Supplementary Material

Proof of Theorem 1 and comments

Consider the BA-UCT algorithm: UCT applied to the Bayes-Adaptive MDP (dynamics are described in Equation 1). Let $\mathcal{D}^\pi(h_T)$ be the *rollout distribution* of BA-UCT: the probability that history h_T is generated when running the BA-UCT search from $\langle s_t, h_t\rangle$, with h_t a prefix of h_T, $T-t$ the effective horizon in the search tree, and π an arbitrary BAMDP policy. Similarly define the similar quantity $\tilde{\mathcal{D}}^\pi(h_T)$: the probability that history h_T is generated when running the BAMCP algorithm. The following lemma shows that these two quantities are in fact equivalent.[3]

Lemma 1. *$\mathcal{D}^\pi(h_T) = \tilde{\mathcal{D}}^\pi(h_T)$ for all BAMDP policies $\pi : \mathcal{H} \to A$.*

Proof. Let π be arbitrary. We show by induction that for all suffix histories h of h_t, $\mathcal{D}^\pi(h) = \tilde{\mathcal{D}}^\pi(h)$; but also $P(\mathcal{P}\,|h) = \tilde{P}_h(\mathcal{P})$ where $P(\mathcal{P}\,|h)$ denotes (as before) the posterior distribution over the dynamics given h and $\tilde{P}_h(\mathcal{P})$ denotes the distribution of $\mathcal{P}$ at node h when running BAMCP.

Base case: At the root ($h = h_t$, suffix history of size 0), it is clear that $\tilde{P}_{h_t}(\mathcal{P}) = P(\mathcal{P}\,|h_t)$ since we are sampling from the posterior at the root node and $D^\pi(h_t) = \tilde{\mathcal{D}}^\pi(h_t) = 1$ since all simulations go through the root node.

Step case:

Assume proposition true for all suffices of size i. Consider any suffix has' of size $i+1$, where $a \in A$ and $s' \in S$ are arbitrary and h is an arbitrary suffix of size i ending in s. The following relation holds:

$$\begin{aligned}
\mathcal{D}^\pi(has') &= \mathcal{D}^\pi(h)\pi(h,a)\int_{\mathcal{P}} \mathrm{d}\mathcal{P}\, P(\mathcal{P}\,|h)\,\mathcal{P}(s,a,s') && (3)\\
&= \tilde{\mathcal{D}}^\pi(h)\pi(h,a)\int_{\mathcal{P}} \mathrm{d}\mathcal{P}\, \tilde{P}_h(\mathcal{P})\,\mathcal{P}(s,a,s') && (4)\\
&= \tilde{\mathcal{D}}^\pi(has'), && (5)
\end{aligned}$$

where the second line is obtained using the induction hypothesis, and the rest from the definitions. In addition, we can match the distribution of the samples $\mathcal{P}$ at node has':

$$\begin{aligned}
P(\mathcal{P}\,|has') &= P(has'|\,\mathcal{P})P(\mathcal{P})/P(has') && (6)\\
&= P(h|\,\mathcal{P})P(\mathcal{P})\,\mathcal{P}(s,a,s')/P(has') && (7)\\
&= P(\mathcal{P}\,|h)P(h)\,\mathcal{P}(s,a,s')/P(has') && (8)\\
&= ZP(\mathcal{P}\,|h)\,\mathcal{P}(s,a,s') && (9)\\
&= Z\tilde{P}_h(\mathcal{P})\,\mathcal{P}(s,a,s') && (10)\\
&= Z\tilde{P}_{ha}(\mathcal{P})\,\mathcal{P}(s,a,s') && (11)\\
&= \tilde{P}_{has'}(\mathcal{P}), && (12)
\end{aligned}$$

where Equation 10 is obtained from the induction hypothesis, Equation 11 is obtained from the fact that the choice of action at each node is made independently of the samples $\mathcal{P}$. Finally, to obtain Equation 12 from Equation 11, consider the probability that a sample $\mathcal{P}$ arrives at node has', it first needs to traverse node ha (this occurs with probability $\tilde{P}_{ha}(\mathcal{P})$) and then, from node ha, the state s' needs to be sampled (this occurs with probability $\mathcal{P}(s,a,s')$); therefore, $\tilde{P}_{has'}(\mathcal{P}) \propto \tilde{P}_{ha}(\mathcal{P})\,\mathcal{P}(s,a,s')$. Z is the normalization constant: $Z = 1/\int_{\mathcal{P}} \mathcal{P}(s,a,s')P(\mathcal{P}\,|h) = 1/\int_{\mathcal{P}} \mathcal{P}(s,a,s')\tilde{P}_h(\mathcal{P})$. This completes the induction. ☐

Proof of Theorem 1. The UCT analysis in Kocsis and Szepesvári [16] applies to the BA-UCT algorithm, since it is vanilla UCT applied to the BAMDP (a particular MDP). By Lemma 1, BAMCP

---

[3] For ease of notation, we refer to a node with its history as opposed to its state and history as done in the rest of the paper.

simulations are equivalent in distribution to BA-UCT simulations. The nodes in BAMCP are therefore being evaluated as in BA-UCT, providing the result. □

Lemma 1 provides some intuition for why belief updates are unnecessary in the search tree: the search tree filters the samples from the root node so that the distribution of samples at each node is equivalent to the distribution obtained when explicitly updating the belief. In particular, the root sampling in POMCP [20] and BAMCP is different from evaluating the tree using the posterior mean. This is illustrated empirically in the section below in the case of simple Bandit problems.

BAMCP versus Gittins indices

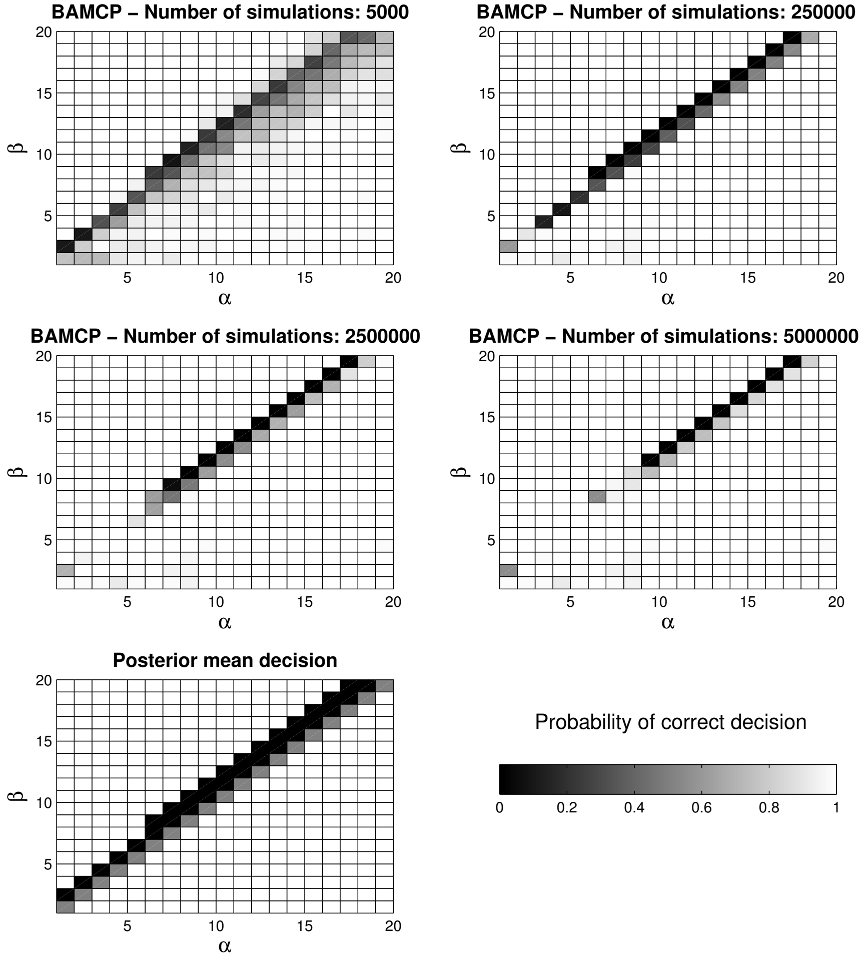
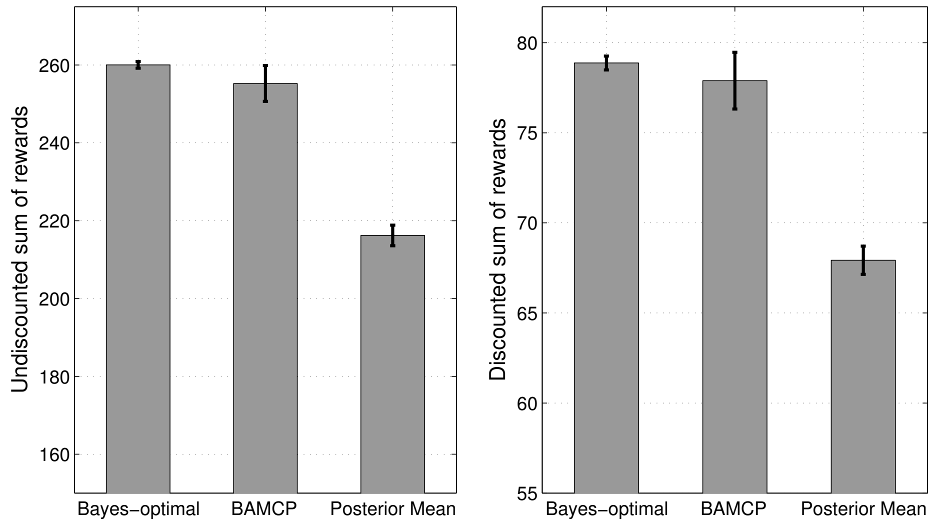


Figure S1: Evaluation of BAMCP against the Bayes-optimal policy, for the case $\gamma = 0.95$, when choosing between a deterministic arm with reward 0.5 and a stochastic arm with reward 1 with posterior probability $p \sim \text{Beta}(\alpha, \beta)$. The result is tabulated for a range of values of α, β, each cell value corresponds to the probability of making the correct decision (computed over 50 runs) when compared to the Gittins indices [14] for the corresponding posterior. The first four tables corresponds to different number of simulations for BAMCP and the last table shows the performance when acting according to the posterior mean. In this range of α, β values, the Gittins indices for the stochastic arm are larger than 0.5 (i.e., selecting the stochastic arm is optimal) for $\beta \le \alpha + 1$ but also $\beta = \alpha + 2$ for $\alpha \ge 6$. Acting according to the posterior mean is different than the Bayes-optimal decision when $\beta >= \alpha$ and the Gittins index is larger than 0.5. BAMCP is guaranteed to converges to the Bayes-optimal decision in all cases, but convergence is slow for the edge cases where the Gittins index is close to 0.5 (e.g., For $\alpha = 17, \beta = 19$, the Gittins index is 0.5044 which implies a value of $0.5044/(1-\gamma) = 10.088$ for the stochastic arm versus a value of $0.5 + \gamma \times 10.088 = 10.0836$ for the deterministic arm).



(a) (b)

Figure S2: Performance comparison of BAMCP (50000 simulations, 100 runs) against the posterior mean decision on an 8-armed Bernoulli bandit with $\gamma = 0.99$ after 300 steps. The arms' success probability are all 0.6 except for one arm which has success probability 0.9. The Bayes-optimal result is obtained from 1000 runs with the Gittins indices [14]. **a.** Mean sum of rewards after 300 steps. **b.** Mean sum of discounted rewards after 300 steps.

Inference details for the infinite 2D grid task of Section 5.2

We construct a Markov Chain using the Metropolis-Hastings algorithm to sample from the posterior distribution of row and column parameters given observed transitions, following the notation introduced in Section 5.2. Let $O = \{(i, j)\}$ be the set of observed reward locations, each associated with an observed reward $r_{ij} \in \{0, 1\}$. The proposal distribution chooses a row-column pair (i_p, j_p) from O uniformly at random, and samples $\tilde{\mathrm{p}}_{i_p} \sim \mathrm{Beta}(\alpha_1 + m_1, \beta_1 + n_1)$ and $\tilde{\mathrm{q}}_{j_p} \sim \mathrm{Beta}(\alpha_2 + m_2, \beta_2 + n_2)$, where $m_1 = \sum_{(i,j)\in O} \mathbf{1}_{i=i_p} r_{ij}$ (i.e., the sum of rewards observed on that column) and $n_1 = (1 - {}^{\beta_2}\!/\!{}_{2(\alpha_2+\beta_2)}) \sum_{(i,j)\in O} \mathbf{1}_{i=i_p}(1 - r_{ij})$, and similarly for m_2, n_2 (mutatis mutandis). The n_1 term for the proposed column parameter $\tilde{\mathrm{p}}_i$ has this rough correction term, based on the prior mean failure of the row parameters, to account for observed 0 rewards on the column due to potentially low row parameters. Since the proposal is biased with respect to the true conditional distribution (from which we cannot sample), we also prevent the proposal distribution from getting too peaked. Better proposals (e.g., taking into account the sampled row parameters) could be devised, but they would likely introduce additional computational cost and the proposal above generated large enough acceptance probabilities (generally above 0.5 for our experiments). All other parameters p_i, q_j such that i or j is present in O are kept from the last accepted samples (i.e., $\tilde{\mathrm{p}}_i = p_i$ and $\tilde{\mathrm{q}}_j = p_j$ for these is and js), and all parameters p_i, q_j that are not linked to observations are (lazily) resampled from the prior — they do not influence the acceptance probability. We denote by $Q(\mathbf{p}, \mathbf{q} \to \tilde{\mathbf{p}}, \tilde{\mathbf{q}})$ the probability of proposing the set of parameters $\tilde{\mathbf{p}}$ and $\tilde{\mathbf{q}}$ from the last accepted sample of column/row parameters $\mathbf{p}$ and $\mathbf{q}$. The acceptance probability A can then be computed as $A = \min(1, A')$ where:

$$
\begin{aligned}
A' &= \frac{P(\tilde{\mathbf{p}}, \tilde{\mathbf{q}}\,|h)Q(\tilde{\mathbf{p}}, \tilde{\mathbf{q}} \to \mathbf{p}, \mathbf{q})}{P(\mathbf{p}, \mathbf{q}\,|h)Q(\mathbf{p}, \mathbf{q} \to \tilde{\mathbf{p}}, \tilde{\mathbf{q}})} \\
&= \frac{P(\tilde{\mathbf{p}}, \tilde{\mathbf{q}})Q(\tilde{\mathbf{p}}, \tilde{\mathbf{q}} \to \mathbf{p}, \mathbf{q})P(h|\,\tilde{\mathbf{p}}, \tilde{\mathbf{q}})}{P(\mathbf{p}, \mathbf{q})Q(\mathbf{p}, \mathbf{q} \to \tilde{\mathbf{p}}, \tilde{\mathbf{q}})P(h|\,\mathbf{p}, \mathbf{q})} \\
&= \frac{p_{i_p}^{m_1}(1 - p_{i_p})^{n_1} q_{j_p}^{m_2}(1 - q_{j_p})^{n_2} \prod_{(i,j)\in O} \mathbb{1}[i = i_p \text{ or } j = j_p](\tilde{\mathrm{p}}_i\, \tilde{\mathrm{q}}_j)^{r_{ij}}(1 - \tilde{\mathrm{p}}_i\, \tilde{\mathrm{q}}_j)^{1-r_{ij}}}{\tilde{\mathrm{p}}_{i_p}^{m_1}(1 - \tilde{\mathrm{p}}_{i_p})^{n_1}\, \tilde{\mathrm{q}}_{j_p}^{m_2}(1 - \tilde{\mathrm{q}}_{j_p})^{n_2} \prod_{(i,j)\in O} \mathbb{1}[i = i_p \text{ or } j = j_p](p_i q_j)^{r_{ij}}(1 - p_i q_j)^{1-r_{ij}}}.
\end{aligned}
$$

The last accepted sampled is employed whenever a sample is rejected. Finally, reward values R_{ij} are resampled lazily based on the last accepted sample of the parameters p_i, q_j, when they have not been observed already. We omit the implicit deterministic mapping to obtain the dynamics $\mathcal{P}$ from these parameters.

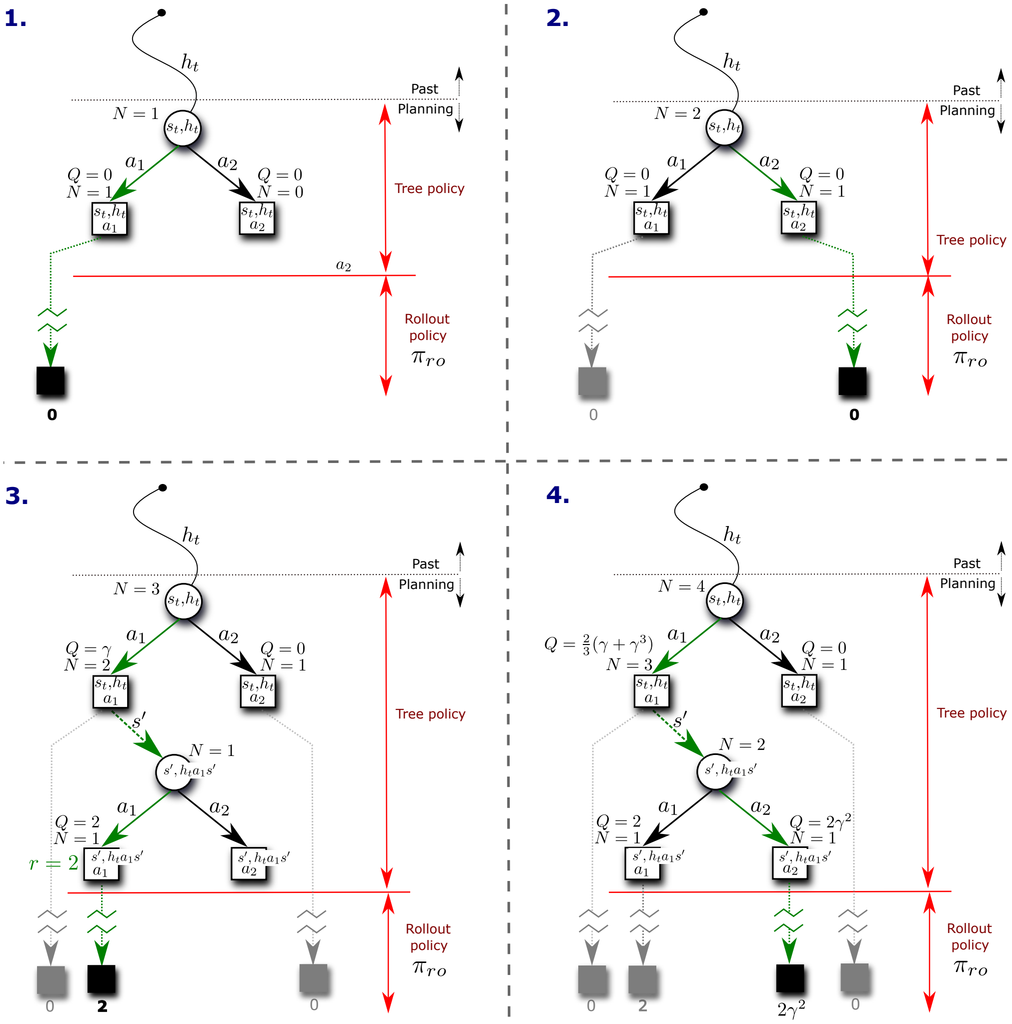


Figure S3: This diagram presents the first 4 simulations of BAMCP in an MDP with 2 actions from state $\langle s_t, h_t \rangle$. The rollout trajectories are represented with dotted lines (green for the current rollouts, and greyed out for past rollouts). **1.** The root node is expanded with two action nodes. Action a_1 is chosen at the root (random tie-breaking) and a rollout is executed in $\mathcal{P}^1$ with a resulting value estimate of 0. Counts $N(\langle s_t, h_t \rangle)$ and $N(\langle s_t, h_t \rangle, a_1)$, and value $Q(\langle s_t, h_t \rangle, a_1)$ get updated. **2.** Action a_2 is chosen at the root and a rollout is executed with value estimate 0. Counts and value get updated. **3.** Action a_1 is chosen (tie-breaking), then s' is sampled from $\mathcal{P}^3(s_t, a_1, \cdot)$. State node $\langle s', h_t a_1 s' \rangle$ gets expanded and action a_1 is selected, incurring a reward of 2, followed by a rollout. **4.** The UCB rule selects action a_1 at the top, the successor state s' is sampled from $\mathcal{P}^4(s_t, a_1, \cdot)$. Action a_2 is chosen from the internal node $\langle s', h_t a_1 s' \rangle$, followed by a rollout using $\mathcal{P}^4$ and π_{ro}. A reward of 2 is obtained after 2 steps from that tree node. Counts for the traversed nodes are updated and the MC backup updates $Q(\langle s', h_t a_1 s' \rangle, a_1)$ to $R = 0 + \gamma 0 + \gamma^2 2 + \gamma^3 0 + \cdots = \gamma^2 2$ and $Q(\langle s_t, h_t \rangle, a_1)$ to $\gamma + {}^{2\gamma^3 - \gamma}/_{3} = \frac{2}{3}(\gamma + \gamma^3)$.

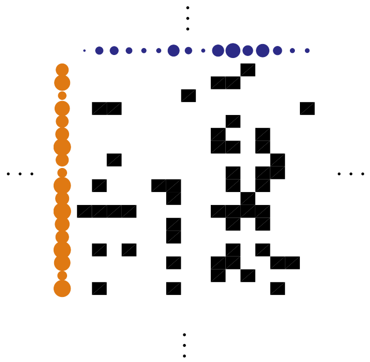


Figure S4: A portion of an infinite 2D grid task generated with Beta distribution parameters $\alpha_1 = 1, \beta_1 = 2$ (columns) and $\alpha_2 = 2, \beta_2 = 1$ (rows). Black squares at location (i,j) indicates a reward of 1, the circles represent the corresponding parameters p_i (blue) and q_j (orange) for each row and column (area of the circle is proportional to the parameter value). One way to interpret these parameters is that following column i implies a collection of $2p_i/3$ reward on average ($2/3$ is the mean of a Beta$(2, 1)$ distribution) whereas following any row j implies a collection of $q_j/3$ reward on average; but high values of parameters p_i are less likely than high values parameters q_j. These parameters are employed for the results presented in Figure 2.

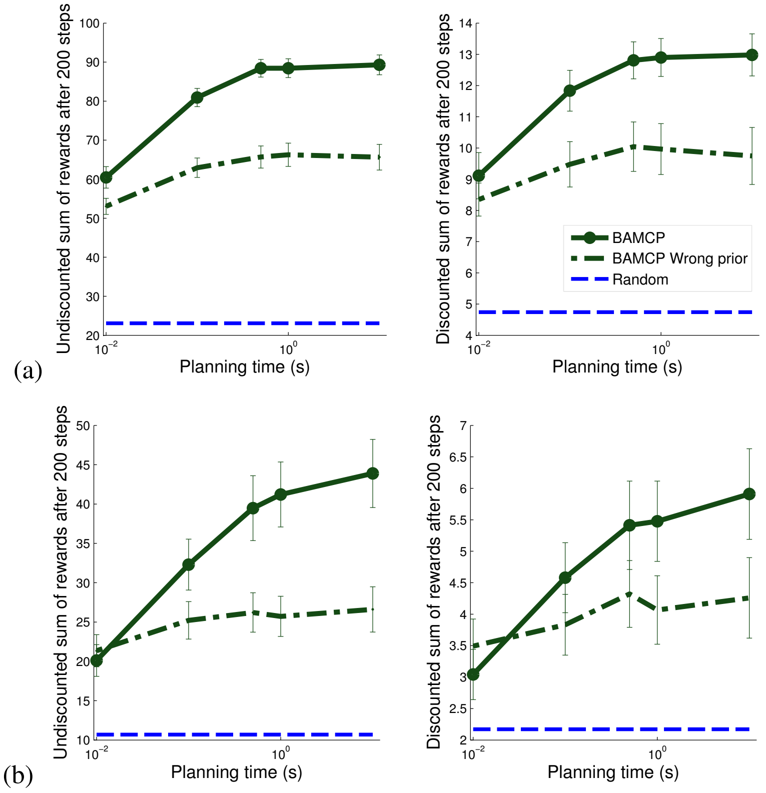


Figure S5: Performance of BAMCP on the Infinite 2D grid task of Section 5.2, for $\gamma = 0.97$, as in Figure 2 but where the grids are generated with Beta parameters **(a)** $\alpha_1 = 0.5, \beta_1 = 0.5, \alpha_2 = 0.5, \beta_2 = 0.5$ and **(b)** $\alpha_1 = 0.5, \beta_1 = 0.5, \alpha_2 = 1, \beta_2 = 3$. In the wrong prior scenario (green dotted line), BAMCP is given the parameters **(a)** $\alpha_1 = 4, \beta_1 = 1, \alpha_2 = 0.5, \beta_2 = 0.5$ and **(b)** $\alpha_1 = 1, \beta_1 = 3, \alpha_2 = 0.5, \beta_2 = 0.5$. The behavior of the agent is qualitatively different depending on the prior parameters employed (see supplementary videos). For example, for the scenario in Figure 2, rewards are often found in relatively dense blocks on the map and the agents exploits this fact when exploring; for the scenario (b) of this Figure, good rewards rates can be obtained by following the rare rows that have high q_j parameters, but finding good rows can be expensive so the agent might settle on sub-optimal rows (as in Bandit problems where the Bayes-optimal agent might settle on sub-optimal arm if it believes it likely is the best arm given past data). It should be pointed out that the actual Bayes-optimal strategy in this domain is not known — the behavior of BAMCP for finite planning time might not qualitatively match the Bayes-optimal strategy.