Reinforcement Learning with Multiple Experts: A Bayesian Model Combination Approach

0712214 陳彥儒 0716007 潘冠蓁
0716308 張千祐 0716312 葉佳翰

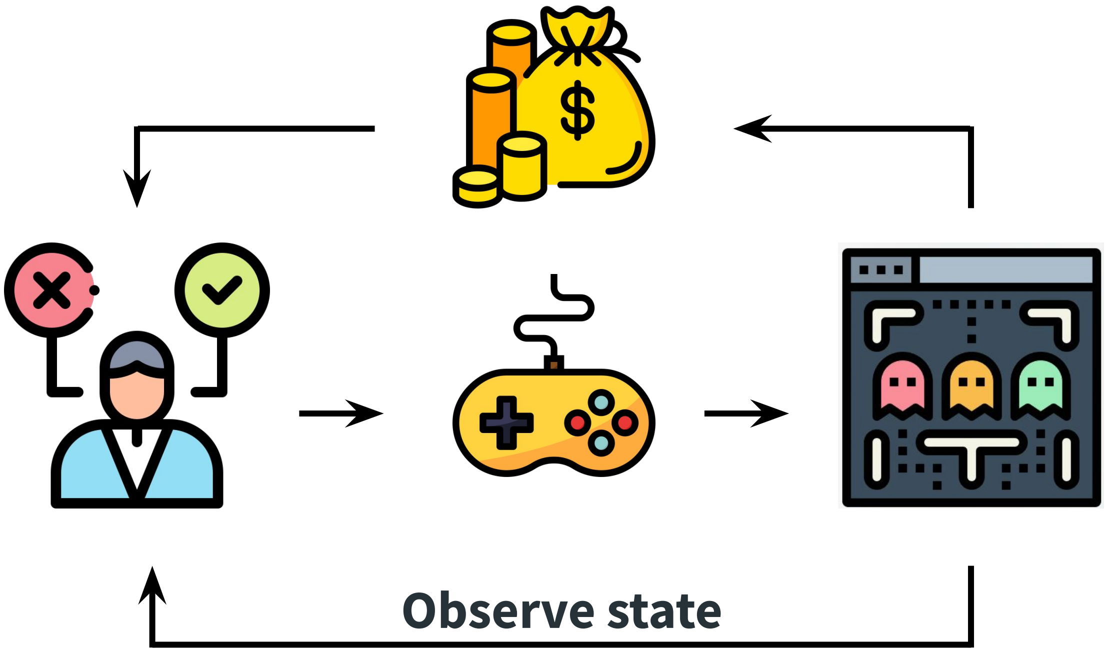
Replication

Outline

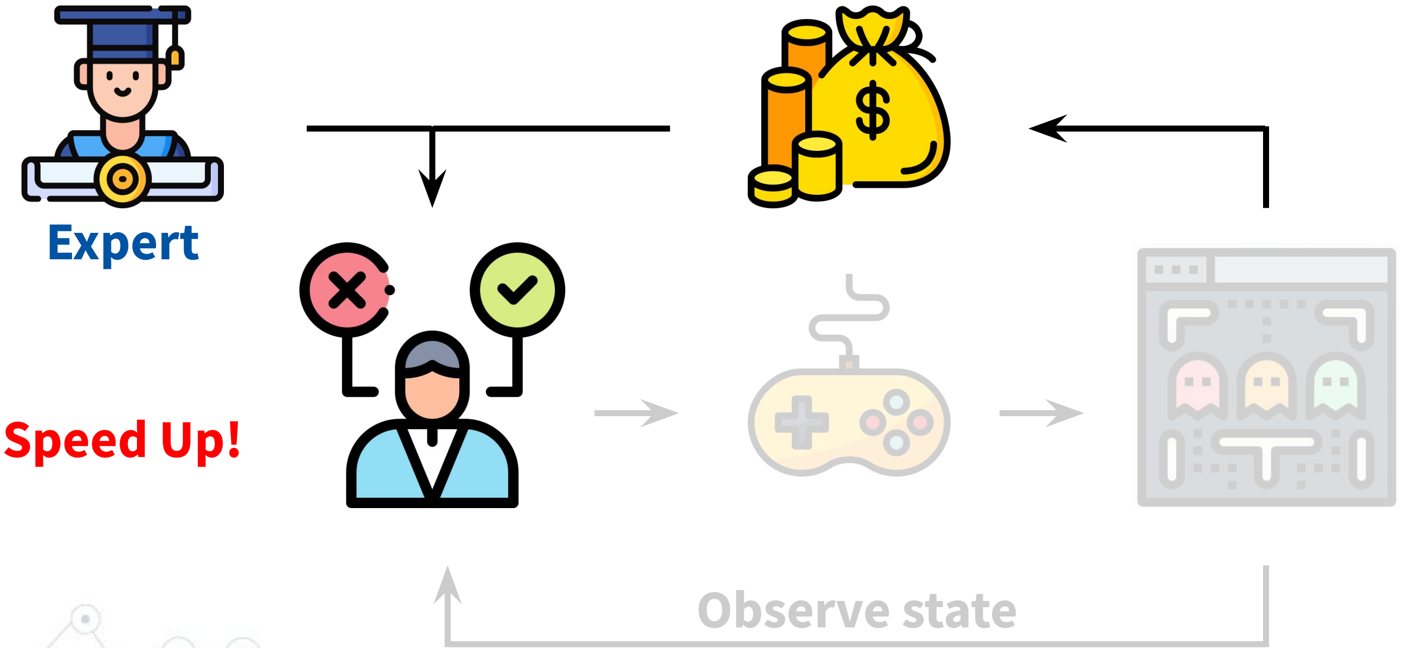
- Introduction
- Bayesian Reward Shaping
- Experimental Setting — Environment
- Experimental Setting — RL algorithm
- Experimental Setting — Experts
- Experimental Results
- Conclusion

Introduction

Introduction



Introduction



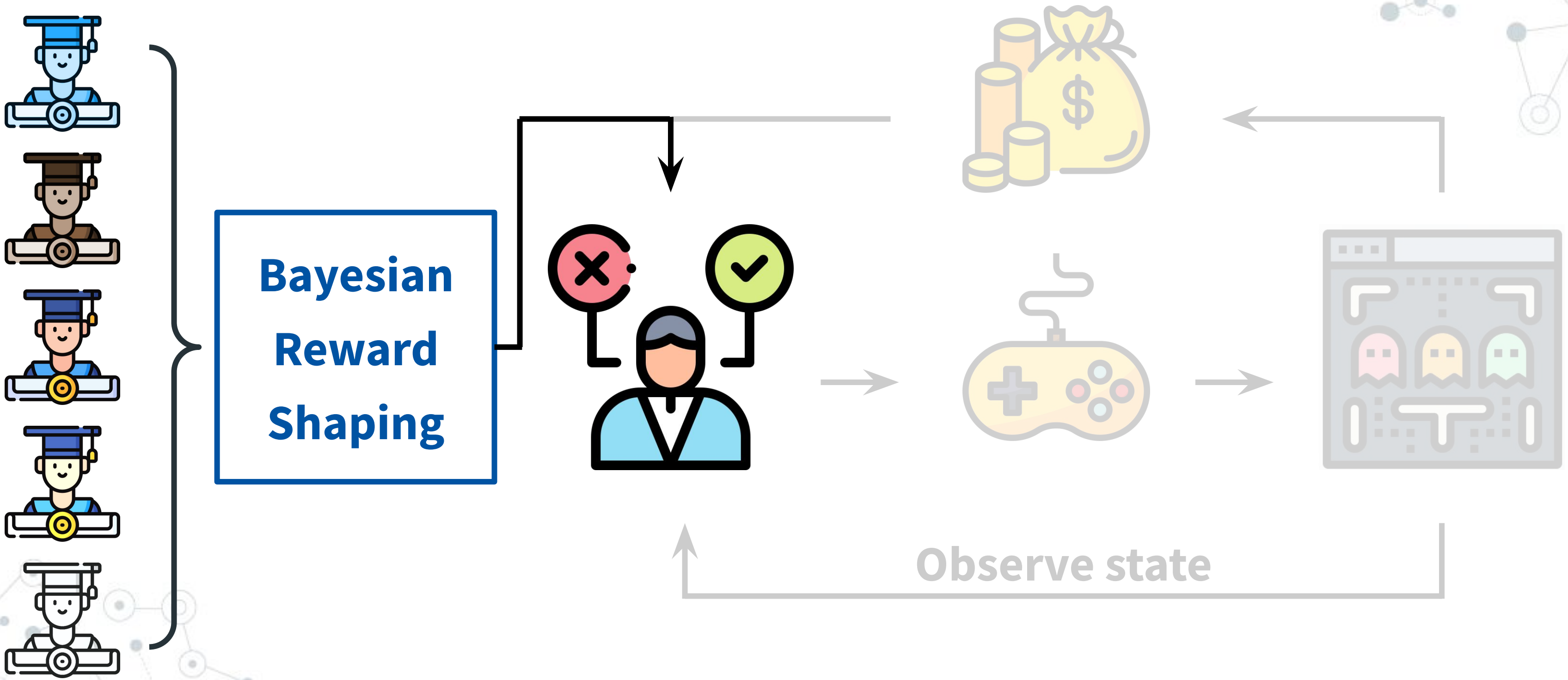
Introduction

But, if you have more than 1 experts…

Expert 1 **Expert 2** **Expert 3** **Expert 4** **Expert 5**

Which one should you trust?

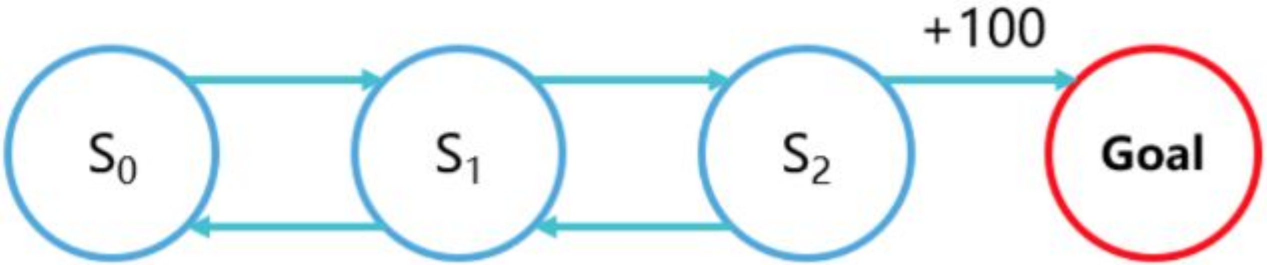
Introduction



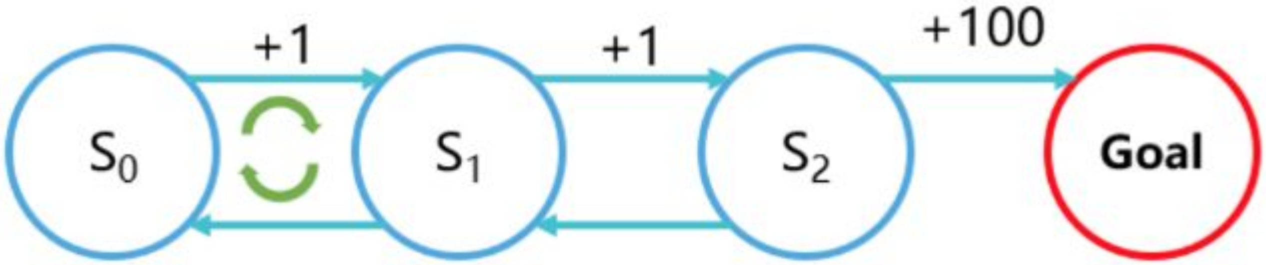
Bayesian Reward Shaping

Bayesian Reward Shaping

Original reward



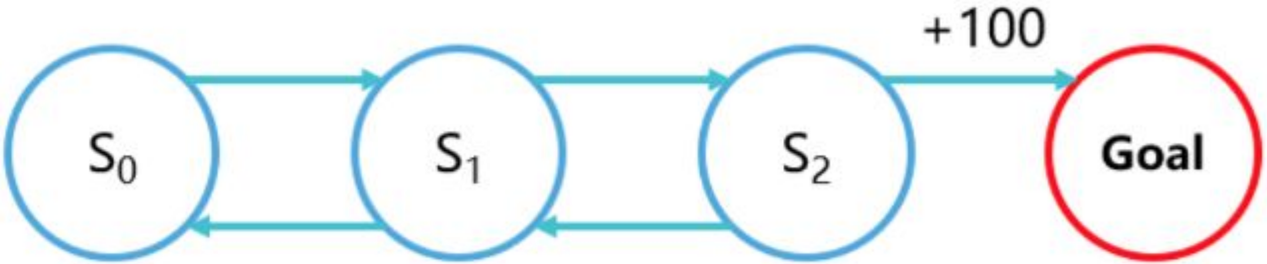
If expert's reward looks like…



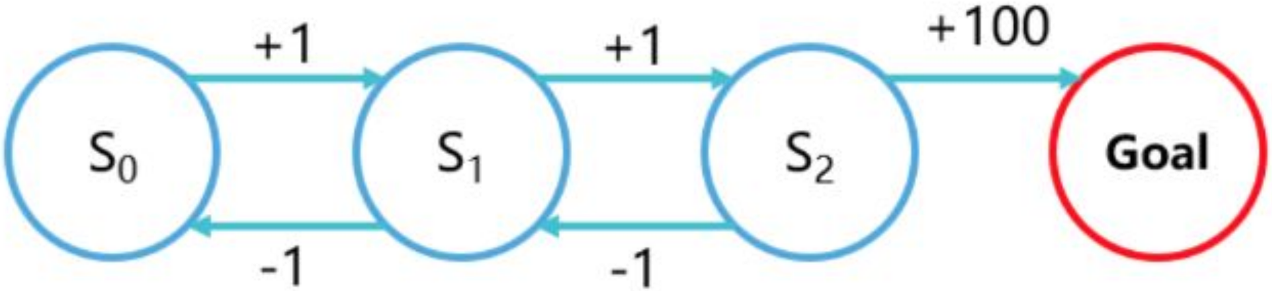
From: https://aijishu.com/a/1060000000143794

Bayesian Reward Shaping

Original reward

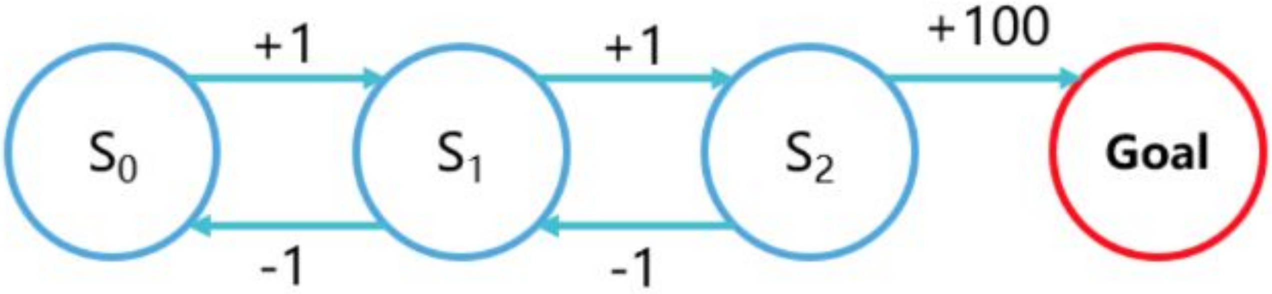


But if we change a little bit…



From: https://aijishu.com/a/1060000000143794

Bayesian Reward Shaping



$$\Phi(s_0) = 0 \qquad \Phi(s_1) = 1 \qquad \Phi(s_2) = 2$$

$$\boxed{F(s, a, s')} = \gamma\Phi(s') - \Phi(s)$$

This might be the proper definition of the additional reward from expert.

Bayesian Reward Shaping

$\Phi_1 \longrightarrow F_1(s, s') = \gamma\Phi_1(s') - \Phi_1(s)$

$\Phi_2 \longrightarrow F_2(s, s') = \gamma\Phi_2(s') - \Phi_2(s)$

$\Phi_3 \longrightarrow F_3(s, s') = \gamma\Phi_3(s') - \Phi_3(s)$

$\Phi_4 \longrightarrow F_4(s, s') = \gamma\Phi_4(s') - \Phi_4(s)$

$\Phi_5 \longrightarrow F_5(s, s') = \gamma\Phi_5(s') - \Phi_5(s)$

Weighting $\longrightarrow$ Add to Reward $\downarrow$ RL !

Bayesian Reward Shaping

Algorithm 2 RL with Bayesian Reward Shaping

initialize $\boldsymbol{\alpha} \in \mathbb{R}_+^N$

for $episode = 0, 1 \ldots M$ **do** ▷ Main loop

$\quad \hat{\Phi} \leftarrow \frac{\sum_{i=1}^N \Phi_i \alpha_i}{\sum_{i=1}^N \alpha_i}$ ▷ Pool experts and compute shaped reward

$\quad F(s, a, s') \leftarrow \gamma \hat{\Phi}(s') - \hat{\Phi}(s)$

$\quad (R_t, s_t)_{t=1 \ldots T} \leftarrow \texttt{TrainRL}(F)$ ▷ Perform one episode of training

$\quad$ **for all** (R_t, s_t) **do** ▷ Posterior update

$\quad\quad$ update $\hat{\sigma}^2$ and compute $\mathbf{e}$

$\quad\quad \boldsymbol{\alpha} \leftarrow \texttt{PosteriorUpdate}(\boldsymbol{\alpha}, \mathbf{e})$

Bayesian Reward Shaping

Algorithm 2 RL with Bayesian Reward Shaping

1: initialize $\boldsymbol{\alpha} \in \mathbb{R}_+^N$
2: **for** *episode* $= 0, 1 \dots M$ **do** $\triangleright$ Main loop
3: $\quad \hat{\Phi} \leftarrow \frac{\sum_{i=1}^N \Phi_i \alpha_i}{\sum_{i=1}^N \alpha_i}$ $\triangleright$ Pool experts and compute shaped reward
4: $\quad F(s, a, s') \leftarrow \gamma \hat{\Phi}(s') - \hat{\Phi}(s)$
5: $\quad (R_t, s_t)_{t=1 \dots T} \leftarrow \mathtt{TrainRL}(F)$ $\triangleright$ Perform one episode of training
6: $\quad$ **for all** (R_t, s_t) **do** $\triangleright$ Posterior update
7: $\quad\quad$ update $\hat{\sigma}^2$ and compute $\mathbf{e}$
8: $\quad\quad \boldsymbol{\alpha} \leftarrow \mathtt{PosteriorUpdate}(\boldsymbol{\alpha}, \mathbf{e})$

How can we update the weights for each expert?

Bayesian Reward Shaping

Answer: lots of math…

Dirichlet Distribution: → 估計 w 的分佈

Target: Model how "properties" vary.

parameter: $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_k)$

$$P(W) \sim D(\alpha) = \underline{\frac{\Gamma(\sum_k \alpha_i)}{\prod_k \Gamma(\alpha_i)} \prod_k W_i^{\alpha_{k-1}}}, \quad \sum_k w_i = 1, \quad W_i > 0$$

$P(W)$ → Weight of experts

Seems like Property of

Hence $E[W_k] = \frac{\alpha_k}{\sum_k \alpha_i}$ ← Mi mentioned in paper Equation (14)

$E[W_k^2] = E[W_k] \frac{1+\alpha_k}{1+\sum_k \alpha_i}$

$\Rightarrow E[W_1^2] = \frac{\alpha_1(\alpha_1+1)}{\alpha_0(\alpha_0+1)}$ ← The "second moment" S_1 mentioned in paper Equation (15)

$\pi_{t+1}(W)$

$$
\begin{aligned}
P(w|D,d) &= \frac{P(w,D,d)}{P(D,d)} && \text{D, d are indep (My Guess)} \\
&= \frac{P(w,D,d)}{P(D)P(d)} && \text{Def of conditional Prob} \\
&= \frac{P(w,D|d)}{P(D)} && \text{Bayes Thm: } P(\theta|x) \propto P(\theta)P(x|\theta) \\
&\propto \frac{P(w,D)\,P(d|w,D)}{P(D)} && \text{Def of conditional Prob} \\
&= P(w|D)\,P(d|w,D) && \text{D, d are indep (My Guess)} \\
&= P(w|D)\,P(d|w) \\
&= \pi_t(w)\,P(d|w) \\
&\propto \sum_{i=1}^{N} P(d|i)\,P(i|w)\,\pi_t(w) && \text{normalization since } \textstyle\sum \pi_{t+1} = 1 \\
&= \frac{1}{C_{t+1}} \sum_{i=1}^{N} P(d|i)\,P(i|w)\,\pi_t(w)
\end{aligned}
$$

$$
\begin{aligned}
C_{t+1} &= \int_{S^{N-1}} \sum_{i=1}^{N} P(d|i)\,P(i|w)\,\pi_t(w)\,dw \\
&= \sum_{i=1}^{N} \underbrace{e_i}_{P(d|i)} \mathop{E}_{\pi_t}[w_i]
\end{aligned}
$$

$$
\begin{aligned}
&\mathop{E}_{q}[q_{sa}|D] \\
&= \int_{\mathbb{R}} q\,P(q|D)\,dq && \text{all possible Q-value} \\
&= \int_{\mathbb{R}} q \int_{S^{N-1}} P(q,w|D)\,dw\,dq && \text{all possible combination of weight} \\
&= \int_{\mathbb{R}} q \int_{S^{N-1}} \frac{P(q,w,D)}{P(D)} \cdot \frac{P(w,D)}{P(w,D)}\,dw\,dq \\
&= \int_{\mathbb{R}} q \int_{S^{N-1}} \underbrace{P(q|w,D)}_{\text{indep of } w} \underbrace{P(w|D)}_{= \pi_t(w)}\,dw\,dq
\end{aligned}
$$

$\pi_t(W)$ → The Probability that weight w emerge

$P(q|D) = \sum_{i=1}^{N} w_i P(q|i)$ → i-th expert's Q-value distribution

summing all experts' Q-value, given weight

$$
\begin{aligned}
&= \int_{\mathbb{R}} q \int_{S^{N-1}} \Big(\sum_{i=1}^{N} \underbrace{P(q|i)}_{\text{indep of } w \Rightarrow \text{ move to outside}} w_i\,\pi_t(w) \Big)\,dw\,dq && \Sigma, \textstyle\int \text{ changing} \\
&= \sum_{i=1}^{N} \int_{\mathbb{R}} q\,P(q|i) \int_{S^{N-1}} w_i\,\pi_t(w)\,dw\,dq && \text{def of } E \\
&= \sum_{i=1}^{N} \int_{\mathbb{R}} q\,P(q|i) \mathop{E}_{\pi_t}[w_i]\,dq \\
&= \sum_{i=1}^{N} \mathop{E}_{\pi_t}[w_i] \int_{\mathbb{R}} q\,P(q|i)\,dq && \text{def of } E \\
&= \sum_{i=1}^{N} \mathop{E}_{\pi_t}[w_i]\,E[q|i]
\end{aligned}
$$

Conclusion: We can compute $q(s,a)$

By computing $\mathop{E}_{\pi_t}[w_i]$ & $E[q|i]$ "separately"

If you are interested:

https://docs.google.com/document/d/1juUxpze40b18kQzbZft8ML3voZ-Fa3aXOyj0Cwo7VFk/edit?usp=sharing

Bayesian Reward Shaping

Don't worry. Let's take it easier...

Algorithm 1 PosteriorUpdate($\boldsymbol{\alpha}_t$, e)

1: **for** $i = 1, 2 \ldots N-1$ **do** ▷ Compute posterior moments

2: $\boxed{m_i \leftarrow \frac{\alpha_{t,i}(e_i + \mathrm{e} \cdot \boldsymbol{\alpha}_t)}{(\mathrm{e} \cdot \boldsymbol{\alpha}_t)(\alpha_{t,0}+1)}}$

3: $s_1 \leftarrow \frac{\alpha_{t,1}(\alpha_{t,1}+1)(2e_1 + \mathrm{e} \cdot \boldsymbol{\alpha}_t)}{(\mathrm{e} \cdot \boldsymbol{\alpha}_t)(\alpha_{t,0}+1)(\alpha_{t,0}+2)}$

4: $\alpha_{t+1,0} \leftarrow \frac{m_1 - s_1}{s_1 - m_1^2}$ ▷ Compute $\boldsymbol{\alpha}_{t+1}$

5: **for** $i = 1, 2 \ldots N-1$ **do**

6: $\alpha_{t+1,i} \leftarrow m_i \alpha_{t+1,0}$

7: $\alpha_{t+1,N} \leftarrow \alpha_{t+1,0} - \sum_{i=1}^{N-1} \alpha_{t+1,i}$

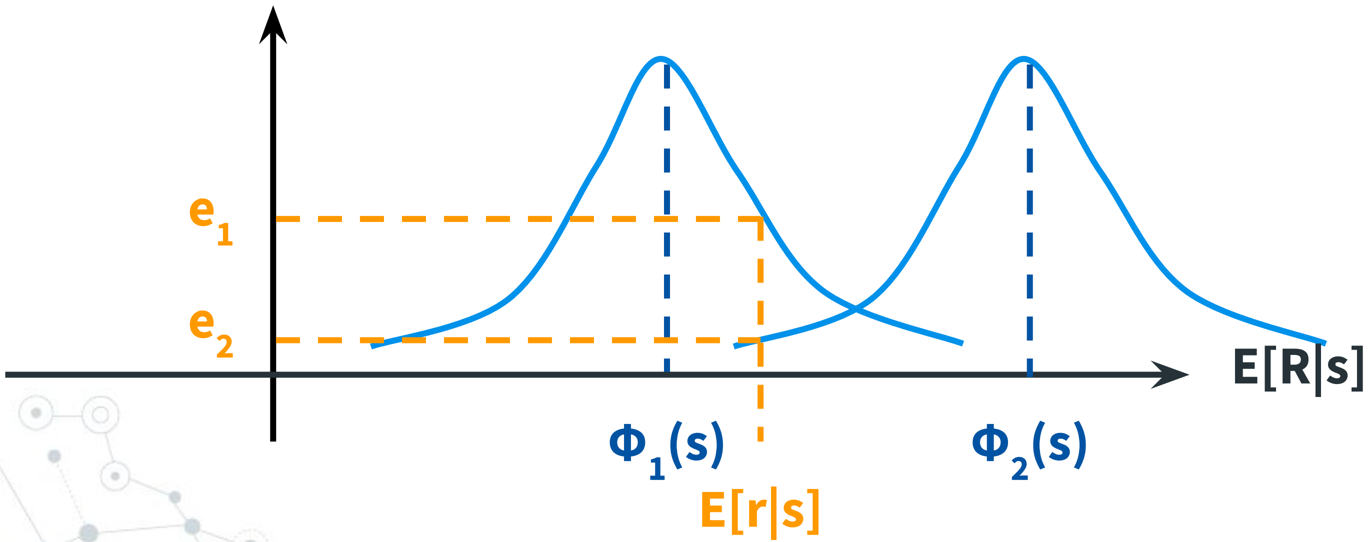
8: **return** $\boldsymbol{\alpha}_{t+1}$

"e" decide how to update the weights.

Bayesian Reward Shaping

What's "e" = "(e_1, e_2, …, e_n)"?

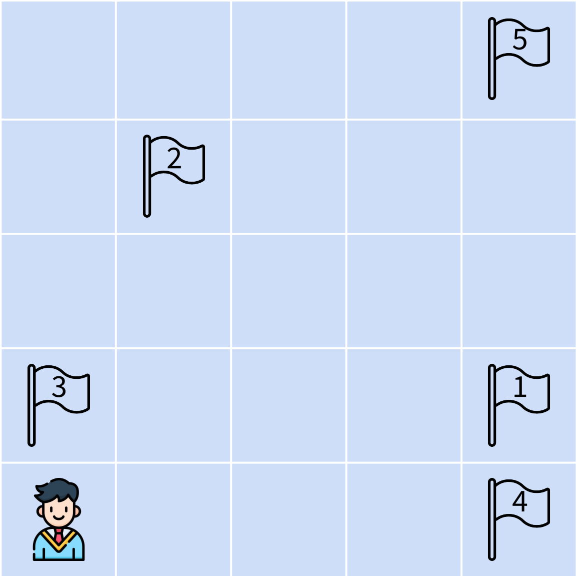
e_i = P(E[r|s] occurs at expert i)



Environment

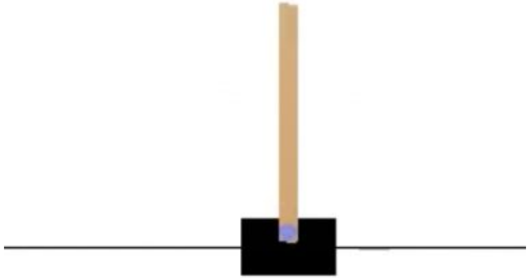
Gridworld

- Rules
  - Every move : -1 point
  - Invalid move : -1 additional point
- How to end this game
  - Until 200 steps
  - Collect all flags in order



Cartpole

- Goal
  - Keep the cartpole balanced
- When the game will end
  - Until 500 steps
  - | The angle between cart and pole | > 12 deg
  - | The position of the cart | > 2.4 deg



RL Algorithm

Tabular Q-learning

Initialize a Q-table

Choose an action a’ and get the next state s’

Observe the reward of all actions in the next state

Use the max Q-value to update the Q-table

Go to the next state s’

SARSA

Initialize a Q-table

Get the next state s’ by the action a

Choose the next action a’

Use Q(s’, a’) to update the Q-table

Go to the next state s’

Deep Q-Learning(DQN)

Choose action by Q network

Store the transition to replay memory

Use the data in replay memory to update Q network

Copy the parameter to target network every n steps

Experts

Experts

For Gridworld:

1. $\Phi_{good}(s)$ = optimal value function
2. $\Phi_{bad}(s)$ = -optimal value function
3. $\Phi_{random}(s) = U(-20,20)$
4. $\Phi_{zero}(s) = 0$
5. $\Phi_{heuristic}(s) = -22 * (5 - c - 0.5) / 5$

Optimal value function:

V(x, y, c) = -distance((x, y), next flag)

- distance(next flag ,final)

Experts

For CartPole:

1. $\Phi_{good}(s)$ = Q Network trained by us
2. $\Phi_{bad}(s)$ = -Q Network trained by us
3. $\Phi_{random}(s) = U(-20,20)$
4. $\Phi_{zero}(s) = 0$
5. $\Phi_{guess}(s) = 20 * (1 - |\theta| / 0.2618)$

$\Phi_{good}(s) = \text{avg } Q(s,a)$

Experimental Results

Experimental Results

Hyperparameters — Gridworld — DQN / Q-learning / SARSA

Learning rate: 0.001 / 0.4 / 0.36

Gamma: 0.99 / 1 / 1

Epsilon: max(0.98^t, 0.01) / 0.98^t / 0.98^t

Episode: 500 / 1000 / 2000

Independent set: 20 / 20 / 20

Experimental Results

Hyperparameters — CartPole — DQN / Q-learning / SARSA

Learning rate: 0.0005 / max(0.5*$lrdc^t$, 0.01) / max(0.5*$lrdc^t$, 0.01)

Gamma: 0.99 / 0.95 / 0.95

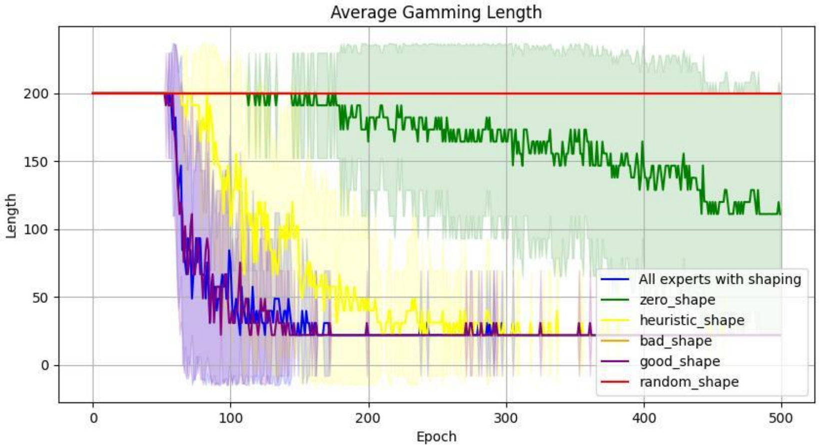
Epsilon: max(0.98^t, 0.01) / max(0.98^t, 0.01) / max(0.98^t, 0.01)

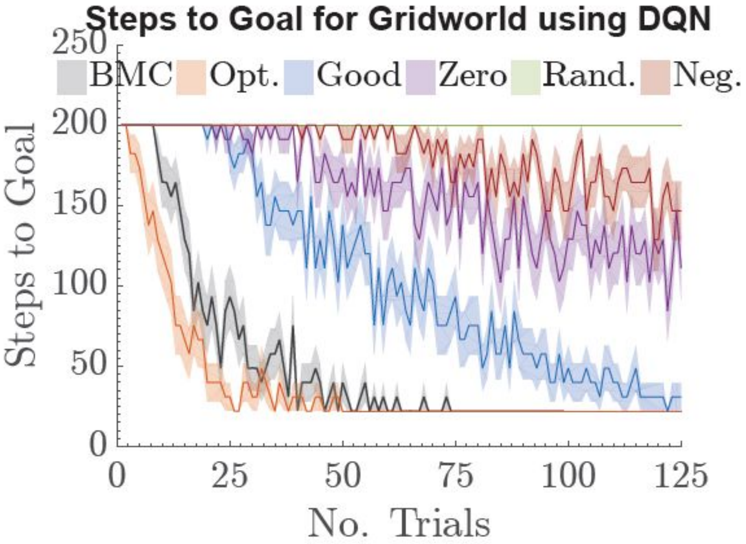
Episode: 2000 / 500 / 2000

Independent set: 20 / 20 / 20

Experimental Results

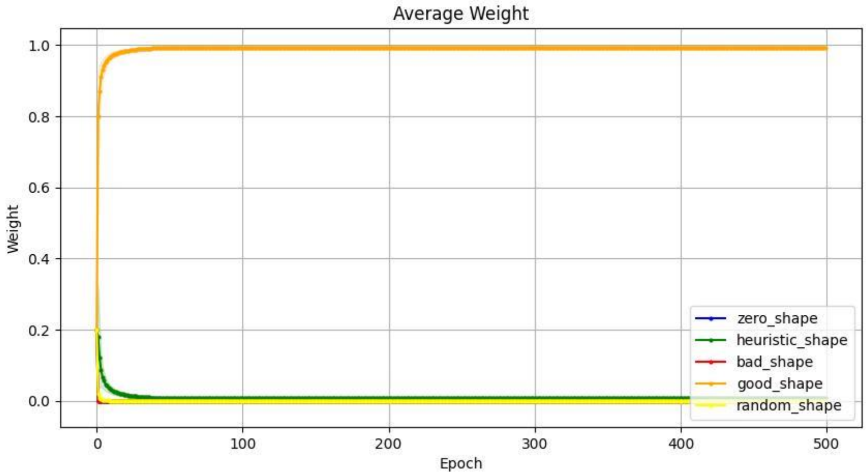
DQN — Gridworld

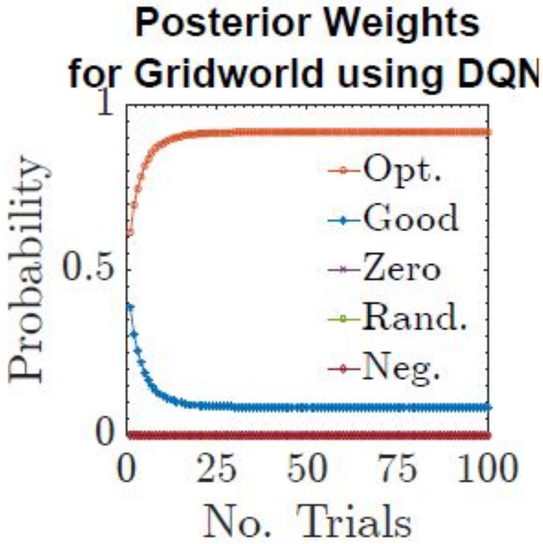




Experimental Results

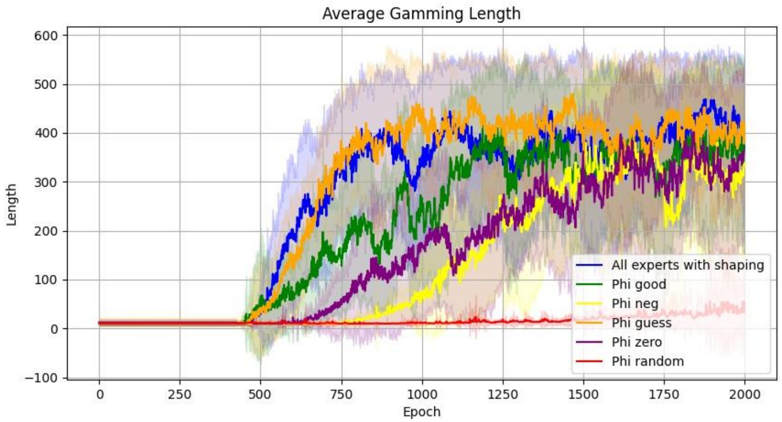
DQN — Gridworld

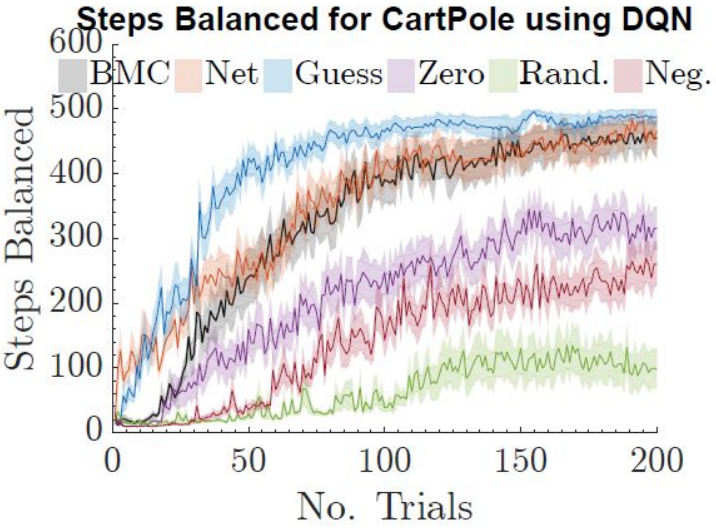




Experimental Results

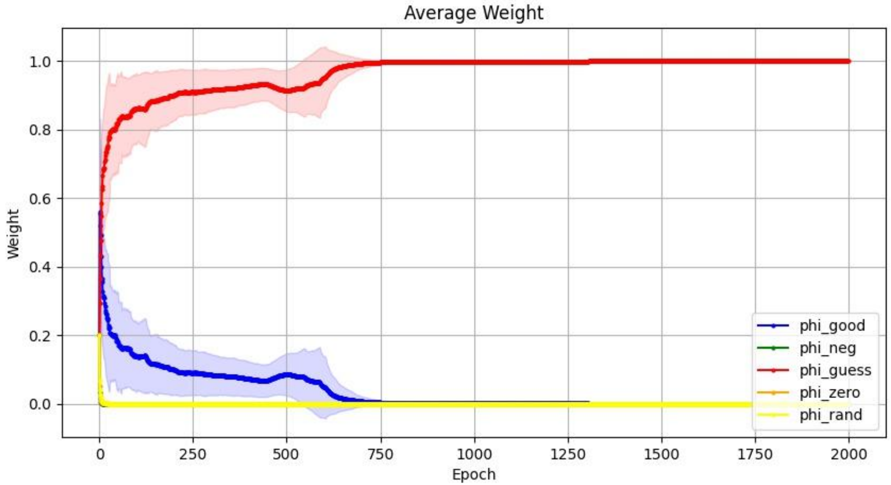
DQN — CartPole

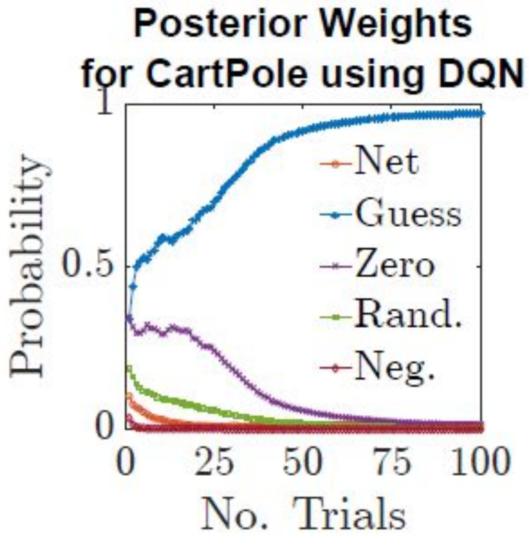




Experimental Results

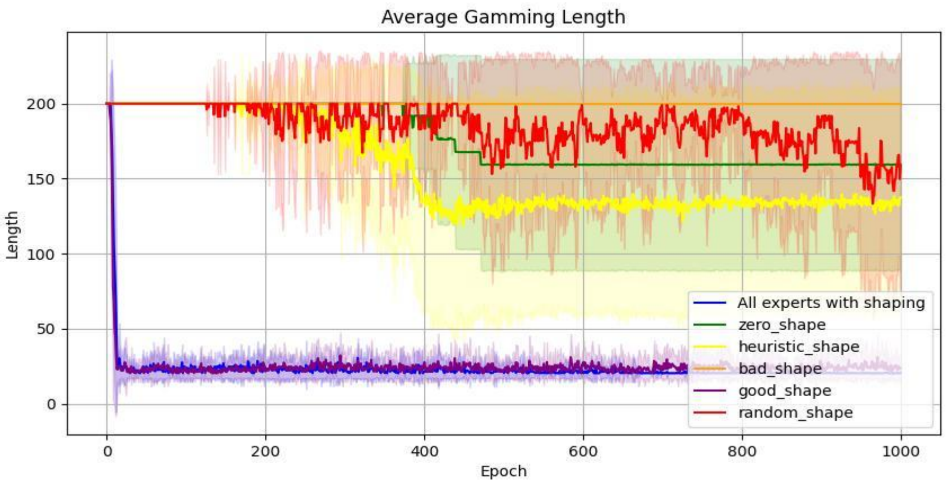
DQN — CartPole

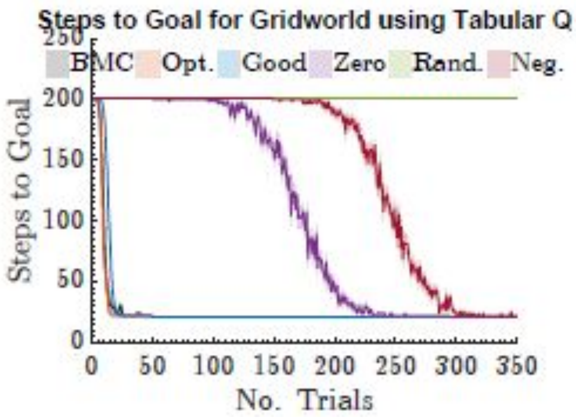




Experimental Results

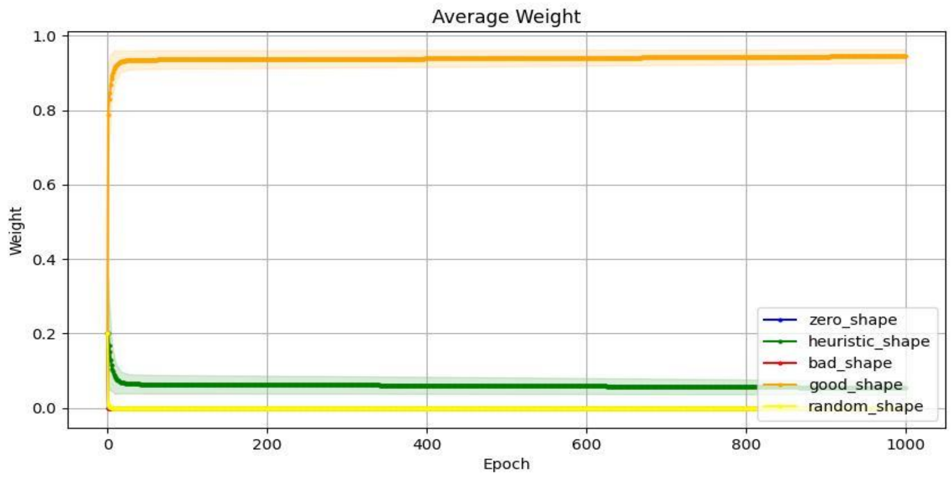
Q-learning — Gridworld <u>without</u> min_eps

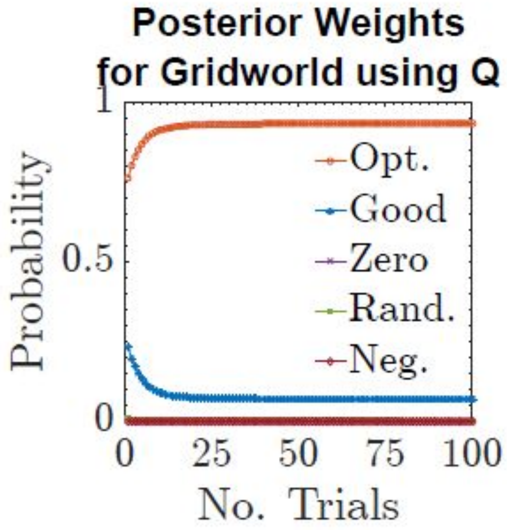




Experimental Results

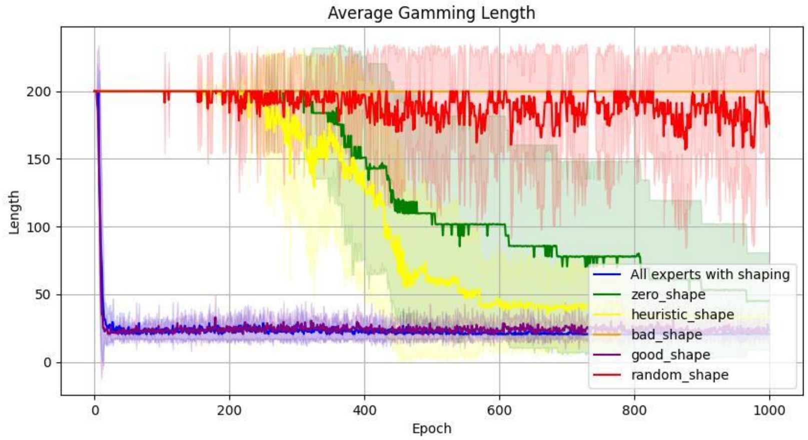
Q-learning — Gridworld without min_eps

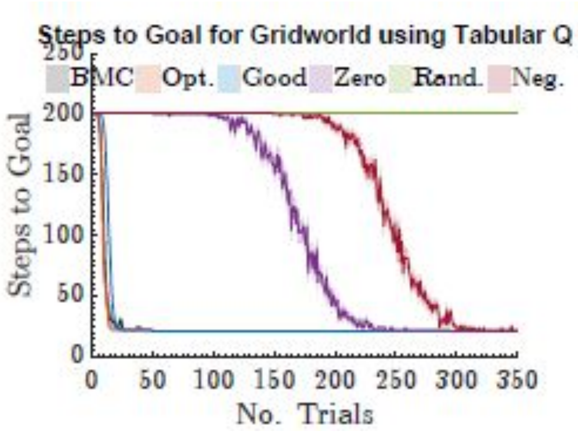




Experimental Results

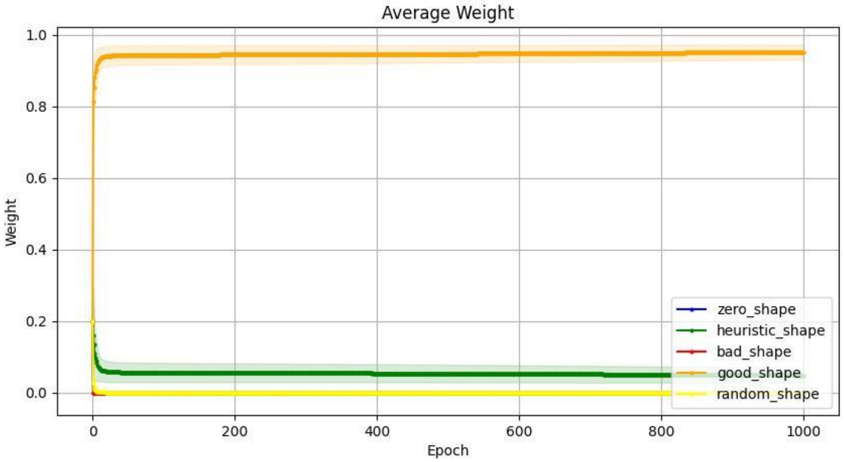
Q-learning — Gridworld <u>with</u> min_eps

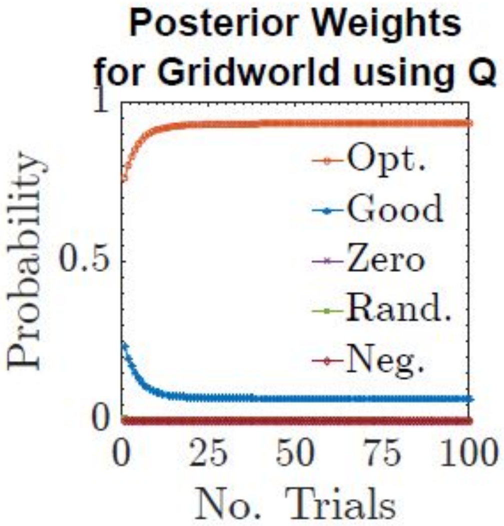




Experimental Results

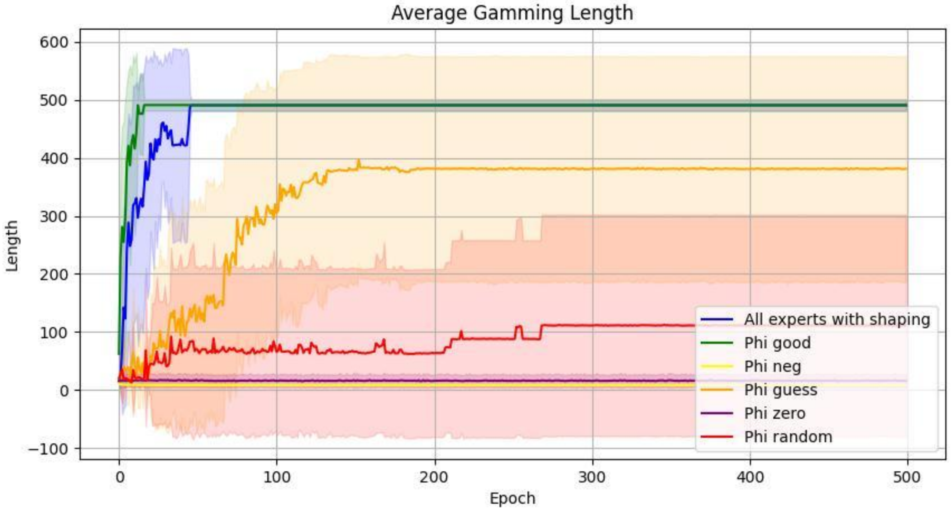
Q-learning — Gridworld with min_eps

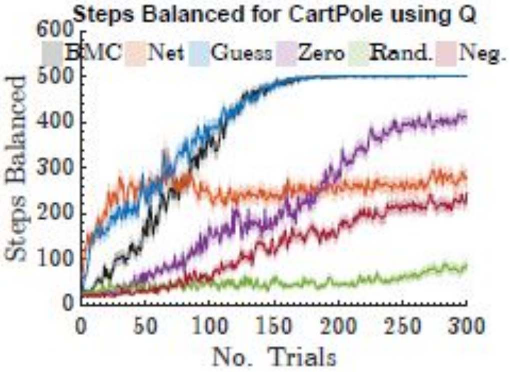




Experimental Results

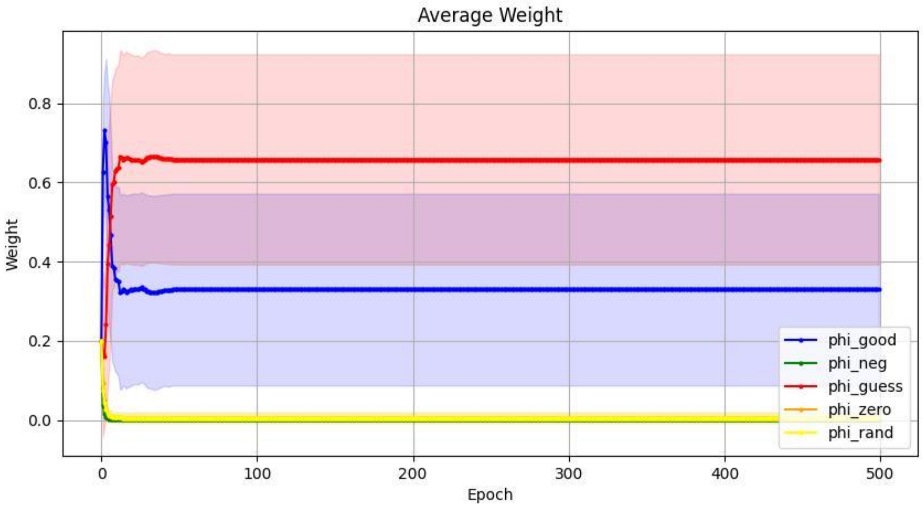
Q-learning — CartPole with early termination

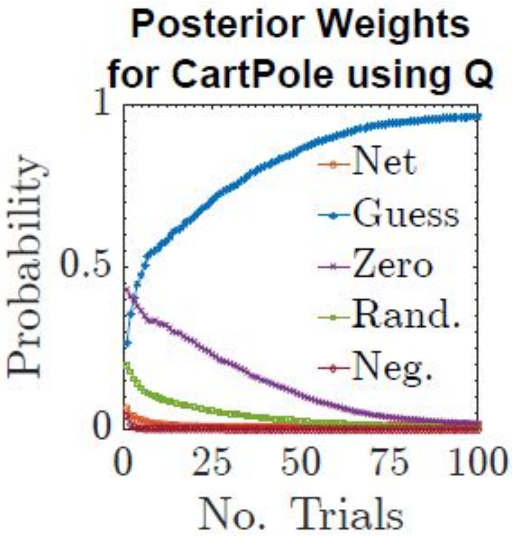




Experimental Results

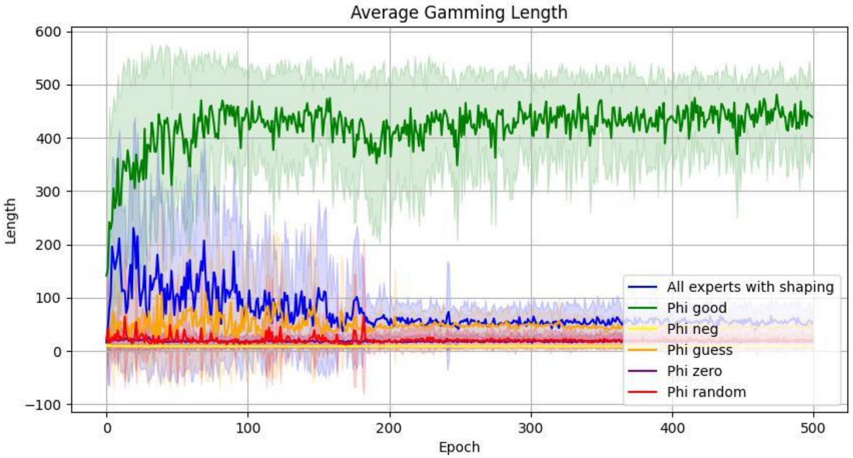
Q-learning — CartPole with early termination

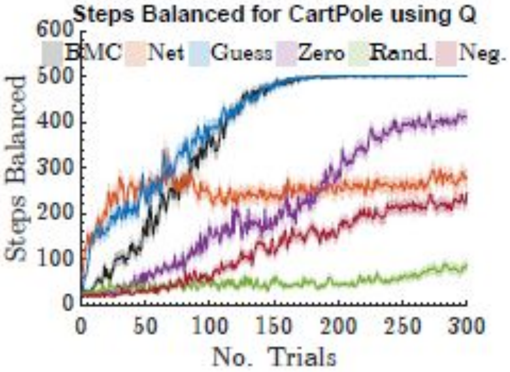




Experimental Results

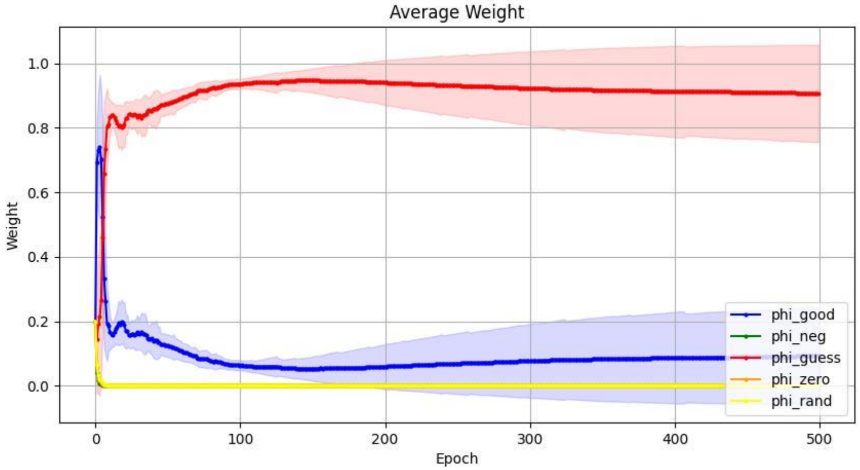
Q-learning — CartPole without early termination

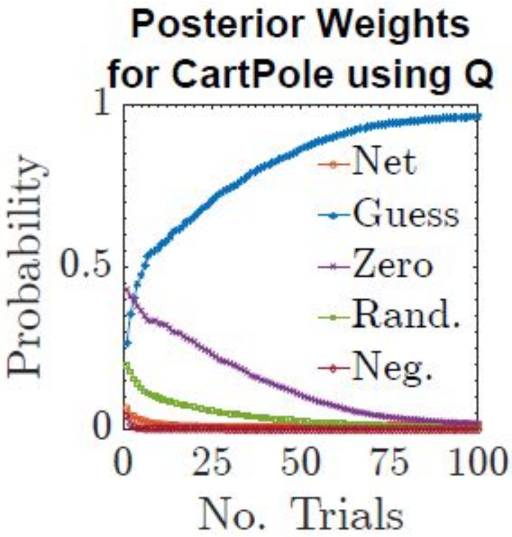




Experimental Results

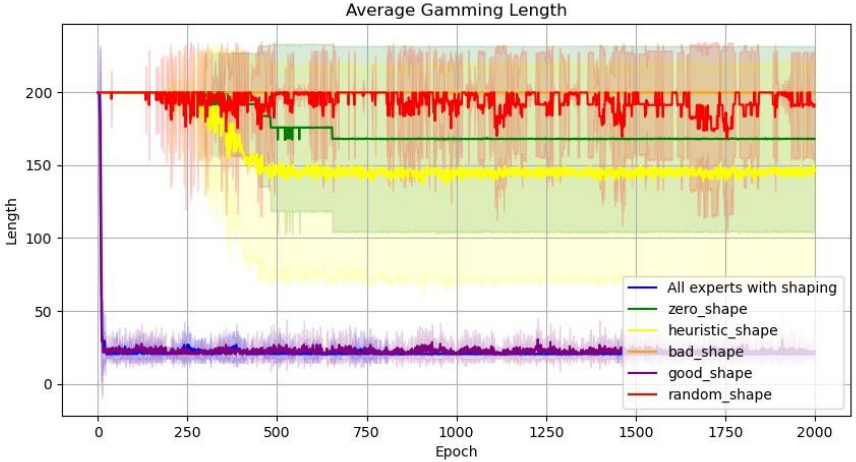
Q-learning — CartPole without early termination

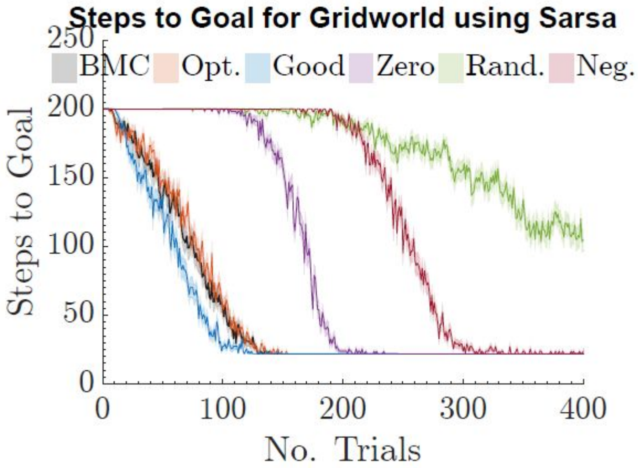




Experimental Results

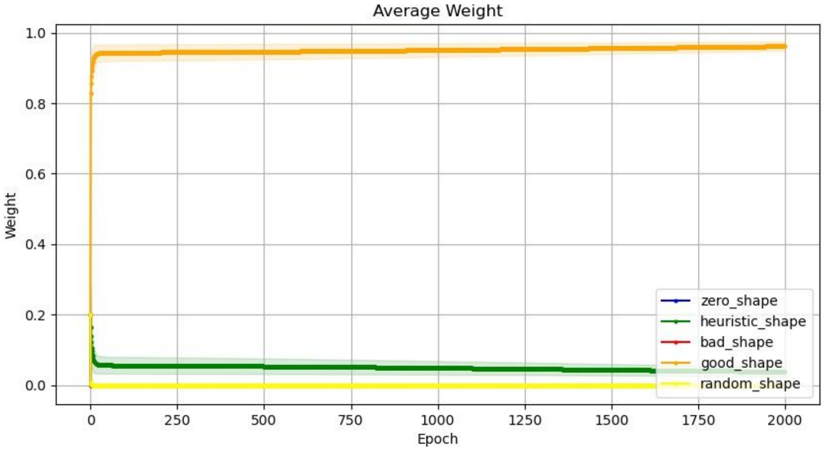
SARSA — Gridworld <u>without</u> min_eps

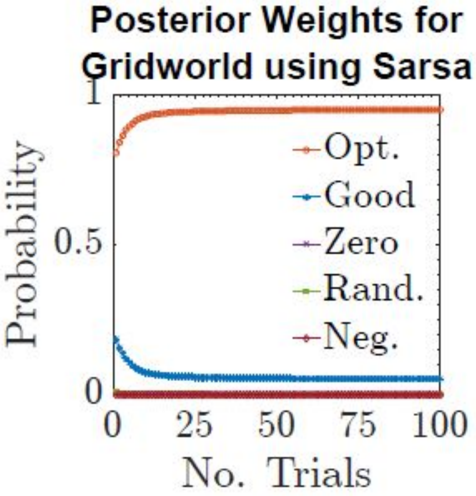




Experimental Results

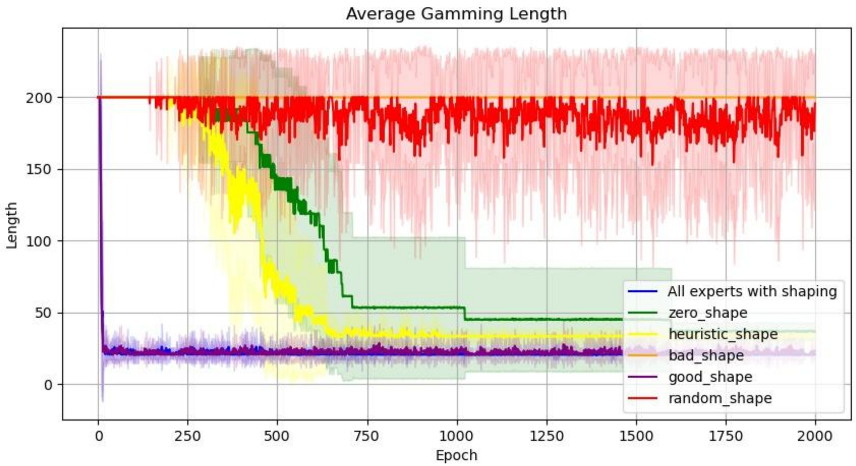
SARSA — Gridworld without min_eps

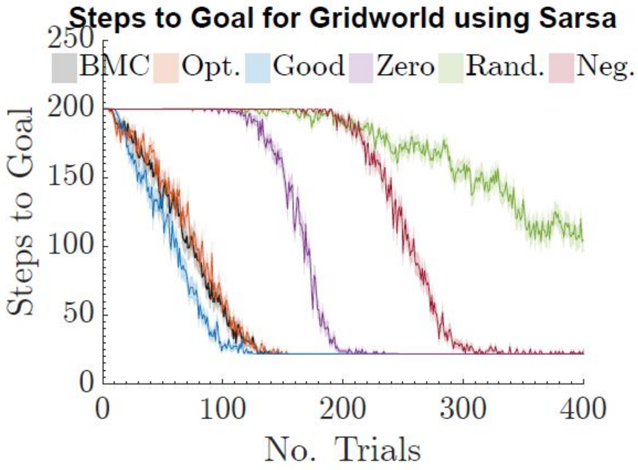




Experimental Results

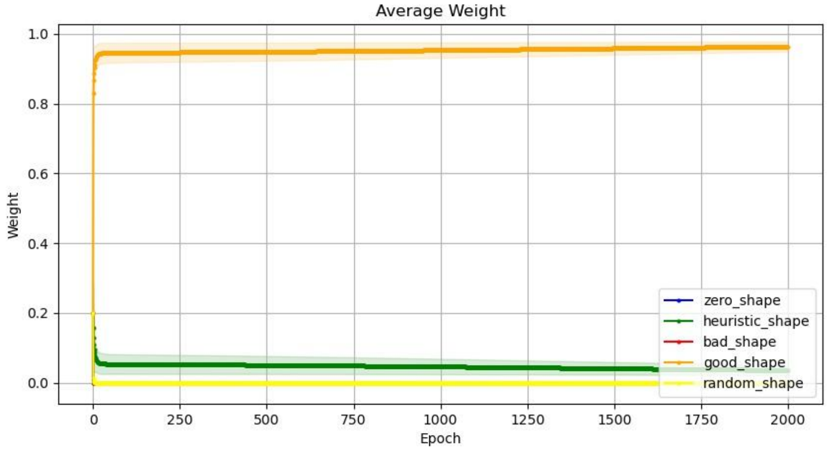
SARSA — Gridworld with min_eps

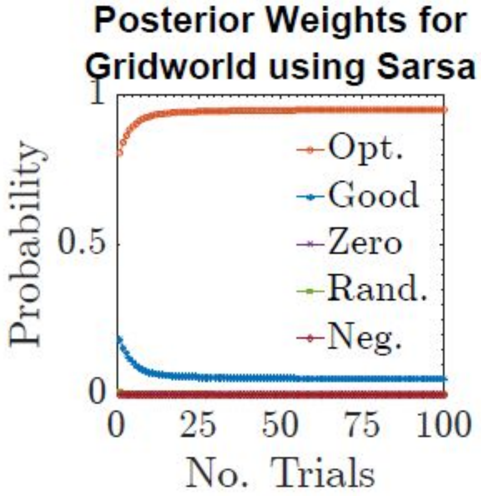




Experimental Results

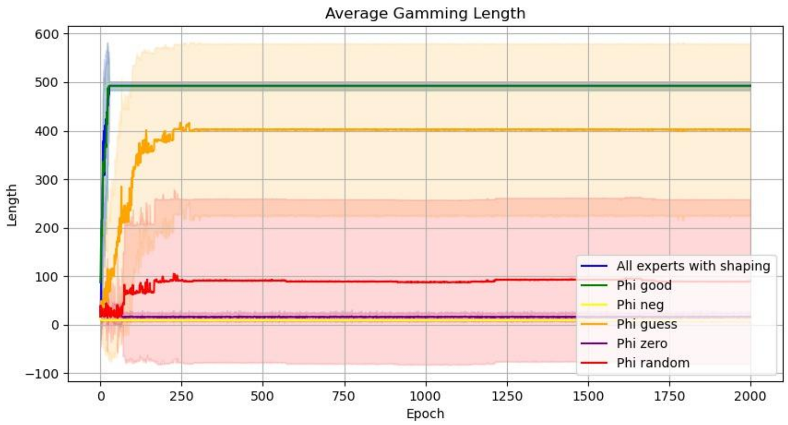
SARSA — Gridworld <u>with</u> min_eps

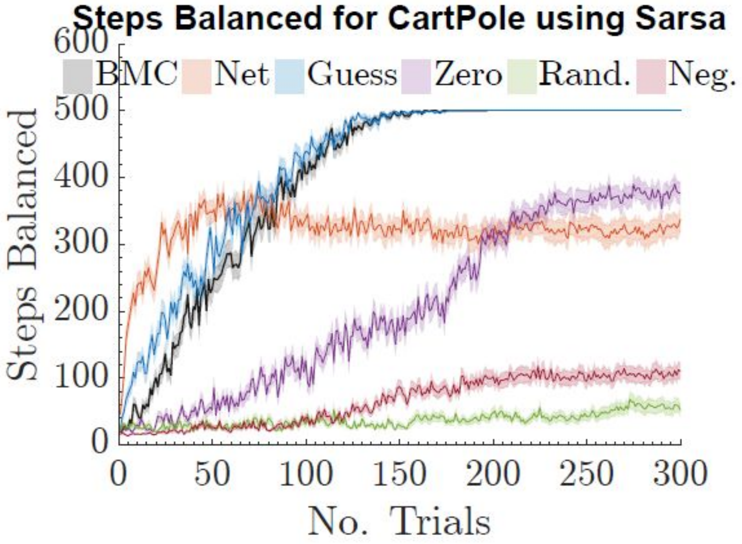




Experimental Results

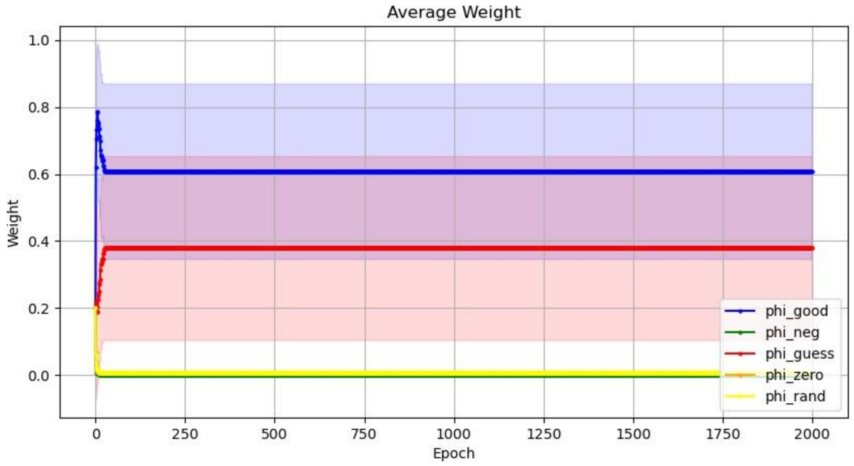
SARSA — CartPole with early termination

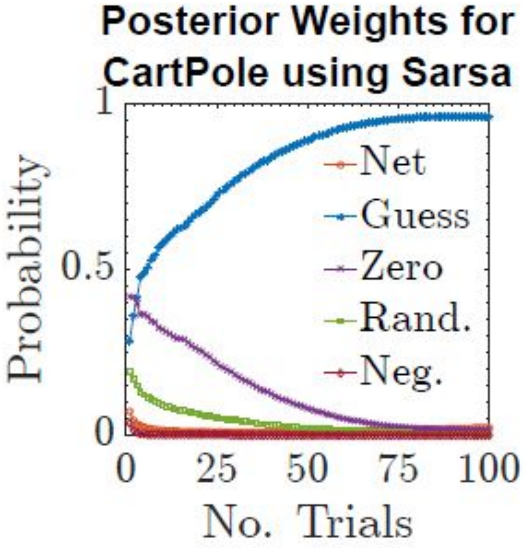




Experimental Results

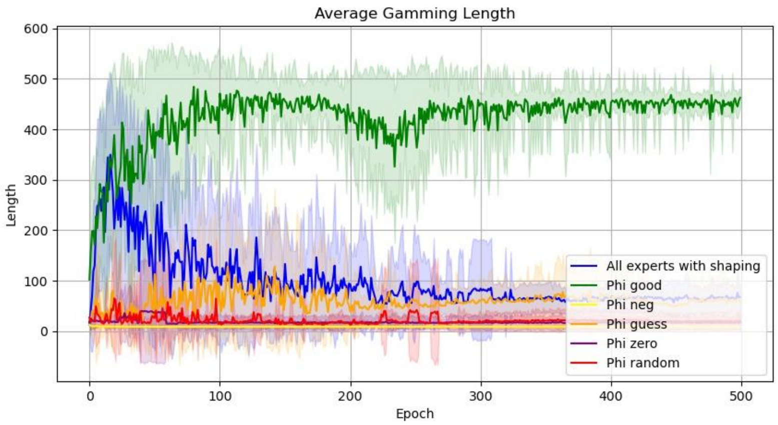
SARSA — CartPole with early termination

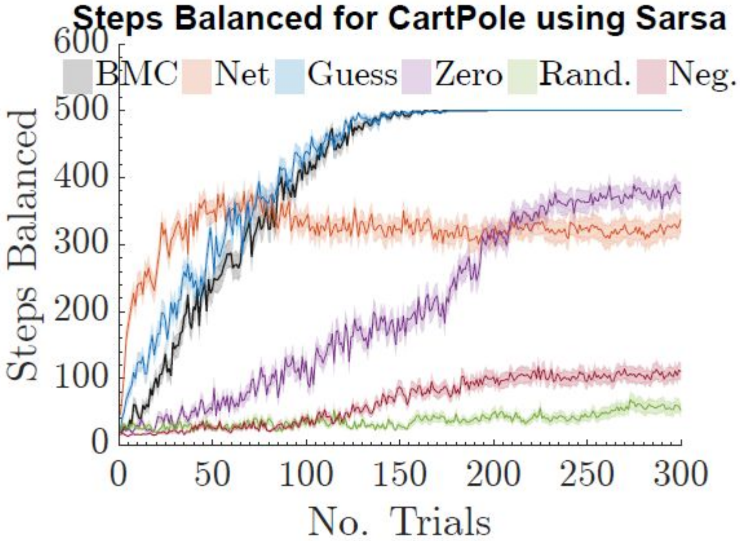




Experimental Results

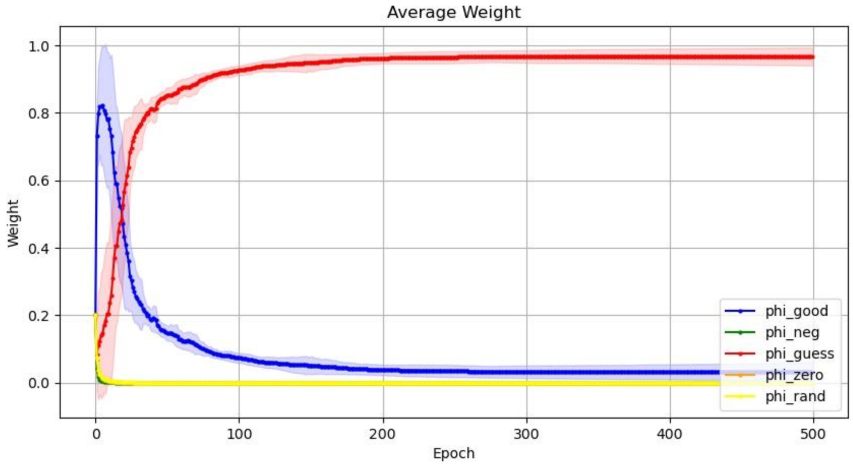
SARSA — CartPole <u>without</u> early termination

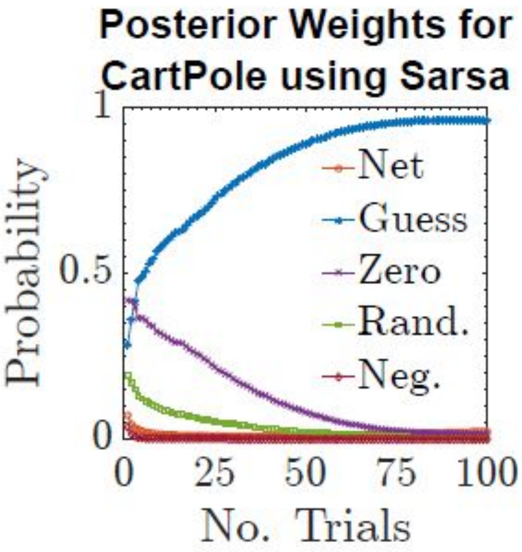




Experimental Results

SARSA — CartPole without early termination





Conclusion

Conclusion

- Bayesian Reward Shaping <u>isn't</u> always giving us a useful combination of experts for better speeding up the learning process.
- Early termination might influence the weights provided by Bayesian Reward Shaping.
- The lower bound of epsilon (for exploration) is important in both tabular methods.

Thanks!

Any questions?