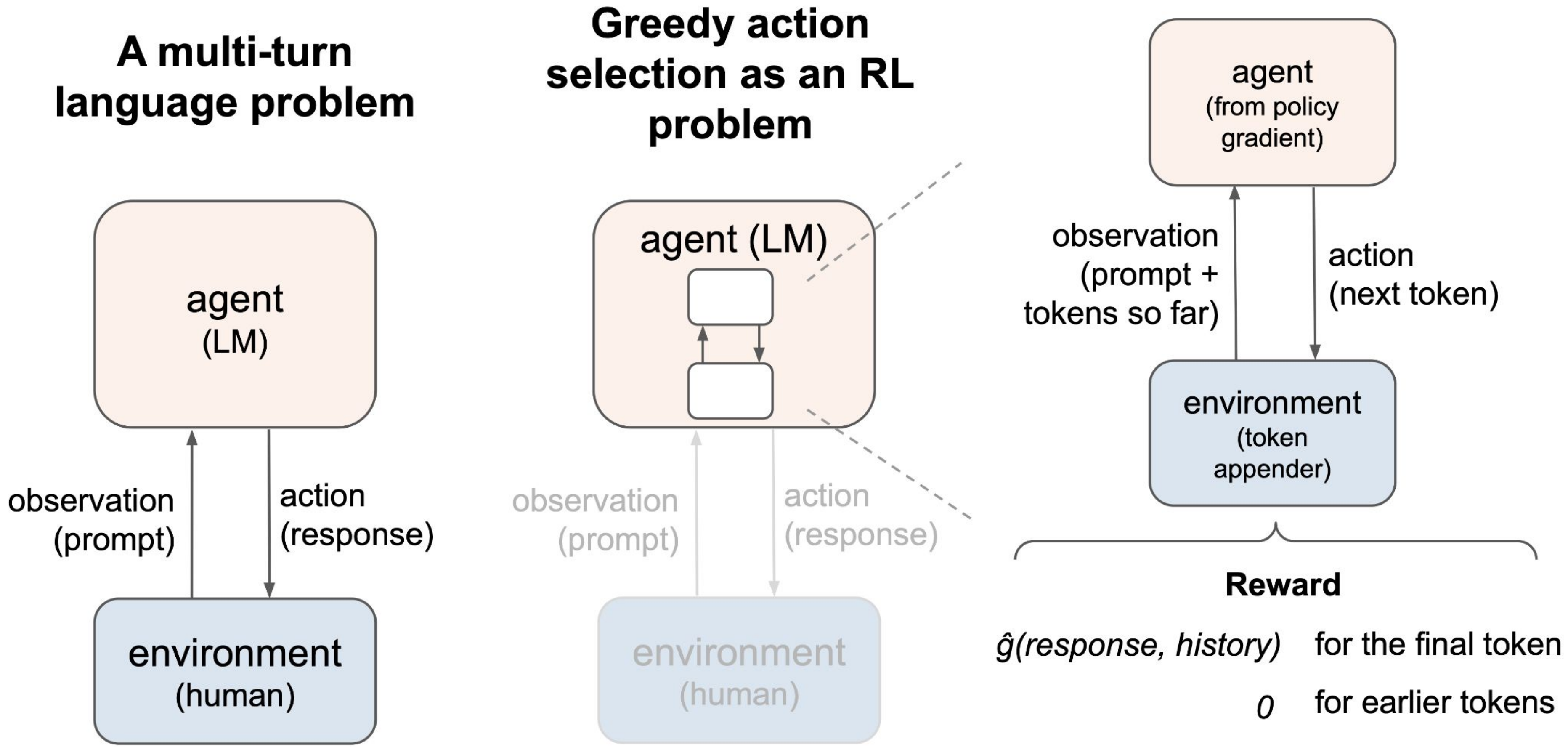
Regret is an improved model that measures a segment's deviation from optimality.

The model of human preference is a critical piece for alignment.

Future work

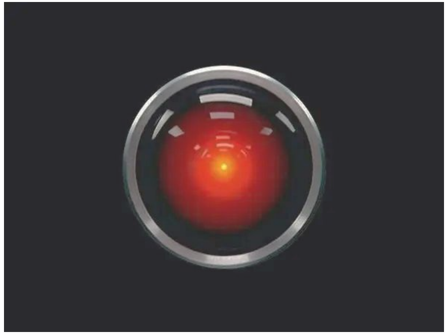
- **Efficient estimation of regret** for complex tasks
- Understand the **partial return preference model's past success**, despite it being a poor model of humans
- Nudging humans towards preference models

Supplemental



A multi-turn language problem
agent
(LM)
observation
(prompt)
action
(response)
environment
(human)
Greedy action selection as an RL problem
agent (LM)
observation
(prompt)
action
(response)
environment
(human)
agent
(from policy gradient)
observation
(prompt + tokens so far)
action
(next token)
environment
(token appender)
Reward
ĝ(response, history) for the final token
0 for earlier tokens

Aligned reward



2001: A Space Odyssey

Blues Brothers

Knox et al., *Reward (Mis)design for Autonomous Driving*
AIJ 2023

BACKGROUND ON REWARD

RL oversimplified: a set of problems and corresponding algorithmic solutions, in which *experience in a task is used to improve an agent's behavior such that it gets more reward*.

More specifically, most RL problems focus on increasing the *expectation* of G(τ), the utility of a trajectory:

$$G(\tau) = \sum_{t=1}^{(T-1)} R(s_t, a_t, s_{t+1})$$

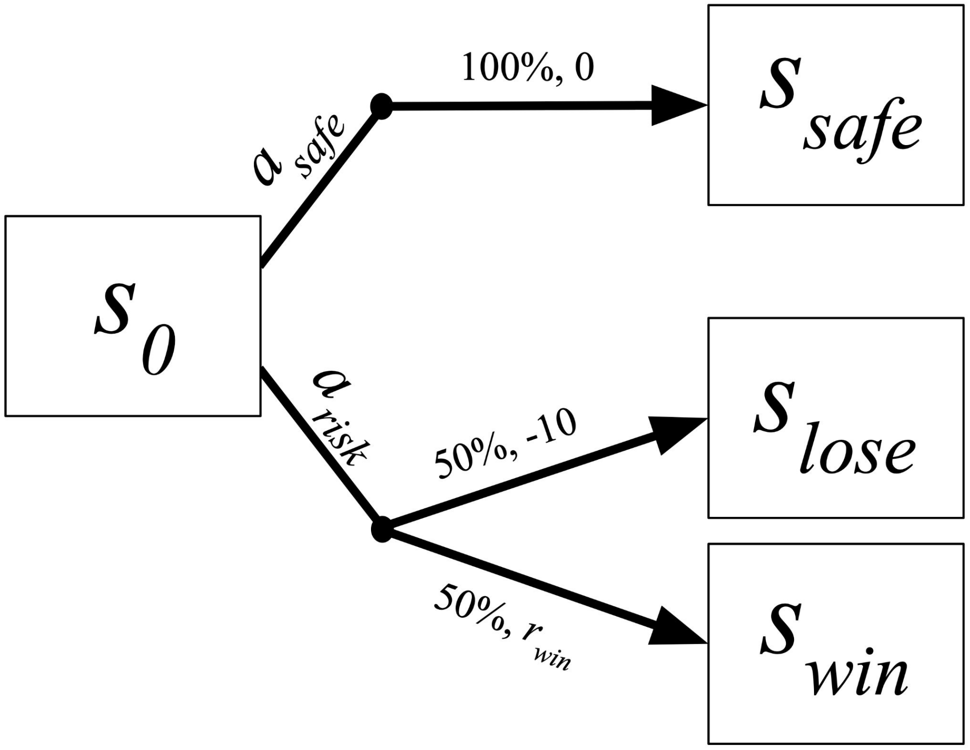
(Assumes undiscounted/episodic setting and an unstated distribution over starting states)

Benefits of the regret preference model
(over the partial return model)

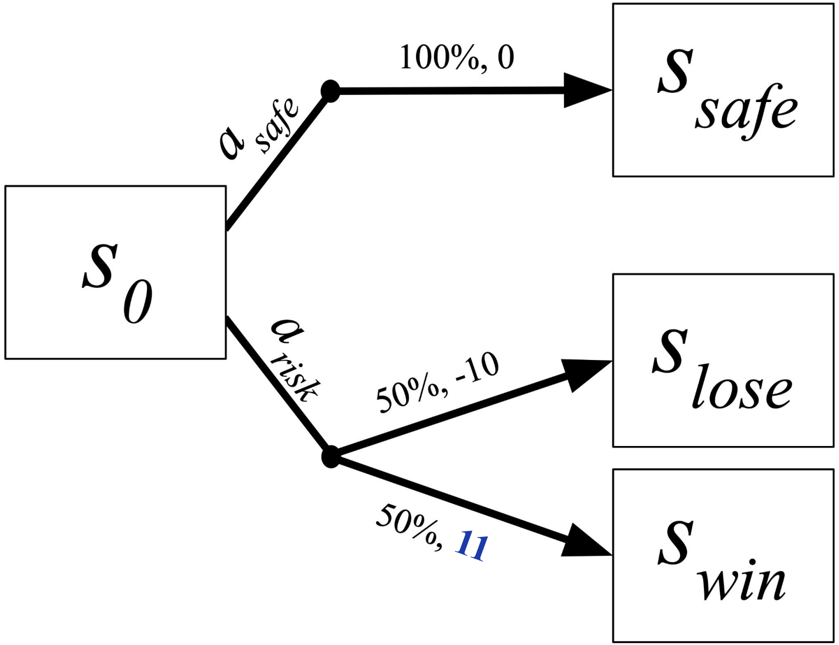
1. Considers consider state value and decision quality, which humans intuitively appear to consider.
2. Always prefers optimal segments over suboptimal segments, making it reward identifiable.
3. Better describes our human preferences dataset.
4. More sample efficient
    - when learning from its own preferences.
    - when learning from human preferences.

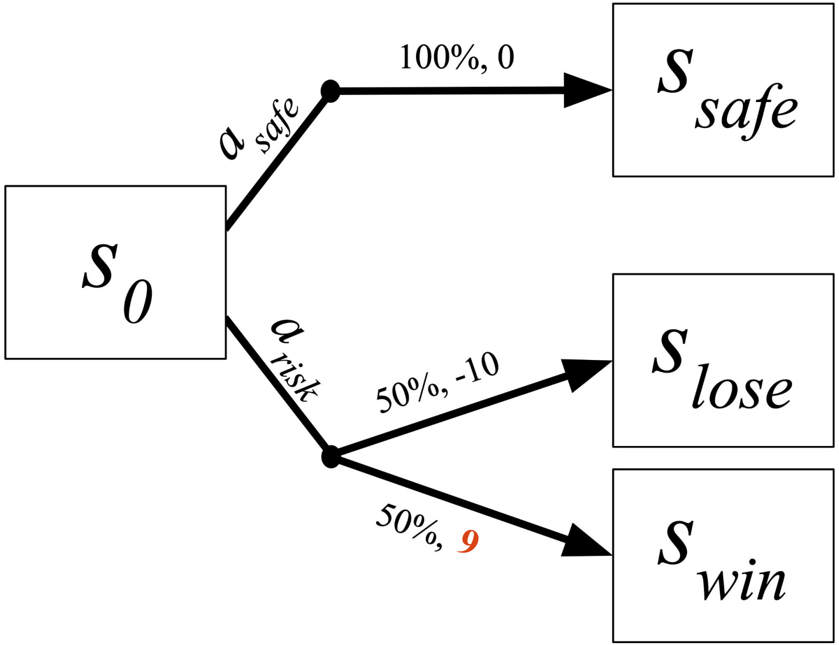
Reward identifiability

With partial return, reward is not generally identifiable without preference noise that reveals rewards' relative proportions.



Reward identifiability



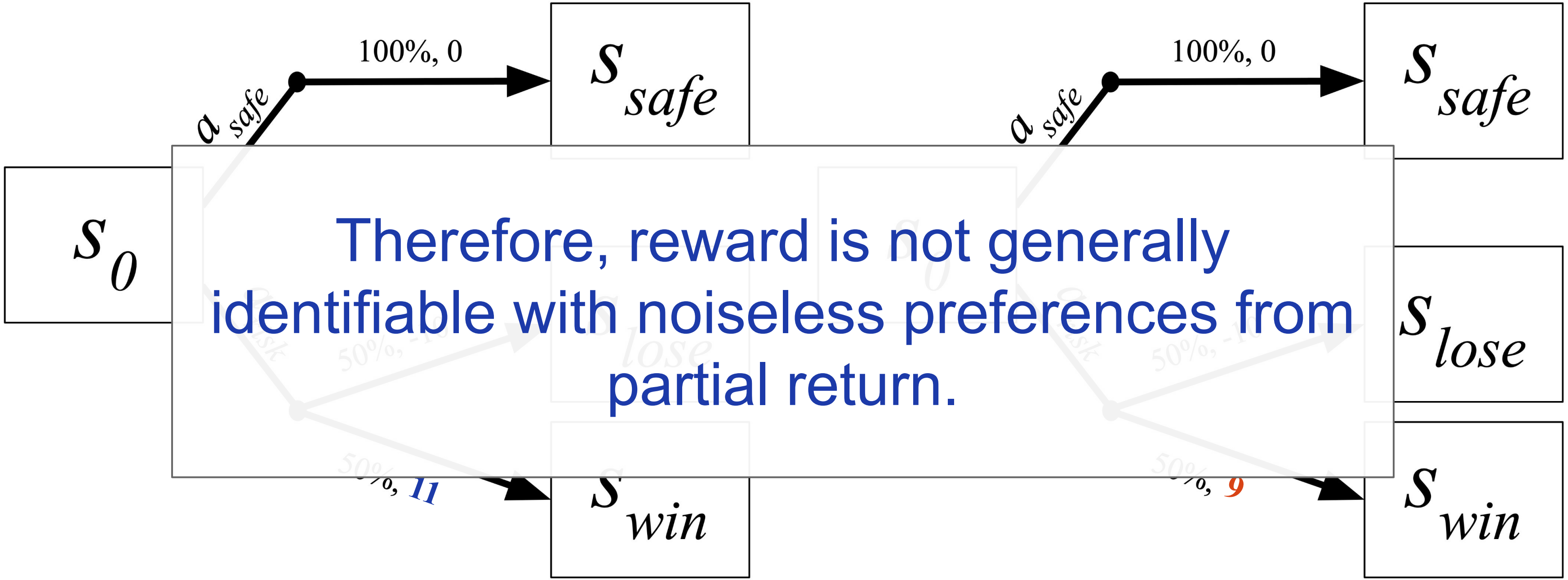


If $r_{win} = 11$, a_{risk} is optimal.

If $r_{win} = 9$, a_{safe} is optimal.

Yet both create the same (noiseless) preferences!!

Reward identifiability



Therefore, reward is not generally identifiable with noiseless preferences from partial return.

If $r_{win} = 11, a_{risk}$ is optimal.

If $r_{win} = 9$, a_{safe} is optimal.

Yet both create the same (noiseless) preferences!!

Reward identifiability

Similarly, reward is **not generally identifiable for inverse reinforcement learning** from (noiseless) demonstrations of optimal behavior.

The missing piece: the model of preference

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Proposed preference model: Regret

$$f(\sigma) = -regret(\sigma)$$

when all transitions are deterministic →

$$regret_d(\sigma|\tilde{r}) \triangleq \sum_{t=0}^{|\sigma|-1} regret_d(\sigma_t|\tilde{r}) = V^*_{\tilde{r}}(s_{\sigma,0}) - (\Sigma_\sigma \tilde{r} + V^*_{\tilde{r}}(s_{\sigma,|\sigma|}))$$

The missing piece: the model of preference

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Proposed preference model: Regret

$$f(\sigma) = -regret(\sigma)$$

when all transitions are deterministic

$$regret_d(\sigma|\tilde{r}) \triangleq \sum_{t=0}^{|\sigma|-1} regret_d(\sigma_t|\tilde{r}) = V^*_{\tilde{r}}(s_{\sigma,0}) - (\boxed{\Sigma_\sigma \tilde{r}} + V^*_{\tilde{r}}(s_{\sigma,|\sigma|}))$$

Partial return

The missing piece: the model of preference

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Proposed preference model: Regret

$$f(\sigma) = -regret(\sigma)$$

when all transitions are deterministic

$$regret_d(\sigma|\tilde{r}) \triangleq \sum_{t=0}^{|\sigma|-1} regret_d(\sigma_t|\tilde{r}) = \boxed{V^*_{\tilde{r}}(s_{\sigma,0})} - (\boxed{\Sigma_\sigma \tilde{r}} + \boxed{V^*_{\tilde{r}}(s_{\sigma,|\sigma|})})$$

Best possible expected return from the *start* state (i.e., by optimal policy)

Partial return

Best possible expected return from the *end* state (i.e., by optimal policy)

The missing piece: the model of preference

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Proposed preference model: Regret

$$f(\sigma) = -regret(\sigma)$$

Best possible expected return from the *start* state given the segment σ

when all transitions are deterministic

$$regret_d(\sigma|\tilde{r}) \triangleq \sum_{t=0}^{|\sigma|-1} regret_d(\sigma_t|\tilde{r}) = \boxed{V^*_{\tilde{r}}(s_{\sigma,0})} - \overbrace{\left(\boxed{\Sigma_\sigma \tilde{r}} + \boxed{V^*_{\tilde{r}}(s_{\sigma,|\sigma|})}\right)}$$

Best possible expected return from the *start* state (i.e., by optimal policy)

Partial return

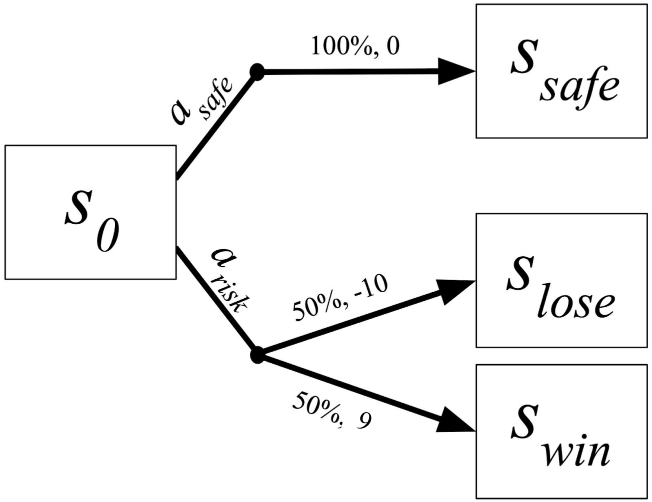
Best possible expected return from the *end* state (i.e., by optimal policy)

What if transitions can be stochastic?

when all transitions are deterministic

$$regret_d(\sigma|\tilde{r}) \triangleq \sum_{t=0}^{|\sigma|-1} regret_d(\sigma_t|\tilde{r}) = V^*_{\tilde{r}}(s_{\sigma,0}) - (\Sigma_\sigma \tilde{r} + V^*_{\tilde{r}}(s_{\sigma,|\sigma|}))$$

The lottery:



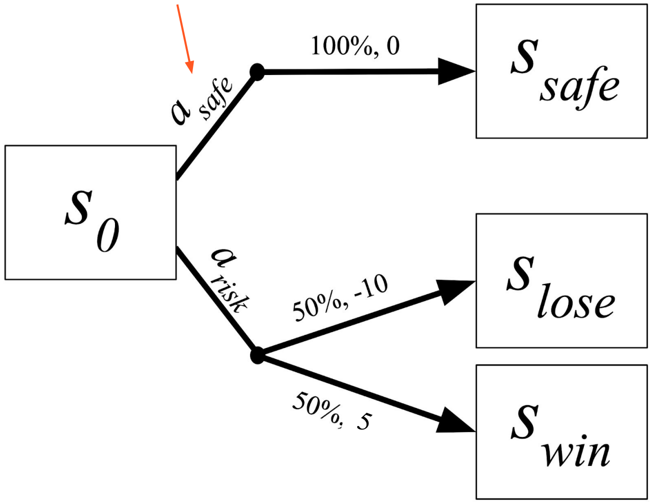
What if transitions can be stochastic?

when all transitions are deterministic →

$$regret_d(\sigma|\tilde{r}) \triangleq \sum_{t=0}^{|\sigma|-1} regret_d(\sigma_t|\tilde{r}) = \cancel{V_{\tilde{r}}^*(s_{\sigma,0})} - (\Sigma_\sigma \tilde{r} + \cancel{V_{\tilde{r}}^*(s_{\sigma,|\sigma|})})$$

Best decision

The lottery:

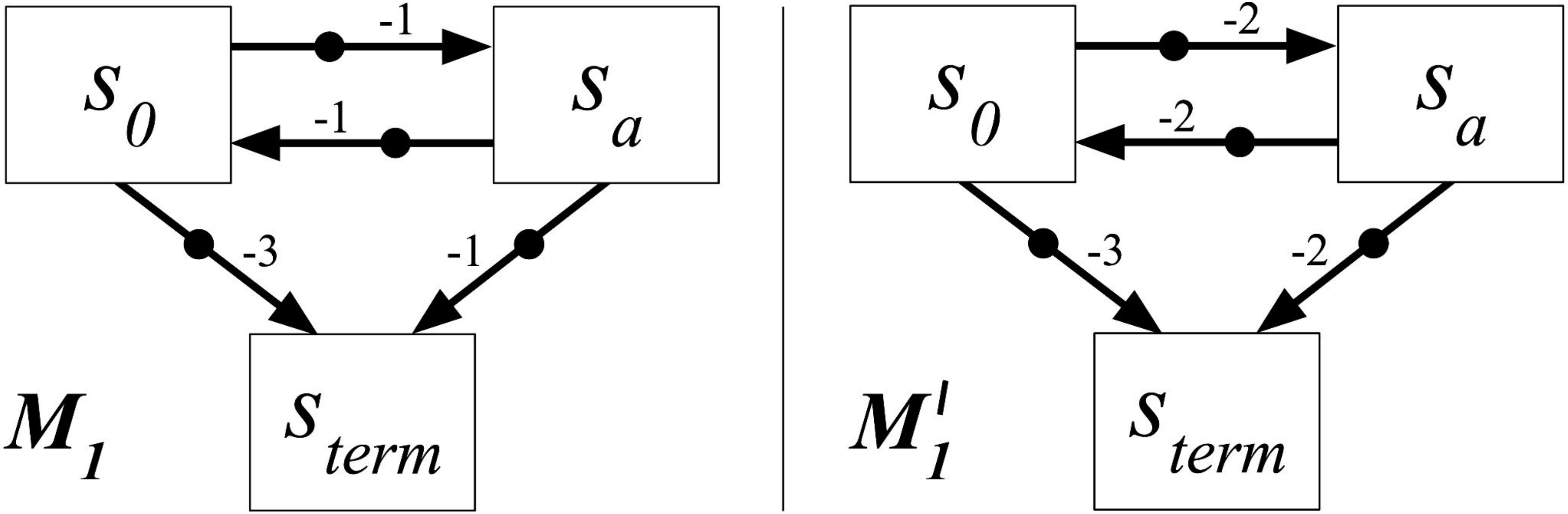


Best outcome

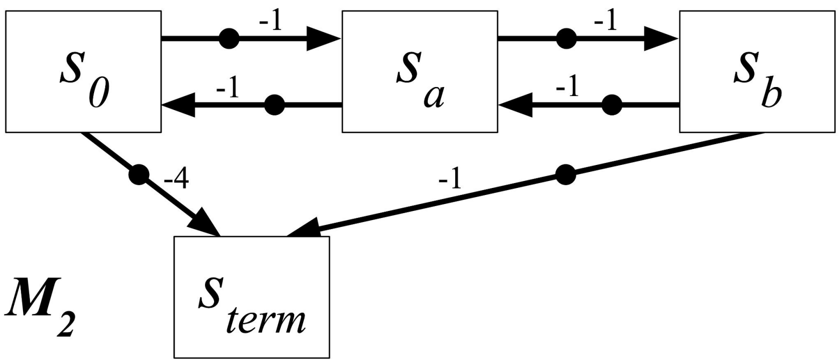
The missing piece: the model of preference

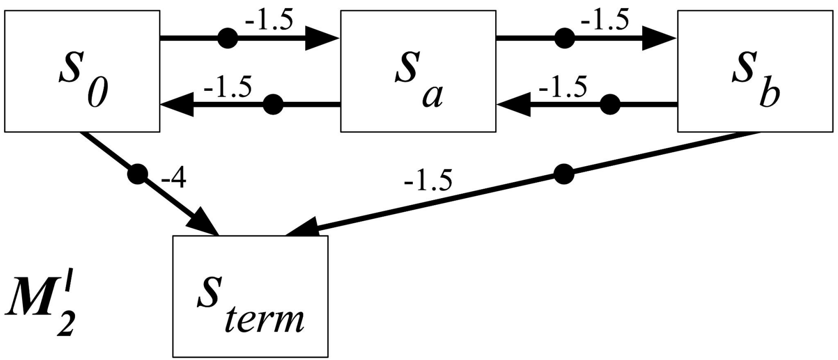
$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Deterministic MDPs with different π* but the same preferences by partial return (for segment size 1)



Deterministic MDPs with different π* but the same preferences by partial return (for segment size 2)

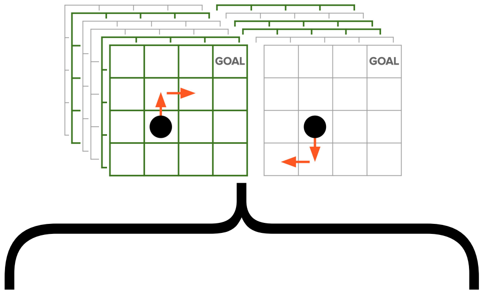




An algorithm for reward learning with *estimated* regret

Learning a reward function from preferences

Given a preference model $P(\sigma_1 \succ \sigma_2 | \hat{r})$,



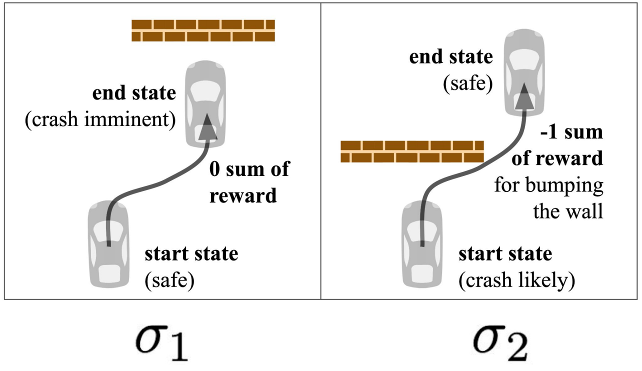
optimize $\hat{r}$ to maximize the likelihood of the *preferences dataset*.

The missing piece: the model of preference

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Partial return: $f(\sigma) = \text{sum of reward in } \sigma$

Which shows better behavior?



Partial return prefers the left segment!

Efficiently estimating value functions

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Regret preference model

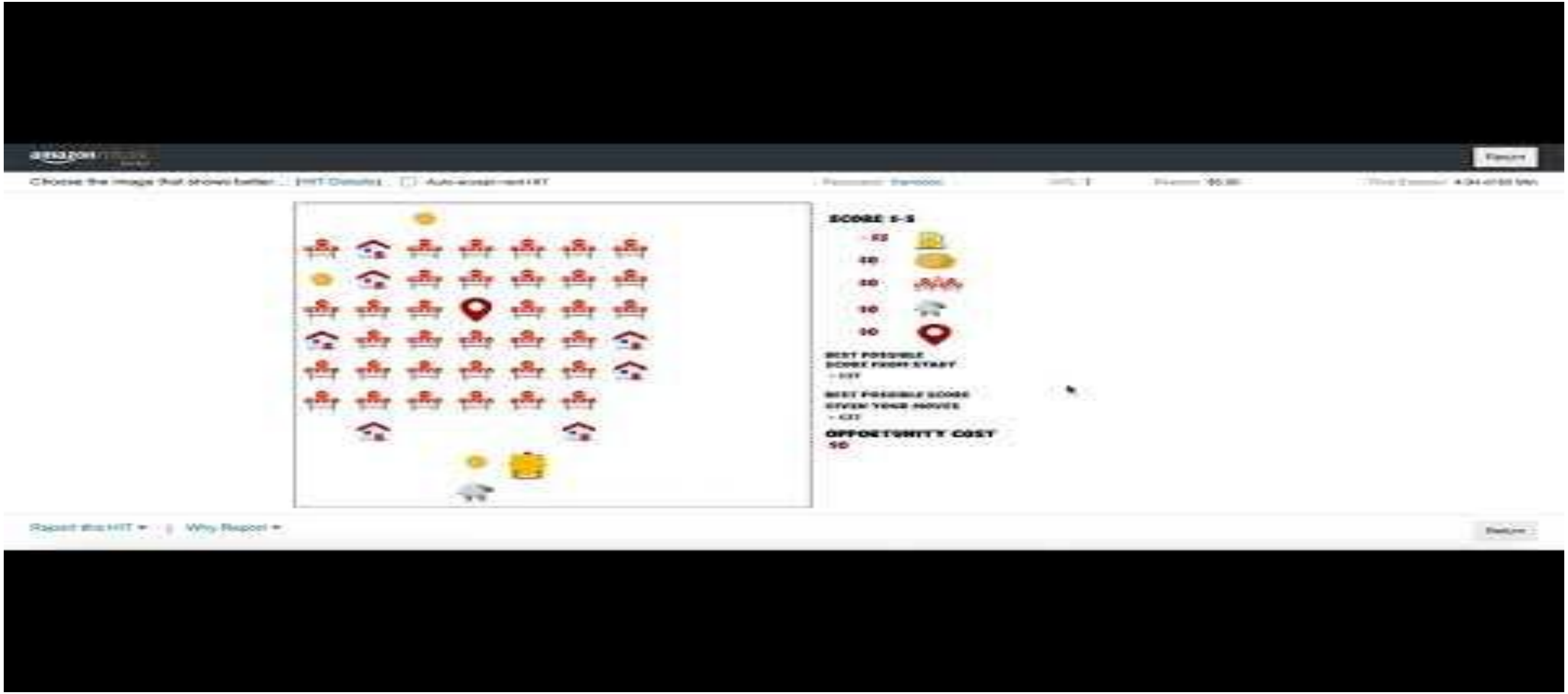
$$\begin{aligned} f(\sigma) &= -regret(\sigma) \\ &= \text{sum of } A^*(s,a) \text{ for each } (s,a) \text{ in } \sigma \end{aligned}$$

$$regret(\sigma|\tilde{r}) = \sum_{t=0}^{|\sigma|-1} regret(\sigma_t|\tilde{r}) = \sum_{t=0}^{|\sigma|-1} \left[V^*_{\tilde{r}}(s_{\sigma,t}) - Q^*_{\tilde{r}}(s_{\sigma,t}, a_{\sigma,t})\right] = \sum_{t=0}^{|\sigma|-1} -A^*_{\tilde{r}}(s_{\sigma,t}, a_{\sigma,t})$$

$$regret_d(\sigma|\tilde{r}) \triangleq \sum_{t=0}^{|\sigma|-1} regret_d(\sigma_t|\tilde{r}) = V^*_{\tilde{r}}(s_{\sigma,0}) - (\Sigma_\sigma \tilde{r} + V^*_{\tilde{r}}(s_{\sigma,|\sigma|}))$$

We assume linear reward functions and use successor features to quickly estimate Q* and V* for new reward parameters.

The delivery task



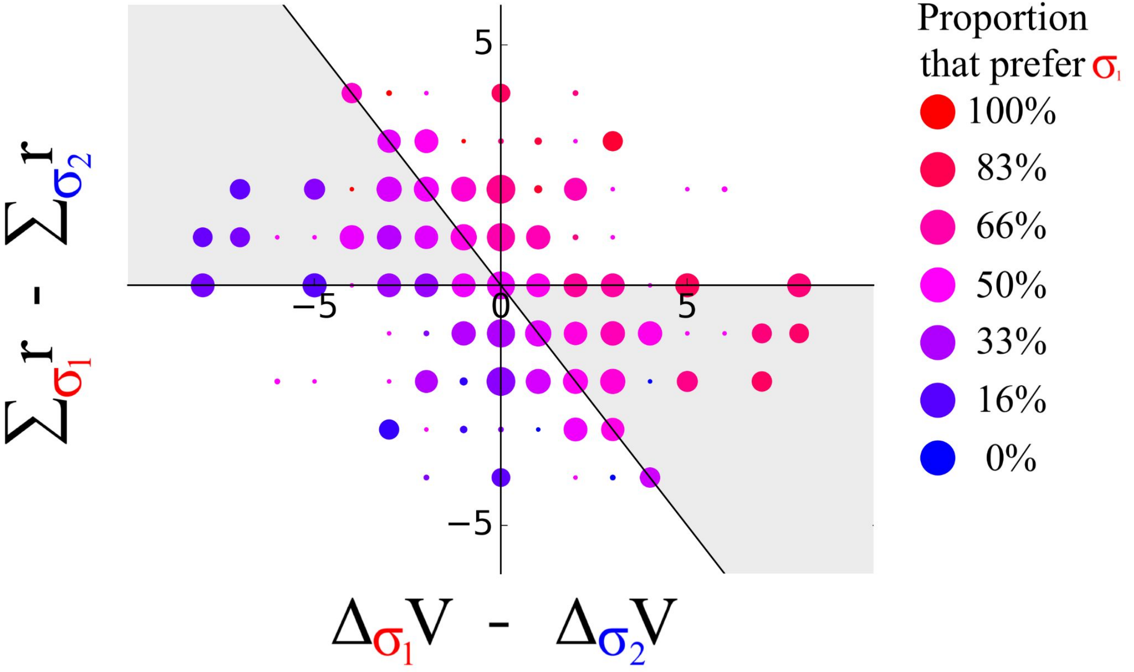
Human preferences visualized

Recall $$regret_d(\sigma|\tilde{r}) \triangleq \sum_{t=0}^{|\sigma|-1} regret_d(\sigma_t|\tilde{r}) = \boxed{V^*_{\tilde{r}}(s_{\sigma,0})} - (\boxed{\Sigma_\sigma \tilde{r}} + \boxed{V^*_{\tilde{r}}(s_{\sigma,|\sigma|})})$$

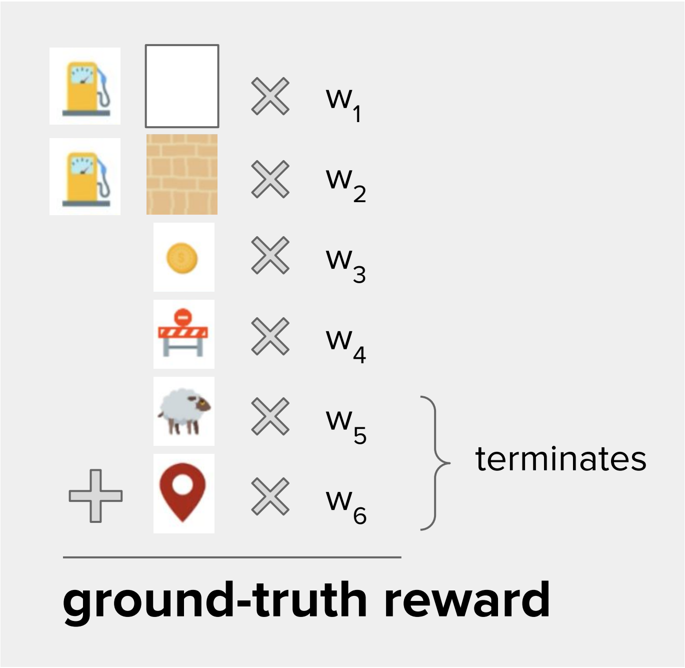
Best possible expected return from the *start* state (i.e., by optimal policy)

Partial return

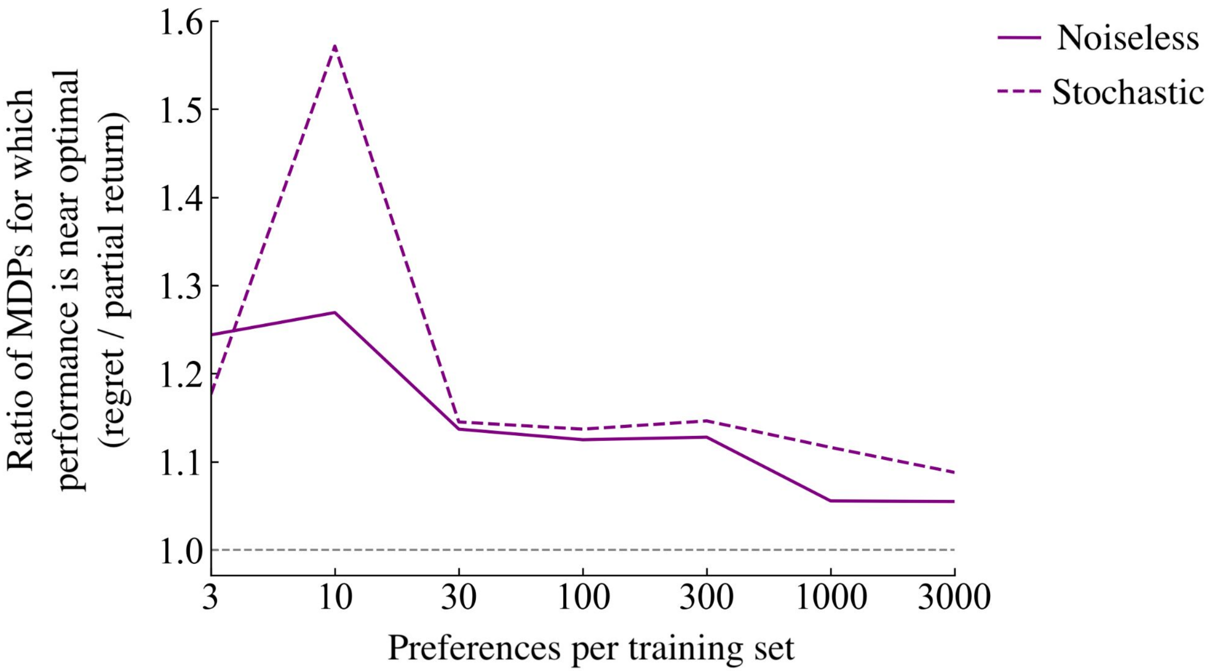
Best possible expected return from the *end* state (i.e., by optimal policy)



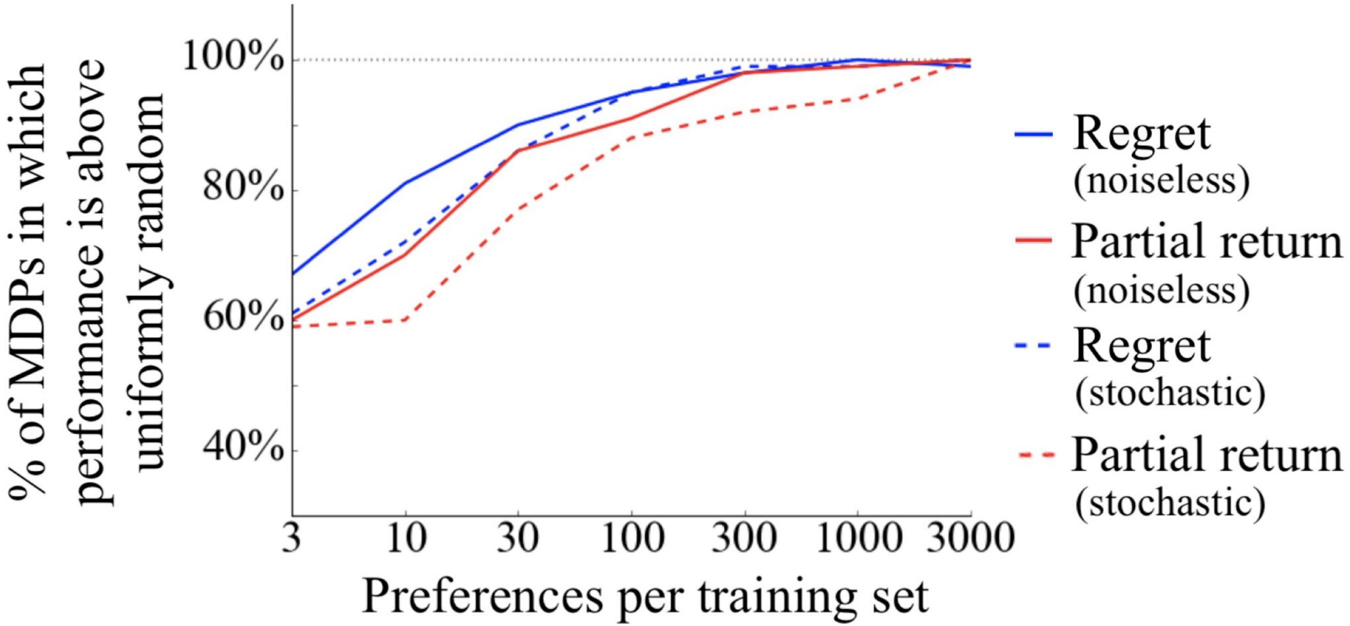
The delivery domain



When each model is perfect, because it creates its own preference dataset



When each model is perfect, because it creates its own preference dataset



Preference Model	$r_{win}=1$ $r_{lose}=-50$	$r_{win}=10^3$ $r_{lose}=-50$	$r_{win}=100$ $r_{lose}=-1$	$r_{win}=100$ $r_{lose}=-10^3$
Noiseless P_{regret}	100%	100%	100%	100%
Stochastic P_{regret}	100%	100%	100%	100%
Noiseless P_{Σ_r}	100%	0%	100%	0%
Stochastic P_{Σ_r}	100%	0%	100%	100%

Problems with the partial return preference model

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Partial return: $f(\sigma) = \text{sum of reward in } \sigma$

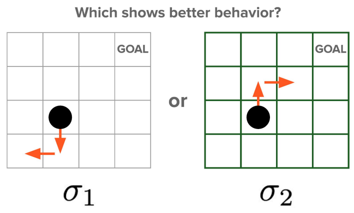
1. Does not always prefer optimal segments over suboptimal segments
2. Humans intuitively appear to consider state value, whereas the partial return preference model does not.
3. Indifferent to a constant shift in the output of the reward function.
4. When $|\sigma| = 1$, the discount factor is not considered, yet the discount factor and the reward function *interact* to determine the set of optimal policies.
5. Lacks identifiability with noiseless preferences
6. Less sample efficient than the regret model when learning from its own preferences.

Problems with the partial return preference model

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Partial return: $f(\sigma) = \text{sum of reward in } \sigma$

1. Does not always prefer optimal segments over suboptimal segments
2. Humans intuitively appear to consider state value, whereas the partial return preference model does not.
3. Indifferent to a constant shift in the output of the reward function.



Problems with the partial return preference model

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Partial return: $f(\sigma) = \text{sum of reward in } \sigma$

1. Does not always prefer optimal segments over suboptimal segments
2. Humans intuitively appear to consider state value, whereas the partial return preference model does not.
3. Indifferent to a constant shift in the output of the reward function.
4. When $|\sigma| = 1$, the discount factor is not considered, yet the discount factor and the reward function *interact* to determine the set of optimal policies.

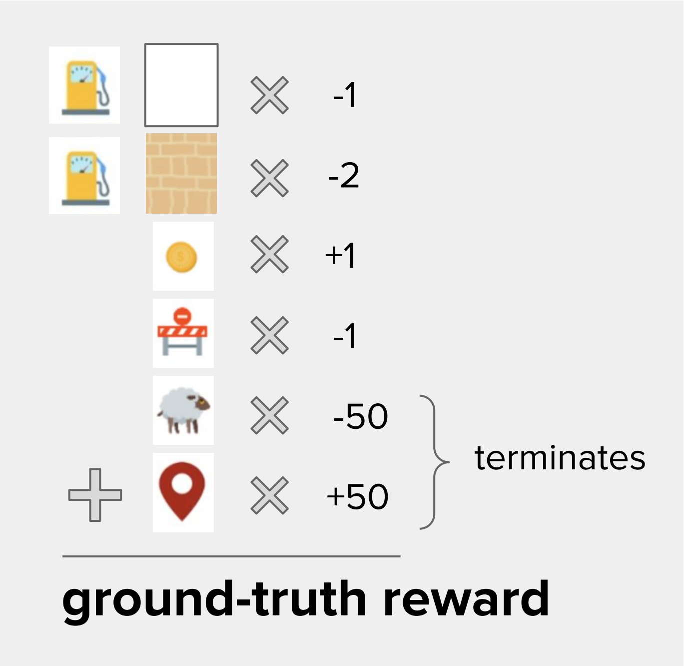
Problems with the partial return preference model

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

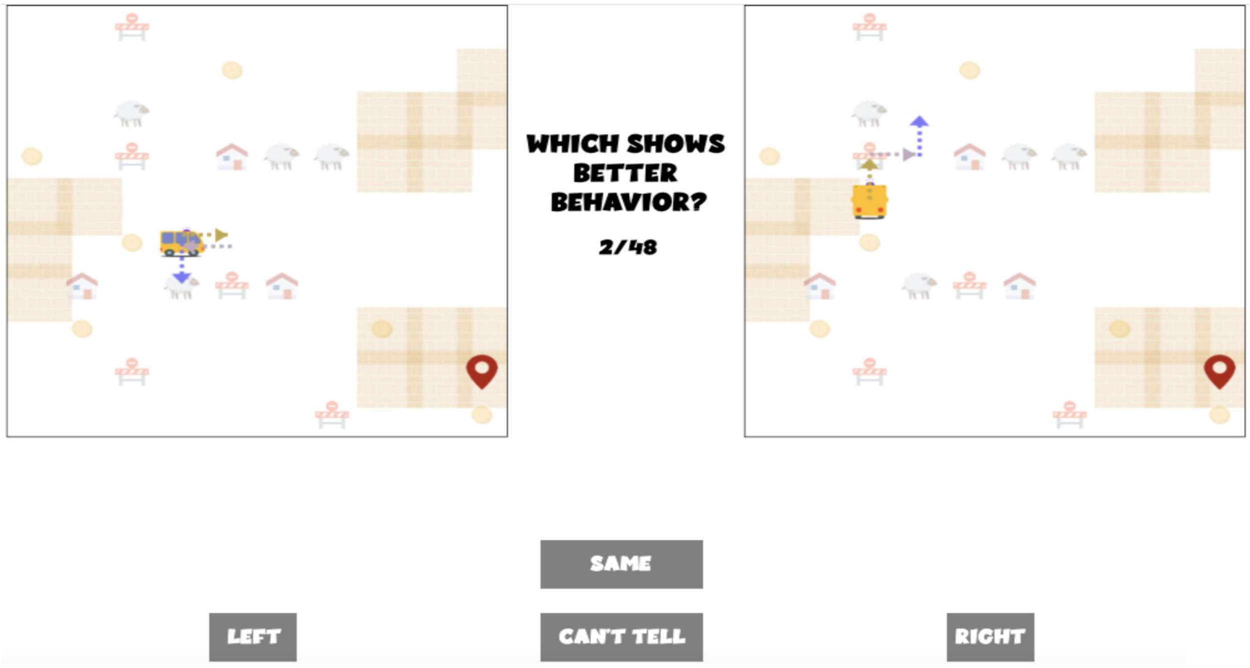
Partial return: $f(\sigma) = \text{sum of reward in } \sigma$

1. Does not always prefer optimal segments over suboptimal segments
2. Humans intuitively appear to consider state value, whereas the partial return preference model does not.
3. Indifferent to a constant shift in the output of the reward function.
4. When $|\sigma| = 1$, the discount factor is not considered, yet the discount factor and the reward function *interact* to determine the set of optimal policies.
5. Lacks identifiability in multiple contexts

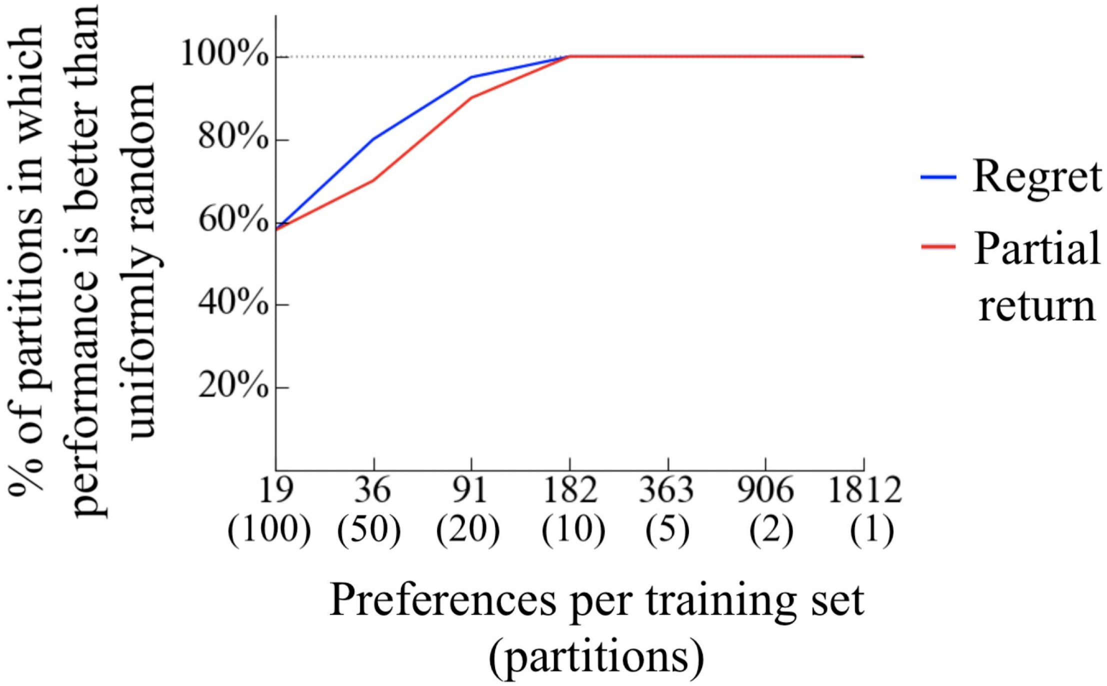
The delivery task



Preference elicitation



Performance with random partitions of human preferences dataset

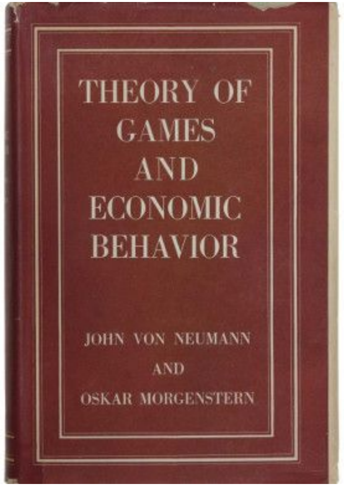


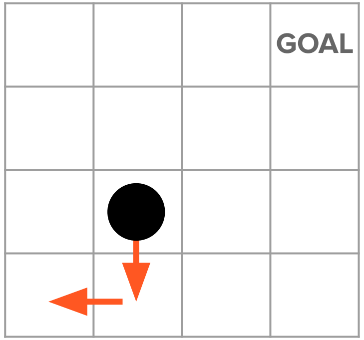
Limitations and future work

- **Efficient estimation of regret** for complex tasks (including deep learning settings).
- **Further test** the regret preference model.
- Understand the **partial return preference model's past success**, despite it being a poor model of humans.
- Develop **prescriptive methods to nudge humans** to conform more to normatively appealing preference models.

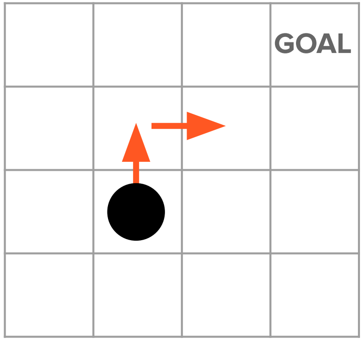
Summary

- A **new preference model with *regret(σ)*** as the segment statistic
    - Normative and descriptive comparisons to previous partial return model
- We show that **the choice of preference model impacts the performance** of learned reward functions.

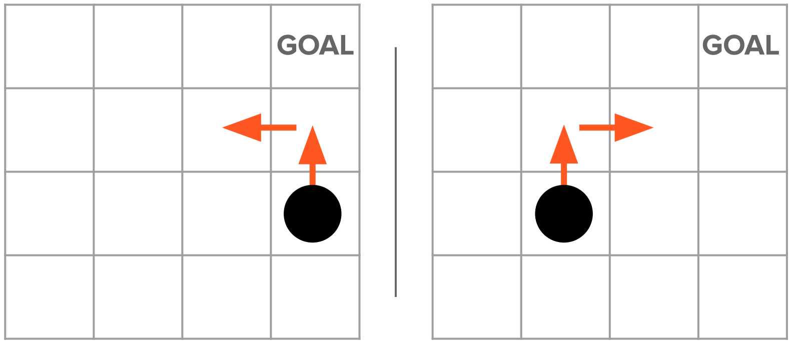

THEORY OF
GAMES
AND
ECONOMIC
BEHAVIOR
JOHN VON NEUMANN
AND
OSKAR MORGENSTERN



Equal partial return
Lower end state value

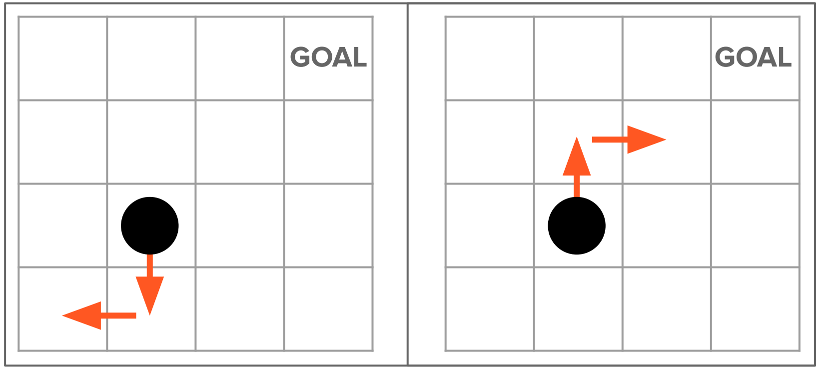


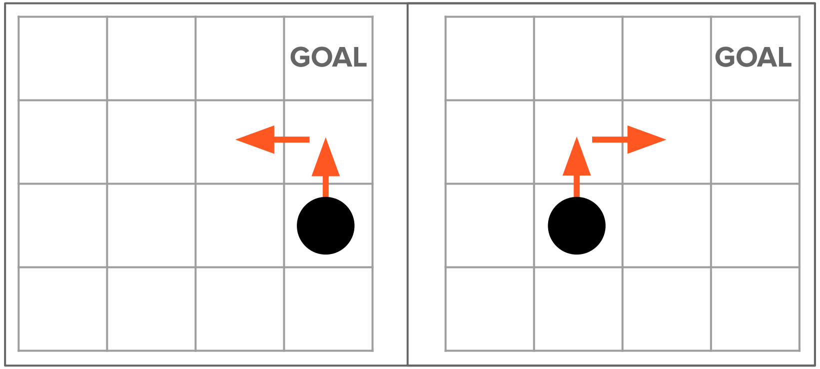
Equal partial return
Higher end state value

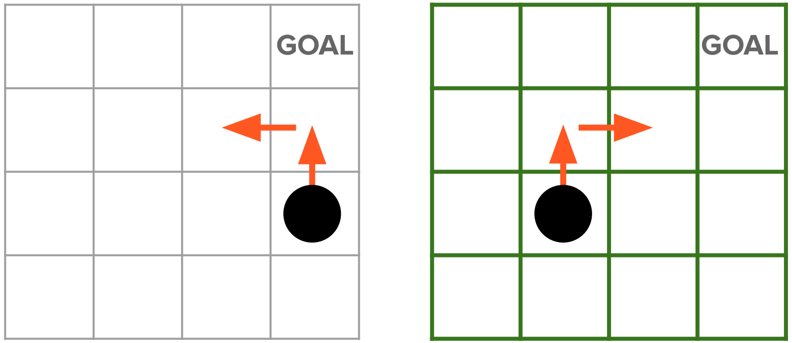


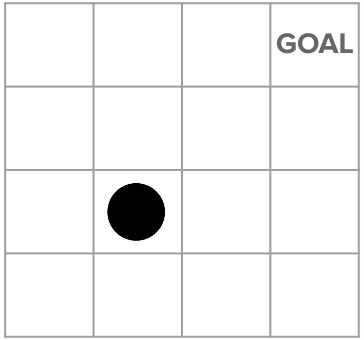
Equal partial return
Higher start state value

Equal partial return
Lower start state value


GOAL
GOAL


GOAL
GOAL


GOAL
GOAL


GOAL

100 randomly generated MDPs