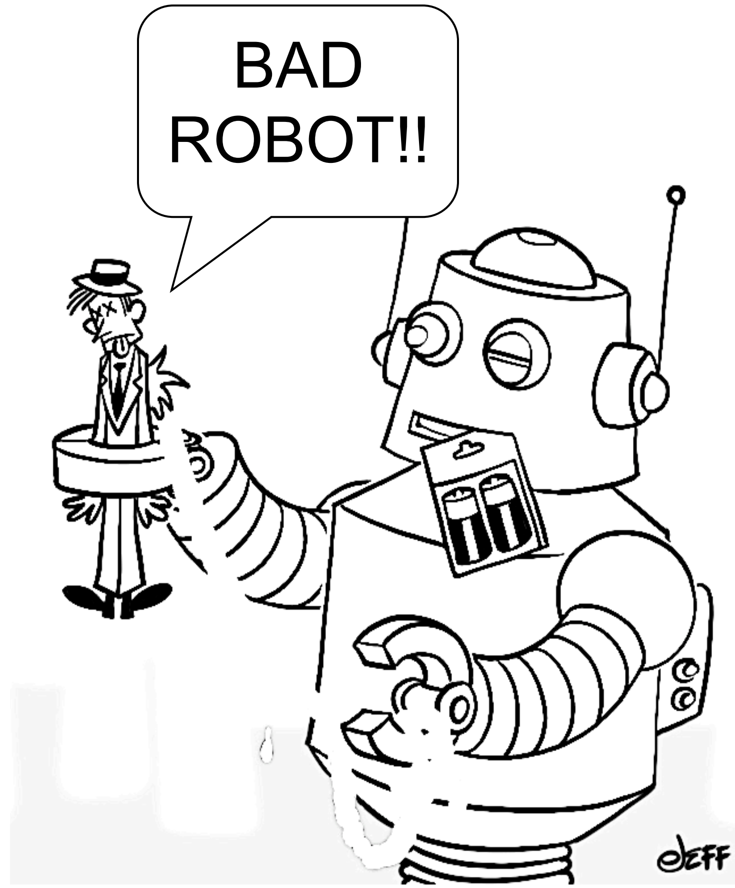
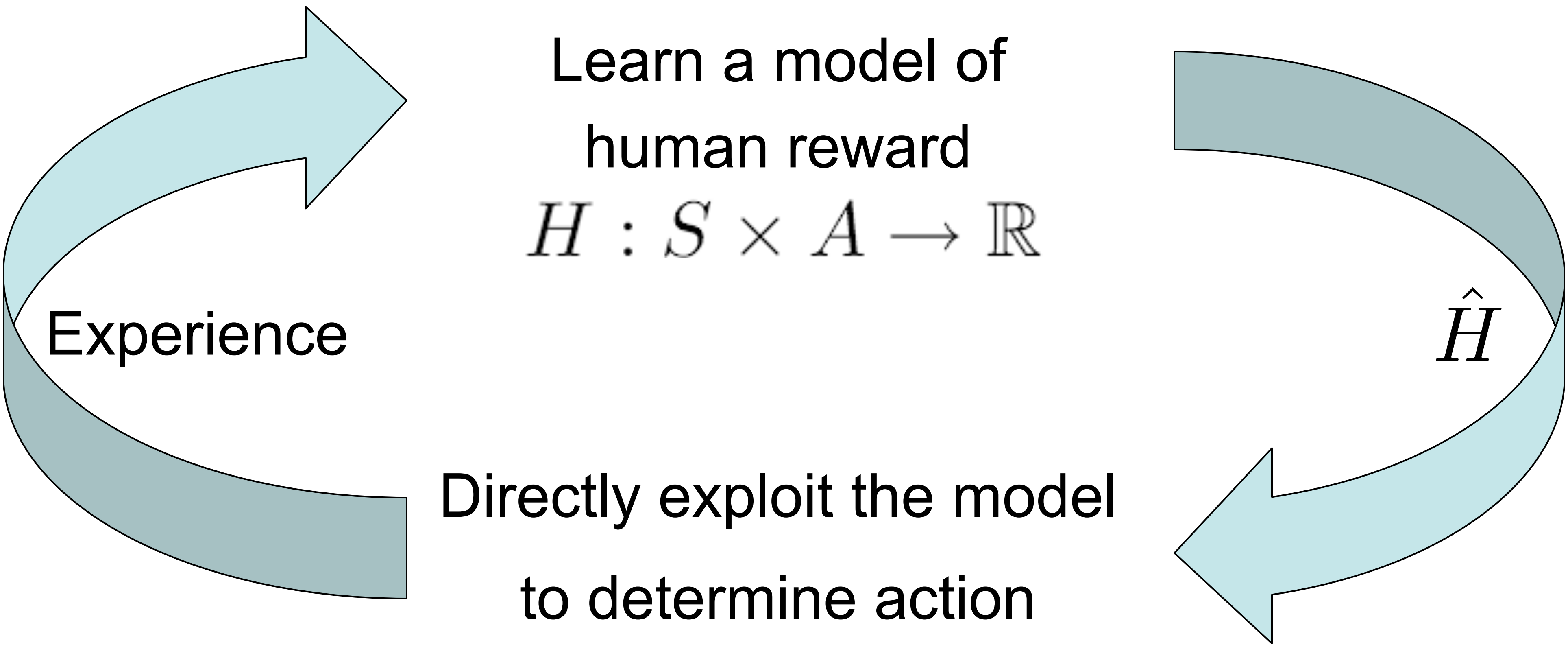
2 One solution to interactive shaping

Reward from a human trainer:

- Trainer has long-term impact in mind.
  - We can consider reward a full judgment of desirability of behavior.
- Trainer can reward with small delay.



Teaching an Agent Manually via Evaluative Reinforcement (**TAMER**)



If greedy: $action = argmax_a \hat{H}(s, a)$

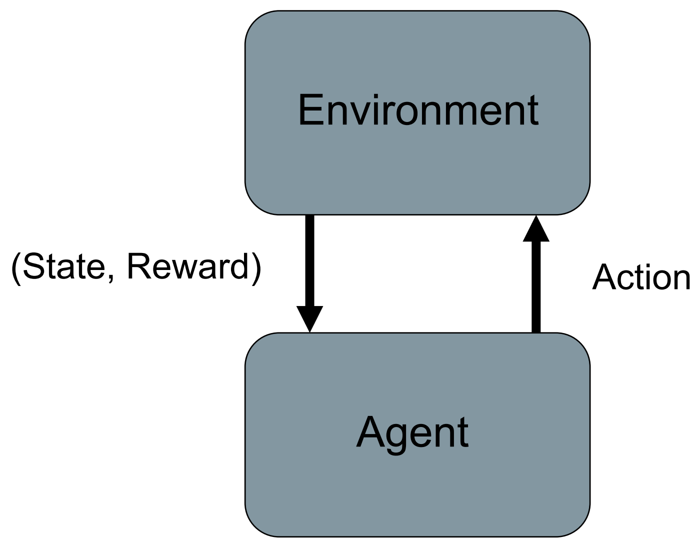
ICDL 2008 and K-CAP 2009

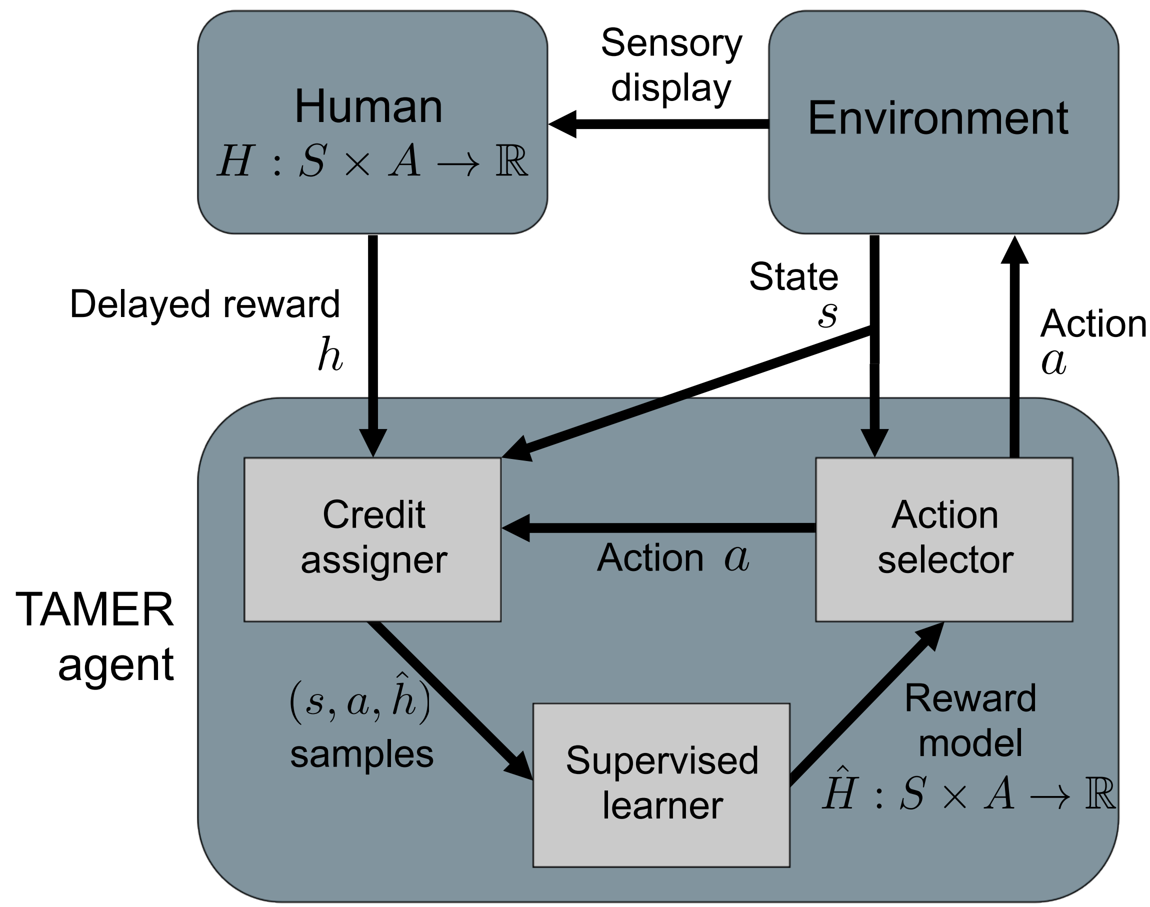
Teaching an Agent Manually via Evaluative Reinforcement (**TAMER**)

$$H : S \times A \rightarrow \mathbb{R}$$

I.e., TAMER **reduces** an apparent reinforcement learning problem **to a supervised learning problem** by setting $\gamma=0$.

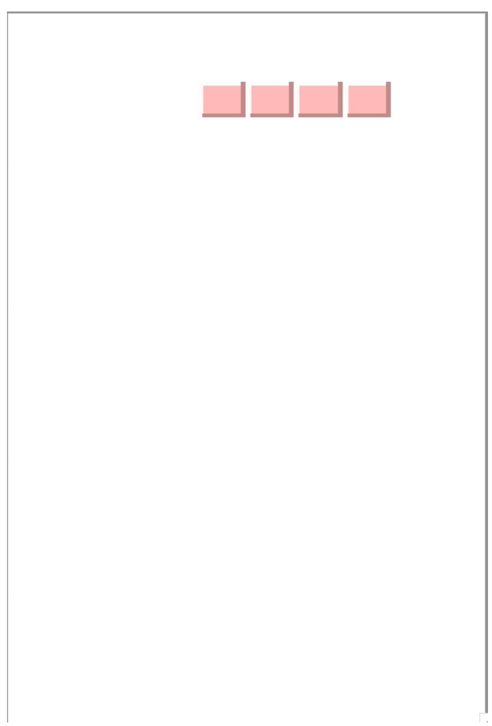
Teaching an Agent Manually via Evaluative Reinforcement (**TAMER**)



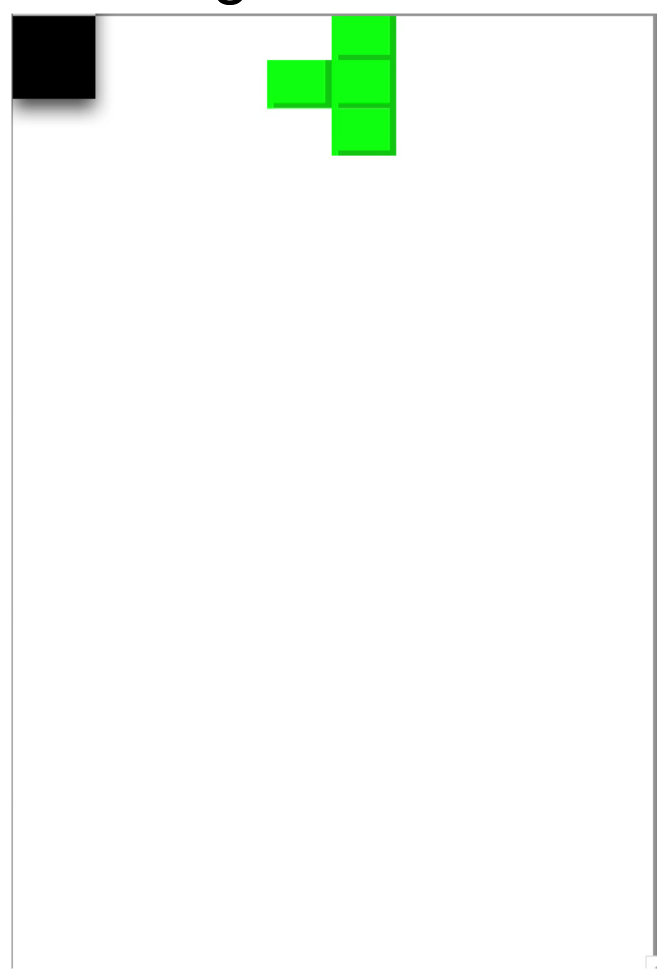


TAMER in action: Tetris

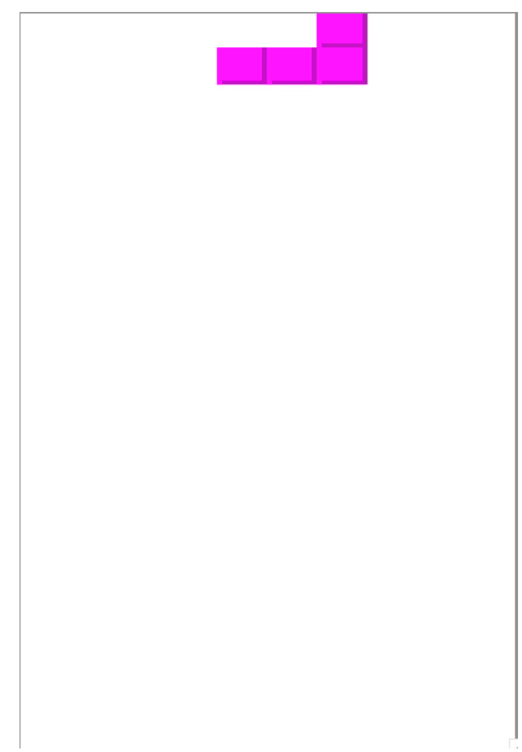
Before training:



Training:

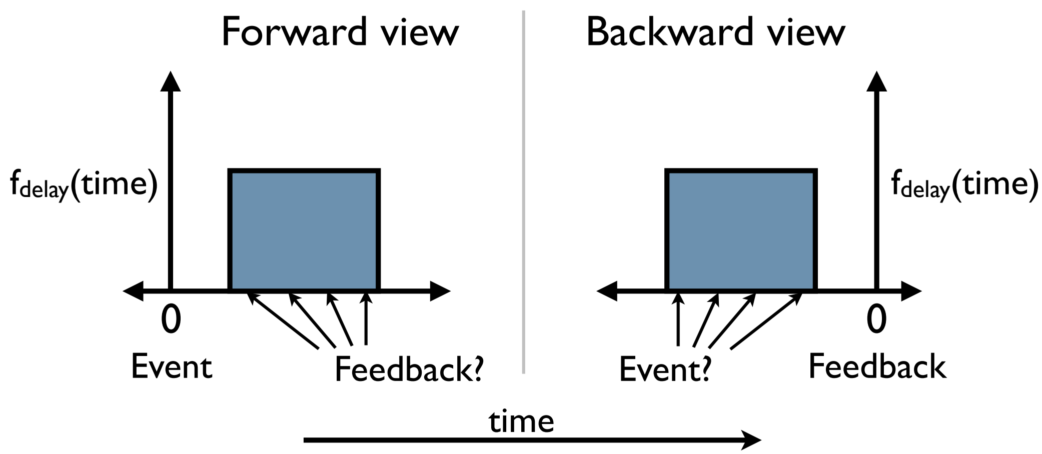


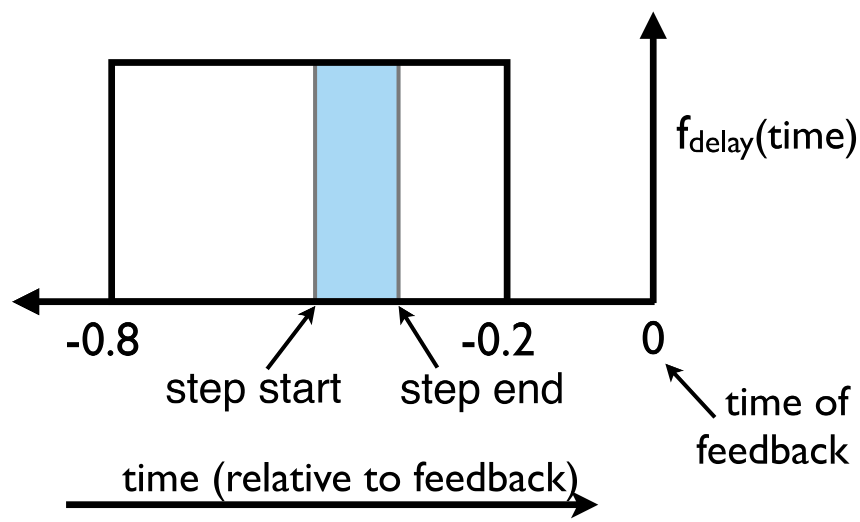
After training:



Environment courtesy of RL-Library and RL-Glue

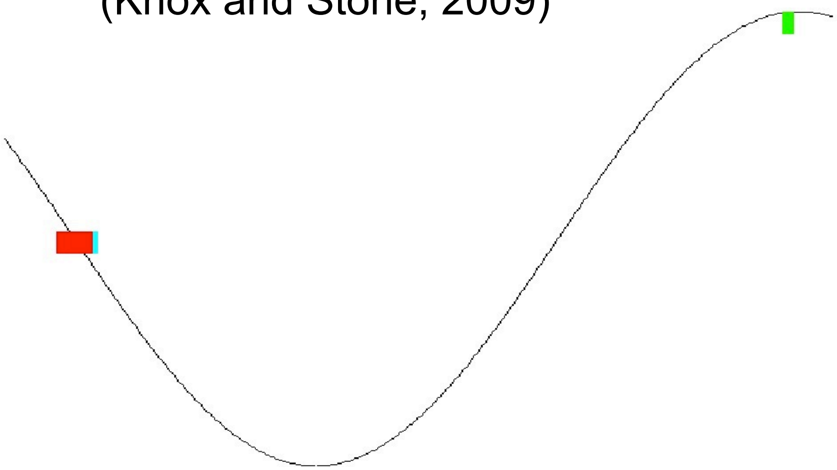
Handling reward delay



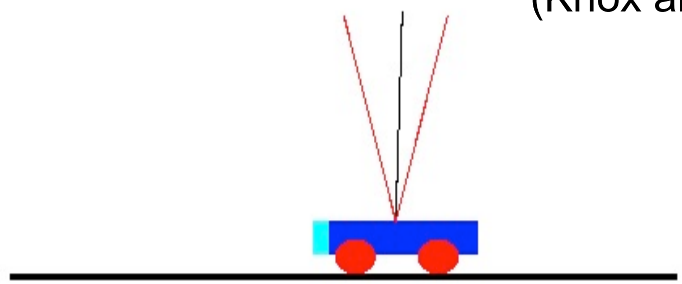


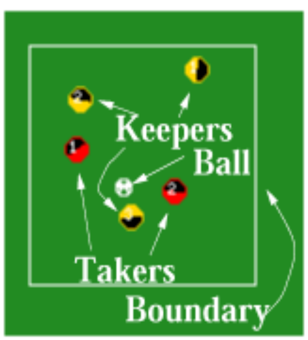
TAMER success on other domains

Mountain Car
(Knox and Stone, 2009)

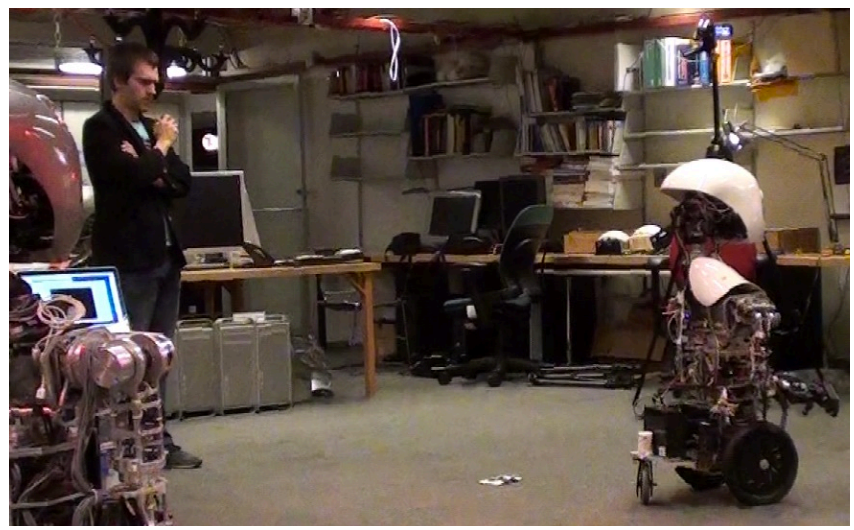


Balancing Cart Pole
(Knox and Stone, 2012)





3 vs 2 Keepaway
(Sridharan, 2011)



Interactive robot navigation
(Knox, Stone, and Breazeal, 2012)

Environments courtesy of RL-Library and RL-Glue (adapted)

Combination Techniques

1. $R'(s,a) = R(s,a) + (\beta * \hat{H}(s,a))$.
2. $\overrightarrow{f'} = \overrightarrow{f}.append(\hat{H}(s,a))$.
3. *Initially train* $Q(s,a)$ *to approximate* $(\beta * \hat{H}(s,a))$.
4. $Q'(s,a) = Q(s,a) + (\beta * \hat{H}(s,a))$.
5. $A' = A \cup argmax_a[\hat{H}(s,a)]$.
6. $Q'(s,a) = Q(s,a) + (\beta * \hat{H}(s,a))$ *only during action selection.*
7. $P(a = argmax_a[\hat{H}(s,a)]) = min(\beta, 1)$. *Otherwise use base RL agent's action selection mechanism.*
8. $R'(s_t,a) = R(s,a) + (\beta * (\phi(s_t) - \phi(s_{t-1})), where$ $\phi(s) = max_a H(s,a)$.

Combination Techniques

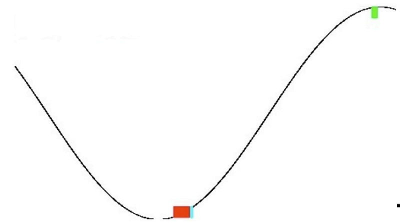
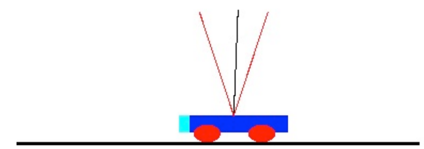
1. $R'(s,a) = R(s,a) + (\beta * \hat{H}(s,a))$.
2. $\overrightarrow{f'} = \overrightarrow{f}.append(\hat{H}(s,a))$.
3. *Initially train* $Q(s,a)$ *to approximate* $(\beta * \hat{H}(s,a))$.
4. $Q'(s,a) = Q(s,a) + (\beta * \hat{H}(s,a))$.
5. $A' = A \cup argmax_a[\hat{H}(s,a)]$.
6. $Q'(s,a) = Q(s,a) + (\beta * \hat{H}(s,a))$ ***only during action selection.***
7. $P(a = argmax_a[\hat{H}(s,a)]) = min(\beta, 1)$. *Otherwise use base RL agent's action selection mechanism.*
8. $R'(s_t,a) = R(s,a) + (\beta * (\phi(s_t) - \phi(s_{t-1}))$, *where* $\phi(s) = max_a H(s,a)$.

action biasing

Combination Techniques

1. $R'(s,a) = R(s,a) + (\beta * \hat{H}(s,a))$.
2. $\overrightarrow{f'} = \overrightarrow{f}.append(\hat{H}(s,a))$.
3. *Initially train* $Q(s,a)$ *to approximate* $(\beta * \hat{H}(s,a))$.
4. $Q'(s,a) = Q(s,a) + (\beta * \hat{H}(s,a))$.
5. $A' = A \cup argmax_a[\hat{H}(s,a)]$.
6. $Q'(s,a) = Q(s,a) + (\beta * \hat{H}(s,a))$ *only during action selection.*
7. $P(a = argmax_a[\hat{H}(s,a)]) = min(\beta, 1)$. *Otherwise use base RL agent's action selection mechanism.*
8. $R'(s_t,a) = R(s,a) + (\beta * (\phi(s_t) - \phi(s_{t-1}))$, *where* $\phi(s) = max_a H(s,a)$.

control sharing

Domains:  

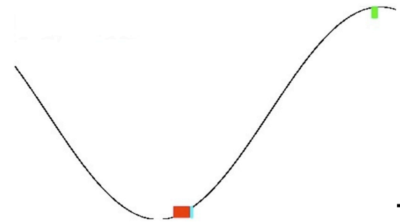
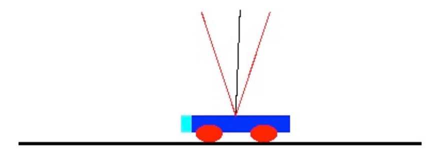
Defining success

Outperforming:

On the metrics:

	TAMER-only	RL-only
cumulative MDP reward	?	?
final performance	?	?

On each tested $\hat{H}$

Domains:  

Defining success

Sarsa(λ) here

Outperforming:

On the metrics:

	TAMER-only	RL-only
cumulative MDP reward	?	?
final performance	?	?

On each tested $\hat{H}$

Complete successes

action biasing

$Q'(s,a) = Q(s,a) + (\beta * \hat{H}(s,a))$ *only during action selection.*

and

control sharing

$P(a = argmax_a[\hat{H}(s,a)]) = min(\beta, 1)$. *Otherwise use base RL agent's action selection mechanism.*

Outperforming:

Manipulating action selection

On the metrics:	TAMER-only	RL-only
cumulative reward	✔	✔
final performance	✔	✔

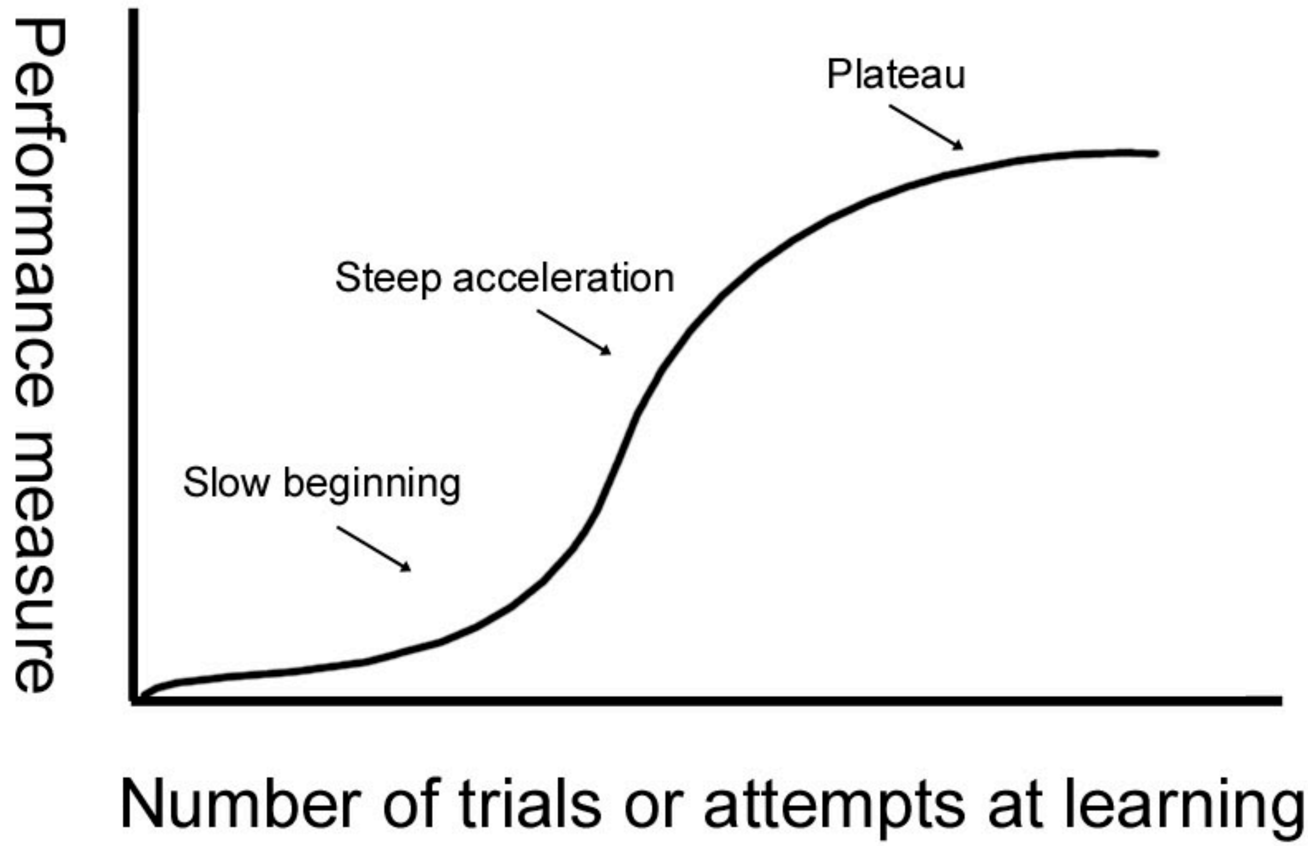
Outline

0 Background and TAMER+RL problem

1 Sequential TAMER+RL

2 Simultaneous TAMER+RL

2 Simultaneous TAMER+RL



Determining when and where human influences

action biasing

$Q'(s,a) = Q(s,a) + (\beta * \hat{H}(s,a))$ *only during action selection.*

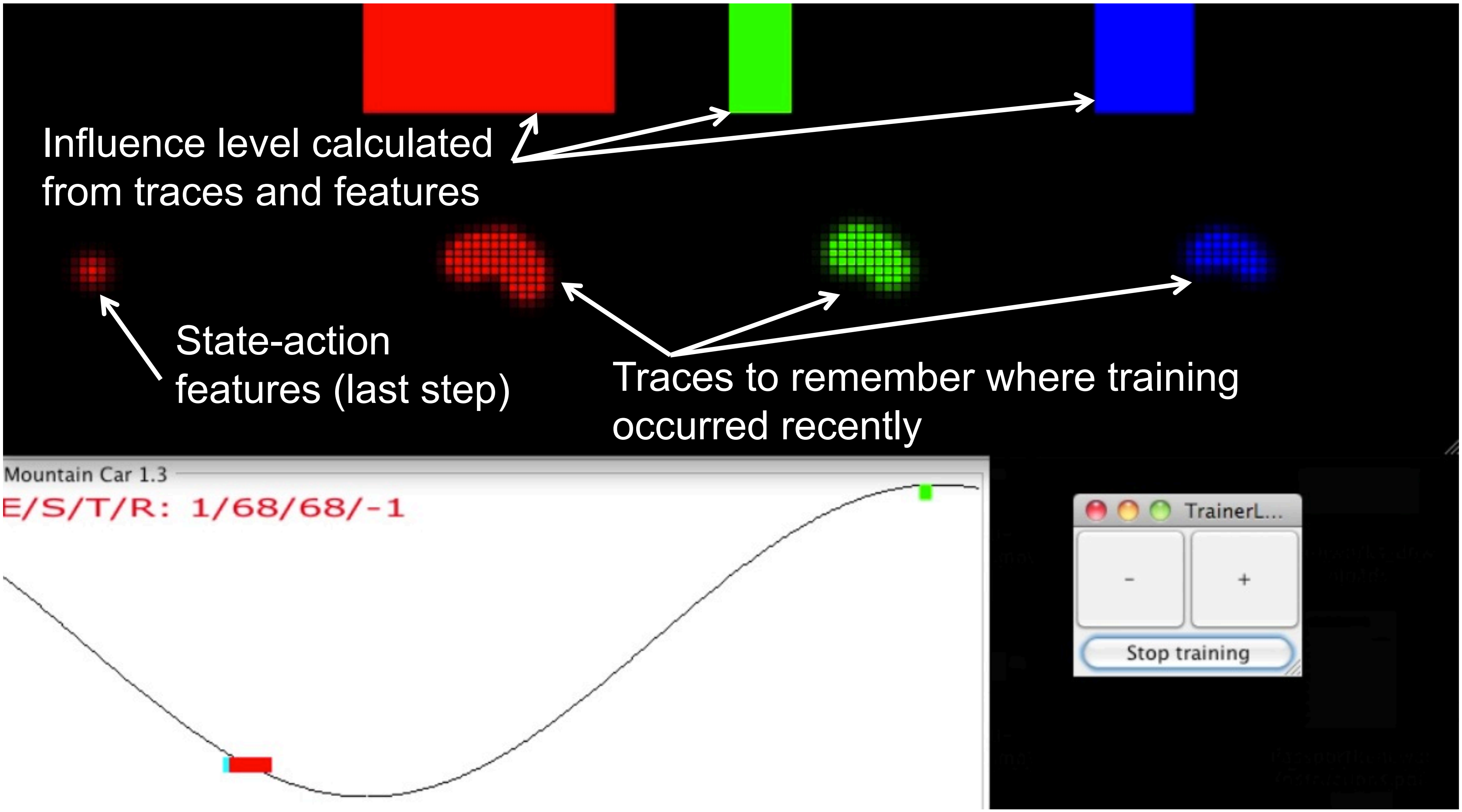
control sharing

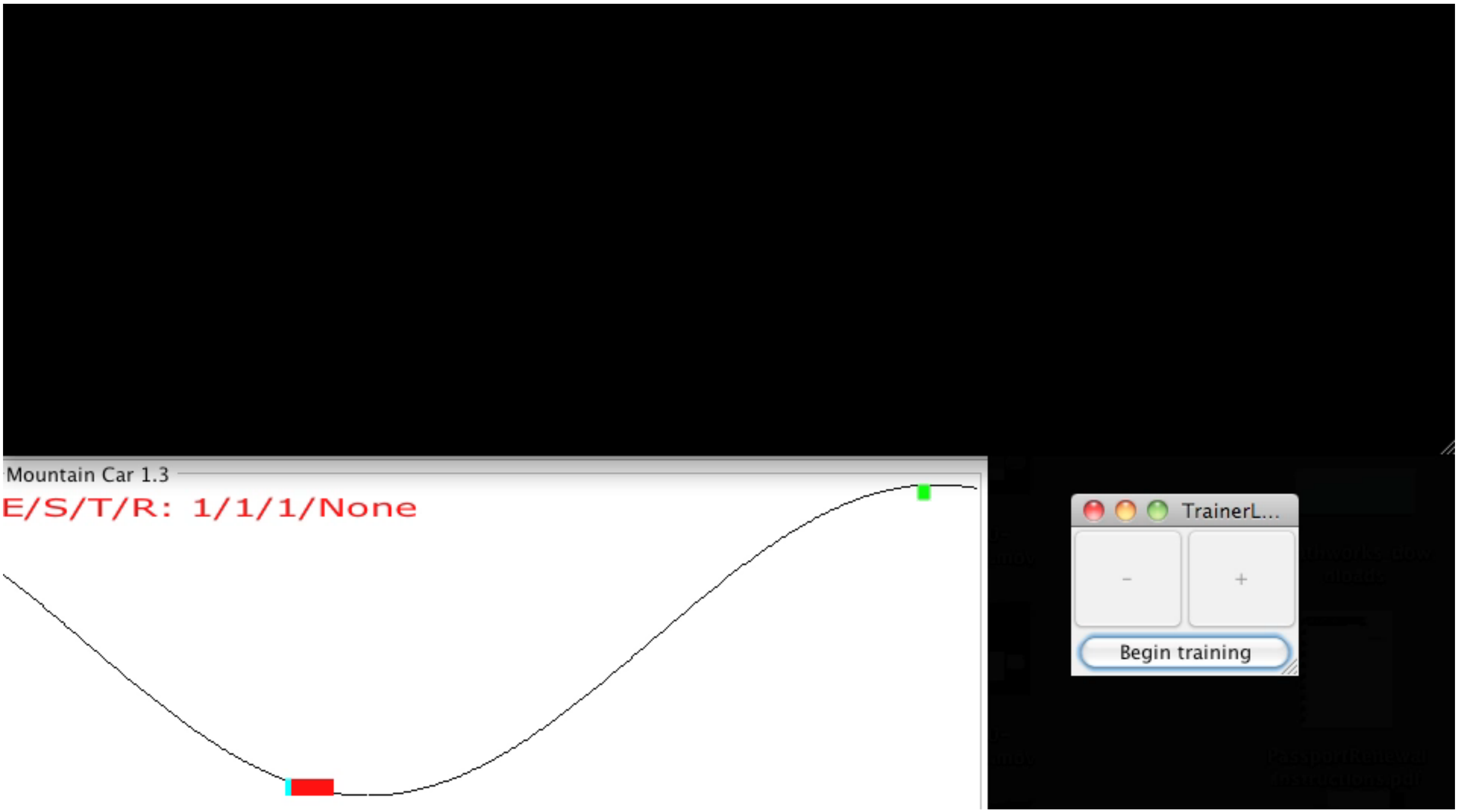
$P(a = argmax_a[\hat{H}(s,a)]) = min(\beta, 1)$. *Otherwise use base RL agent's action selection mechanism.*

Sequential – reduce influence of by annealing β as learning progresses

Simultaneous – influence of (as regulated by β) *should*

1. increase after training in nearby state-action space, and
2. decrease in the absence of training.


Influence level calculated
from traces and features
State-action
features (last step)
Traces to remember where training
occurred recently
Mountain Car 1.3
E/S/T/R: 1/68/68/-1
TrainerL...
-
+
Stop training


Mountain Car 1.3
E/S/T/R: 1/1/1/None
TrainerL...
-
+
Begin training

Determining when and where human influences

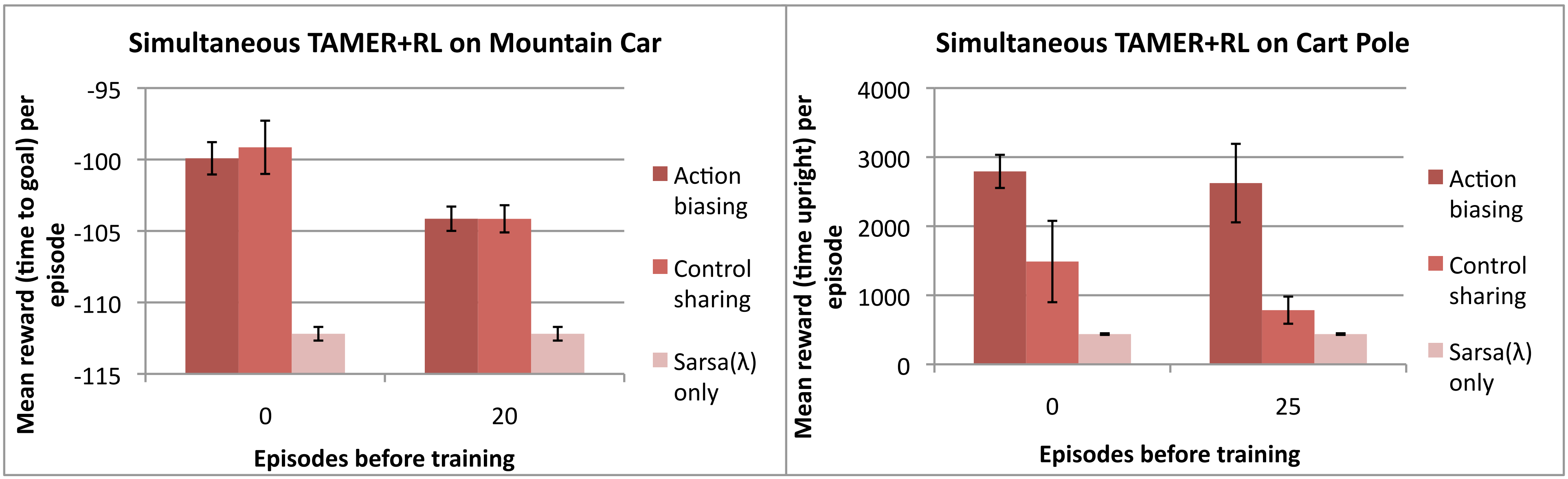
$\hat{H}$ Eligibility Module – qualitative characteristics

1. Scales up influence in areas of recent training
2. Slowly reduces influence in the absence of training

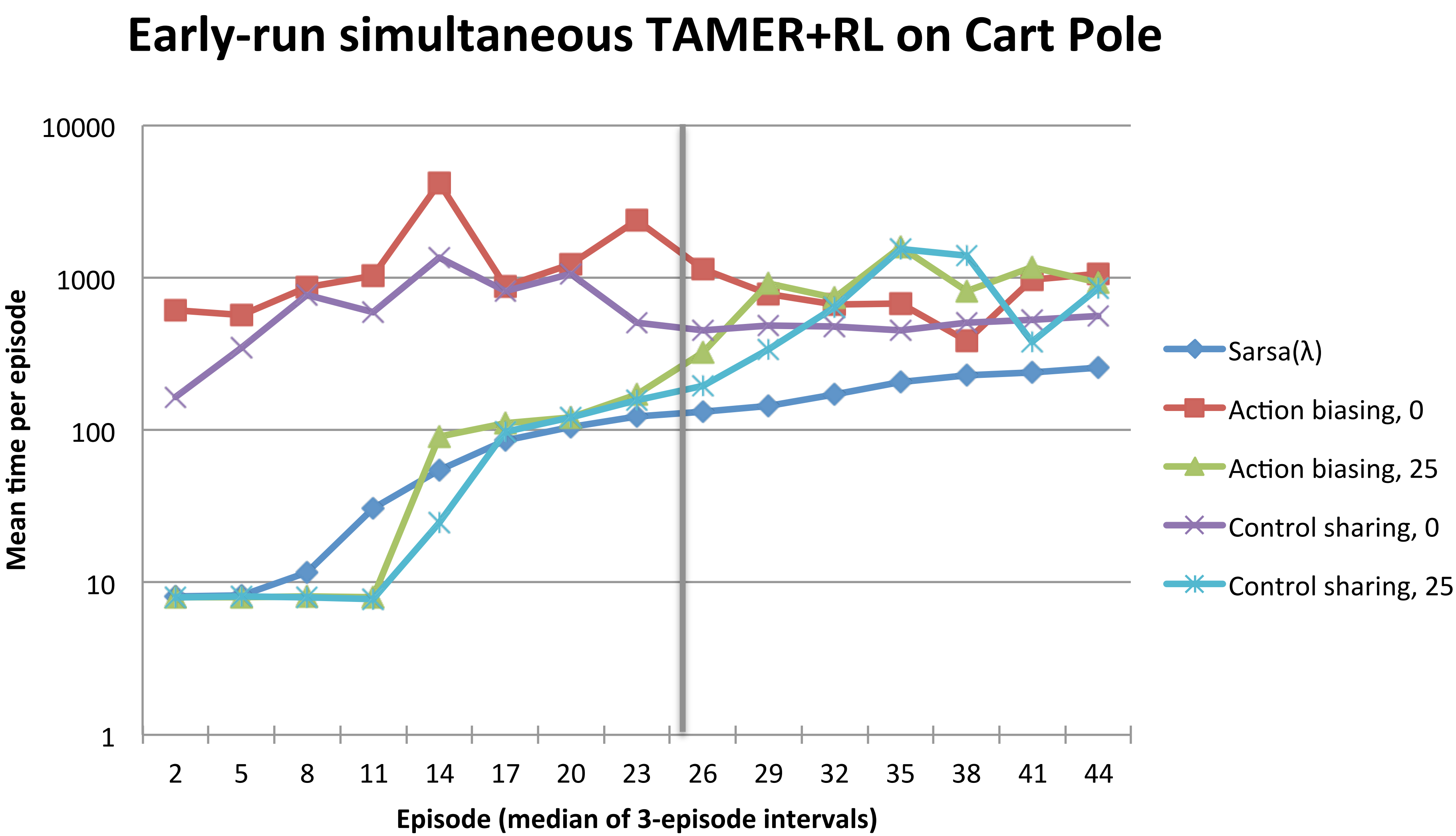
$$\beta := c\,\overrightarrow{e} \cdot (\overrightarrow{f_n} \,/\, \| \overrightarrow{f_n} \|_1)$$

Experiments

Mountain Car and Balancing Cart-Pole



Experiments



Related work on learning from MDP reward and human input

- alternating stages of autonomous action and human critique (Judah et. al, 2010)
- learning from demonstration (Smart and Kaelbling, 2000; Taylor et al., 2011)
- learning options from demonstration (Subramanian et al., 2011)
- feature selection from demonstration (Cobo et al. 2011, 2012)

TAMER+RL **Conclusions**

Human reward can be combined with MDP reward to improve upon learning from either alone.

Manipulating action selection – highest, most consistent gains and robust to changes in weights

Mixing human and MDP reward in a single value function – sometimes helps, brittle to weight values

Can learn simultaneously through an adaptation of eligibility traces