SUPPLEMENTARY MATERIAL

THE RECURSIVELY BLINKERED POLICY

The blinkered policy of Hay et al. (2012) was defined for problems where each computation informs the value of only one action. This assumption of "independent actions" is crucial to the efficiency of the blinkered approximation because it allows the problem to be decomposed into independent (and easily solved) subproblems for each action. However, the assumption does not hold for the Bernoulli metalevel tree because the reward at a given state affects the value of multiple policies. This is because in the context of sequential decision making, "actions" become policies, and the reward at one state affects the values of all policies visiting that state. Thus, a single computation affects the value of many policies. An intuitive generalization would be to approximate the value of a computation c_k by assuming that future computations will be limited to those that are informative about *any* of the policies the initial computation is relevant to, a set we call $\mathcal{E}_{c_k}$,. However, for large trees, this only modestly reduces the size of the initial problem. This suggests a recursive generalization: Rather than applying the blinkered approximation once and solving the resulting subproblem exactly, we recursively apply the approximation to the resulting subproblems. Finally, to ensure that the subproblems decrease in size monotonically, we remove from $\mathcal{E}_{c_k}$ the computations about rewards on the path from the agent's current state to the state s_k inspected by computation c_k and call the resulting set $\mathcal{E}'_{c_k}$. Thus, we define the *recursively blinkered policy* as $\pi^{\text{RB}}(b) = \arg\max_c Q^{\text{RB}}(b, c)$ with $Q^{\text{RB}}(b_t, \perp) = r_{\text{meta}}(b_t, \perp)$ and $Q^{\text{RB}}_{\text{meta}}(b, c) =$

$$r_{\text{meta}}(b, c) + \mathbb{E}_{B' \sim T_{\text{meta}}(b,c,\cdot)} \left[\max_{c' \in \mathcal{E}'_{c'}} Q^{\text{RB}}(B', c') \right]$$

DETAILS ON SIMULATIONS REPORTED IN SECTION 5

We found the computational cost of metareasoning for the tornado problem to be several orders of magnitude lower than realistic costs of *object*-level computations (i.e. weather simulations). Thus, the simulations leave open the question of whether BMPS can also be usefully applied when metareasoning costs are non-negligible. To answer this question, we ran additional simulations for the tornado problem with *un*realistically low values of T and t_{sim}.

The simulations summarized in Figure 1 investigated hypothetical scenarios where the metareasoning cost incurred by the BMPS policy considerably reduces the amount of object-level computation it can perform. This reduction is greatest when object-level computations are fast and the total amount of available time T is high. Nevertheless, as shown in Figure 2, BMPS still often outperforms allocating computation time uniformly. This is often true even when BMPS can perform only half as many simulations (e.g. $T = 0.03$; $k = 30$; $t_{\text{sim}} = 2^{-10}$). As expected, when the time to run a simulation is much less than the time to metareason about which simulation to run, metareasoning does not pay off anymore. Overall, we see that the benefit of metareasoning increases with the costliness of object-level reasoning and the number of computations that must be considered, but decreases with increased total computation time.

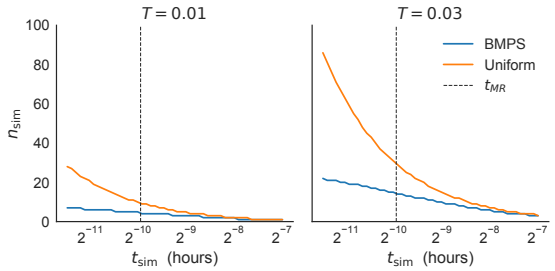

Figure 1: The number of simulations that can be run with versus without metareasoning as a function of the total time T and the cost of each simulation t_{sim}.

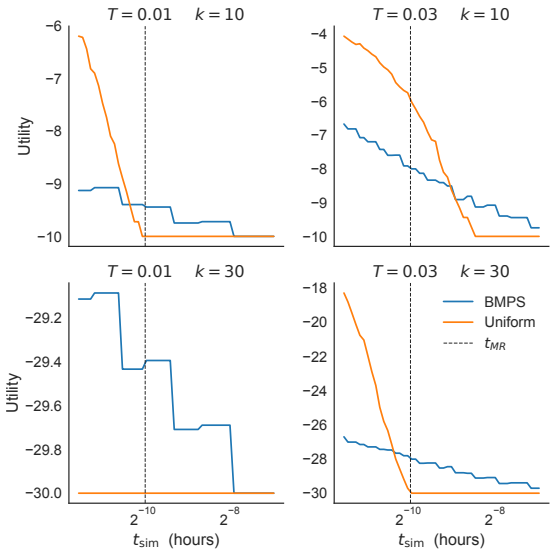

Figure 2: Utility of BMPS vs. allocating computation uniformly as a function of the total time T, the cost of each simulation t_{sim}, and the number of possible computations (i.e. the number of cities) k.

$\mathcal{M}_{\text{meta}}$	meta-level Markov Decision Process
$\mathcal{B}$	Set of possible belief states
$\mathcal{A}$	Set of meta-level actions $\mathcal{C}, \cup\{\perp\}$
$\mathcal{C}$	set of possible computations
$\perp$	meta-level action that terminates deliberation and initiates an object-level action
$r_{\text{meta}}(b, c)$	reward function of the meta-level MDP, $r_{\text{meta}}(b, c) = -\text{cost}(c) = -\lambda$ for $c \in \mathcal{C}$ and $r_{\text{meta}}(b, \perp) = \max_\pi \mathbb{E}_{\theta \sim b}[U_\pi(\theta)]$
λ	cost of a single computation
$T_{\text{meta}}(b, c, b')$	probability that performing computation c in belief state b leads to belief state b'
θ	parameters of the agent's model of the environment
π	object-level policy for selecting physical actions
$U_\pi(\theta)$	expected return of acting according to the object-level policy π if θ is the correct model of the environment
$U(b)$	expected value of terminating computation with the belief b, $r_{\text{meta}}(b, \perp)$
π_{meta}	meta-level policy for selecting computational actions
$\pi^\star_{\text{meta}}$	optimal meta-level policy, see Equation 1
$\text{VOC}(c, b)$	Value of Computation, the expected improvement in decision quality that can be achieved by performing computation c in belief state b and continuing optimally, minus the cost of the optimal sequence of computations
$\text{VOI}_1(c, b)$	myopic Value of Information, expected improvement in decision quality from taking a single computation c before terminating computation, see Equation 2
$\text{VPI}(b)$	Value of Perfect Information, the expected improvement in decision quality from attaining a maximally informed belief state beginning in belief state b, see Equation 3
$\text{VPI}_{\text{sub}}(c, b)$	value of attaining perfect information about the subset of components of θ that are most relevant to computation c, see Equation 4

Table 1: Mathematical notation