Policy Shaping: Integrating Human Feedback with Reinforcement Learning

Shane Griffith

Kaushik Subramanian

Jonathan Scholz

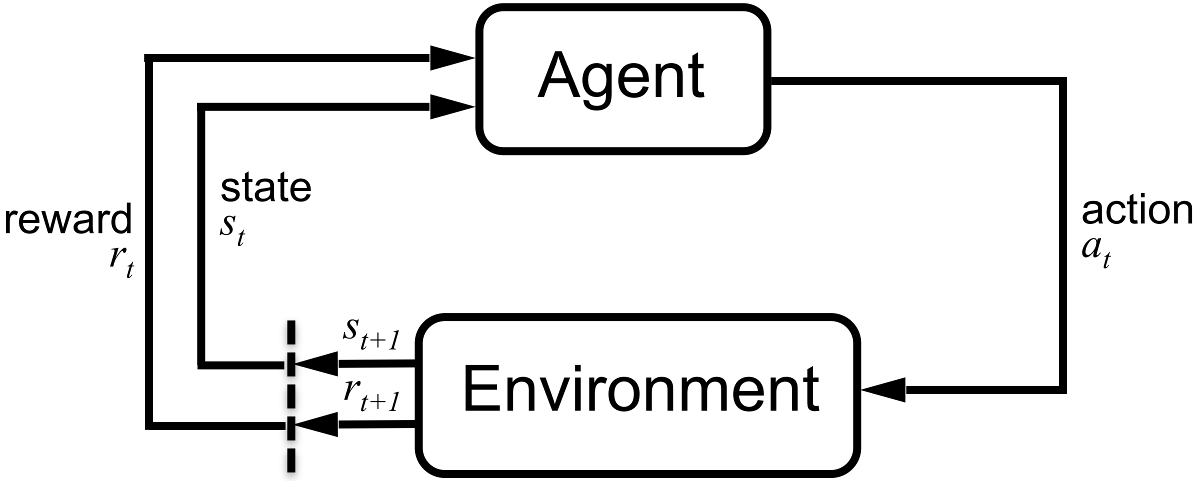
Charles L. Isbell

Andrea Thomaz

Institute for Robotics and Intelligent Machines
Georgia Institute of Technology

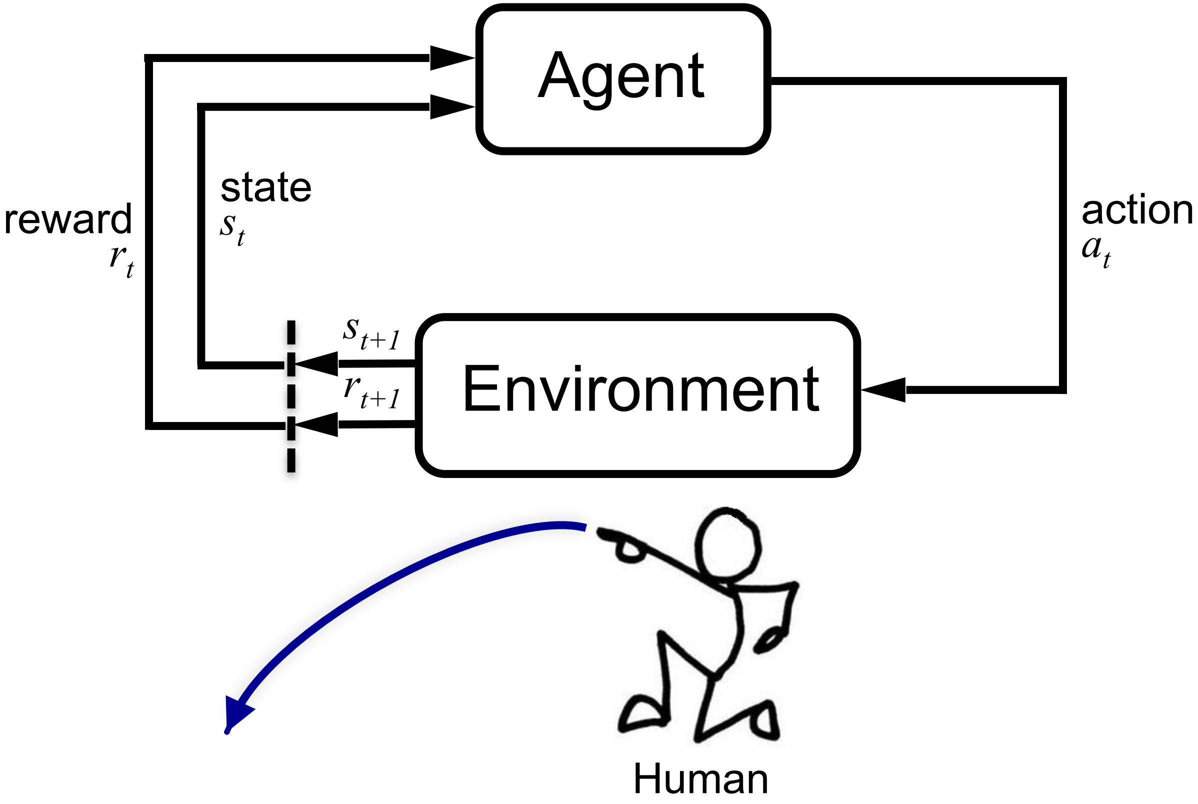
Presented at RLDM 2013 and published in NIPS 2013.

The Agent-Environment Interaction

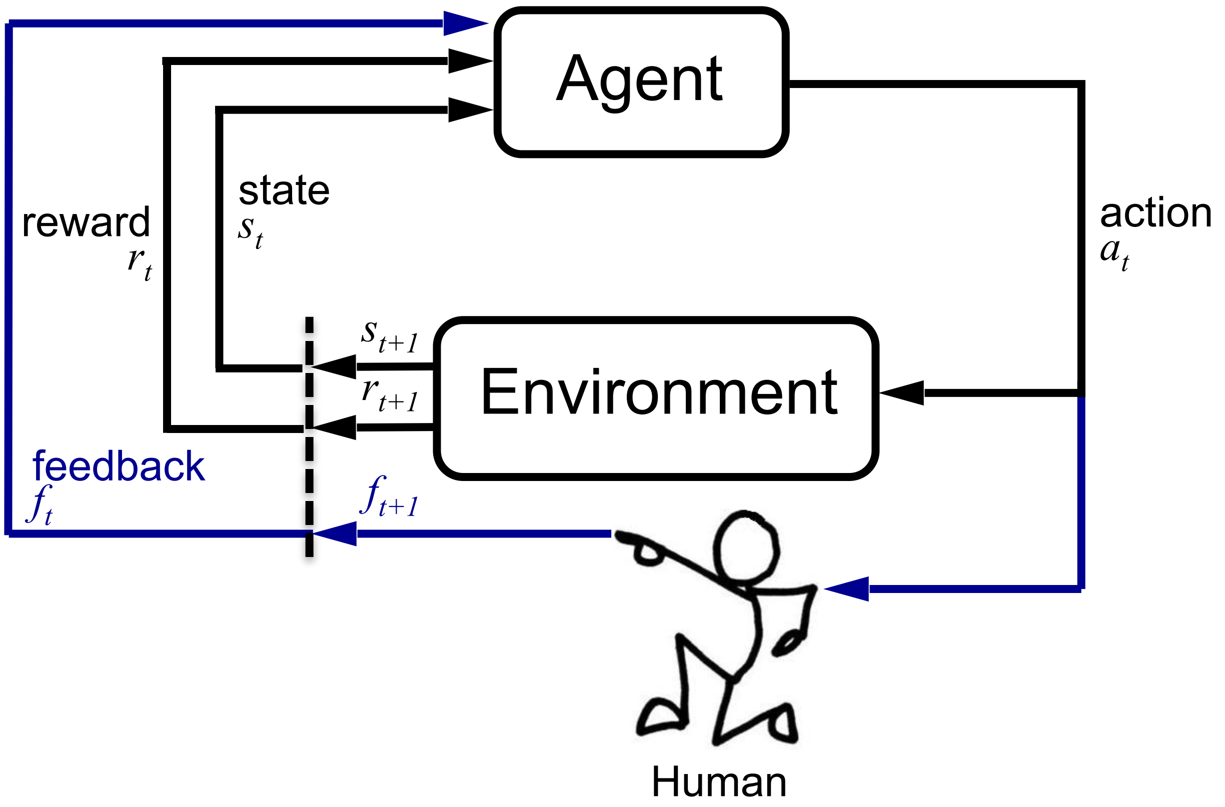


From Sutton and Barto. 1998

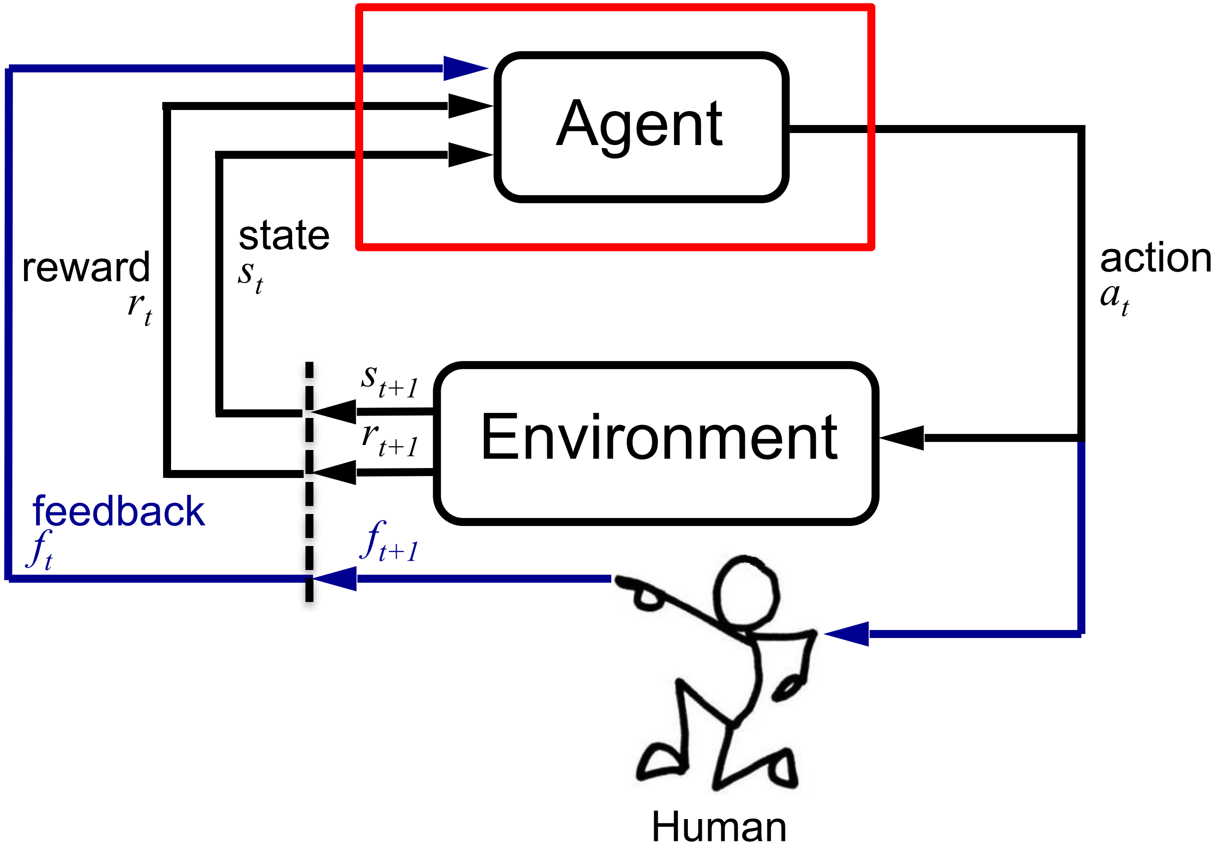
The Agent-Environment Interaction



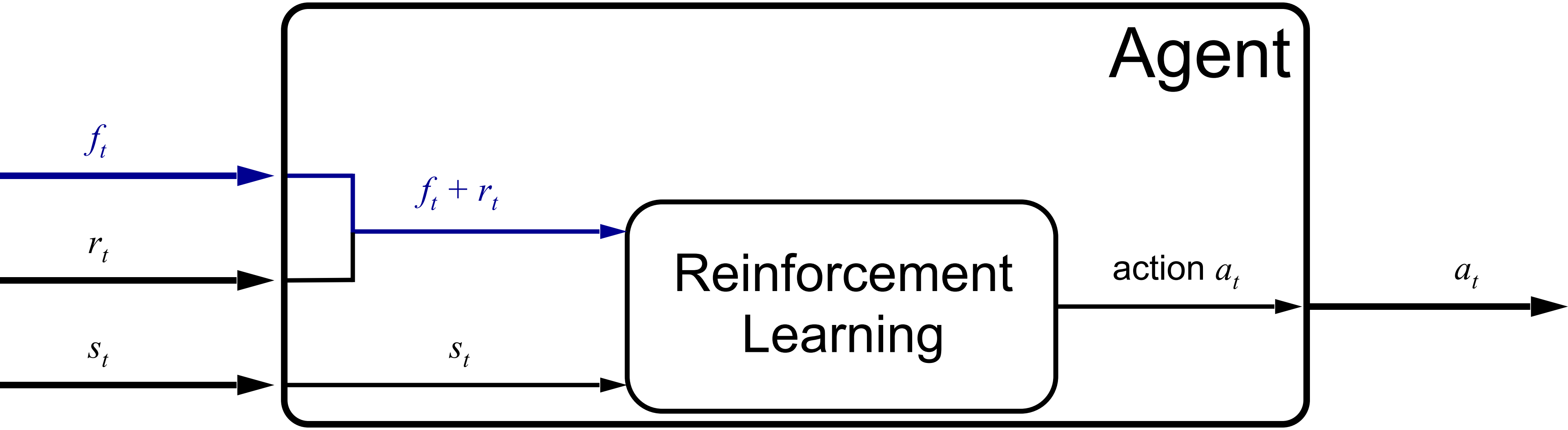
Integrating Human Feedback



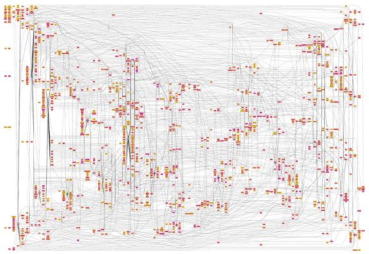
Integrating Human Feedback



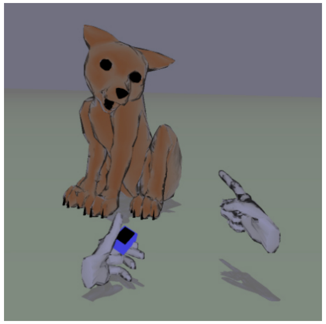
Adding the Feedback Channel



(Reward Shaping)

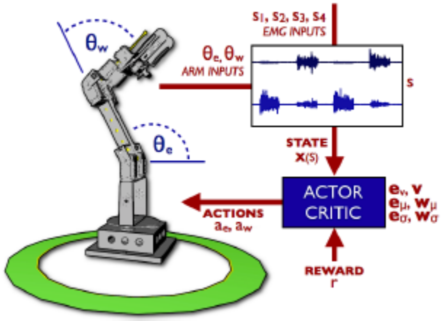


Isbell *et al.*; 2001



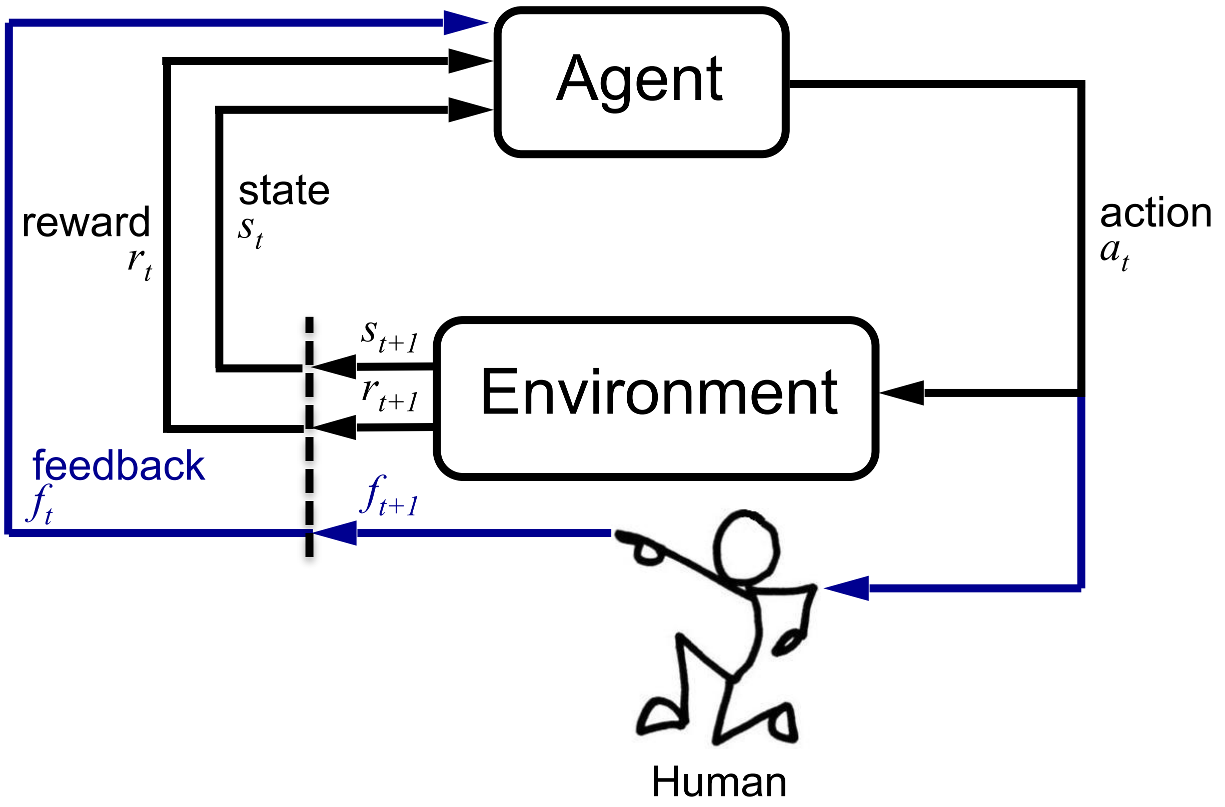
Blumberg *et al.*; 2002

Tenorio-Gonzalez *et al.*; 2010



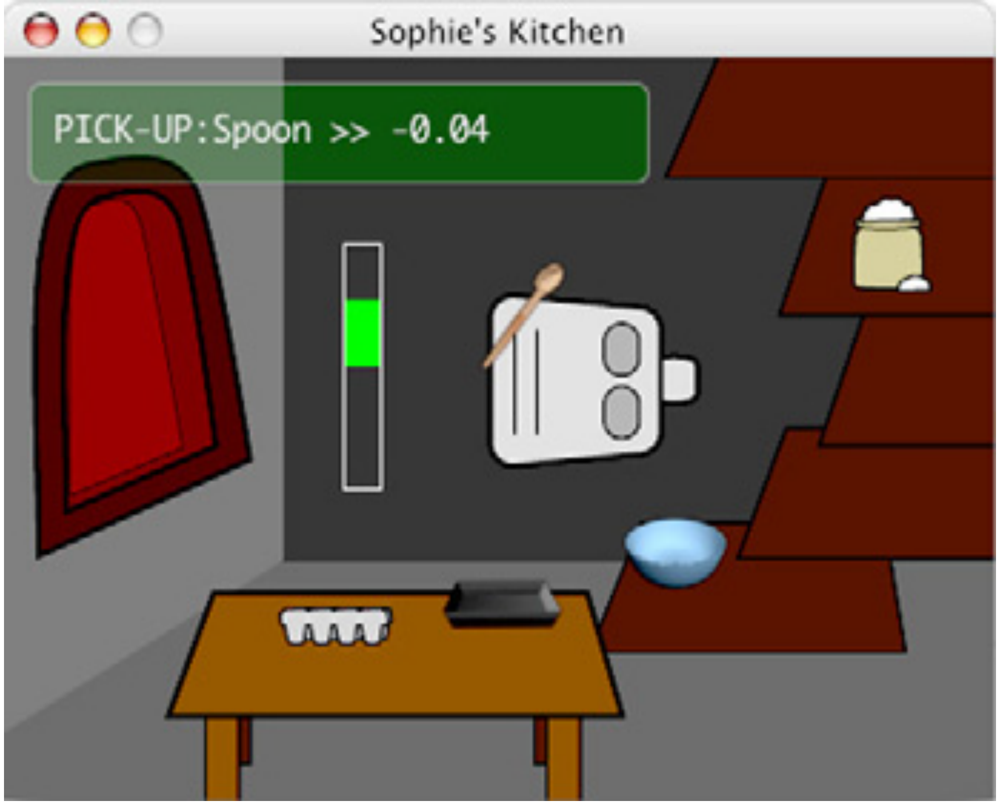
Pilarski *et al.*; 2011

Doesn’t the RL Loop Already Encapsulate Human Feedback?



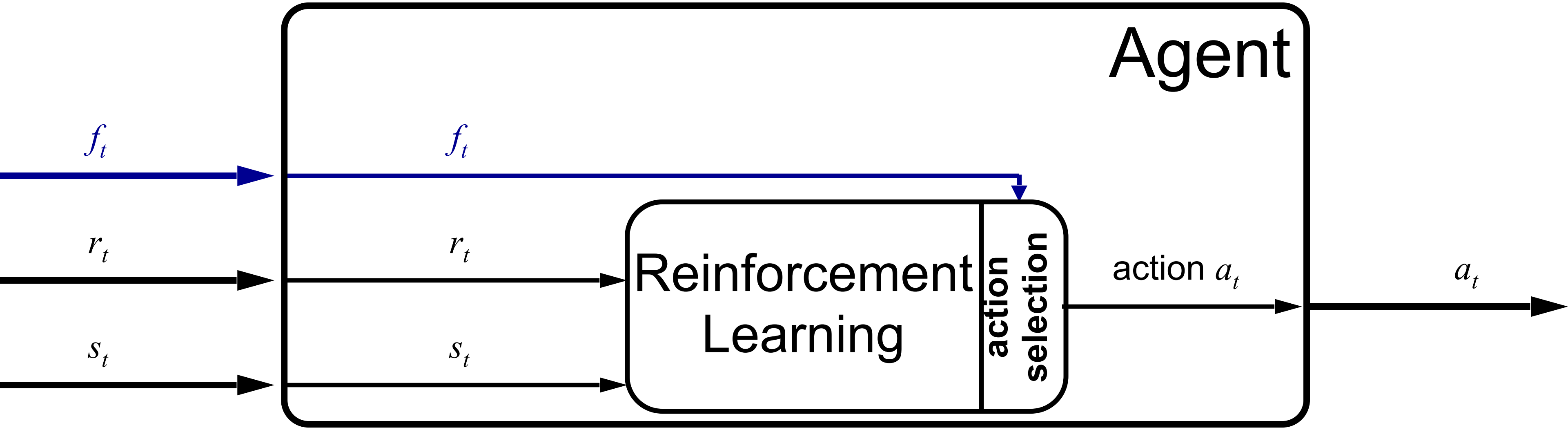
It’s not so simple.

(Thomaz and Breazeal; 2008)



“The communication from the human teaching partner cannot be merged into one single reward signal.”

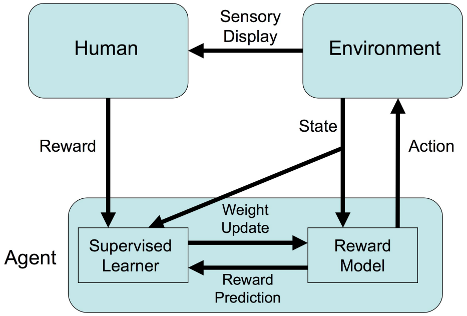
Separating Feedback from MDP Reward



(e.g., Action Biasing and Control Sharing)

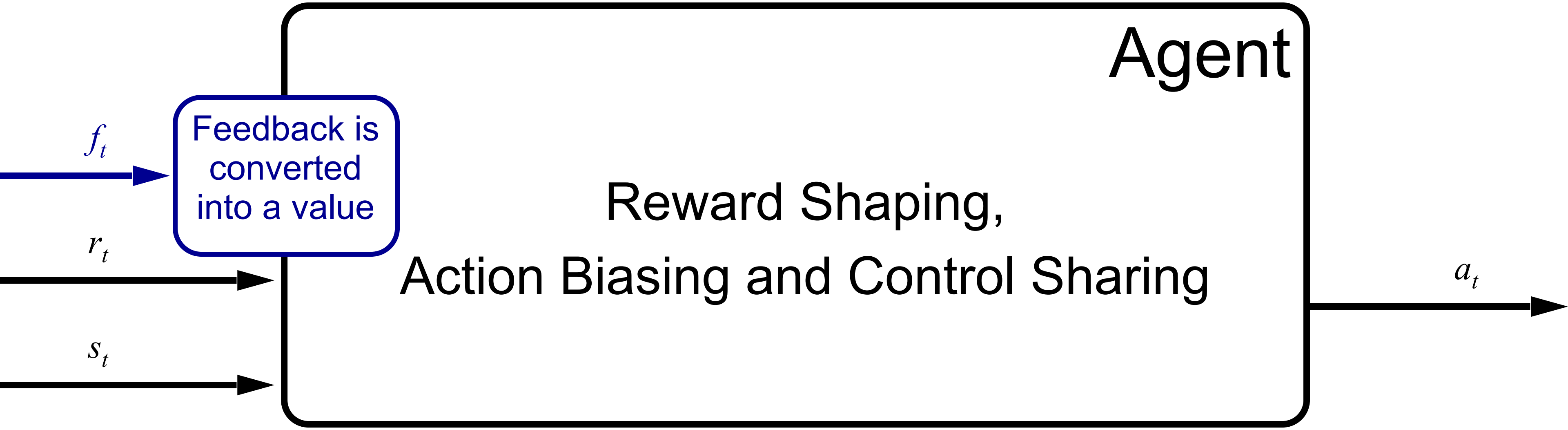
PICK-UP:Spoon >> -0.04

Thomaz and Breazeal; 2008



Knox and Stone; 2010

The Hidden Step In These Methods



f_t

r_t

s_t

Feedback is converted into a value

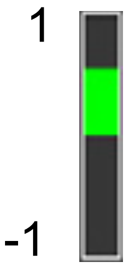
Agent

Reward Shaping,

Action Biasing and Control Sharing

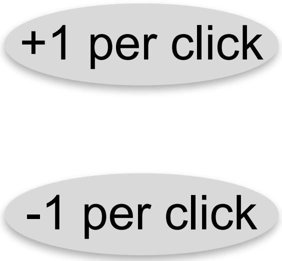
a_t

Reward slider



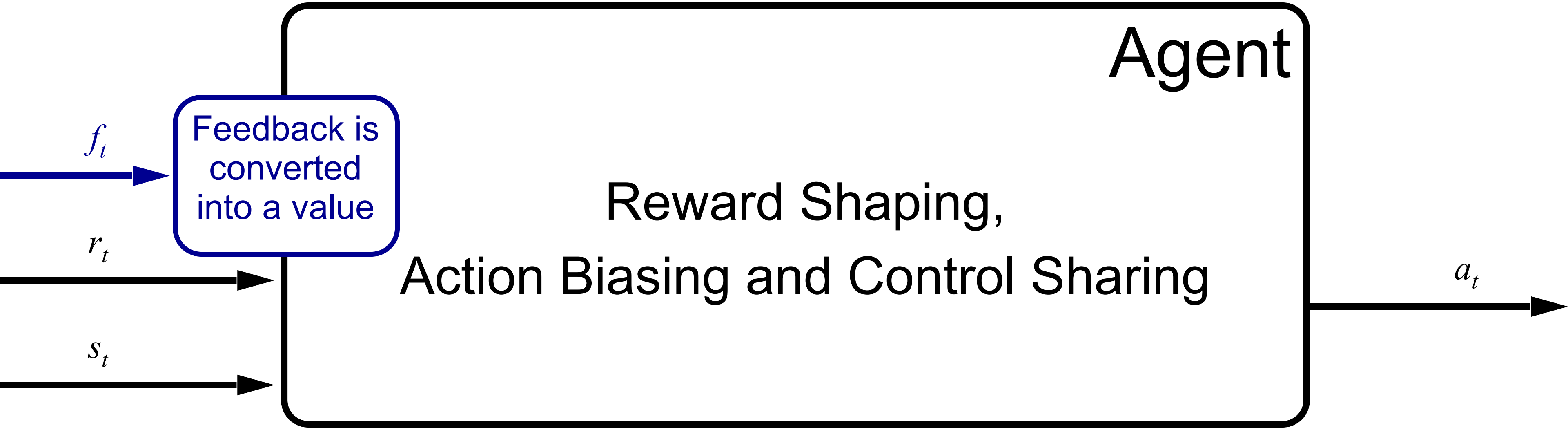
Thomaz and Breazeal; 2008

Human reinforcement buttons



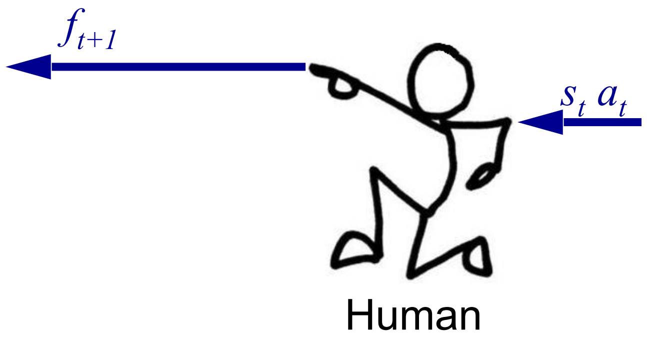
Knox and Stone; 2010

The Hidden Step In These Methods


Agent
f_t
Feedback is converted into a value
r_t
s_t
Reward Shaping,
Action Biasing and Control Sharing
a_t

- The conversion from feedback into a reward is *ad hoc*.
- Identifying a good reward requires solving the learning problem beforehand, which defeats the purpose.
- Feedback can have a delayed effect on exploration.

Policy Shaping



f_{t+1}

s_t a_t

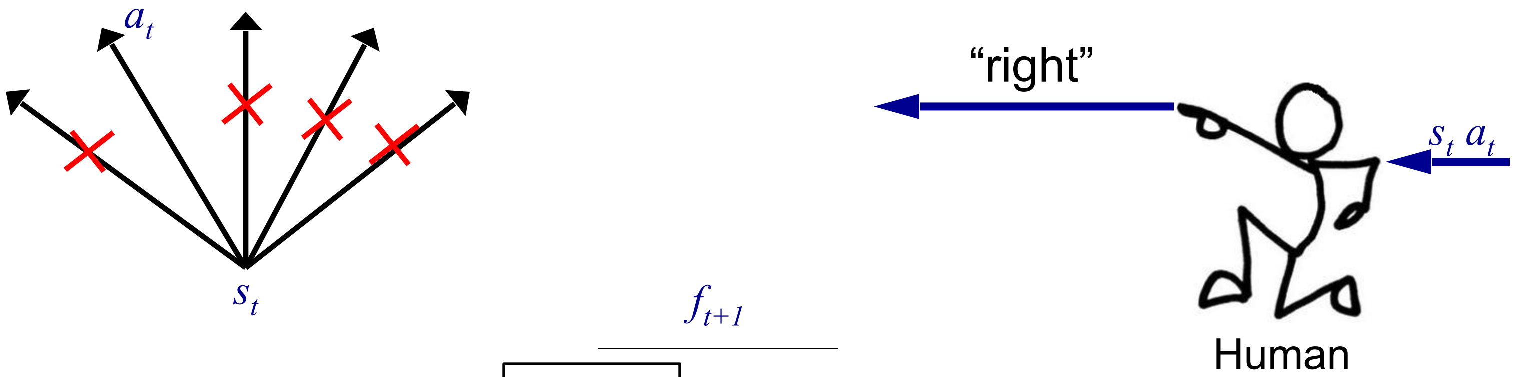
Human

f_{t+1}

“right” “wrong” …other labels?

What These Labels Mean

(assuming there's a single optimal action per state)



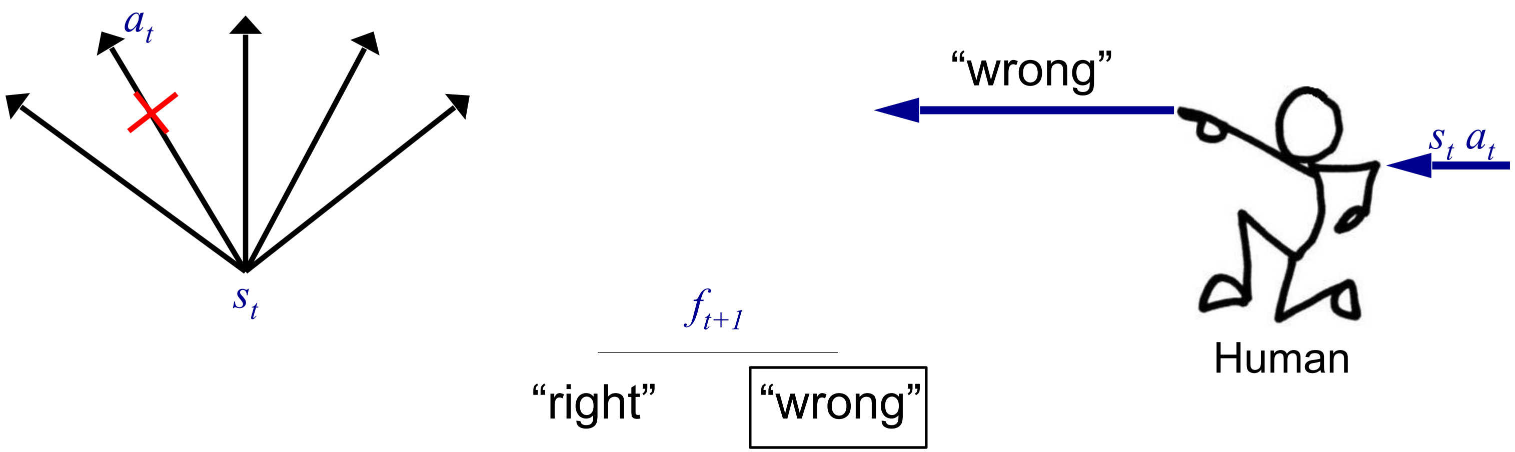
f_{t+1}

"right" "wrong"

$s_t\,a_t$ is "right": No further exploration is needed in state s_t

What These Labels Mean

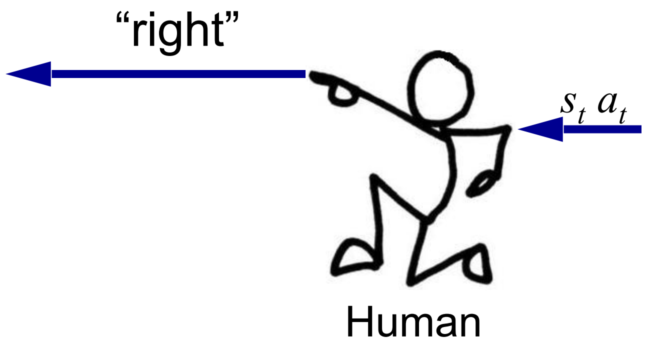
(assuming there's a single optimal action per state)



$s_t\, a_t$ is "right": No further exploration is needed in state s_t

$s_t\, a_t$ is "wrong": The agent should cease exploration down the path through action a_t in state s_t.

Feedback Consistency



Feedback History for $s_t\ a_t$

“right”	“right”
“right”	“wrong”
“right”	“right”
“right”	“right”
“right”	“right”

Noise in the feedback channel means we cannot simply prune actions from the search tree

Here feedback has consistency $\mathcal{C}$=0.9

(cf. Pradalier *et al.*, 2003)

Information Theoretic ‘Pruning’

BAYES RULE

$$\mathbf{P(H|D) = \frac{P(D|H)\ P(H)}{P(D)}}$$

DATA

Feedback History for $s_t\,a_t$	
“right”	“right”
“right”	“wrong”
“right”	“right”
“right”	“right”
“right”	“right”

HYPOTHESES

$s_t\,a_t$ is optimal

$s_t\,a_t$ is suboptimal

Advise

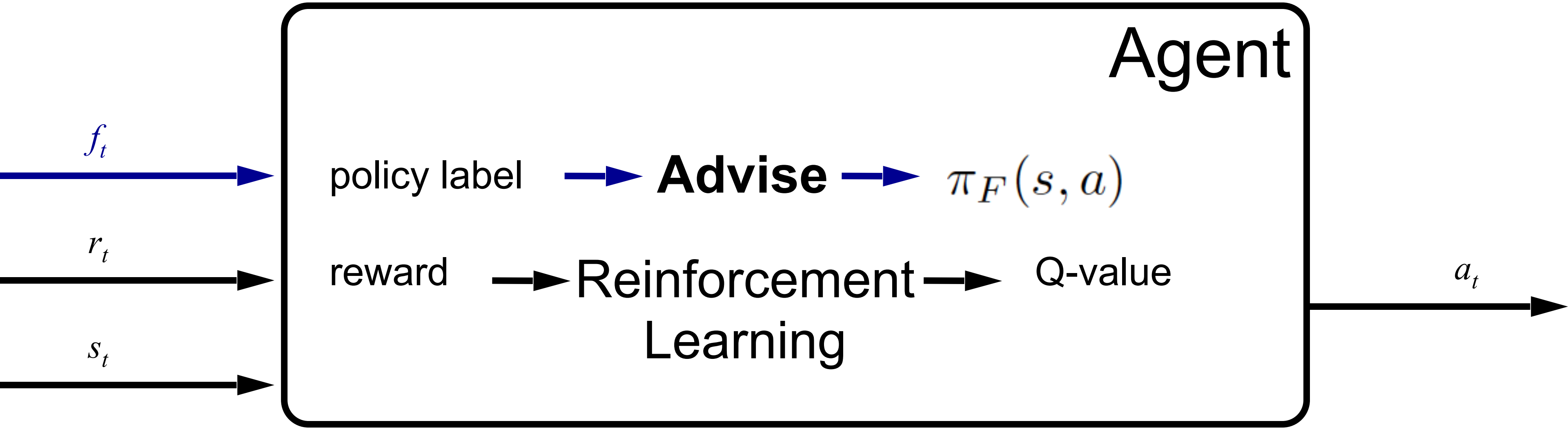
The probability the state-action pair, s,a, is optimal:

$$\pi_F(s,a) \propto \mathcal{C}^{\Delta_{s,a}}(1-\mathcal{C})^{\sum_{j \neq a} \Delta_{s,j}}$$

$\Delta_{s,a}$ - the difference between # right and # wrong labels

$\mathcal{C}$ - the feedback consistency

The Information Is Still Incompatible

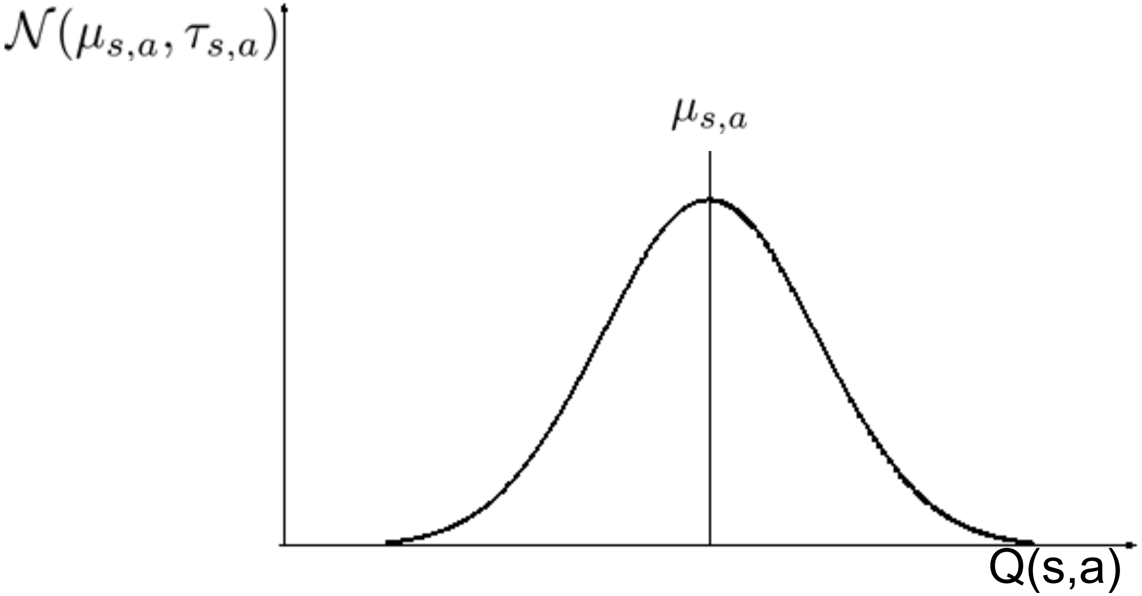


Top: a probability distribution over hypotheses about which action is optimal.

Bottom: an estimate of the long—term expected discounted reward for a state—action pair.

Can We Get Probabilities From Q-values?

- We can estimate the probability an action is optimal using *Pr(Q(s,a) > Q(s,!a))*
- The uncertainty in a Q-value can be modeled using a normal distribution.



(Dearden *et al.* 1998)

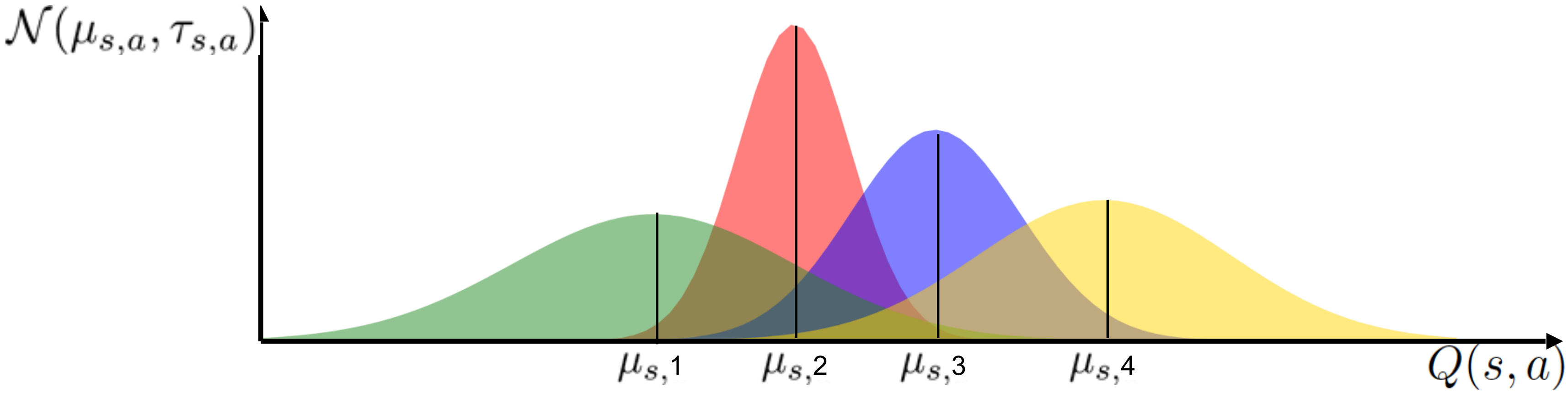
Bayesian Q-learning

(Dearden *et al.* 1998)

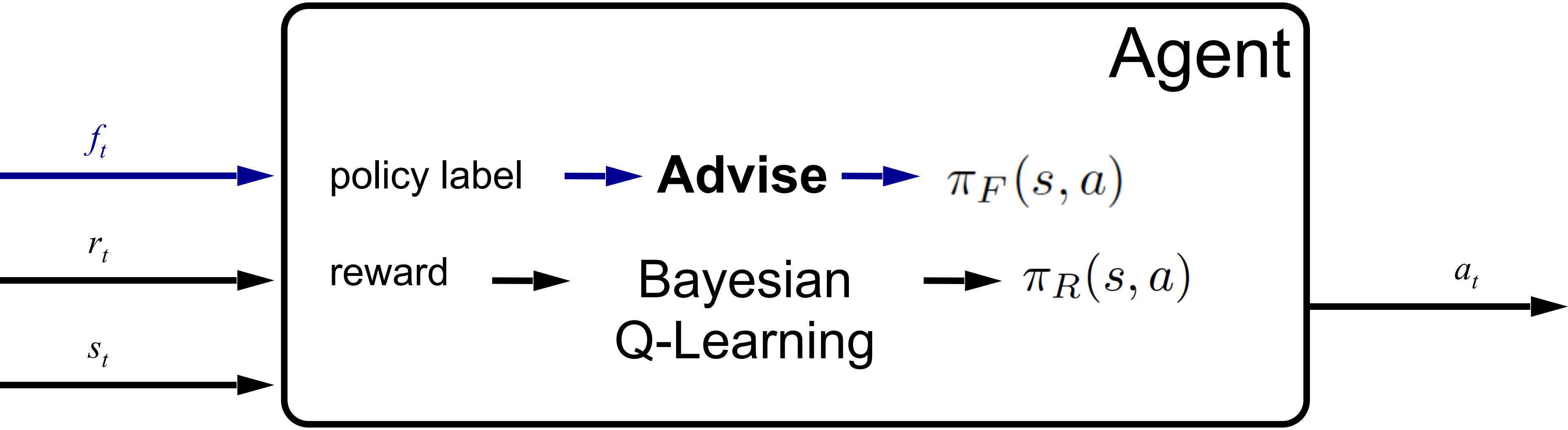
- Maintain parameters that specify a normal-gamma distribution for each state-action pair:

$$\mathcal{N}(\mu_{s,a},\tau_{s,a}) \sim Normal-Gamma(\mu_0^{s,a},\lambda^{s,a},\alpha^{s,a},\beta^{s,a})$$

- Sample each distribution, and then take the max 100 times to obtain $p(Q(s,a) > Q(s,!a))$.
- This gives: $\pi_R(s,a)$



Now The Signals Are Compatible



Top: a probability distribution over hypotheses about which action is optimal.

Bottom: a probability distribution over hypotheses about which action is optimal.

Learning From Both Sources of Information

$$p(H|rewards, feedback, C)$$

H is the hypothesis that s, a is optimal and $s, !a$ is suboptimal

$$\propto p(rewards, feedback|H, C)$$

$$\propto p(feedback|rewards, H, C) \times p(rewards|H, C)$$

$$\propto p(feedback|H, C) \times p(rewards|H, C)$$

$$\propto p(H|feedback, C) \times p(H|rewards)$$

$$\pi_F \qquad \times \qquad \pi_R$$

(Pradalier *et al.*, 2003)

(Bailer-Jones and Smith. 2011.)

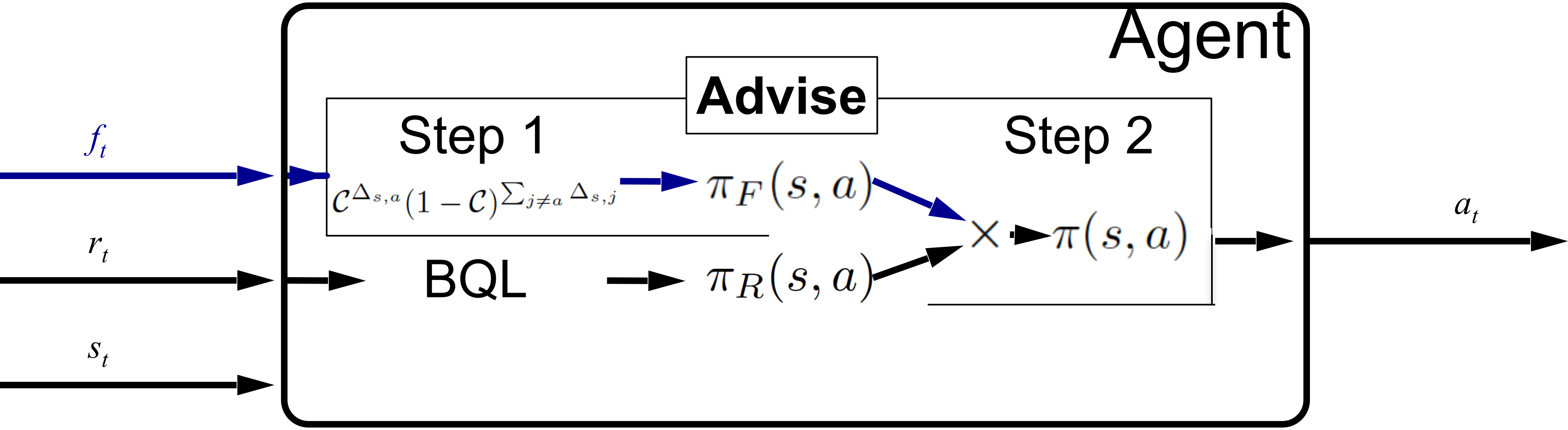
Learning From Both Sources of Information

$$\pi(s,a) \propto \pi_F(s,a) \times \pi_R(s,a)$$

(Pradalier *et al.*, 2003)

(Bailer-Jones and Smith. 2011.)

The Complete Advise Algorithm

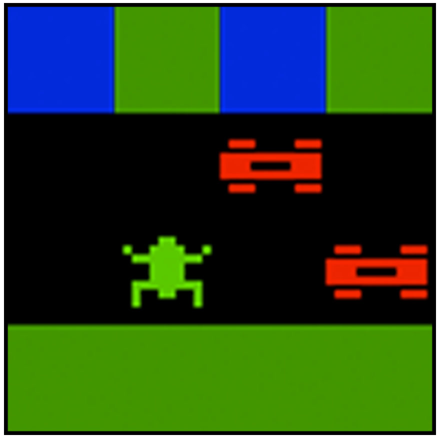


Step 1: Create the Human Feedback policy.

Step 2: Combine both policies into one.

The Domains We Used

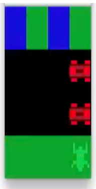
Frogger



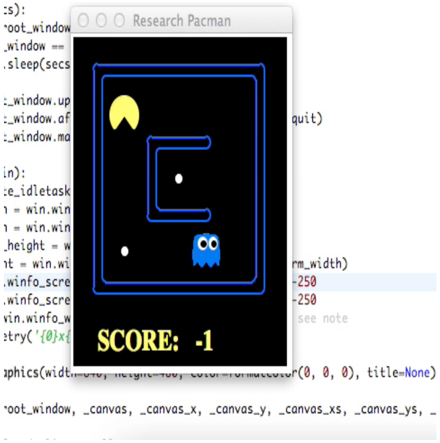
Pac-Man

The Domains We Used

Frogger



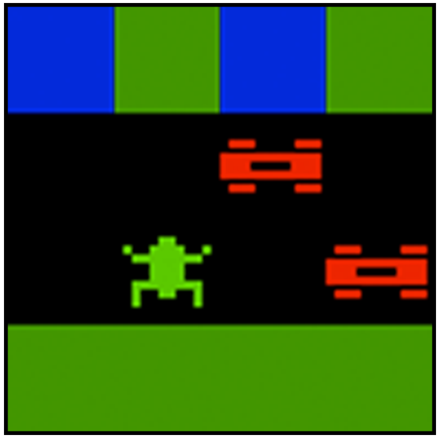
Pac-Man



```
:s):
root_window
_window ==
.sleep(secs

:_window.up
:_window.af                                   quit)
:_window.ma

in):
:e_idletask
1 = win.win
1 = win.win
_height = w
1t = win.wi                                   rm_width)
.winfo_scre                                   -250
.winfo_scre                                   -250
vin.winfo_w                                    see note
etry('{0}x{

aphics(widt                                  r(0, 0, 0), title=None)

root_window, _canvas, _canvas_x, _canvas_y, _canvas_xs, _canvas_ys, _
```

The Domains We Used

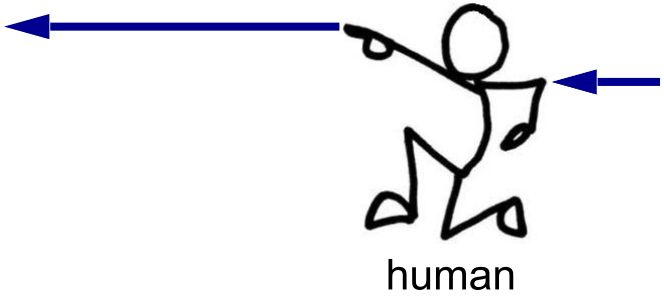
Frogger



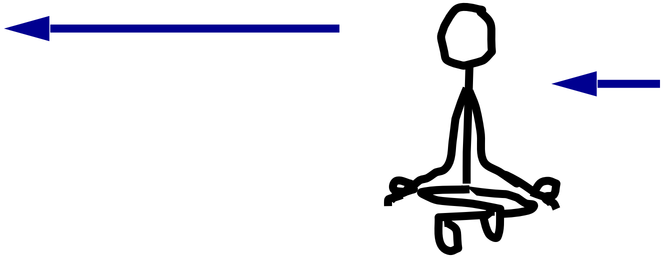
Pac-Man

States	**160**	**1890**
Actions per state	**5**	**2-3**
Episodes to converge	**~300**	**~300**

Source of Human Feedback



Source of Human Feedback



Oracle

- Instead of humans, we used an *oracle* to provide feedback.
- An oracle *simulates* the feedback from a real human.
- The oracle was a database consisting of the optimal action for each state.
- This allowed us to test several scenarios with different feedback likelihood and consistency.

The Four Scenarios We Tested

Ideal Case

$$\mathcal{L} = 1.0; \mathcal{C} = 1.0$$

Reduced Feedback

$$\mathcal{L} = \underline{0.1}; \mathcal{C} = 1.0$$

Reduced Consistency

$$\mathcal{L} = 1.0; \mathcal{C} = \underline{0.55}$$

Moderate Case

$$\mathcal{L} = \underline{0.5}; \mathcal{C} = \underline{0.8}$$

Methods We Evaluated

Reward Shaping | Action Biasing | Control Sharing | **Advise**

Feedback is Reward: Parameters

$H[s,a]$ Stores the accumulated human reward.

$r_h, -r_h$ Maps feedback to reward.

$\mathrm{B}[s,a]$ Stores the human influence value.

b The amount $\mathrm{B}[s,a]$ is incremented each time feedback is received for s,a.

d The decay rate of $\mathrm{B}[s,a]$.

Methods We Evaluated

Reward Shaping | Action Biasing | Control Sharing | **Advise**

$H[s,a]$ accumulated reward

$\mathrm{B}[s,a]$ human influence

$$R'(s,a) \leftarrow R(s,a) + \mathrm{B}[s,a] \times H[s,a]$$

Information in feedback is input into the RL algorithm by adding it to the MDP reward.

Methods We Evaluated

Reward Shaping | Action Biasing | Control Sharing | **Advise**

$H[s,a]$ accumulated reward

$\mathrm{B}[s,a]$ human influence

$$\operatorname{argmax}_a \hat{Q}(s,a) + \mathrm{B}[s,a] \times H[s,a]$$

Information in feedback is accumulated and used to bias the RL policy at decision making time.

Methods We Evaluated

Reward Shaping | Action Biasing | Control Sharing | **Advise**

$H[s,a]$ accumulated reward

$\mathrm{B}[s,a]$ human influence

$$P(a = \operatorname{argmax}_a H[s,a]) = min(\mathrm{B}[s,a], 1.0)$$

The probability of choosing an action from the feedback policy is equal to the human influence value.

Methods We Evaluated

Reward Shaping | Action Biasing | Control Sharing | **Advise**

Feedback is Policy Labels: Parameters

$\pi_F(s,a)$ Stores the feedback policy.

$\hat{C}$ The estimated feedback consistency.

Methods We Evaluated

Reward Shaping | Action Biasing | Control Sharing | **Advise**

$\mathcal{C}$ feedback consistency

Step 1: $\pi_F(s,a) \propto \mathcal{C}^{\Delta_{s,a}}(1-\mathcal{C})^{\sum_{j\neq a}\Delta_{s,j}}$

$\Delta_{s,a}$ - difference between # right and # wrong labels.

Step 2: $\pi(s,a) \propto \pi_F(s,a) \times \pi_R(s,a)$

Construct a separate policy from feedback, combine the feedback policy and the RL policy, and then sample it.

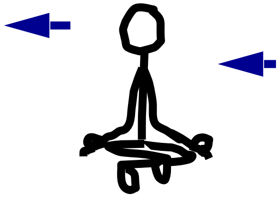
Experimental Setup Summary

Domains

Frogger

Pac-Man

Feedback



Oracle

Scenarios

Ideal Case

Reduced Feedback

Reduced Consistency

Moderate Case

Methods

Reward Shaping

Action Biasing

Control Sharing

Advise

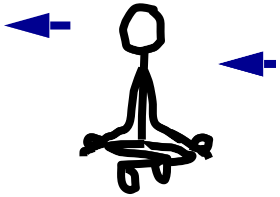
Comparing **Advise** to Alternative Methods

Domains

Frogger

Pac-Man

Feedback



Oracle

Scenarios

Ideal Case

Reduced Feedback

Reduced Consistency

Moderate Case

Methods

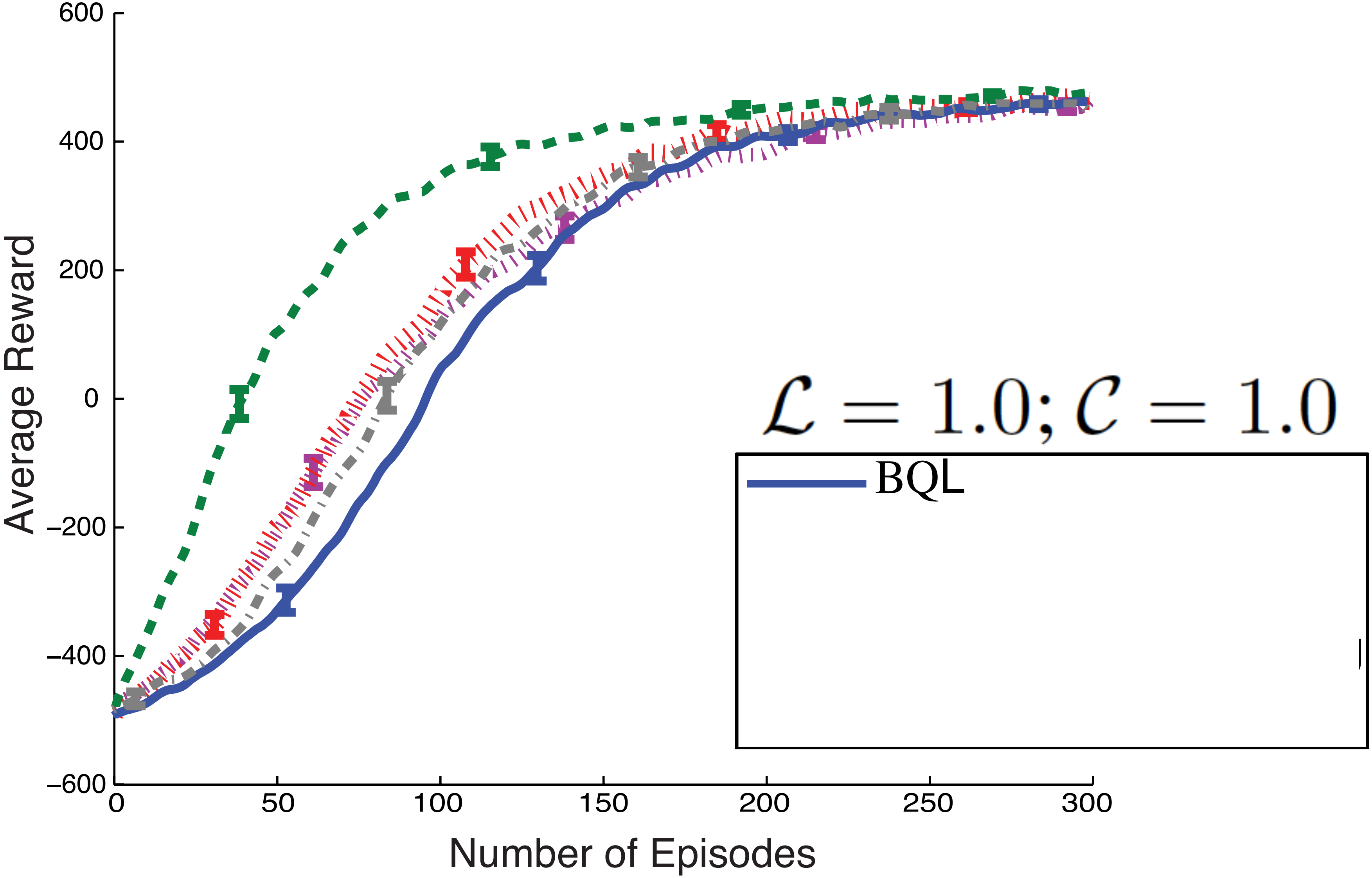
Reward Shaping

Action Biasing

Control Sharing

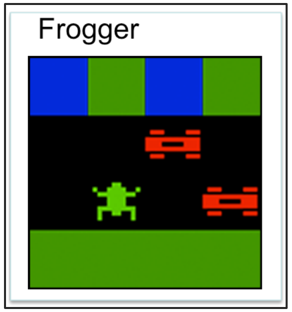
Advise

Learning with Ideal Feedback in Frogger

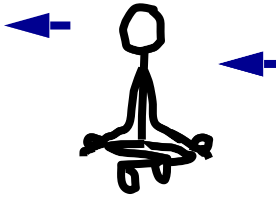


Comparing **Advise** to Alternative Methods

Domains



Feedback



Oracle

Scenarios

Ideal Case

Reduced Feedback

Reduced Consistency

Moderate Case

Methods

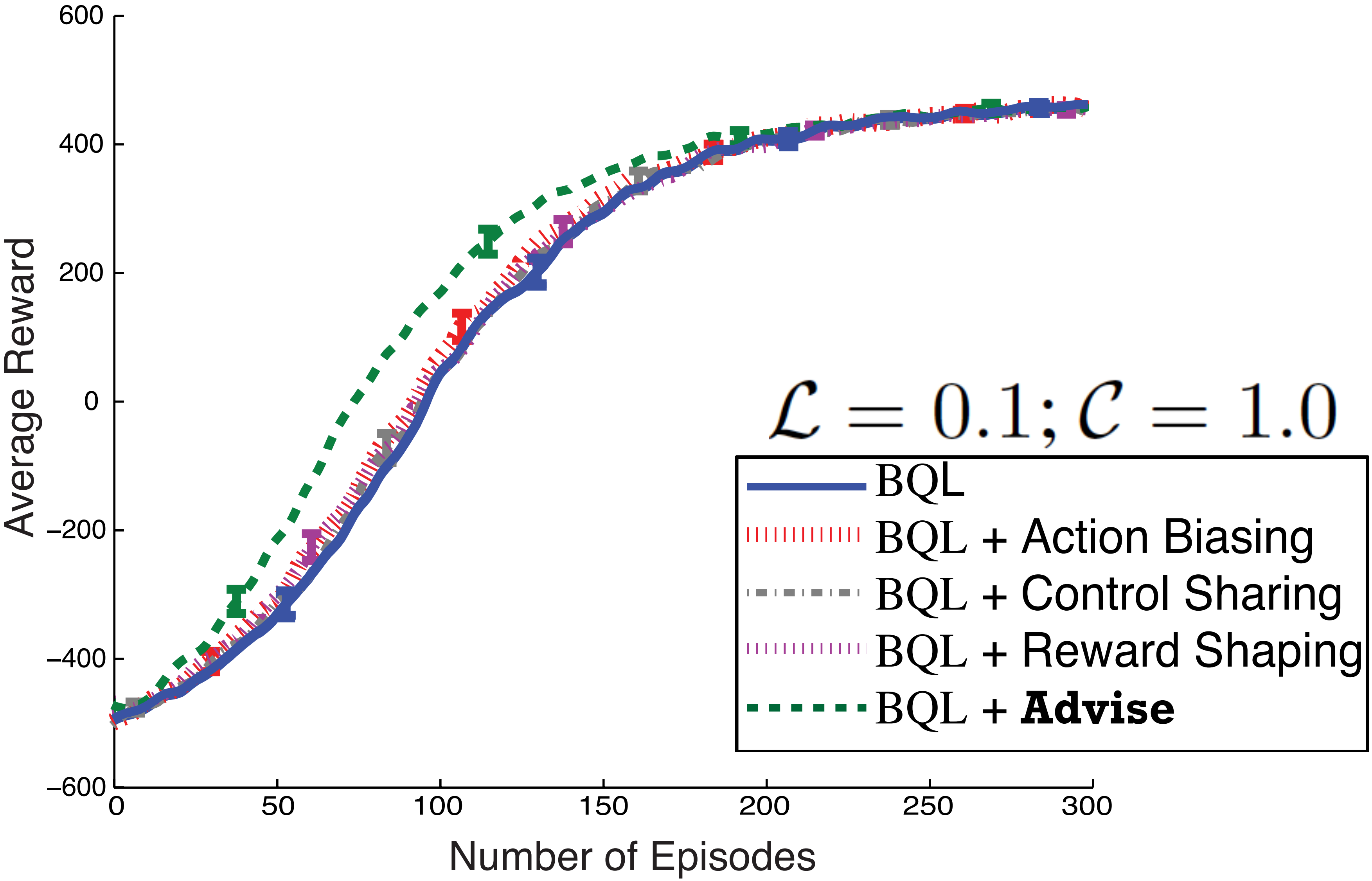
Reward Shaping

Action Biasing

Control Sharing

Advise

Reducing the Feedback Likelihood



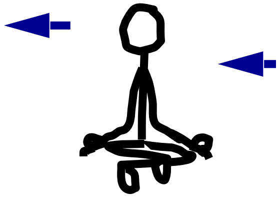
Comparing **Advise** to Alternative Methods

Domains

Frogger

Pac-Man

Feedback



Oracle

Scenarios

Ideal Case

Reduced Feedback

Reduced Consistency

Moderate Case

Methods

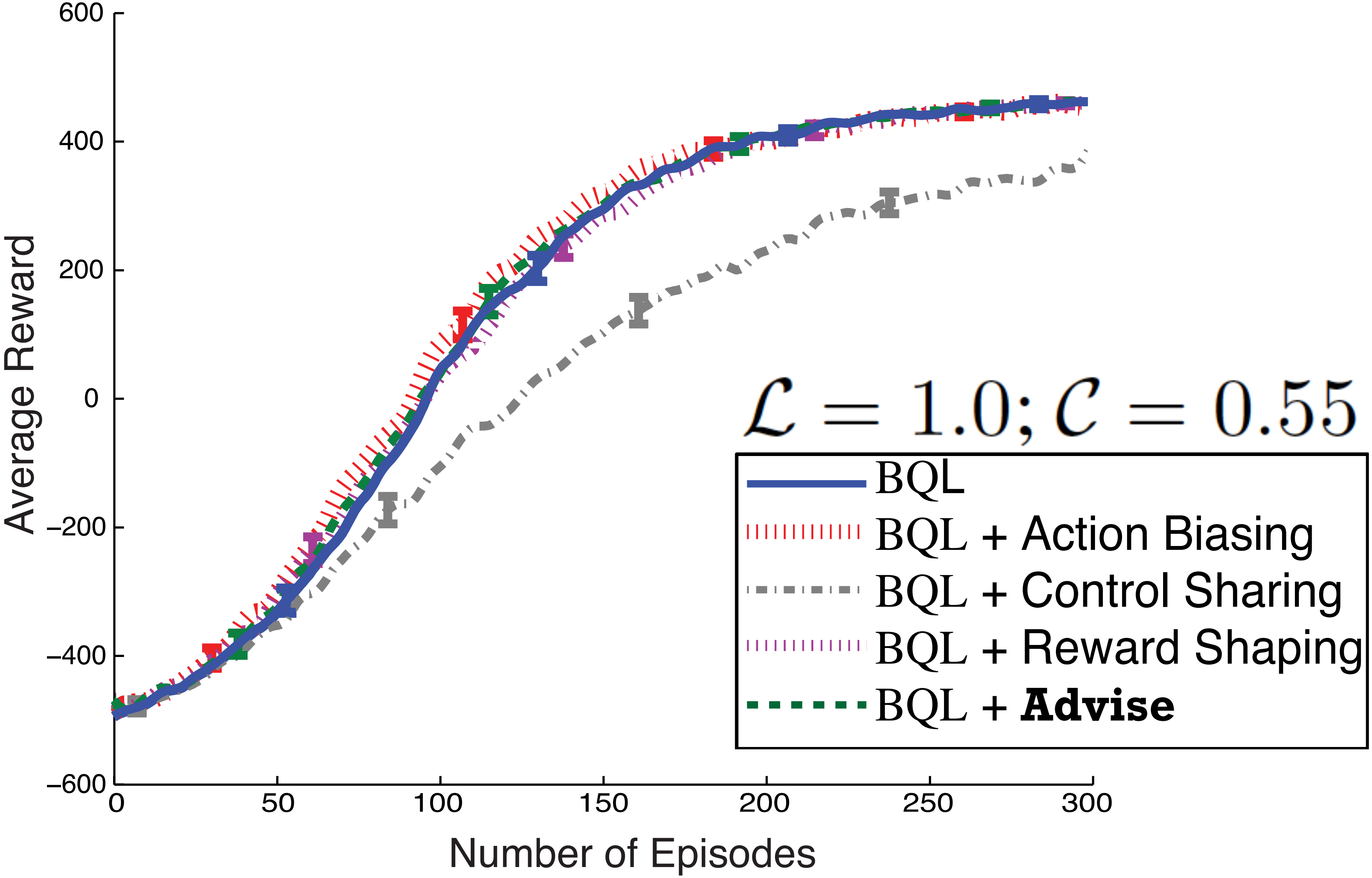
Reward Shaping

Action Biasing

Control Sharing

Advise

Reducing the Feedback Consistency



$\mathcal{L} = 1.0; \mathcal{C} = 0.55$

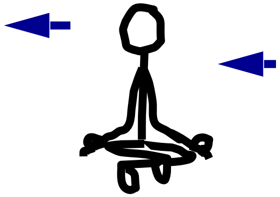
Comparing **Advise** to Alternative Methods

Domains

Frogger

Pac-Man

Feedback



Oracle

Scenarios

Ideal Case

Reduced Feedback

Reduced Consistency

Moderate Case

Methods

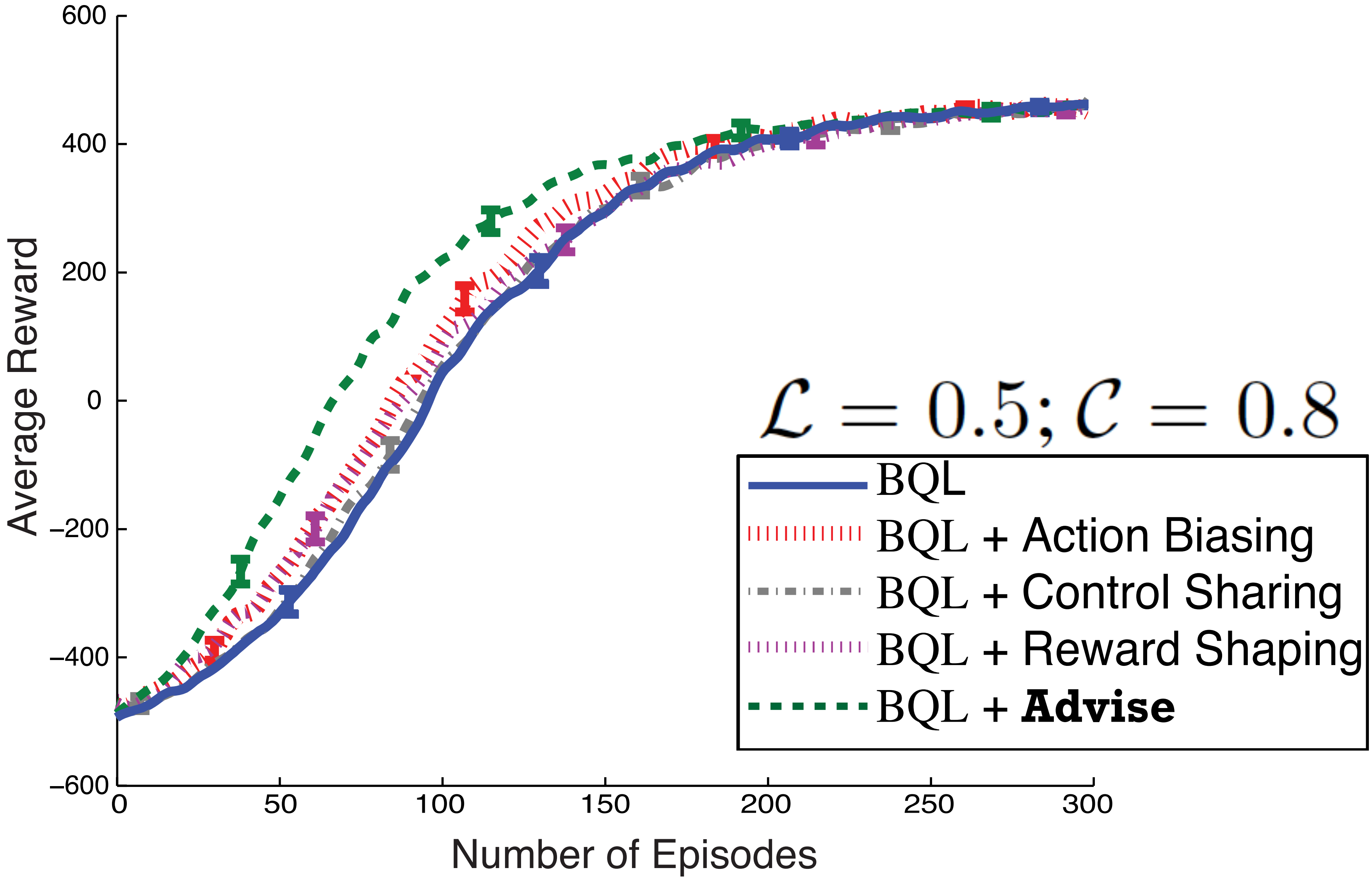
Reward Shaping

Action Biasing

Control Sharing

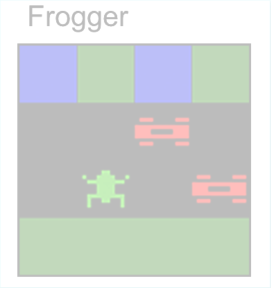
Advise

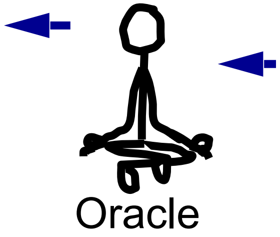
Moderate Likelihood and Consistency



Comparing **Advise** to Alternative Methods

Domains **Feedback** **Scenarios** **Methods**





- Ideal Case
- Reduced Feedback
- Reduced Consistency
- Moderate Case

- Reward Shaping
- Action Biasing
- Control Sharing
- **Advise**

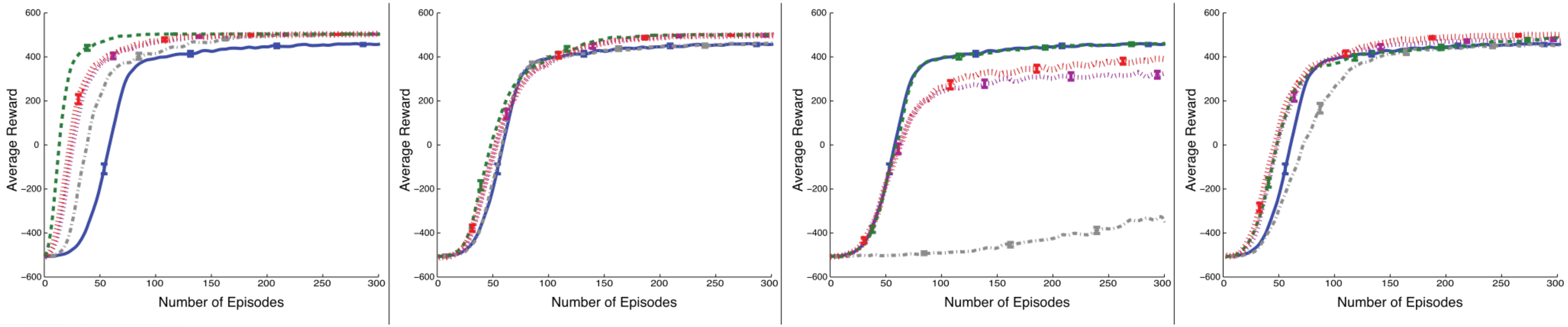
We Observed Similar Trends in Pac-Man

Ideal Case $\mathcal{L} = 1.0; \mathcal{C} = 1.0$

↓ Likelihood $\mathcal{L} = 0.1; \mathcal{C} = 1.0$

↓ Consistency $\mathcal{L} = 1.0; \mathcal{C} = 0.55$

Moderate $\mathcal{L} = 0.5; \mathcal{C} = 0.8$



▬ BQL

‖‖‖ BQL + Action Biasing

-·-· BQL + Control Sharing

‖‖‖ BQL + Reward Shaping

- - - BQL + **Advise**

A Quantitative Look at Performance

	Ideal Case		Reduced Feedback	
	($\mathcal{L} = 1.0, \mathcal{C} = 1.0$)		($\mathcal{L} = 0.1, \mathcal{C} = 1.0$)	
	Pac-Man	Frogger	Pac-Man	Frogger
BQL + Action Biasing	0.58 ± 0.02	0.16 ± 0.05	0.16 ± 0.04	0.04 ± 0.06
BQL + Control Sharing	0.34 ± 0.03	0.07 ± 0.06	0.01 ± 0.12	0.02 ± 0.07
BQL + Reward Shaping	0.54 ± 0.02	0.11 ± 0.07	0.14 ± 0.04	0.03 ± 0.07
BQL + **Advise**	$\mathbf{0.77 \pm 0.02}$	$\mathbf{0.45 \pm 0.04}$	$\mathbf{0.21 \pm 0.05}$	$\mathbf{0.16 \pm 0.06}$

	Reduced Consistency		Moderate Case	
	($\mathcal{L} = 1.0, \mathcal{C} = 0.55$)		($\mathcal{L} = 0.5, \mathcal{C} = 0.8$)	
	Pac-Man	Frogger	Pac-Man	Frogger
BQL + Action Biasing	-0.33 ± 0.17	0.05 ± 0.06	$\mathbf{0.25 \pm 0.04}$	0.09 ± 0.06
BQL + Control Sharing	-2.87 ± 0.12	-0.32 ± 0.13	-0.18 ± 0.19	0.01 ± 0.07
BQL + Reward Shaping	-0.47 ± 0.30	0 ± 0.08	0.17 ± 0.12	0.05 ± 0.07
BQL + **Advise**	$\mathbf{-0.01 \pm 0.11}$	0.02 ± 0.07	0.13 ± 0.08	$\mathbf{0.22 \pm 0.06}$

A Follow-up Experiment

Action Biasing used an optimized conversion from feedback into reward.

$r_h, -r_h$

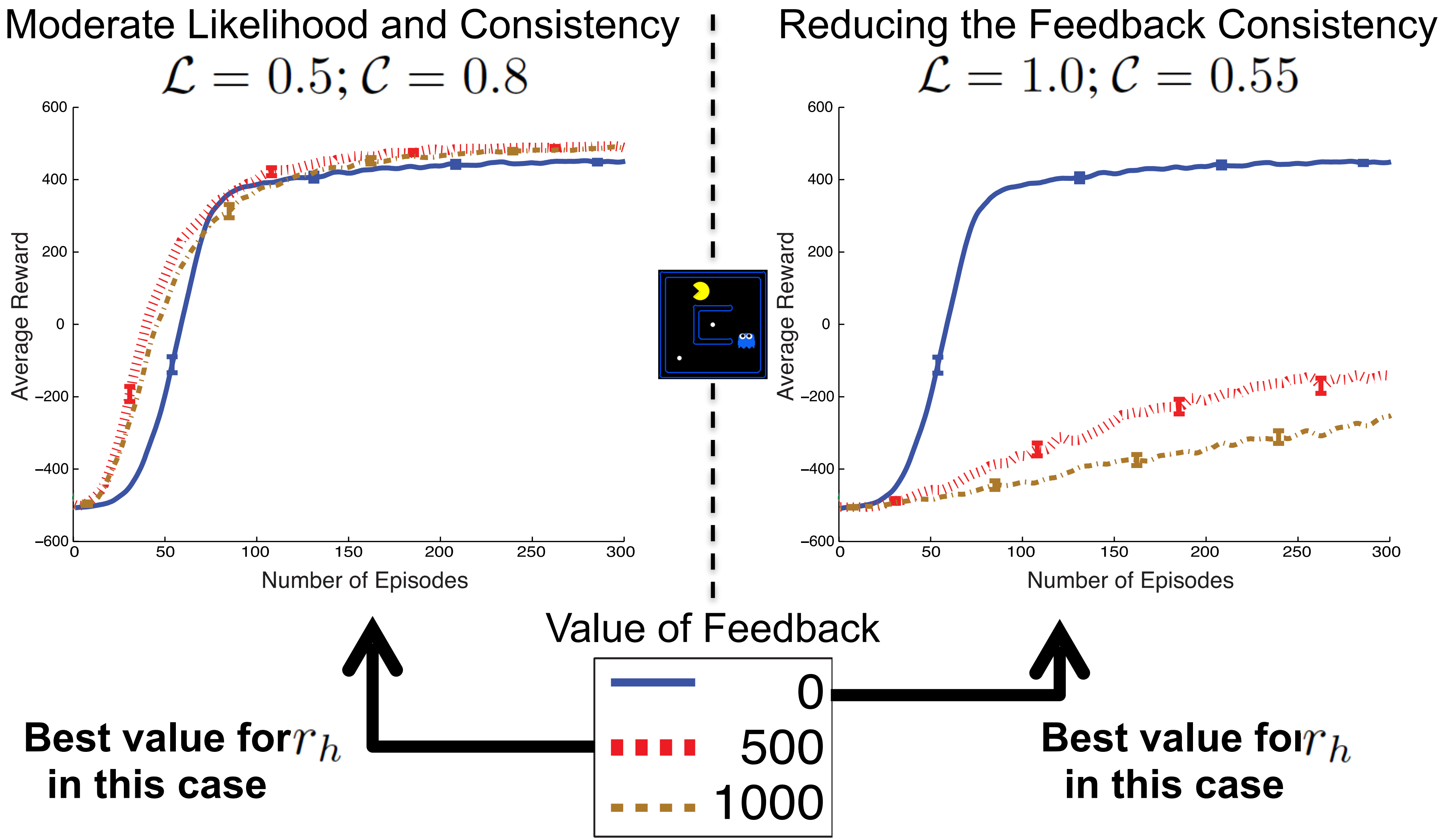
Depends on:

MDP reward

the feedback consistency

Our next experiment tested how action biasing performed if we varied the value of r.

How the Reward Parameter Affects Learning



Moderate Likelihood and Consistency

$\mathcal{L} = 0.5; \mathcal{C} = 0.8$

Reducing the Feedback Consistency

$\mathcal{L} = 1.0; \mathcal{C} = 0.55$

Value of Feedback

▬	0
■ ■ ■ ■	500
▪ ▪ ▪ ▪	1000

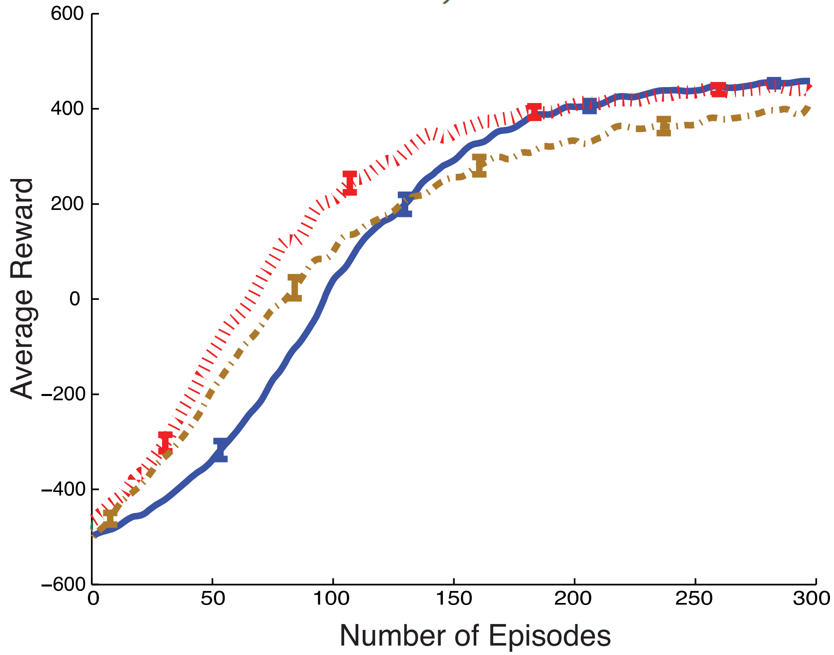
Best value for r_h in this case

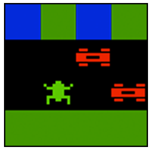
Best value for r_h in this case

How the Reward Parameter Affects Learning

Moderate Likelihood and Consistency

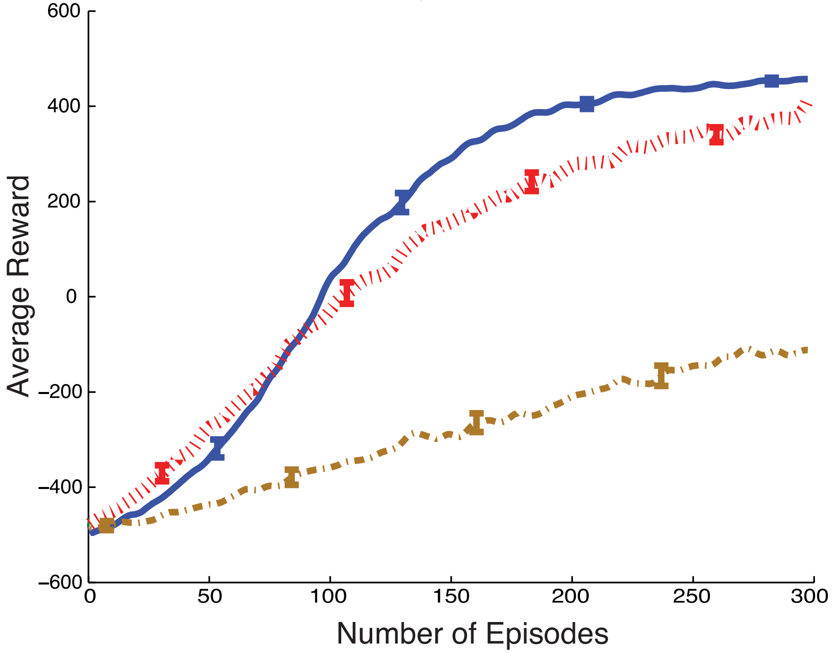
$\mathcal{L} = 0.5; \mathcal{C} = 0.8$





Reducing the Feedback Consistency

$\mathcal{L} = 1.0; \mathcal{C} = 0.55$



Value of Feedback

▬	0
▪▪▪▪	500
▪ ▪ ▪ ▪	1000

Best value for r_h **in this case**

Best value for r_h **in this case**

Another Follow-up Experiment

Reward Shaping, Action Biasing, and Control sharing used optimized human influence parameters.

$b, \ d$

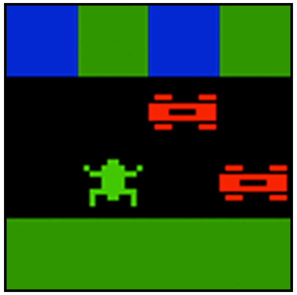
Depends on:

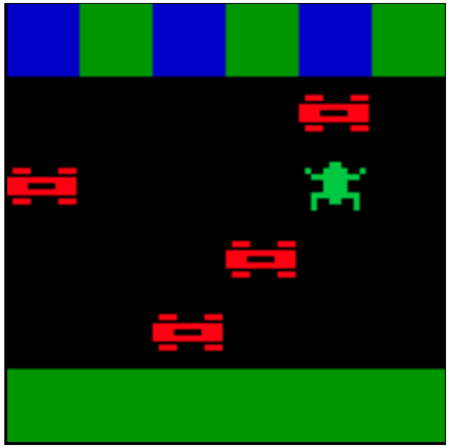
the size of the domain

the feedback consistency

Our next experiment varied the domain size to show that these parameters depend more on that than the information in human feedback.

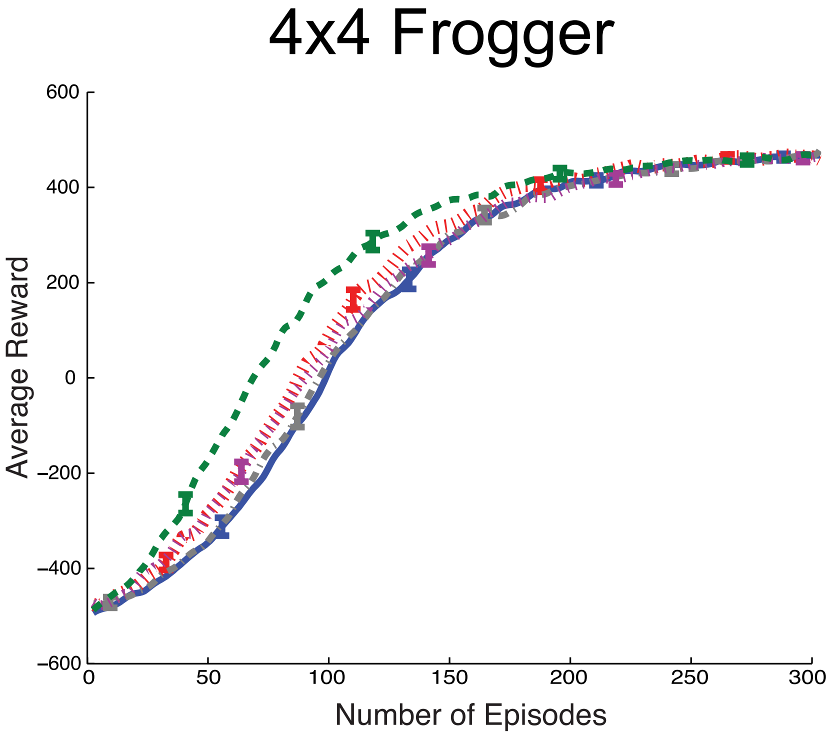
Enlarging Frogger

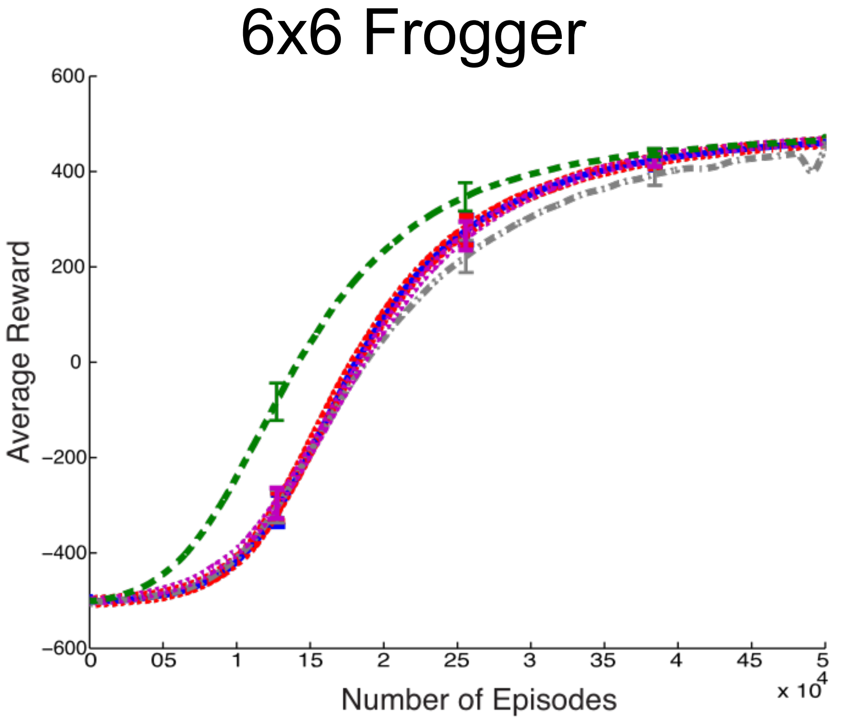


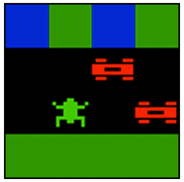


Domain Size	**4x4**	**6x6**
States	**160**	**33,360**
Episodes to Converge	**~300**	**~50,000**

How the Domain Size Affects Learning







4x4 Frogger

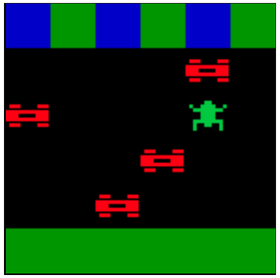
━━━ BQL

┆┆┆┆ BQL + Action Biasing

-·-·- BQL + Control Sharing

┆┆┆┆ BQL + Reward Shaping

- - - - BQL + **Advise**



6x6 Frogger

Advise Parameters

It is clear that the other algorithms perform inferior to **Advise** with suboptimal parameter values, but what about **Advise?**

$\hat{\mathcal{C}}$ Depends on:

The value of $\mathcal{C}$, the true feedback consistency

Our next experiment tested how well **Advise** performed with a suboptimal estimate of $\mathcal{C}$.

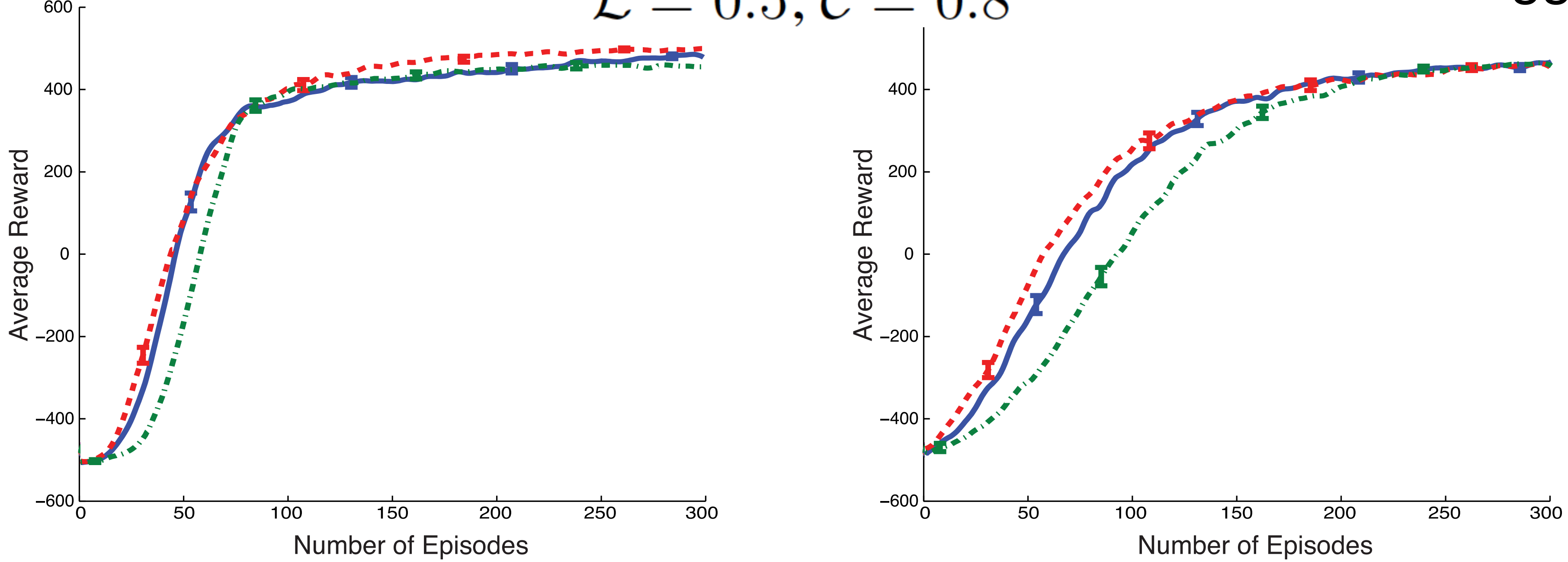
Using an Inaccurate Estimate of C

Moderate Likelihood and Consistency

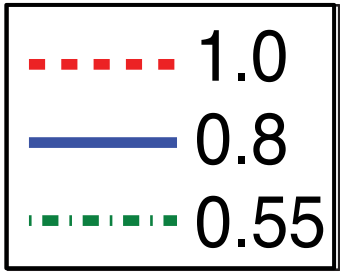
$$\mathcal{L} = 0.5; \mathcal{C} = 0.8$$

Pac-Man

Frogger



Estimated Feedback Consistency



Overestimate

True value

Underestimate

Discussion:

Summary of the experiments

- Control Sharing and Action Biasing depend on β which is decoupled from the information in each policy.

- Action Biasing depends on r, which is domain specific.
- **Advise** depends on $\mathcal{C}$, its single input parameter.

Conclusion

- This work introduced *Policy Shaping*.
- Advise is comparable to or outperforms state of the art techniques for integrating human feedback with RL.
- We avoid ad hoc parameter settings and are robust to infrequent and inconsistent feedback.
- There are many directions for future work: credit assignment; how to estimate C online; etc.