A Loss bounds for ATPO

This section provides a bound on the loss incurred by our approach. The bound can be derived from a standard compression lemma from Banerjee [2], by comparing the performance of an agent using an arbitrary constant distribution over tasks. Let p denote a distribution over $\mathcal{M}$, and define

$$L_t(p) = \sum_{k=1}^{K} p(m_k)\ell_t(\hat{\pi}_k \mid m^*),$$

where m^* is the (unknown) target task, and

$$\ell_t(\hat{\pi}_k \mid m^*) = \sum_{a^\alpha \in \mathcal{A}^\alpha} \hat{\pi}_k(a^\alpha \mid b_{k,t})\ell_t(a^\alpha \mid m^*).$$

We have the following result, the proof of which is provided in the supplementary material.

Proposition 1. *Let q denote an arbitrary stationary distribution over the tasks in $\mathcal{M}$, and $\{p_t\}$ the sequence of beliefs over tasks generated by ATPO. Then,*

$$\sum_{t=0}^{T-1} L_t(p_t) \leq \sum_{t=0}^{T-1} L_t(q) + \sqrt{\frac{2}{T}} \sum_{t=0}^{T-1} \mathrm{KL}(q \parallel p_t) + \sqrt{\frac{T}{2}} \cdot \frac{R_{\max}^2}{(1-\gamma)^2}. \tag{14}$$

Unsurprisingly—and aside from the term $\sqrt{\frac{T}{2}} \cdot \frac{R_{\max}^2}{(1-\gamma)^2}$, which is independent of q and grows sublinearly with T—the bound in (14) states that the difference between the performance of ATPO and that obtained using a constant distribution (for example, the distribution concentrated on m^*) is similar to those reported by Banerjee [2] for Bayesian online prediction with bounded loss, noting that

$$\sum_{t=0}^{T-1} \mathrm{KL}(q \parallel p_t) = \mathrm{KL}(\boldsymbol{q} \parallel \boldsymbol{p}_{0:T-1}), \tag{15}$$

where we write $\boldsymbol{q}$ and $\boldsymbol{p}_{0:T-1}$ refer to distributions over sequences in $\mathcal{M}^T$.

A.1 Proof of Proposition 1

We use the following compression lemma from Banerjee [2].

Lemma 2. *Given a set of hypothesis $\mathcal{H} = \{1, \ldots, H\}$, for any measurable function $\phi : \mathcal{H} \to \mathbb{R}$ and any distributions p and q on $\mathcal{H}$,*

$$\mathbb{E}_{h\sim q}\left[\phi(h)\right] - \log \mathbb{E}_{h\sim p}\left[\exp(\phi(h))\right] \leq \mathrm{KL}(q \parallel p). \tag{16}$$

We want to bound the loss incurred by our agent after T time steps. Before introducing our result, we require some auxiliary notation. Let m^* denote the (unknown) target task at time step t. The expected loss of our agent at time step t is given by

$$\begin{aligned} L_t(\pi_t) &= \mathbb{E}\left[\ell_t(A^\alpha \mid m^*)\right] \\ &= \sum_{a^\alpha \in \mathcal{A}^\alpha} \pi_t(a^\alpha)\ell_t(a^\alpha \mid m^*) \\ &= \sum_{k=1}^{K} p_t(m_k) \sum_{a^\alpha \in \mathcal{A}^\alpha} \hat{\pi}_k(a^\alpha \mid b_{k,t})\ell_t(a^\alpha \mid m^*) \\ &= \sum_{k=1}^{K} p_t(m_k)\ell_t(\hat{\pi}_k \mid m^*), \end{aligned}$$

where, for compactness, we wrote

$$\ell_t(\hat{\pi}_k \mid m^*) = \sum_{a^\alpha \in \mathcal{A}^\alpha} \hat{\pi}_k(a^\alpha \mid b_{k,t})\ell_t(a^\alpha \mid m^*). \tag{17}$$

Let q denote an arbitrary distribution over $\mathcal{M}$, and define

$$L_t(q) = \sum_{k=1}^{K} q(m_k)\ell_t(\hat{\pi}_k \mid m^*). \tag{18}$$

Then, setting $\phi(m_k) = -\eta\ell_t(\hat{\pi}_{k_t} \mid m^*)$, for some $\eta > 0$, and using Lemma 2, we have that

$$\mathbb{E}_{m\sim q}\left[\phi(m)\right] - \log \mathbb{E}_{m\sim p_t}\left[\exp(\phi(m))\right] \leq \mathrm{KL}(q \parallel p_t) \tag{19}$$

which is equivalent to

$$-\log \mathbb{E}_{m\sim p_t}\left[\exp(\phi(m))\right] \leq \eta L_t(q) + \mathrm{KL}(q \parallel p_t). \tag{20}$$

Noting that $-2\eta\frac{R_{\max}}{1-\gamma} \leq \phi(m) \leq 0$ and using Hoeffding's Lemma,[1] we have that

$$-\log \mathbb{E}_{m\sim p_t}\left[\exp(\phi(m))\right] \geq \eta L_t(p_t) - \frac{\eta^2 R_{\max}^2}{2(1-\gamma)^2}. \tag{22}$$

Combining (20) and (22), yields

$$L_t(p_t) \leq L_t(q) + \frac{1}{\eta}\mathrm{KL}(q \parallel p_t) + \frac{\eta R_{\max}^2}{2(1-\gamma)^2} \tag{23}$$

which, summing for all t, yields

$$\sum_{t=0}^{T-1} L_t(p_t) \leq \sum_{t=0}^{T-1} L_t(q) + \frac{1}{\eta}\sum_{t=0}^{T-1} \mathrm{KL}(q \parallel p_t) + \frac{T\eta R_{\max}^2}{2(1-\gamma)^2}. \tag{24}$$

Since η can be selected arbitrarily, setting $\eta = \sqrt{\frac{T}{2}}$ we finally get

$$\sum_{t=0}^{T-1} L_t(p_t) \leq \sum_{t=0}^{T-1} L_t(q) + \sqrt{\frac{2}{T}} \sum_{t=0}^{T-1} \mathrm{KL}(q \parallel p_t) + \sqrt{\frac{T}{2}} \cdot \frac{R_{\max}^2}{(1-\gamma)^2}. \tag{25}$$

B Domain Descriptions

We now provide detailed descriptions of our three test scenarios.

B.1 Gridworld

In the gridworld domain (Fig. 2), the ad hoc agent can move up, down, left and right, or stay in its current cell. The teammate follows the shortest path to its closest goal. Each moving action succeeds with probability $1 - \varepsilon$, for some $\varepsilon \geq 0$, except if there is a wall in the corresponding direction, in which case the position of the agent remains unchanged. When an action fails, the position of the agent remains unchanged.

[1] Hoeffding's lemma states that, given a real-valued random variable X, where $a \leq X \leq b$ almost surely,

$$\mathbb{E}\left[e^{\lambda X}\right] \leq \exp\left(\lambda\mathbb{E}\left[X\right] + \frac{\lambda^2(b-a)^2}{8}\right), \tag{21}$$

for any $\lambda \in \mathbb{R}$.

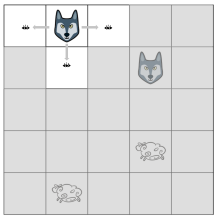

Figure 2: The gridworld domain. Two agents must each navigate to two goal cells.

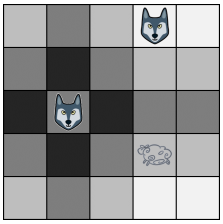

Figure 3: The pursuit domain. Two agents must capture a moving prey.

The ad hoc agent can only observe the neighboring cells. *It cannot observe its current position*. Observations are also not flawless: whenever an element (teammate, wall) is in a neighboring cell, there is a probability ε that the agent will fail to observe it.

We model each possible model $m_k \in \mathcal{M}$ as a POMDP

$$(\mathcal{X}, \mathcal{A}^\alpha, \mathcal{Z}, \{\mathbf{P}_{a^\alpha}, a^\alpha \in \mathcal{A}^\alpha\}, \{\mathbf{O}_{a^\alpha}, a^\alpha \in \mathcal{A}^\alpha\}, r_k, \gamma).$$

A state $x \in \mathcal{X}$ contains information regarding the positions of both agents, and is therefore represented as a tuple $x = (c_1, r_1, c_2, r_2)$, where (c_n, r_n) represents the cell (column and row) where agent n is located.

Each observation $z \in \mathcal{Z}$ are also represented by a tuple $z = (\hat{u}, \hat{d}, \hat{l}, \hat{r})$, where each entry represents what is observed, respectively, above, below, to the left, and to the right of the agent. For each entry there are three possible values: *Nothing*, *Teammate*, *Wall*. The action space $\mathcal{A}^\alpha$ contains five possible actions, *Up, Down, Left, Right* and *Stay*. The transition probabilities $\{\mathbf{P}_{a^\alpha}, a^\alpha \in \mathcal{A}^\alpha\}$ map a state x and action a^α to every possible next state x', taking into account the probability ϵ of the agent failing to move (considering that the teammate actions always succeed and move the teammate towards its closest goal cell). Similarly, the observation probabilities $\{\mathbf{O}^\alpha_a, a^\alpha \in \mathcal{A}^\alpha\}$ map a state x and previous action a^α to every an observation z, taking into account the probability ϵ of the agent failing to observe any given element. The reward function r_k assigns the reward of -1 for all time steps except those where both destination have been reached, in which case it assigns a reward of 100 (and an absorbent state assigned a reward of 0 afterwards). In other words, since where dealing with an horizon of 50 steps, we do not reset the environment whenever the goals have been reached. Finally, we consider a discount factor $\gamma = 0.95$.

B.2 Pursuit

In the pursuit domain (Fig. 3), the ad hoc agent is able to observe the elements located in its neighborhood. Specifically, if either the teammate or the prey are located in one of the neighbouring nine cells surrounding the agent, it may be able to observe them with probability $1-\epsilon$, where ϵ represents the probability of failing the observation.

We model each possible model $m_k \in \mathcal{M}$ as a POMDP

$$(\mathcal{X}, \mathcal{A}^\alpha, \mathcal{Z}, \{\mathbf{P}_{k,a^\alpha}, a^\alpha \in \mathcal{A}^\alpha\}, \{\mathbf{O}_{a^\alpha}, a^\alpha \in \mathcal{A}^\alpha\}, r_k, \gamma).$$

Different prey capture configurations vary the reward function r and different teammate policies vary the transition probabilities $\mathbf{P}$. We consider four different capture combinations: (1) *north of prey* + *south of prey*; (2) *west of prey* + *east of prey*; (3) *southwest of prey* + *northeast of prey*; and (4) *northwest of prey* + *southeast of prey*. The capture positions may be used interchangeably between the agents (i.e., either the ad hoc agent or the teammate may occupy each of the two required positions). We also consider two possible teammate policies: (1) greedy policy; and (2) teammate aware policy. The greedy policy follows the shortest path to its closest capture position, not taking into account the position of the ad hoc agent and therefore not acting as efficiently as possible, while the teammate aware policy runs an A search taking into account the position of the teammate. Each state $x \in \mathcal{X}$ contain information regarding the relative distances to the teammate and prey and is therefore represented by a tuple $x = (d^{a_1}{}_x, d^{a_1}{}_y, d^p{}_x, d^p{}_y)$, where $d^{a_1}{}_x, d^{a_1}{}_y$ represents the relative distance (in units) to the teammate and $d^p{}_x, d^p{}_y$ represents the relative distance (in units) to the prey. Each observation $z \in \mathcal{Z}$ is also represented as a tuple $z = (\hat{a_1}, \hat{p})$, where $\hat{a_1}$ represents an observation cell identifier of the teammate and $\hat{p}$ represents an observation cell identifier of the prey. A cell identifier is a value between $0 and 8$ which represents the nine surrounding cells of the agent. When an observation is failed or the teammate/prey is not in the nine surrounding cells, the ad hoc agent sees the other agent as it were standing in its own position. The action space $\mathcal{A}^\alpha$ contains five possible actions, *Up, Down, Left, Right, Stay*. For each teammate type k, the transition probabilities $\{\mathbf{P}_{k,a^\alpha}, a^\alpha \in \mathcal{A}^\alpha\}$ map a state x and action a to every possible next state x', taking into account the probability of the teammate executing each possible action on x given their policy for task k. Similarly, the observation probabilities $\{\mathbf{O}^\alpha_a, a^\alpha \in \mathcal{A}\}$ map a state x and previous action a^α to every possible observation z, taking into account the probability $\epsilon(d)$ of the agent failing to observe the position of the other agents. The reward function r_k assigns the reward of -1 for all time steps except those where the prey has been cornered, in which case it assigns a reward of 100 for the one the prey was cornered in and 0 afterwards (in other words, since where dealing with a finite horizon, we do not reset the environment whenever the prey have been cornered). Finally, we consider a discount factor $\gamma = 0.95$.

B.3 Abandoned Power Plant

We model each possible model $m_k \in \mathcal{M}$ of the abandoned power plant domain as a POMDP

$$(\mathcal{X}_k, \mathcal{A}^\alpha, \mathcal{Z}, \{\mathbf{P}_{a^\alpha}, a^\alpha \in \mathcal{A}^\alpha\}, \{\mathbf{O}_{a^\alpha}, a^\alpha \in \mathcal{A}^\alpha\}, r_k, \gamma).$$

POMDPs from both tasks vary in both state space $|\mathcal{X}|$ and reward function r. In POMDPs from the first task—exploration—each state $x \in \mathcal{X}$ is represented as a tuple $x = (\text{room}_{\text{robot}}, \text{room}_{\text{human}}, e_1, e_2, e_3)$ where $\text{room}_{\text{robot}}$ and $\text{room}_{\text{human}}$ represent the identifier of the room in which the robot and human are located (respectively) and e_i represents the status of room i (explored, unexplored). In POMDPs from the second task—cleanup—each state $x \in \mathcal{X}$ is represented as a tuple $x = (\text{room}_{\text{robot}}, \text{room}_{\text{human}}, d_1, d_2)$, where $\text{room}_{\text{robot}}$ and $\text{room}_{\text{human}}$ represent the identifier of the room in which the robot and human are located (respectively) and d_i represent the status of the room i (dirty,

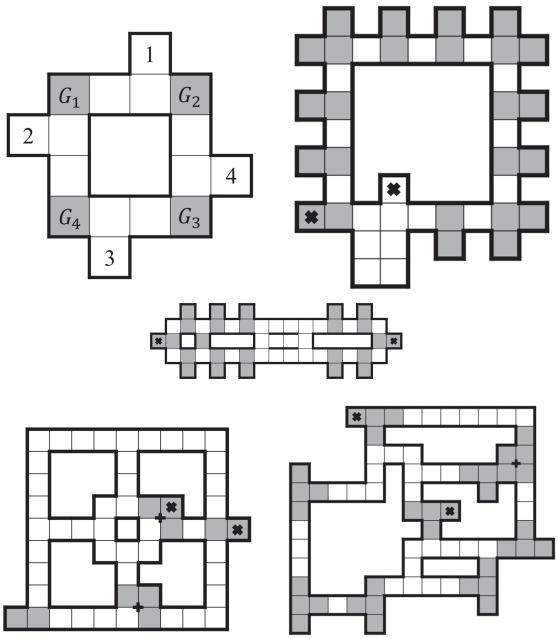

Figure 4: The ntu, isr, mit, pentagon and cit domains. Two agents must navigate in close quarters in order to each reach an exit (while taking into account collisions).

clean). The power plant is modeled as a topological graph with six nodes representing the rooms. The edges between nodes represent the connections between the rooms, and therefore, the action space for both the robot and the human contain four base actions—*Move to lowest-index node, Move to second lowest-index node, Move to third lowest-index node and Stay*. The robot has to itself available two additional actions—*Query human for human location and Query human for own location*. The observation $z \in \mathcal{Z}$ is represented by a a value. Whenever the robot moves (instead of querying the human), it is able to observe, with a probability of failure ϵ, its own location, with z representing the id of the room it is in (and -1 when failing). Whenever the robot queries the human for the human's or its own location, the human may or may not reply, also with probability ϵ. If the query is successful, z contains the room of the human or the room of the robot (respectively according to the query type). If the query fails, $z = -1$. Finally, we consider a discount factor γ of 0.95 and the reward function r assigns a value of -1 the one where, in the case of exploration tasks, all unexplored rooms have been explored and, in the case of the cleanup tasks, all dirty rooms have been cleaned. Afterwards, an absorbent state is assigned a reward of 0.

B.4 NTU, ISR, MIT, PENTAGON and CIT

We model each possible model $m_k \in \mathcal{M}$ of the ntu, isr, mit, pentagon and cit domains (Fig. 4) as a POMDP

$$(\mathcal{X}, \mathcal{A}^\alpha, \mathcal{Z}, \{\mathbf{P}_{a^\alpha}, a^\alpha \in \mathcal{A}^\alpha\}, \{\mathbf{O}_{a^\alpha}, a^\alpha \in \mathcal{A}^\alpha\}, r_k, \gamma).$$

Different tasks have different goal locations, which translate to different reward functions r. A state $x \in \mathcal{X}$ contains information regarding the positions of both agents, and is therefore represented as a tuple $x = (c_1, r_1, c_2, r_2)$, where (c_n, r_n) represents the cell (column and row) where agent n is located.

Each observation $z \in \mathcal{Z}$ are also represented by a tuple $z = (\hat{u}, \hat{d}, \hat{l}, \hat{r})$, where each entry represents what is observed, respectively, above, below, to the left, and to the right of the agent. For each entry there are three possible values: *Nothing*, *Teammate*, *Wall*. The action space $\mathcal{A}^\alpha$ contains five possible actions, *Up, Down, Left, Right* and *Stay*. The transition probabilities $\{\mathbf{P}_{a^\alpha}, a^\alpha \in \mathcal{A}^\alpha\}$ map a state x and action a^α to every possible next state x', taking into account the probability ϵ of the agent failing to move (considering that the teammate actions always succeed and move the teammate towards its closest goal cell). Similarly, the observation probabilities $\{\mathbf{O}_a^\alpha, a^\alpha \in \mathcal{A}^\alpha\}$ map a state x and previous action a^α to every an observation z, taking into account the probability ϵ of the agent failing to observe any given element. The reward function r_k assigns the reward of -1 for all time steps except those where both destination have been reached, in which case it assigns a reward of 100 (and an absorbent state assigned a reward of 0 afterwards). In other words, since where dealing with an horizon of 50 steps, we do not reset the environment whenever the goals have been reached. Finally, we consider a discount factor $\gamma = 0.95$.

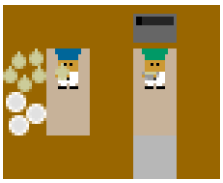

Figure 5: The overcooked domain. Two agents, a cook and an assistant, are required to deliver soups as fast as possible.

B.5 Overcooked

We model each possible model $m_k \in \mathcal{M}$ of the Overcooked domain (Fig. 5) as a POMDP

$$(\mathcal{X}, \mathcal{A}^\alpha, \mathcal{Z}, \{\mathbf{P}_{k,a^\alpha}, a^\alpha \in \mathcal{A}^\alpha\}, \{\mathbf{O}_{a^\alpha}, a^\alpha \in \mathcal{A}^\alpha\}, r, \gamma).$$

Varying the teammates varies the transition probabilities. Since the state is fully observable in this domain, observations correspond to the states themselves. Since the policy of the teammate may vary, each different model has distinct transition probabilities. We consider four different cook teammate policies: (1) a teammate which has the optimal policy; (2) a teammate which acts randomly; (3) a teammate which has the optimal policy but always stands near the bottom balcony when receiving ingredients and plates; and (4) a teammate which has the optimal policy but always stands near the upper balcony when receiving ingredients and plates. Each state $x \in \mathcal{X}$ is represented as a tuple $x = (p_a, p_c, h_a, h_c, tb, bb, s)$, where, respectively, (p_a, p_c) represents the cell (top or bottom) and (h_a, h_c) represents the objects (nothing, onion, plate or soup) in hand of the helper and the cook. (tb, bb) represents the objects on the top kitchen counter balcony and bottom kitchen counter balcony (respectively). Finally, s represents the contents of the soup pan (empty, one onion, two onions, cooked soup). The action space $\mathcal{A}$ contains four possible actions, *Up, Down, Noop* and *Act*. The transition probabilities $\{\mathbf{P}_{k,a^\alpha}, a^\alpha \in \mathcal{A}^\alpha\}$ map a state x and action a^α to every possible next state x', taking into account the probability of the teammate executing each action on x. The reward function r assigns the reward of 100 to states x where the cook delivers a cooked soup through the kitchen window, and -1 otherwise. After the soup is delivered, the task is considered complete, leading to an absorbent state which is assigned a reward of 0. We consider a discount factor $\gamma = 0.95$.

Table 5: Hyperparameters used for the Perseus [25] algorithm. The discount factor was also used for the Value Iteration algorithm.

Environment	Horizon	Beliefs	Tolerance	Discount Factor
gridworld	50	5000	0.01	0.95
pursuit-task	75	5000	0.01	0.95
pursuit-teammate	85	5000	0.01	0.95
pursuit-both	85	5000	0.01	0.95
abandoned power plant	50	2500	0.01	0.95
ntu	75	5000	0.01	0.95
overcooked	50	1800	0.01	0.95
isr	75	5000	0.01	0.95
mit	75	5000	0.01	0.95
pentagon	75	5000	0.01	0.95
cit	85	8000	0.01	0.95

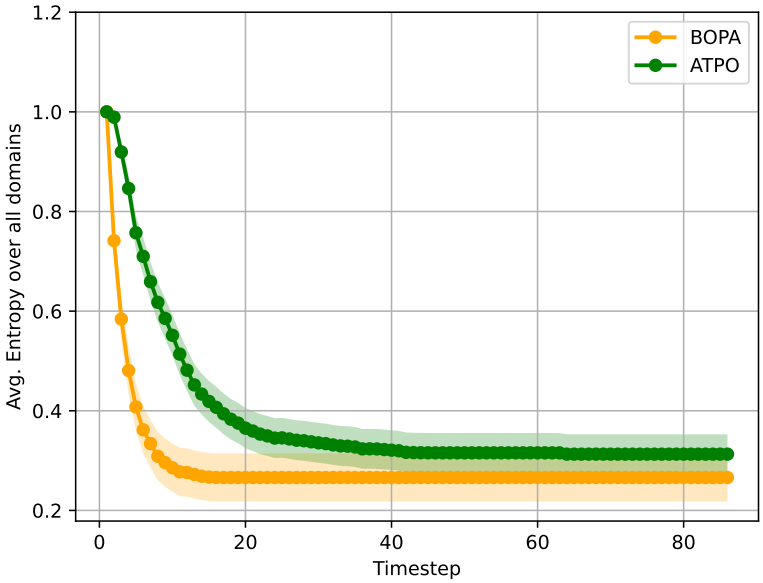

Figure 6: Average belief entropy over all environments for ATPO and BOPA. Both Bayesian inference agents are successfully able to identify the correct model as they interact with the environment. Compared to BOPA, ATPO, on average, requires extra timesteps to identify the correct model. This is an expected result, given that BOPA is able to fully observe the environment and therefore has less uncertainty when performing the inference.