Learning Who to Trust: Policy Learning in Single-Stage Decision Problems with Unreliable Expert Advice - Supplementary Material

Anonymous Author(s)
Affiliation
Affiliation Line 2
email@email.com

1 Derivations

1.1 Derivation of Decision-Making Rule

Let a^* denote the optimal action for state s_t and $v_t^{(e)}$ denote the advice utterance given by expert e for s_t, with V_t denoting the set $\{v_t^{(e)} | e \in E_t\}$, where E_t is the set of all experts who have offered advice for s_t at some in trials $[0, ..., t-1]$. Our aim is to calculate $P(a_j = a^* | V_t)$ for each $a_j \in A$. By Bayes' rule,

$$P(a_j = a^* | V_t) = \frac{P(V_t | a_j = a^*) P(a_j = a^*)}{\sum_{k=0}^{|A|} P(V_t | a_k = a^*) P(a_k = a^*)}. \tag{1}$$

Under the assumption that each expert gives advice independently of every other expert, Equation 1 can be expressed as

$$P(a_j = a^* | V_t) = \frac{\prod_{e \in E_t} P(v_t^{(e)} | a_j = a^*) P(a_j = a^*)}{\sum_{k=0}^{|A|} \prod_{e \in E_t} P(v_t^{(e)} | a_k = a^*) P(a_k = a^*)}, \tag{2}$$

which, under the assumption that the prior probability $P(a = a^*)$ is equal for all $a \in A$, reduces to

$$P(a_j = a^* | V_t) = \frac{\prod_{e \in E_t} P(v_t^{(e)} | a_j = a^*)}{\sum_{k=0}^{|A|} \prod_{e \in E_t} P(v_t^{(e)} | a_k = a^*)}. \tag{3}$$

1.2 Derivation of Reliability Estimate Update Rule

For each expert $e \in E$, let $a^{(e)}$ denote the action advised by e for state s_t in trial t. Let $n^{(e)}$ denote the total number of times e has offered advice in trials $[0, ..., t]$, $x^{(e)}$ denote the number of times the agent has judged the expert's advice to be correct, and $\rho^{(e)}$ denote the reliability of e. For ease of readability, we omit the superscript denoting expert e. We wish to calculate $P(\rho | x)$, which, by Bayes' rule,

$$P(\rho | x) = \frac{P(x | \rho) P(\rho)}{\int_0^1 P(x | \rho) P(\rho) d\rho}. \tag{4}$$

We model the likelihood $P(x | \rho)$ as a binomial distribution,

$$P(x | \rho) = B_x[n, \rho] = \binom{n}{x} \rho^x (1 - \rho)^{n-x}, \tag{5}$$

and thus the prior $P(\rho)$ is modelled as a beta distribution,

$$P(\rho) = Beta_\rho[\alpha_0, \beta_0] = \frac{\Gamma(\alpha_0 + \beta_0)}{\Gamma(\alpha_0)\Gamma(\beta_0)} \rho^{\alpha_0 - 1} (1 - \rho)^{\beta_0 - 1}, \tag{6}$$

as a beta distribution is conjugate to a binomial distribution (Etz 2018). Substituting Equations 5 and 6 into Equation 4, we arrive at

$$P(\rho | x) = \frac{B_x[n, \rho] Beta_\rho[\alpha_0, \beta_0]}{\int_0^1 B_x[n, \rho] Beta_\rho[\alpha_0, \beta_0] d\rho} \tag{7}$$

$$= \frac{\binom{n}{x} \rho^x (1 - \rho)^{n-x} \frac{\Gamma(\alpha_0 + \beta_0)}{\Gamma(\alpha_0)\Gamma(\beta_0)} \rho^{\alpha_0 - 1} (1 - \rho)^{\beta_0 - 1}}{\int_0^1 \binom{n}{x} \rho^x (1 - \rho)^{n-x} \frac{\Gamma(\alpha_0 + \beta_0)}{\Gamma(\alpha_0)\Gamma(\beta_0)} \rho^{\alpha_0 - 1} (1 - \rho)^{\beta_0 - 1} d\rho} \tag{8}$$

$$= \frac{\rho^{x + \alpha_0 - 1} (1 - \rho)^{n - x + \beta_0 - 1}}{\int_0^1 \rho^{x + \alpha_0 - 1} (1 - \rho)^{n - x + \beta_0 - 1} d\rho}. \tag{9}$$

Now, because a beta distribution is a valid probability distribution,

$$1 = \int_0^1 Beta_\rho[\alpha, \beta] d\rho \tag{10}$$

$$= \int_0^1 \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \rho^{\alpha - 1} (1 - \rho)^{\beta - 1} d\rho, \tag{11}$$

and thus

$$\int_0^1 \rho^{\alpha - 1} (1 - \rho)^{\beta - 1} d\rho = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha + \beta)}. \tag{12}$$

Substituting Equation 12 into Equation 9,

$$P(\rho | x) = \frac{\rho^{x + \alpha_0 - 1} (1 - \rho)^{n - x + \beta_0 - 1}}{\frac{\Gamma(x + \alpha_0)\Gamma(n - x + \beta_0)}{\Gamma(x + \alpha_0 + n - x + \beta_0)}} \tag{13}$$

$$= \frac{\Gamma(n + \alpha_0 + \beta_0)}{\Gamma(x + \alpha_0)\Gamma(n - x + \beta_0)} \rho^{x + \alpha_0 - 1} (1 - \rho)^{n - x + \beta_0 - 1} \tag{14}$$

$$= Beta_\rho[x + \alpha_0, n - x + \beta_0] \tag{15}$$

2 Pseudocode

Algorithm 1 outlines the standard process for solving SSDPs without any external information, as described in Section 2.1. of the paper. This can be compared to Algorithm 2, which outlines the CLUE process for solving SSDPs with the advice of multiple experts.

Algorithm 3 outlines the decision-making and learning procedures of an action-value ϵ-greedy agent which does not incorporate any expert advice, as described in Section 2.1. of the paper.

Algorithm 4 outlines the process by which a CLUE agent selects an action in trial t for state s_t, given the threshold

Algorithm 1: Single Stage Decision Problem

```
procedure STANDARD_SSDP($environment, N$) ▷ $N =$ number of trials
    for $t \in [0, ..., N-1]$ do
        $s_t \leftarrow$sample_state($environment$)
        $a_t \leftarrow$select_action($s_t$) ▷ agent acts (e.g. Alg. 3)
        $r_t \leftarrow$execute_action($a_t, environment$)
        learn($s_t, a_t, r_t$) ▷ agent learns (e.g. Alg. 3)
```

Algorithm 2: Cautiously Learning with Unreliable Experts

```
procedure CLUE($environment, E, N$) ▷ $E =$ panel of experts, $N =$ number of trials
    for $t \in [0, ..., N-1]$ do
        $s_t \leftarrow$sample_state($environment$)
        $a_t \leftarrow$select_action($s_t$) ▷ Alg. 4
        $r_t \leftarrow$execute_action($a_t, environment$)
        $advice_t \leftarrow$advise($E, s_t, a_t, r_t$) ▷ e.g. Alg 6
        learn($s_t, a_t, r_t$) ▷ e.g. Alg. 3
        update_estimates($s_t, a_t, r_t, advice_t$) ▷ Alg. 5
```

parameter T, exploration parameter ϵ, and the reliability estimate $\mathbb{E}[\rho^{(e)}]$ of each expert e in E.

Algorithm 5 outlines the process by which a CLUE agent updates the beta distributions modelling the reliability of each expert, where $v_t^{(e)}$ denotes the advice given by expert e in trial t, and V_t denotes the set of all advice given in trial t.

Algorithm 6 outlines the process by which each expert $e \in E$ decides whether or not to offer advice to the agent and whether or not the advice is optimal, as discussed in Section 4.1. in the paper.

Algorithm 3: Baseline Approach for Solving SSDPs

```
procedure ACT($s_t$,$t$,$\epsilon$)
    $p \leftarrow$random()
    if $p < \epsilon$ then
        return random $a \in A$ ▷ agent "explores"
    else
        return $\arg\max_a Q(s_t, a)$ ▷ agent "exploits"
procedure LEARN($s_t$,$a_t$,$r_t$,$t$)
    $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_t - Q(s_t, a_t))$
```

Algorithm 4: Acting with Advice from a Panel of Potentially Unreliable Experts

```
procedure ACT_WITH_ADVICE($s_t$,$t$,$T$,$\epsilon$,$\mathbb{E}[\rho^{(e)}]$)
    $p \leftarrow$random()
    if $p < \epsilon$ then
        $E_t \leftarrow \{e | e$ advised for $s_t$ in $\tau \in [0, ..., t-1]\}$
        if $|E_t| = \varnothing$ then
            return random $a \in A$
        else
            for $a \in A$ do
                $L_a \leftarrow 0$
                for $e \in E_t$ do
                    if expert $e$ advised $(s_t, a)$ then
                        $L_a \leftarrow L_a \times \mathbb{E}[\rho^{(e)}]$
                    else
                        $L_a \leftarrow L_a \times \frac{1-\mathbb{E}[\rho^{(e)}]}{|A|-1}$
        for $a \in A$ do
            $P(a = a_*) \leftarrow \frac{L_a}{\sum_{i=0}^{|A|} L_{a_i}}$
        $a_{best} \leftarrow \underset{a}{\arg\max} P(a = a_*)$
        if $P(a_{best} = a_*) < T$ then
            return random $a \in A$
        else
            $q \leftarrow$random()
            if $q < P(a_{best} = a_*)$ then
                return $a_{best}$
            else
                return random $a \in A$
    else
        return $\arg\max_a Q(s_t, a)$
```

3 Additional Experiments

3.1 Initial Reliability Estimates

In this set of experiments, we investigate the effect of varying the α_0 and β_0 parameters which determine the prior reliability distribution (see Section 3.4 of the main paper). Recall that α_0 and β_0 can be thought of as prior counts of the expert giving incorrect and correct advice respectively. Thus $\alpha_0 > \beta_0$ biases ρ towards 0, and $\alpha_0 < \beta_0$ towards 1, with $\alpha_0 + \beta_0$ determining the strength of that prior against trial data. To measure their effect, we plot the difference between the average total regret of the Baseline Agent (Algorithm 3) and the average total regret of CLUE (Algorithm 2) for a number of different α_0 and β_0 values, summed over $10,000$ trials and averaged over 100 runs for an environment with $|V_S| = 7$ and $|V_A| = 3$, where the total regret is equal to

$$R_{Agent} = \sum_{0 \leq t < N} r(s_t, \pi^*(s_t)) - r(s_t, a_t), \tag{16}$$

and the difference in regret is

$$R_{CLUE} - R_{Baseline}. \tag{17}$$

Therefore, a value of 0 indicates performance equal to the Baseline Agent. Minimising regret is equivalent to maximising reward, and thus a lower difference in regret indicates better performance. This process was repeated for each of the panels described in Section 4.2 of the main paper, and plotted in Figure 1. The average total regret obtained by the Baseline Agent was 3505.0, and the average total regret obtained by NAF was 411.1 for the single reliable expert (difference: -3093.9), 9449.7 for the single unreliable expert (difference: 5944.7), and 4994.5 for the varied panel (difference: 1489.5).

For the single reliable expert, the best performance occurs when the parameters heavily bias the estimate towards 1. However, all tested values result in improved performance over the Baseline Agent, and the gain in performance between $\alpha_0 = 1, \beta_0 = 1$ and $\alpha_0 = 1, \beta_0 = 1000$ is minimal.

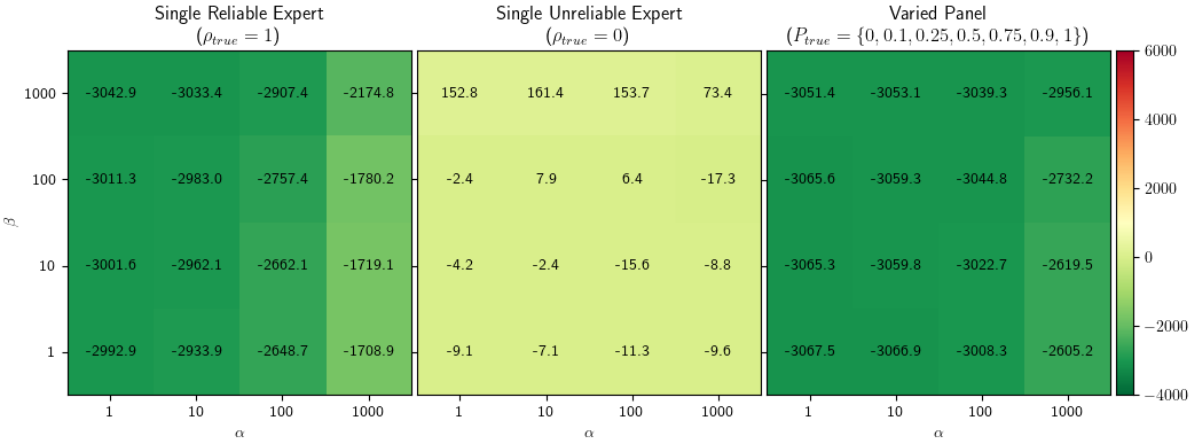

Figure 1: The total difference in regret between the Baseline Agent and CLUE for each value of α_0 and β_0 and for each panel. Lower values indicate better performance of CLUE.

Algorithm 5: Updating Unreliability Estimates

```
procedure UPDATE_ESTIMATES($s_t$,$a_t$,$r_t$,$V_t$)
    $E'_t \leftarrow \{e|e$ advised for $s_t$ in $\tau \in [0, ..., t]\}$
    for $e \in E$ do
        if $e$ advised some action in trial $t$ then
            store_advice($e$,$s_t$,$v_t^{(e)}$)
        if $e \in E'_t$ then
            $best\_reward \leftarrow \max_a EU(s_t, a)$
            $advice\_reward \leftarrow EU(s_t, v_t^{(e)})$
            if $advice\_reward \geq best\_reward$ then
                $\alpha_{t+1}^{(e)} \leftarrow \alpha_t^{(e)} + 1$
                $\beta_{t+1}^{(e)} \leftarrow \beta_t^{(e)}$
            else
                $\alpha_{t+1}^{(e)} \leftarrow \alpha_t^{(e)}$
                $\beta_{t+1}^{(e)} \leftarrow \beta_t^{(e)} + 1$
            $\mathbb{E}[\rho_{t+1}^{(e)}] \leftarrow \frac{\alpha_{t+1}^{(e)}}{\alpha_{t+1}^{(e)}+\beta_{t+1}^{(e)}}$
```

Results for the single unreliable expert are less varied, with performance close to the Baseline Agent for all tested values. Performance is only degraded when β_0 is large.

The best performance for the varied panel occurs when both parameters are low, as the variety in ρ_{true} means that no single strong prior can bias the reliability distributions correctly for all experts at a time. Across all panels, the performance with a relatively uninformative prior is close to, if not equal to, the best performance, making it a reasonable choice in the absence of strong belief about an expert's reliability. These results also demonstrate that CLUE is robust to the choice of prior, except where $\alpha_0 + \beta_0$ approaches the order of magnitude of the total number of trials.

3.2 Expert Parameters

In this set of experiments, we investigate the effect of varying the number of interactions between the agent and each expert. Recall from Section 4.1 of the main paper that the number of interactions is determined by the values of μ and γ, with lower values of both resulting in more interactions and higher values of both resulting in fewer interactions. Similar to the previous section, we plot the difference in average total regret between the Baseline Agent and CLUE, and between the Baseline Agent and NAF, for varying values of μ and γ, as depicted in Figure 2. The environment and number of trials and runs is identical to Section 3.1, and $\alpha_0 = 1$, $\beta_0 = 1$, as in Section 4.2 of the main paper.

Algorithm 6: Expert Advice Process

```
procedure ADVISE($s_t, a_t, r_t, E$)
    $advice \leftarrow []$
    for $e \in E$ do
        $a_t^* \leftarrow$get_optimal_action($s_t$)
        $t' \leftarrow$last_advice_trial()
        if $t - t' \geq \mu^{(e)}$ then
            if $\sum_{t'<i\leq t} \frac{EU(s_t,a_i^*)-EU(s_t,a_i)}{t-t'} \geq \gamma^{(e)}$ then
                $p \leftarrow$random()
                if $p < \rho_{true}^{(e)}$ then
                    $advice[e] \leftarrow a_t^*$
                else
                    $advice[e] \leftarrow$random $a \in A \setminus \{a_t^*\}$
    return $advice$
```

With NAF, more advice results in better performance when the advice is always correct, but worse performance when the advice is sometimes incorrect. CLUE on the other hand is robust to the presence of incorrect advice; more correct advice results in better performance, but more incorrect advice has no adverse effect on performance.

3.3 Adversarial Advice

To illustrate how CLUE can benefit from adversarial advice (advice from an expert that is consistently wrong), we compare the average reward obtained by the agent advised by a

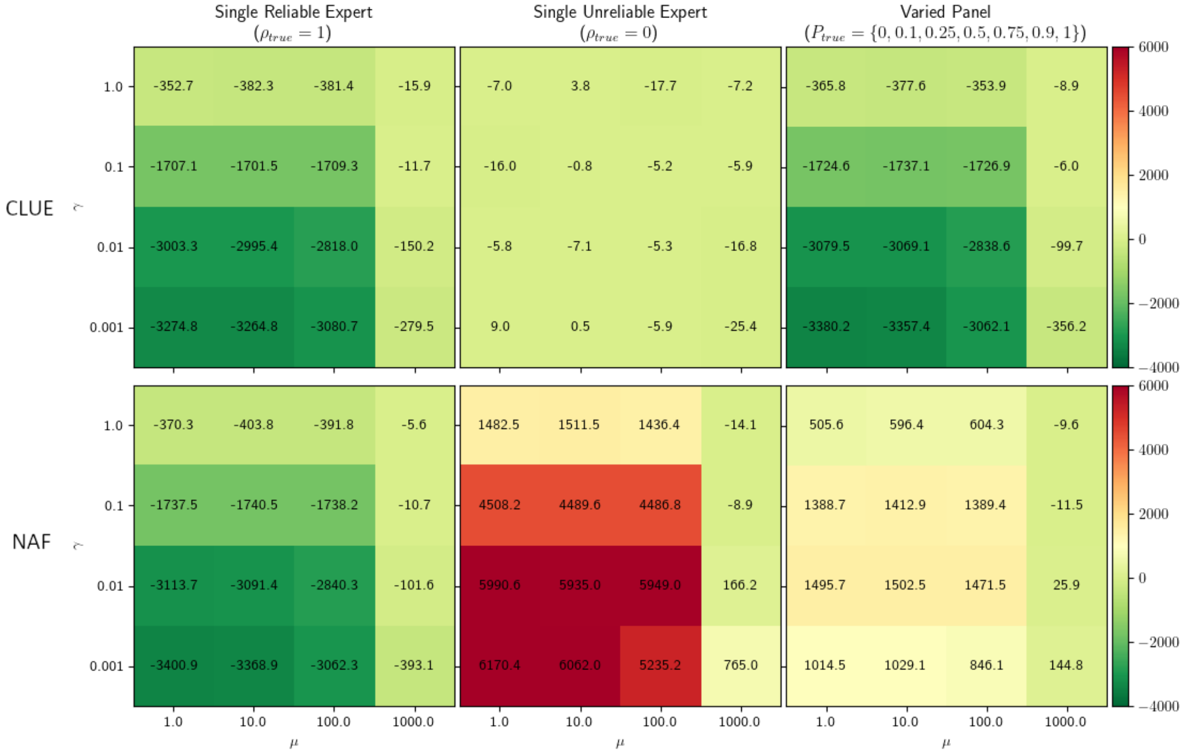

Figure 2: The total difference in regret between the Baseline Agent and CLUE and Baseline Agent and NAF for each value of μ and γ and for each panel. Lower values indicate better performance.

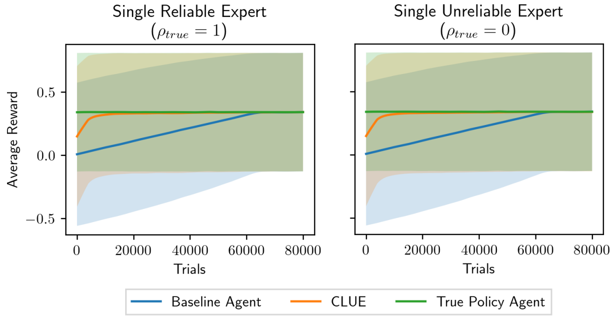

Figure 3: A comparison of the performance of CLUE with a single reliable expert ($\rho_{true} = 1$) and a single unreliable expert ($\rho_{true} = 0$) for an environment where $|A| = 2$.

single reliable expert ($\rho_{true} = 1$) and a single unreliable expert ($\rho_{true} = 0$), in an environment where $|V_S| = 10$ and $|V_A| = 1$, and thus $|A| = 2$, averaged over 100 runs. Results are plotted in Figure 3.

For both experts, CLUE shows a similar improvement in performance over the Baseline Agent. This improvement is possible in the case of the single unreliable expert because, having estimated the reliability of the expert to be low, the suboptimal action advised by the expert is deemed to have a low probability of being optimal, thus improving the probability of the other action being optimal.

3.4 Comparison with Probabilistic Policy Reuse

In this set of experiments, we compare the average reward per trial obtained by a number of agents in an environment with $|V_S| = 7$, $|A| = 3$, averaged over 100 runs. The panels compared include the three described in Section 4.2 of the main paper, as well as a *Single Random Expert* ($\rho_{true} = 0.5$). In addition to the agents described in Section 4.1 of the main paper, we compare adaptations of the π-reuse and PRQ algorithms of Fernández and Veloso (2006), hereafter referred to as the *Decayed Reliance Agent* and *PRQ Agent* respectively.

The Decayed Reliance Agent maintains a parameter $\psi \in [0, 1]$ that decays over the course of learning. With probability ψ, the agent randomly follows advice it has received from the experts, in the same fashion as NAF. Otherwise, the agent acts unassisted, in the same fashion as the Baseline Agent. In these experiments, ψ decays from 1 to 0 across the first 80% of trials.

The PRQ Agent maintains a list of policies $L = \{\Pi^{(e)} | \forall e \in E\}$ which retain the advice it has received from each expert, as well as its own learned policy $\Pi^{(0)}$. Every time it selects an action, it follows a policy with probability

$$P(\Pi^{(j)}) = \frac{e^{\tau^{W^{(j)}}}}{\sum_{p=0}^{|E|} e^{\tau^{W^{(p)}}}}, \tag{18}$$

where τ is a temperature parameter and $W^{(j)}$ is the average reward obtained by following expert j, with $W^{(0)}$ denoting the average reward obtained by following the agent's policy. Every trial, the temperature parameter is incremented by $\Delta\tau$. In these experiments, $\tau = 0$ and $\Delta\tau = 0.05$.

The Decayed Reliance Agent is outperformed by CLUE in all tested panels, as either its performance is hampered due to following suboptimal advice (as with the single unreliable expert), or it stops relying on useful advice before its policy has converged (as with the other panels).

The PRQ agent performs exceptionally well when a reliable expert is present, even outperforming CLUE with the single reliable expert, as it learns to identify and follow the optimal policy. However, its performance is hampered with the single unreliable expert, as it takes longer for it to learn to identify and ignore the suboptimal policy. When the expert's policy is not optimal, but is nevertheless better than a random policy, as with the single random expert, the PRQ agent learns early on that following the expert's advice is better than its initial policy, resulting in an initial boost to performance, but is unable to surpass the performance of the policy. CLUE, on the other hand, is able to benefit from the higher-than-random chance of receiving optimal advice, but is still able to surpass the expert's performance.

These results demonstrate that CLUE is more robust to unreliable experts, while still being able to benefit from advice from reliable experts.

4 Theoretical Analysis

In this section, we show the conditions for which CLUE, when exploring, will have a higher probability of selecting the optimal action for a given state than some default unassisted exploration strategy, when acting in an environment where $|A| = 2$ and being advised by a single expert that operates under the following assumption

Assumption 1. *Assume an expert has a true reliability $\rho_{true} \in [0, 1]$ such that, when giving advice, it advises the optimal action with probability ρ_{true} and otherwise advises some suboptimal action with probability $\frac{1-\rho_{true}}{|A|-1}$.*

To do this, we define a function $W(a)$ which represents the probability of selecting a given action when exploring. For example, if selecting actions with uniform random probability, $W(a) = \frac{1}{|A|}$ $\forall a \in A$. We show the values of $\mathbb{E}[\rho]$ (the agent's estimate of the reliability, which may or may not be accurate) for which the probability of selecting the optimal action a^* is greater than or equal to $W(a^*)$, given W and ρ_{true}.

To aid in the proof of this theorem (Theorem 1), we first prove Lemma 1, which shows the conditions for which a CLUE agent is guaranteed to identify a given action as optimal.

For a uniform random exploration strategy ($W(a) = \frac{1}{|A|}$ $\forall a \in A$), the implication of Theorem 1 is that there will be improved performance provided that the estimate $\mathbb{E}[\rho]$ is on the same side of $\frac{1}{2}$ as the true reliability ρ_{true}. For another exploration strategy, the improvement may increase or decrease depending on the probability of selecting a^* under that strategy.

Lemma 1. *Suppose an environment with $|A| = 2$ and a panel consisting of a single expert. Let $\mathbb{E}[\rho]$ denote the agent's estimate of the reliability of the expert. For any given state the expert has advised for, the optimal action a^* will be identified as such by the agent if one of the following holds*

- *The expert advised a^* and $\mathbb{E}[\rho] > \frac{1}{2}$*
- *The expert did not advise a^* and $\mathbb{E}[\rho] < \frac{1}{2}$*

If $\mathbb{E}[\rho] = \frac{1}{2}$, the agent is equally likely to identify

Proof. From Equation 3 of the main paper, we have that

$$P(a_j = a^*|V_t) = \frac{\prod_{e\in E_t} P(v_t^{(e)}|a_j = a^*)}{\sum_{k=0}^{|A|}\prod_{e\in E_t} P(v_t^{(e)}|a_k = a^*)}, \tag{19}$$

which, given that $|E_t| = 1$ and $|A| = 2$, reduces to

$$P(a_j = a^*|V_t) = \frac{P(v_t^{(e)}|a_j = a^*)}{P(v_t^{(e)}|a_0 = a^*) + P(v_t^{(e)}|a_1 = a^*)}. \tag{20}$$

Without loss of generality, let a_0 denote the optimal action for s_t. Substituting in Equation 4 of the main paper, Equation 20 is equal to

$$\begin{aligned} P(a_j = a^*|V_t) &= \frac{P(v_t^{(e)}|a_j = a^*)}{\mathbb{E}[\rho] + 1 - \mathbb{E}[\rho]} \\ &= P(v_t^{(e)}|a_j = a^*), \end{aligned} \tag{21}$$

which is equal to $\mathbb{E}[\rho]$ if the expert advised a_j and $1 - \mathbb{E}[\rho]$ otherwise. Let a_{best} denote the action that maximises $P(a_j = a^*|V_t)$.

We consider 2 cases.

Case 1: The expert has advised a_0. Thus,

$$P(a_0 = a^*|V_t) = \mathbb{E}[\rho]$$
$$P(a_1 = a^*|V_t) = 1 - \mathbb{E}[\rho].$$

$P(a_0 = a^*|V_t) > P(a_1 = a^*|V_t)$ is therefore only true when $\mathbb{E}[\rho] > \frac{1}{2}$, and thus the agent will identify a_0 as the optimal action if $\mathbb{E}[\rho] > \frac{1}{2}$. If $\mathbb{E}[\rho] = \frac{1}{2}$, the agent will do so with probability $\frac{1}{2}$.

Case 2: The expert has advised a_1. Thus,

$$P(a_0 = a^*|V_t) = 1 - \mathbb{E}[\rho]$$
$$P(a_1 = a^*|V_t) = \mathbb{E}[\rho].$$

$P(a_0 = a^*|V_t) > P(a_1 = a^*|V_t)$ is therefore only true when $\mathbb{E}[\rho] < \frac{1}{2}$, and thus the agent will identify a_0 as the optimal action if $\mathbb{E}[\rho] < \frac{1}{2}$. If $\mathbb{E}[\rho] = \frac{1}{2}$, the agent will do so with probability $\frac{1}{2}$. □

Theorem 1. *Suppose an environment with $|A| = 2$ and a panel consisting of a single expert. Let $W(a)$ denote the probability of selecting action a when exploring unassisted. Then the probability of a CLUE agent selecting the optimal action a^* when exploring is greater than or equal to $W(a^*)$ if one of the following holds*

- $\mathbb{E}[\rho] = \frac{1}{2}$ *and* $W(a^*) \leq \frac{1}{2}$
- $\mathbb{E}[\rho] < \frac{1}{2}$ *and* $W(a^*) \leq 1 - \rho_{true}$
- $\mathbb{E}[\rho] > \frac{1}{2}$ *and* $W(a^*) \leq \rho_{true}$

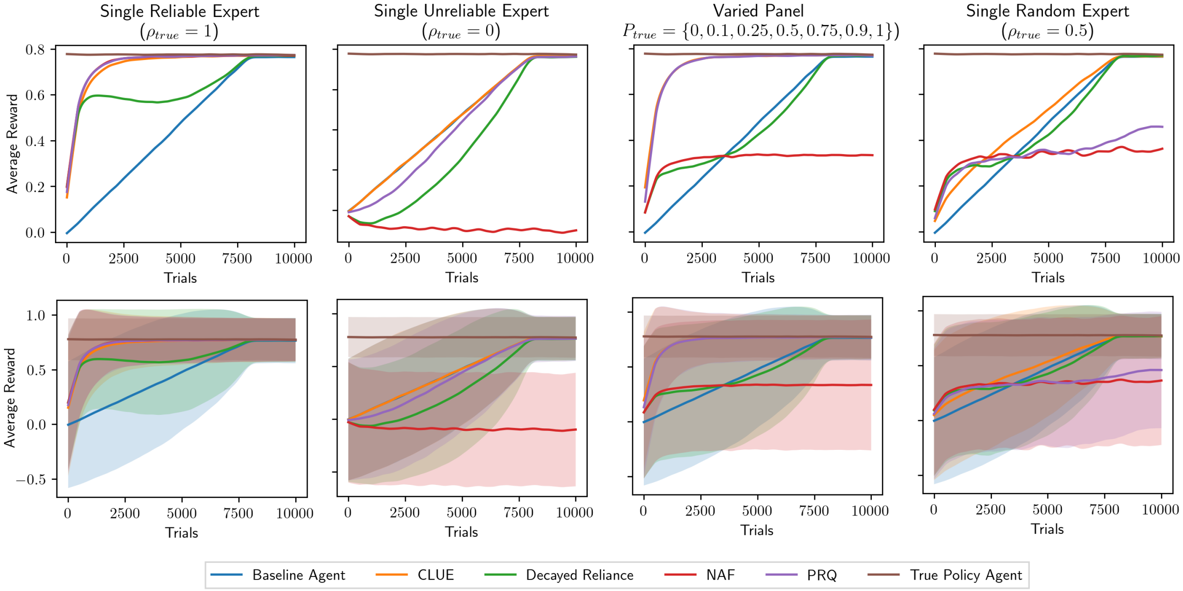

Figure 4: Comparisons of CLUE with the Decayed Reliance and PRQ agents, adapted from the π-reuse and PRQ algorithms respectively (Fernández and Veloso 2006). For the sake of clarity, a version without the shaded areas representing one standard deviation is provided. Note that for the varied panel, CLUE and PRQ have near-identical performance, and thus the PRQ curve lies on top of the CLUE curve.

Proof. Let $P(a)$ denote the probability of selecting action a. Let $a^{(e)}$ denote the action advised by the expert. From the decision-making process described in Section 3.3 of the main paper, we have that

$$P(a^*) = \mathbb{E}[\rho] + (1-\mathbb{E}[\rho])W(a^*) \qquad \text{if } a_{best} = a^* \tag{22}$$

$$P(a^*) = (1-\mathbb{E}[\rho])W(a^*) \qquad \text{if } a_{best} \neq a^* \tag{23}$$

We consider three cases.

Case 1: Let $\mathbb{E}[\rho] = \frac{1}{2}$. From Equations 22 and 23, we have that

$$P(a^*) = \frac{1}{2} + \frac{1}{2}W(a^*) \qquad \text{if } a_{best} = a^*$$

$$P(a^*) = \frac{1}{2}W(a^*) \qquad \text{if } a_{best} \neq a^*$$

From Theorem 1, $P(a_{best} = a^*) = \frac{1}{2} = P(a_{best} \neq a^*)$ and thus

$$\begin{aligned} P(a^*) &= \frac{1}{2}(\frac{1}{2} + \frac{1}{2}W(a^*)) + \frac{1}{2}(\frac{1}{2}W(a^*)) \\ &= \frac{1}{4} + \frac{1}{2}W(a^*), \end{aligned}$$

which is greater than or equal to $W(a^*)$ if and only if $W(a^*) \leq \frac{1}{2}$.

Case 2: Let $\mathbb{E}[\rho] < \frac{1}{2}$. From Equations 22 and 23, and from Theorem 1, we have that

$$P(a^*) = (1-\mathbb{E}[\rho])W(a^*) \qquad a^{(e)} = a^*$$

$$P(a^*) = \mathbb{E}[\rho] + (1-\mathbb{E}[\rho])W(a^*) \qquad a^{(e)} \neq a^*$$

From Assumption 1, $P(a^{(e)} = a^*) = \rho_{true}$ and $P(a^{(e)} \neq a^*) = 1 - \rho_{true}$. Therefore,

$$\begin{aligned} P(a^*) = P(a^{(e)} &= a^*)(1-\mathbb{E}[\rho])W(a^*) + \\ P(a^{(e)} &\neq a^*)(\mathbb{E}[\rho] + (1-\mathbb{E}[\rho])W(a^*)) \\ &= \rho_{true}(1-\mathbb{E}[\rho])W(a^*) + \\ (1 &- \rho_{true})(\mathbb{E}[\rho] + (1-\mathbb{E}[\rho])W(a^*)) \\ &= \mathbb{E}[\rho](1-\rho_{true}) + W(a^*)(1-\mathbb{E}[\rho]), \end{aligned}$$

which is greater than or equal to $W(a^*)$ if and only if $\mathbb{E}[\rho](1 - W(a^*) - \rho_{true}) \geq 0$. As $\mathbb{E}[\rho] \geq 0$, it is sufficient to prove that $1 - W(a^*) - \rho_{true} \geq 0$.

$$\begin{aligned} 1 - W(a^*) - \rho_{true} &\geq 0 \\ -W(a^*) &\geq \rho_{true} - 1 \\ W(a^*) &\leq 1 - \rho_{true} \end{aligned}$$

Thus $P(a^*) \geq W(a^*)$ if and only if $W(a^*) \leq 1 - \rho_{true}$.

Case 3: Let $\mathbb{E}[\rho] > \frac{1}{2}$. From Equations 22 and 23, and from Theorem 1, we have that

$$P(a^*) = \mathbb{E}[\rho] + (1-\mathbb{E}[\rho])W(a^*) \qquad a^{(e)} = a^*$$

$$P(a^*) = (1-\mathbb{E}[\rho])W(a^*) \qquad a^{(e)} \neq a^*$$

From Assumption 1, $P(a^{(e)} = a^*) = \rho_{true}$ and $P(a^{(e)} \neq a^*) = 1 - \rho_{true}$. Therefore,

$$\begin{aligned} P(a^*) &= P(a^{(e)} = a^*)(\mathbb{E}[\rho] + (1-\mathbb{E}[\rho])W(a^*)) + \\ &\quad P(a^{(e)} \neq a^*)(1-\mathbb{E}[\rho])W(a^*) \\ &\quad = \rho_{true}(\mathbb{E}[\rho] + (1-\mathbb{E}[\rho])W(a^*)) + (1-\rho_{true})(1-\mathbb{E}[\rho]), \end{aligned}$$

which is greater than or equal to $W(a^*)$ if and only if $\mathbb{E}[\rho](\rho_{true} - W(a^*)) \geq 0$. As $\mathbb{E}[\rho] \geq 0$, $P(a^*) \geq W(a^*)$ if and only if $W(a^*) \leq \rho_{true}$.

□

5 Hardware and Software Specifications

Experiments were run on Ubuntu 18.04 machines with Intel(R) Core(TM) i9-10940X CPU @ 3.30GHz, with 125GiB of RAM.

All code was written and executed in python 3.8.6, with the following libraries:

- *numpy* version 1.19.2
- *matplotlib* version 3.3.2
- *statsmodels* version 0.12.0
- *networkx* version 2.5

Additionally, the implementations of graphical models, influence diagrams, factors and variable elimination are adapted from Poole and Mackworth (2017).

References

Etz, A. 2018. Introduction to the concept of likelihood and its applications. *Advances in Methods and Practices in Psychological Science*, 1(1): 60–69.

Fernández, F.; and Veloso, M. 2006. Probabilistic policy reuse in a reinforcement learning agent. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, 720–727.

Poole, D. L.; and Mackworth, A. K. 2017. Python code for Artificial Intelligence: Foundations of Computational Agents. *Version 0.7*, 6.