Empirical convergence analysis

Replay threshold

The plots below show how the value of the root state changes with the EVB threshold ξ at multiple horizons, inverse temperatures, and for various prior beliefs. Note that the root Q-values (i.e, the current MF Q-values) were set to 0 everywhere, and the Q-values at all other belief states were set to the expected immediate reward.

The prioritisation was determined by:

$$\text{EVB(b, a)} = p(b_\rho \to b, \pi) \times \mathbb{E}_{b' \sim p(b'|b,a,\pi)}\big[v(b') - v(b)\big]$$

where b_ρ is the root belief and π is a softmax policy.

A couple of interesting observations. First, it looks like higher inverse temperature (β) needs fewer replays t approach the Bayes-optimal value at the root (it doesn't completely get there though because of the softmax). Second, there is an interesting horizon - ξ trade-off – higher horizons need fewer replays to get a better root value estimate than lower horizons; this means that one can (potentially) increase the horizon at the expense of setting a stricter (higher) ξ threshold. As we have seen before, there is also a somewhat "emergent" depth-breadth trade-off which depends on uncertainty – i.e., for the more uncertain arm the algorithm tends to backup values "in breadth", and for the more certain arm it goes more "in depth".

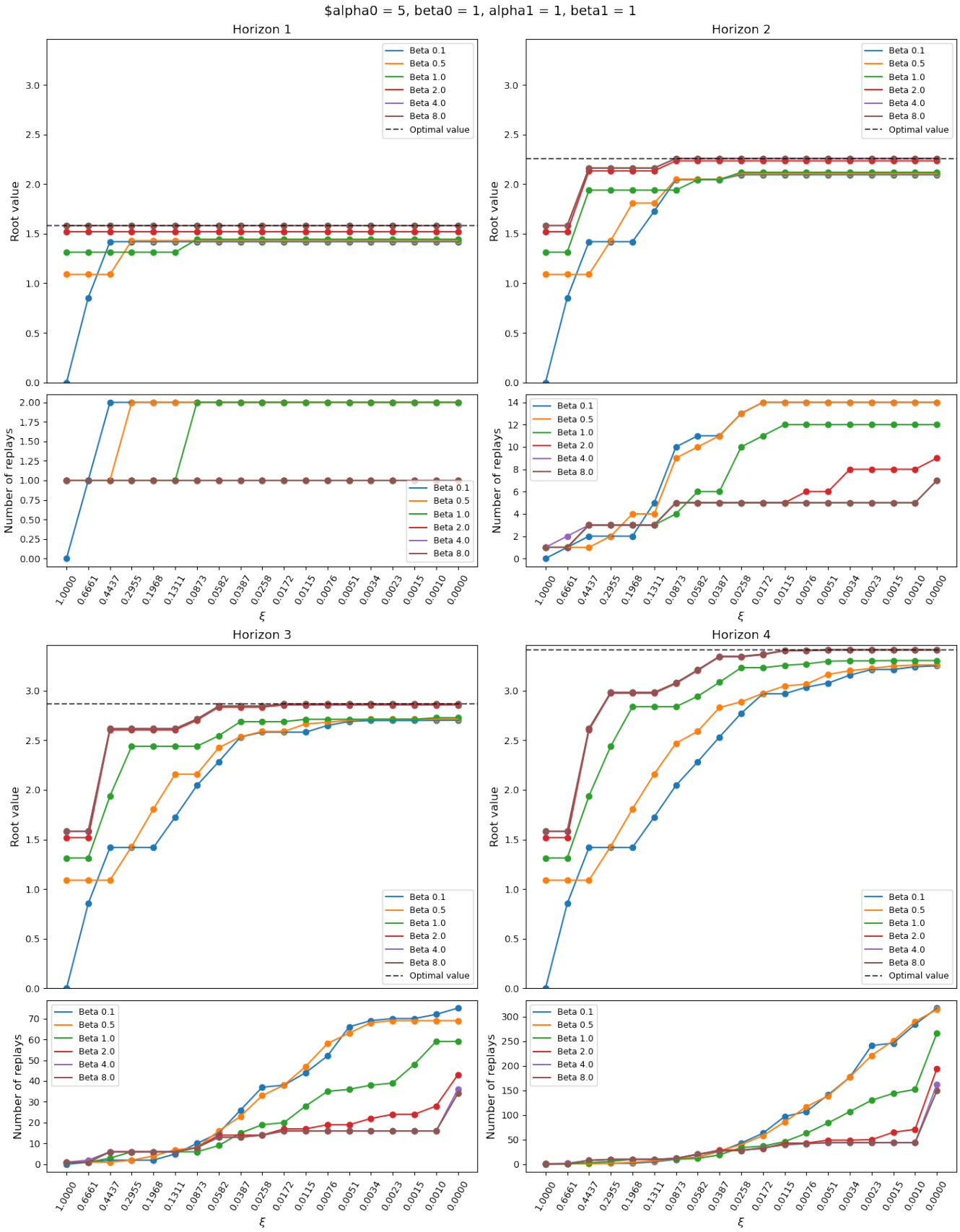
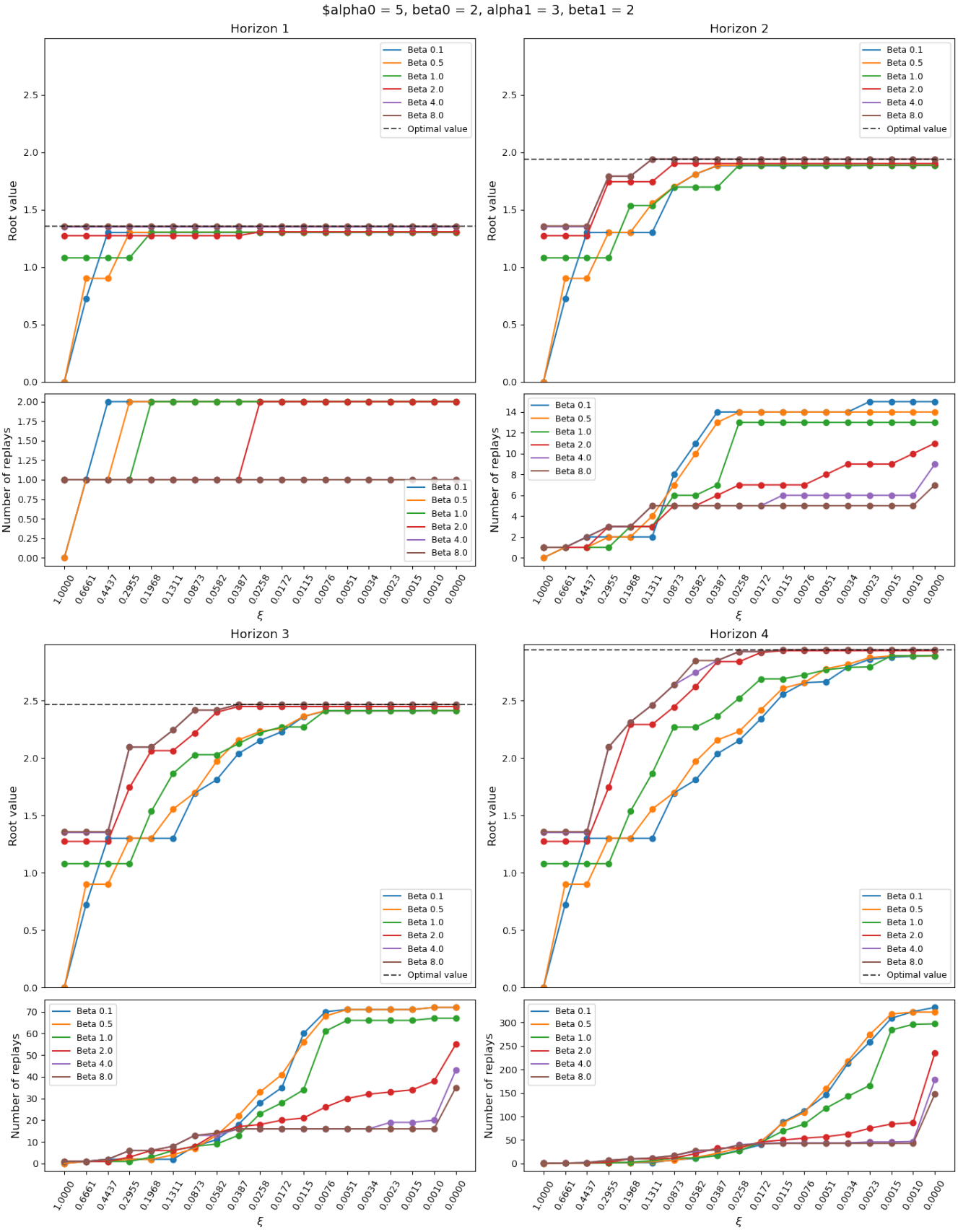
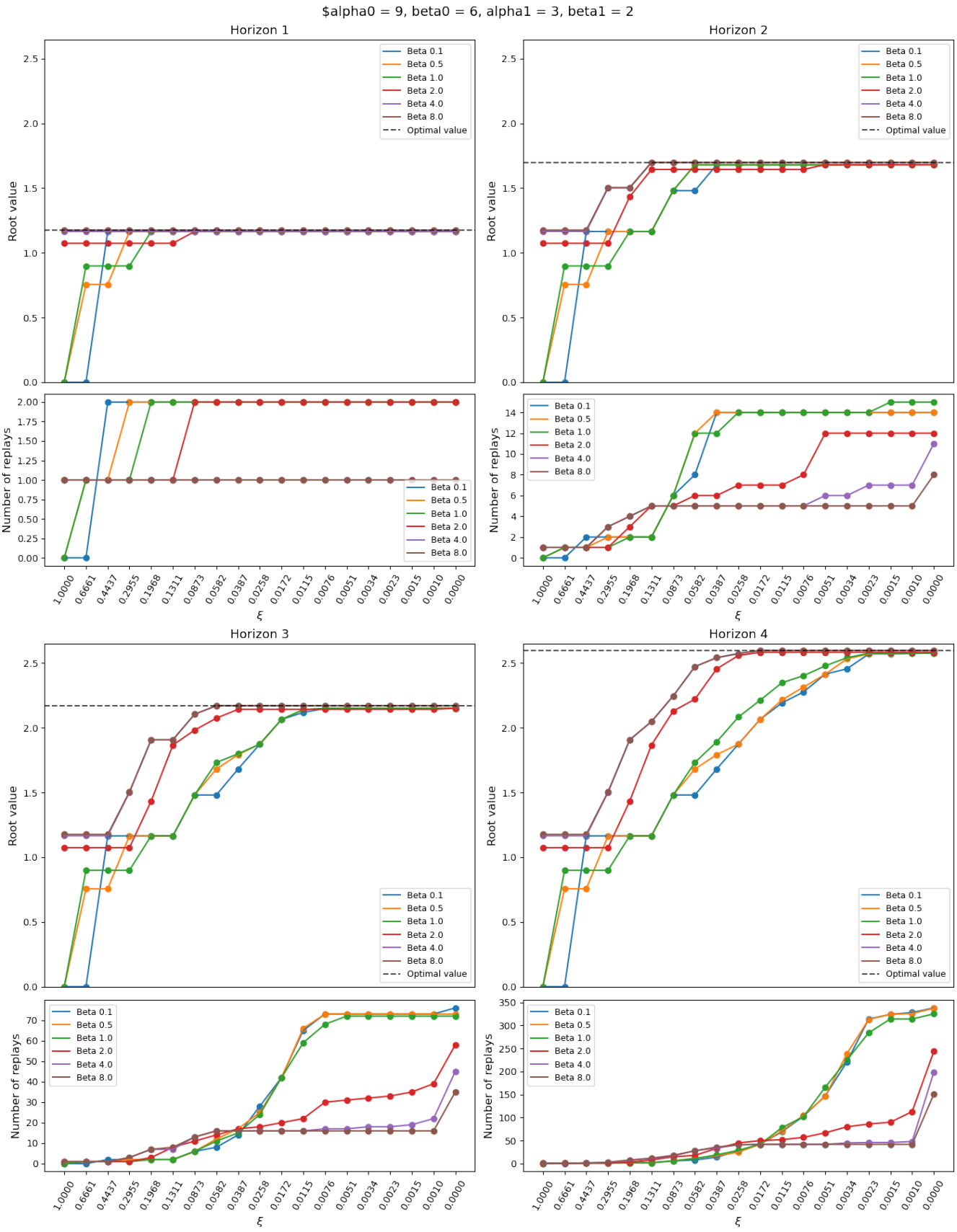
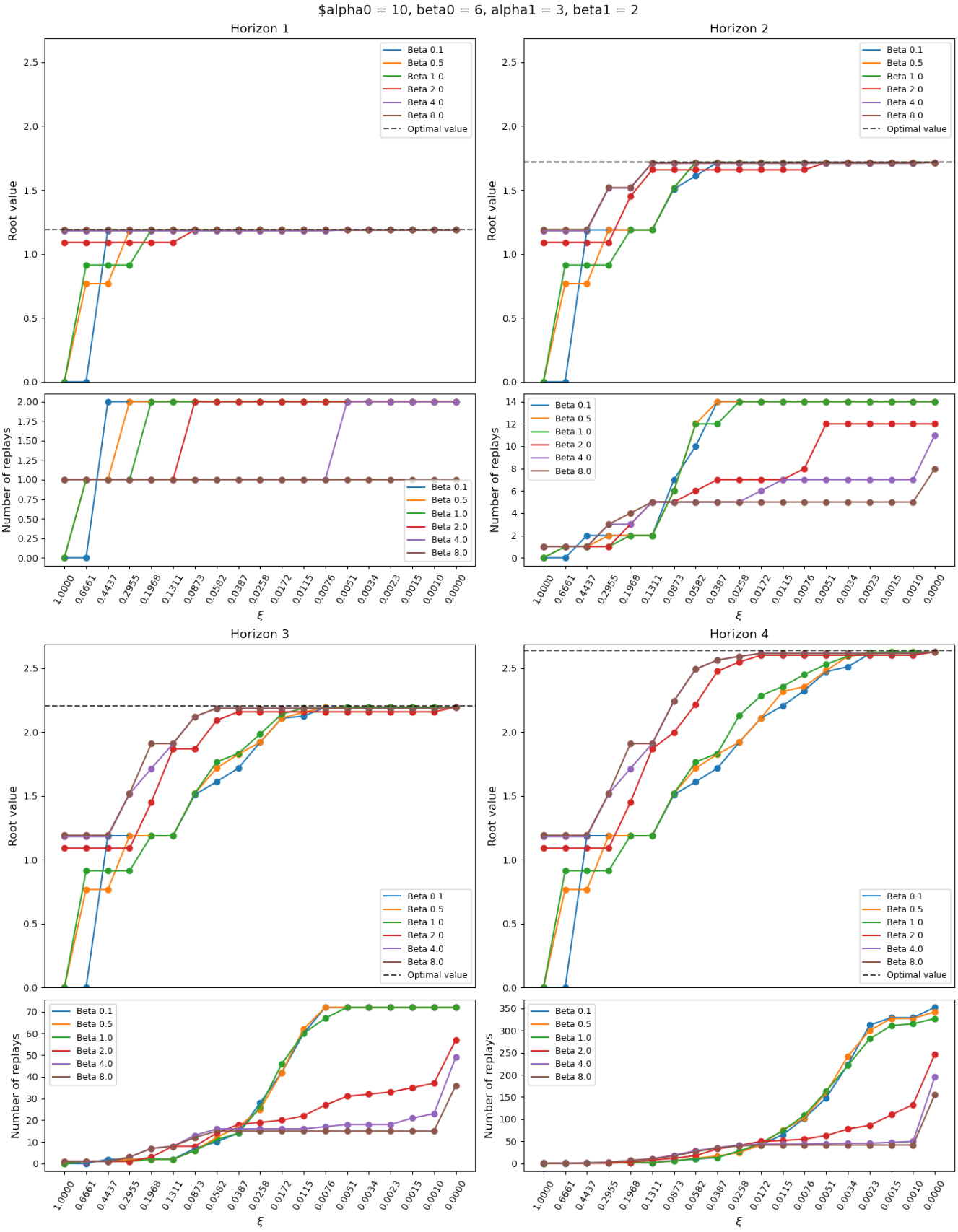


Figure 1: Note that the same inverse temperature parameter β was used within the tree and at the root (which is the plotted root value).


$alpha0 = 5, beta0 = 2, alpha1 = 3, beta1 = 2
Horizon 1
Horizon 2
Horizon 3
Horizon 4
Root value
Number of replays
ξ
Beta 0.1
Beta 0.5
Beta 1.0
Beta 2.0
Beta 4.0
Beta 8.0
Optimal value
1.0000
0.6661
0.4437
0.2955
0.1968
0.1311
0.0873
0.0582
0.0387
0.0258
0.0172
0.0115
0.0076
0.0051
0.0034
0.0023
0.0015
0.0010
0.0000


$alpha0 = 9, beta0 = 6, alpha1 = 3, beta1 = 2
Horizon 1
Horizon 2
Horizon 3
Horizon 4
Root value
Number of replays
ξ
Beta 0.1
Beta 0.5
Beta 1.0
Beta 2.0
Beta 4.0
Beta 8.0
Optimal value
1.0000
0.6661
0.4437
0.2955
0.1968
0.1311
0.0873
0.0582
0.0387
0.0258
0.0172
0.0115
0.0076
0.0051
0.0034
0.0023
0.0015
0.0010
0.0000


$alpha0 = 10, beta0 = 6, alpha1 = 3, beta1 = 2
Horizon 1
Horizon 2
Horizon 3
Horizon 4
Root value
Number of replays
ξ
Beta 0.1
Beta 0.5
Beta 1.0
Beta 2.0
Beta 4.0
Beta 8.0
Optimal value
1.0000
0.6661
0.4437
0.2955
0.1968
0.1311
0.0873
0.0582
0.0387
0.0258
0.0172
0.0115
0.0076
0.0051
0.0034
0.0023
0.0015
0.0010
0.0000

Maze simulations

The prioritisation of experiences at **distal** information states was detemined by:

$$\text{EVB}(< s, b >, a) = p(< s_\rho, b_\rho > \rightarrow < s, b >, \pi) \times \mathbb{E}_{< s', b' > \sim p(< s', b' > | < s, b >, a, \pi)} \left[v(< s', b' >) - v(< s, b >) \right]$$

where s_ρ is the root physical state and b_ρ is the root belief.

The implementation is very similar to that of the bandit case – during planning (replay), the agent generates a planning tree from every state and for every action. The updates can then happen either in distal parts of the trees (only 1 update is allowed at each iteration), or at the root. The value of each tree node is initialised to the agent's current MF Q-values.

For the **root** action updates, the EVB is computed in the standard M&D fashion:

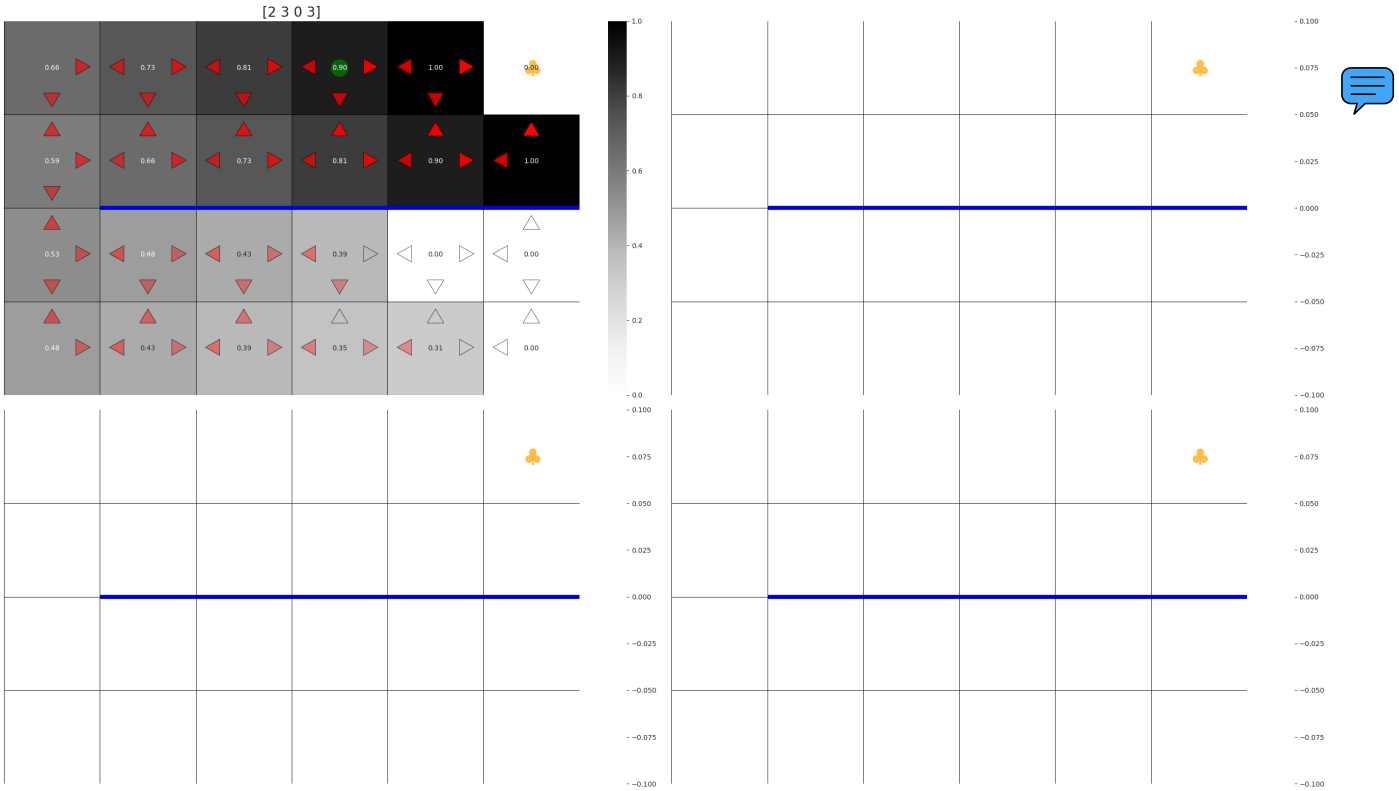
$$\text{EVB}(< s_\rho, b_\rho >, a) = \text{Need}(< x, b_\rho > \rightarrow < s_\rho, b_\rho >, \pi) \times \text{Gain}(< s_\rho, b_\rho >, a)$$

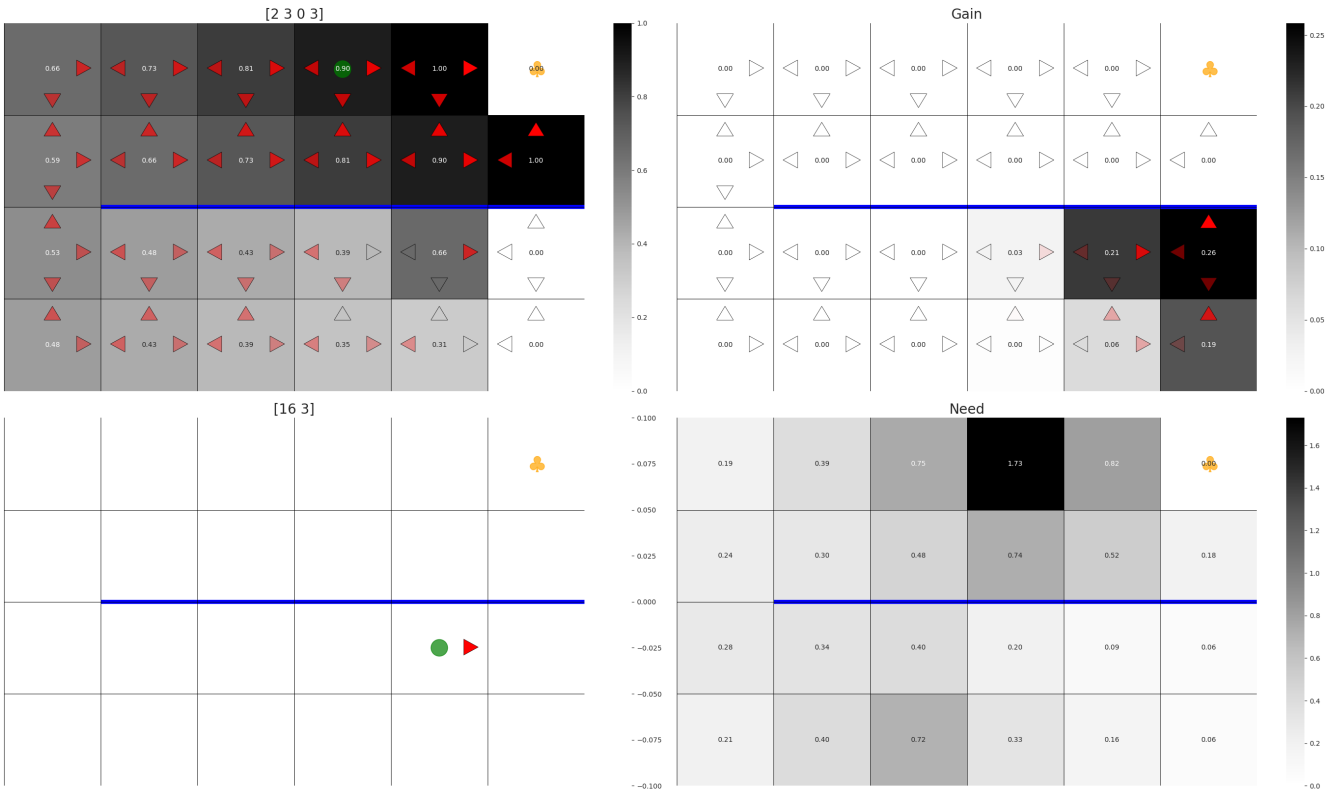
This is a bit tricky to write down in a less cryptic way. Essentially, the assumption is that state s_ρ can be reached from the agent's current location x without belief dynamics – that is, Need in this case is exactly the same as in M&D.

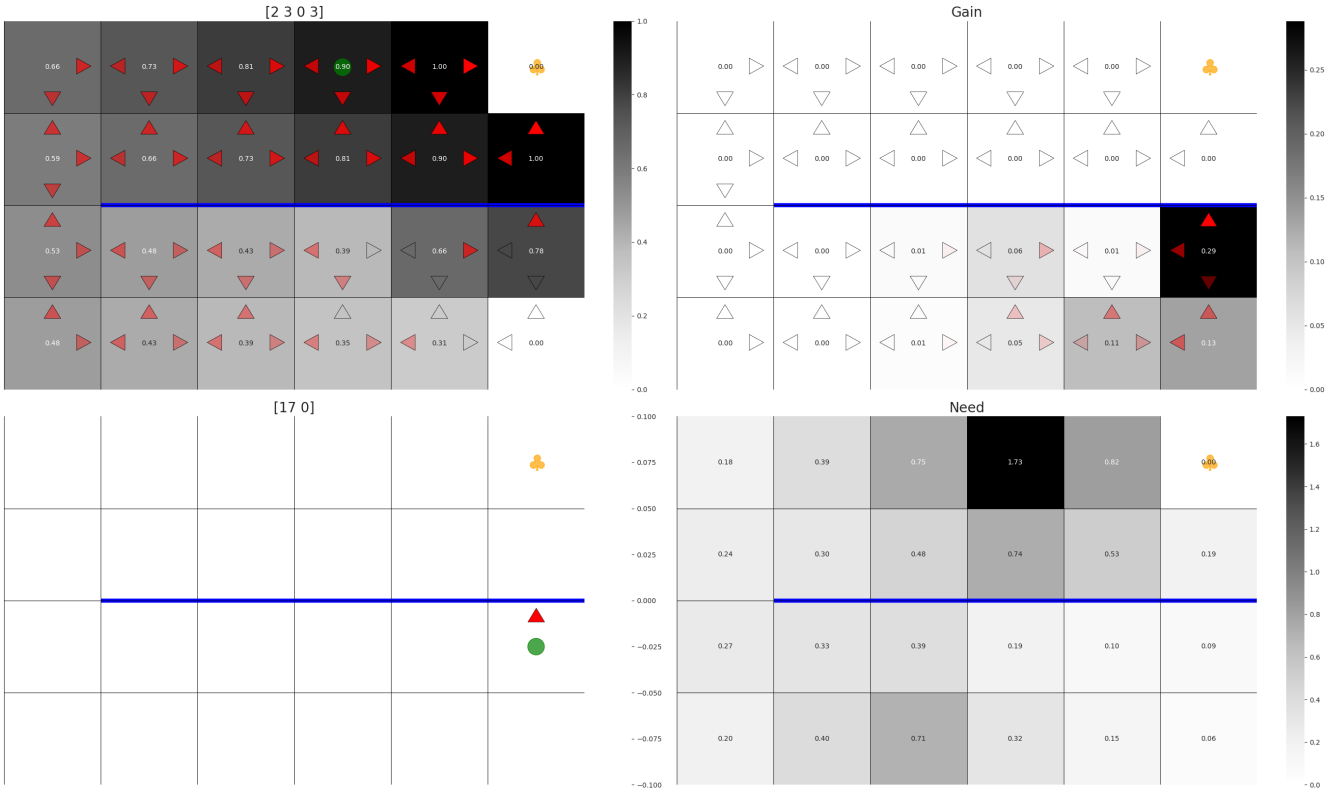
The crucial difference is the Gain term. Our Bayes-adaptive prioritised sweeping procedure effectively "chooses" which computation to perform at the root. Since it approximates a Bayes-optimal value (yet with a heuristic at the distal bits in the form of the current MF Q-values), the future information gain is subsumed by M&D's Gain at the root.

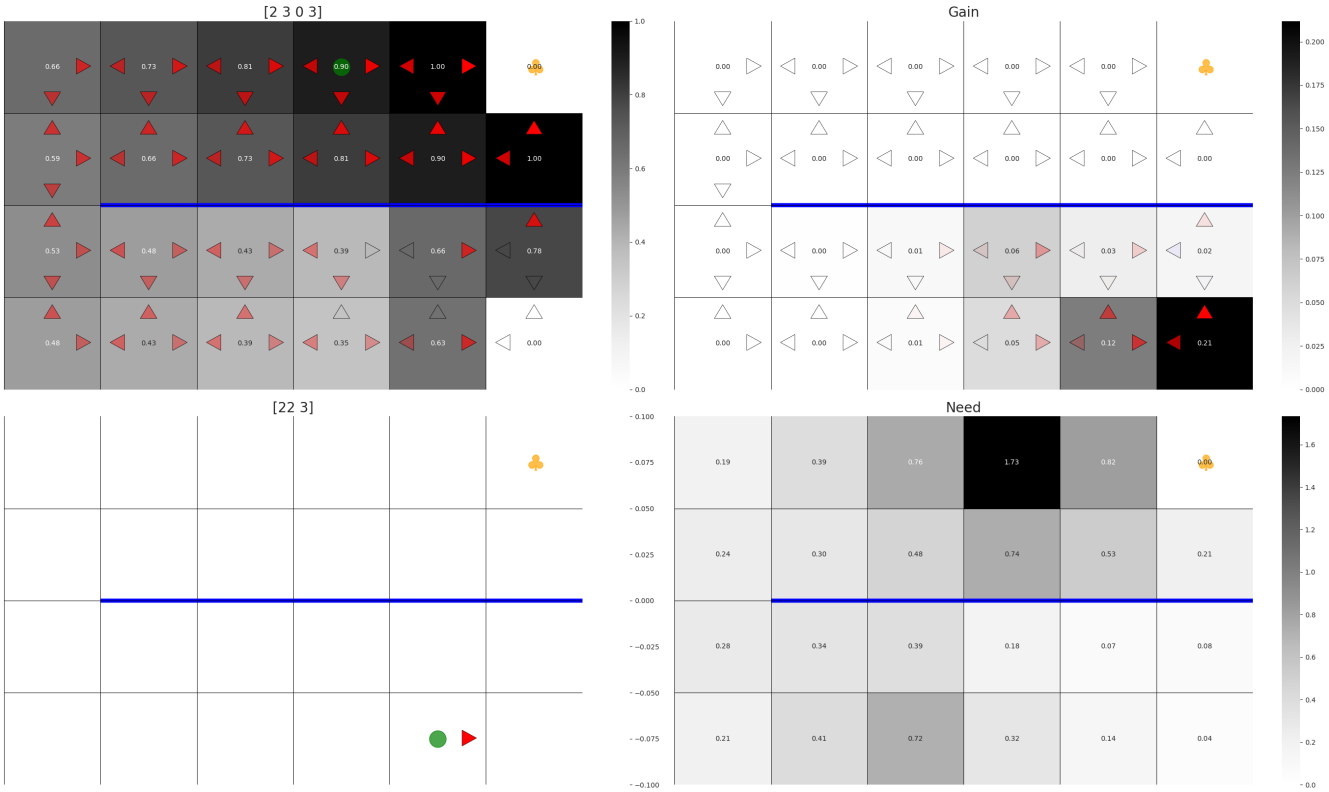
Below I show an example exploratory replay from a toy simulation. First, the agent was simulated in the environment for 3000 moves without being allowed to replay (so only MF learning) – and luckily, with this seed, it never got to experience some of the bottom rightmost states (the ones where MF Q-values are all 0). The agent is completely certain about all transitions; however, at move 3000 I set its Beta posterior to $(1, 1)$ for the action 'up' below the rightmost barrier – hence making it expect that the probability of that barrier being removed is 0.5. I then let it replay with horizon 2. The start state is the square which is the 3rd from the left at the very bottom.

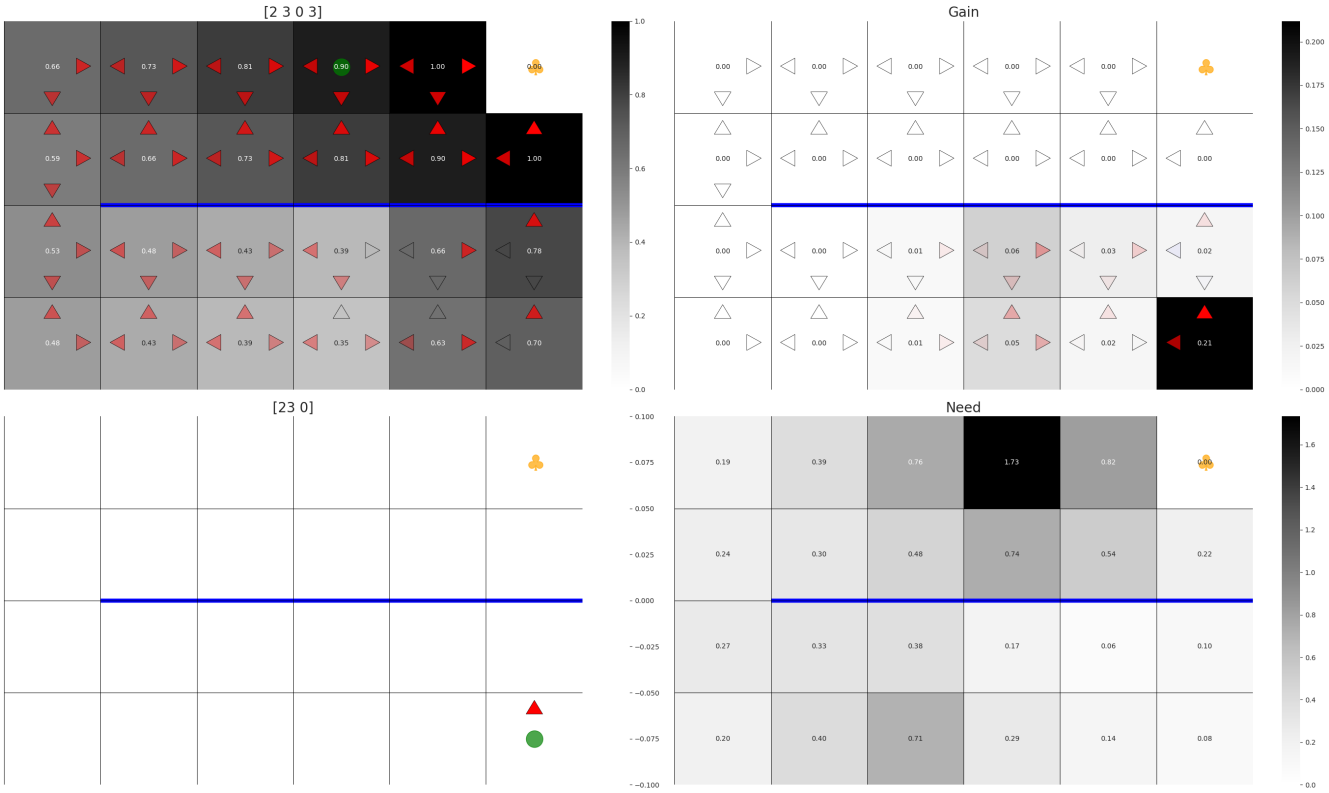
The top left plots show MF Q-values and the agent's current location (green dot); the bottom left plots show the replay chosen by the agent; the top right plots show Gain; and the bottom right plots show Need. Note that only root updates are shown.

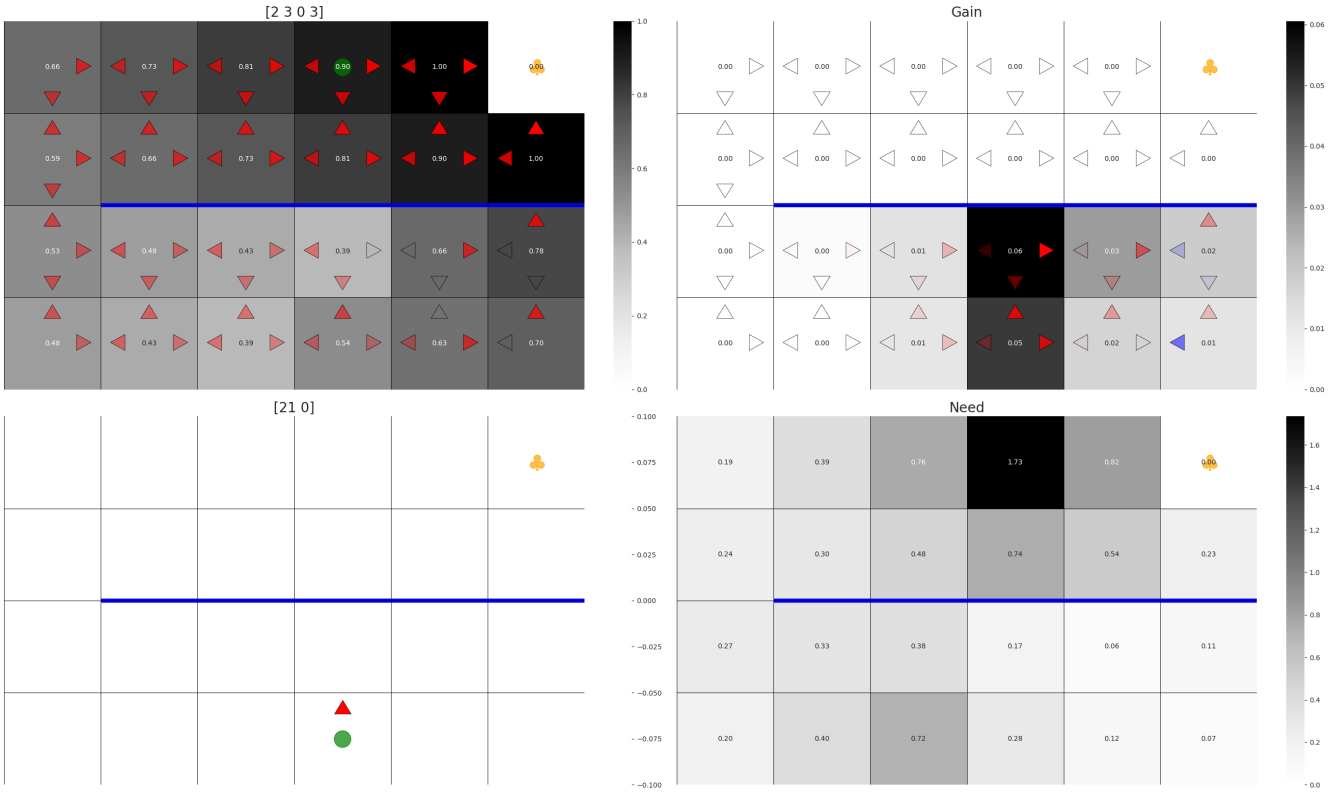

[2 3 0 3]

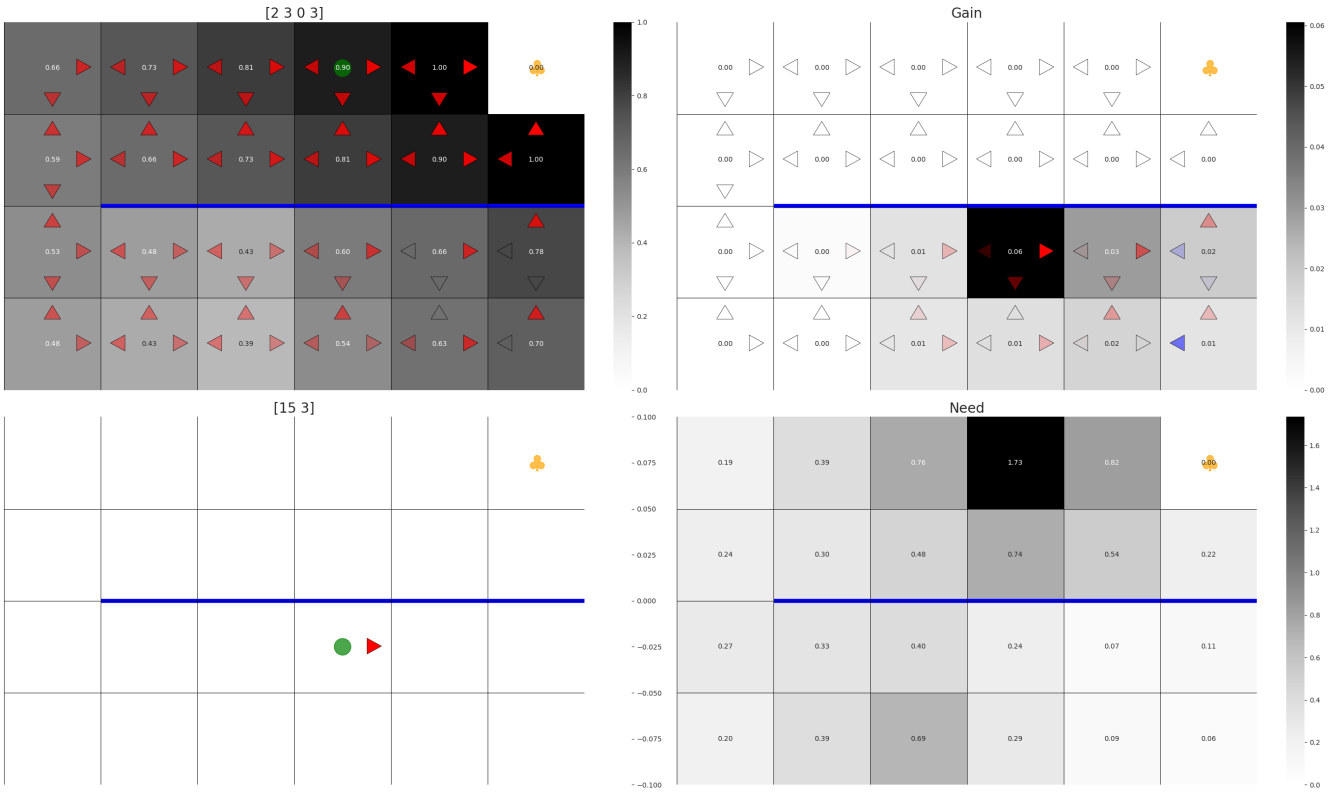

[2 3 0 3]
Gain
[16 3]
Need


[2 3 0 3]
Gain
[17 0]
Need


[2 3 0 3]
Gain
[22 3]
Need


[2 3 0 3]
Gain
[23 0]
Need


[2 3 0 3]
Gain
[21 0]
Need


[2 3 0 3]
Gain
[15 3]
Need

I think this is very interesting and there are quite a few important considerations here. First, the horizon really matters – since it helps overcome the very low Need at (currenty) unexpected states. In the example shown above, for instance, if the horizon was set to 1, there would be no exploratory replays at all. This is because Need at the state adjacent to the shortcut is very small. However, for a larger horizon (2), the agent could plan crossing the barrier from a state which had a slightly higher Need – just enough so that the expected value of that replay crossed the threshold, even though Gain was slightly lower due to γ (which was set to 0.9 in this simulation). That replay, in turn, increased the Need for the subsequent state (the one adjacent to the barrier), and hence that replay followed next. This is the reason why the model predicts such exploratory replays to proceed in a forward manner.

One thing which feels a bit uncomfo e is that Gain turns out to be much more powerful than Need (because Gain can dramatically change Need). This, of course, depends on the horizon and learning rates (replay learning rate was set to 1 in this simulation), as well as the discount factor.

For clarity, I also wrote down a very high-level pseudocode of the algorithm on the next page.

Replay pseudocode

Initialise
x $\leftarrow$ agent's current physical state
b_ρ $\leftarrow$ agent's current belief
H $\leftarrow$ horizon
α_r $\leftarrow$ replay learning rate
γ $\leftarrow$ discount factor
ξ $\leftarrow$ EVB replay threshold

for each $s \in \mathcal{S}$ **do**
    build belief tree t_s with root information state $z_\rho =< s_\rho = s, b_\rho >$ up to horizon H
    initialise $Q_{t_s(h)}(z =< k, b >, a) \leftarrow Q^{MF}(< k, b_\rho >, a) \;\; \forall h \in H, z \in t_s(h)$
done
while $\max_{z,a,t_s,h} \mathrm{EVB}_{t_s(h)}(z, a) > \xi$ **do**
    for each h in $(H-1)$ **do**
        for each information state $z =< k, b >\in t_s(h)$ **do**
            for each action $a \in \mathcal{A}$ **do**
                if $(h > 1)$ **do**
                    $\mathrm{EVB}_{t_s(h)}(z, a) \leftarrow p(z_\rho \rightarrow z, \pi) \times \mathbb{E}_{z' \sim p(z'|z,a,\pi)}\big[v(z') - v(z)\big]$
                else if $(h == 1)$ **do**
                    $\mathrm{EVB}_{t_s(h)}(z_\rho, a) \leftarrow \mathrm{Need}(< x, b_\rho > \rightarrow z_\rho, \pi) \times \mathrm{Gain}(z_\rho, a)$
                done
            done
        done
    done
    $z, a, t_s, h \leftarrow \mathrm{argmax}_{z,a,t_s,h}\ \mathrm{EVB}_{t_s(h)}(z, a)$
    if $(h == 1)$ **do**
        $Q^{MF}(z, a) \leftarrow Q_{t_s(h)}(z, a) + \alpha_r \mathbb{E}_{z' \sim p(z'|z,a,\pi)}\big[R(z') + \gamma \max_{a' \in \mathcal{A}} Q_{t_s(h+1)}(z', a') - Q(z, a)\big]$
        for each $s \in \mathcal{S}$ **do**
            build belief tree t_s with root information state $z_\rho =< s_\rho = s, b_\rho >$ up to horizon H
            initialise $Q_{t_s(h)}(z =< k, b >, a) \leftarrow Q^{MF}(< k, b_\rho >, a) \;\; \forall h \in H, z \in t_s(h)$
        done
    else if $(h! = 1)$ **do**
        $Q_{t_s(h)}(z, a) \leftarrow Q_{t_s(h)}(z, a) + \alpha_r \mathbb{E}_{z' \sim p(z'|z,a,\pi)}\big[R(z') + \gamma \max_{a' \in \mathcal{A}} Q_{t_s(h+1)}(z', a') - Q(z, a)\big]$
    done
repeat