A Notations, definitions and other useful facts

In this section, we define certain standard notations and state some facts that were used in the main paper.

Policy of an algorithm. In PAC-MDP models, an algorithm is considered to be a non-stationary policy $\mathcal{A}$ which, at any instant t, takes as input the path taken so far $p_t := s_0, a_0, \ldots, a_{t-1}, s_t$ and outputs an action. More formally, $\mathcal{A} : \{S \times A\}^* \times S \to A$. Note that since the algorithm is already given the true reward function, we do not provide rewards as input to this non-stationary policy. Then the value of the policy is formally defined as given below.

Definition 7. *For any p_t, we define the value of the non-stationary policy $\mathcal{A}$ of our algorithm on the MDP M as:*

$$V_M^{\mathcal{A}}(p_t) = \mathbb{E}\left[\sum_{t'=t}^{\infty} R(s_{t'}, a_{t'}) \,\middle|\, p_t, \mathcal{A}\right].$$

For any $H > 0$, we denote the truncated value function of $\mathcal{A}$ as:

$$V_M^{\mathcal{A}}(p_t, H) = \mathbb{E}\left[\sum_{t'=t}^{t+H} R(s_{t'}, a_{t'}) \,\middle|\, p_t, \mathcal{A}\right].$$

Following $\pi \in \Pi(Z)$. Below we state the fact that for a closed set Z, by following $\pi \in \Pi(Z)$, the agent always remains in Z with probability 1.

Fact 1. *For any closed set of state-action pairs Z, and for any policy $\pi \in \Pi(Z)$ and for any initial state $s_0 \in Z$:*

$$\mathrm{P}[\forall t, s_t \in Z \mid \pi, s_0 \in Z] = 1$$

Proof. For any t, if $s_t \in Z$ (and this is true for $t = 0$), we have that $(s_t, \pi(s_t)) \in Z$ since $\pi \in \Pi(Z)$. Since Z is closed, this means that $s_{t+1} \in Z$. Hence, by induction the above claim is true. $\square$

Communicatingness. We now discuss the standard notion of communicating and argue that it is equivalent to the notion defined in the paper (Definition 2). Recall that the standard notion of communicatingness of an MDP (Puterman, 2014) is that, for any pair of states in the MDP, there exists a stationary policy that takes the agent from one to the other with positive probability in finite steps. This can be easily generalized to a subset of closed states as follows:

Definition 8. *A closed subset of state-action pairs, $Z \subset S \times A$ is said to be **communicating** if for any two states, $s, s' \in Z$, there exists a stationary policy $\pi_{s \to s'} \in \Pi(Z)$ such that for some $n \geq 1$*

$$\mathrm{P}[s_n = s' \mid \pi_{s \to s'}, s_0 = s] > 0$$

There are two key differences between this definition and Definition 2. (1) Recall that in Definition 2, we defined communicating to mean that for a particular destination state, there exists a single stationary policy that can take the agent from *any* state inside Z to that destination state, whereas in Definition 8 there is a specific policy for every pair of states. (2) Definition 8 only requires the probability of reaching s' to be positive as opposed to 1. Below, we note why these definitions are equivalent:

Fact 2. *Definition 2 and Definition 8 are equivalent.*

Proof. Informally, we need to show that "for any pair of states in Z, there exists a policy that takes the agent from one to the other with positive probability" if and only if "there exists a single policy that reaches a destination state from anywhere in Z with probability 1".

The sufficient direction is clearly true: if such a $\pi_{s'}$ and t from Definition 2 exist, then we know that for any $s \in S$, we can set $\pi_{s \to s'}$ and n in Definition 8 to be $\pi_{s'}$ and t to prove communicatingness (since if it holds with probability 1, it also holds with positive probability).

For the necessary direction, we will show that for a communicating $Z = S \times A$, for any $s' \in Z$, the optimal policy of another MDP satisfies the requirements of $\pi_{s'}$. For a communicating Z with Definition 8, we know that for any $s, s' \in S$, there exists a policy $\pi_{s \to s'}$ and $n \geq 1$ such that $\mathrm{P}[s_n = s' \mid \pi_{s \to s'}, s_0 = s] > 0$.

Now we construct the MDP. For a given s', define an MDP $M_{s'} = \langle S, A, T, R_{s'}, \gamma_{s'} \rangle$ where $\gamma_{s'} = 1$, $R(s) = \mathbf{1}\{s = s'\}$, and s' is terminal. Note that, since $R(s') = 1$ and is zero everywhere else and s' is terminal, for any policy π, the state value function $V_{M_{s'}}^\pi(s) = \mathrm{P}[\exists\ t,\ s.t.\ s_t = s' | \pi, s_0 = s]$. Now define $\pi_{s'}$ to be the optimal policy of this MDP, i.e. the policy that maximizes this value function. This implies that for any s, $V_{M_{s'}}^{\pi_{s'}}(s) \geq V_{M_{s'}}^{\pi_{s,s'}}(s)$. Then,

$$\mathrm{P}[\exists\ t,\ s.t.\ s_t = s' | \pi_{s'}, s_0 = s] = V_{M_{s'}}^{\pi_{s'}}(s)$$
$$\geq V_{M_{s'}}^{\pi_{s,s'}}(s)$$
$$= \mathrm{P}[\exists\ t,\ s.t.\ s_t = s' | \pi_{s,s'}, s_0 = s]$$
$$\geq \mathrm{P}[s_n = s' | \pi_{s,s'}, s_0 = s]$$
$$> 0.$$

Thus, $\mathrm{P}[\exists\ t,\ s.t.\ s_t = s' | \pi_{s'}, s_0 = s] > 0$. Since this condition is satisfied for all starting states $s \in S$ and since we know that Z is closed, we can use Lemma 5 to show that in fact $\mathrm{P}[\exists\ t,\ s.t.\ s_t = s' | \pi_{s'}, s_0 = s] = 1$, proving the existence of $\pi_{s'}$ as claimed.

$\square$

$Z_{\mathbf{safe}}$ is safe. Below, we prove that the set Z_{safe} defined in Definition 5 is indeed a safe set.

Fact 3. *Z_{safe} is a safe set.*

Proof. For any $(s, a) \in Z_{\mathrm{safe}}$, we have by definition that $R(s, a) \geq 0$. Now, if for some s' for which $T(s, a, s') > 0$, consider $a' = \pi_{\mathrm{return}}(s')$. Then, by definition, if the agent were to start at (s', a') and continue following π_{return}, with probability 1, it would reach Z_0 while experiencing only positive rewards. Hence, $(s', a') \in Z_{\mathrm{safe}}$. Thus, Z_{safe} is closed. $\square$

B Methodology

In this section, we provide detailed definitions of our algorithms and the notation used for our proofs. We start by giving detailed a description of the main algorithm and the algorithms which it calls. We then detail how our confidence intervals are computed and how we use these confidence intervals to construct the set of all candidate transition functions. The next section details how these candidate transition functions are used for computing optimistic policies $\overline{\pi}_{\mathrm{goal}}$, $\overline{\pi}_{\mathrm{explore}}$ and $\overline{\pi}_{\mathrm{switch}}$. Specifically, we define new MDPs for each of these policies and define how an optimistic policy is computed on an arbitrary MDP. Finally, we formalize the discounted state distribution and discuss how this is computed in practice.

Here we provide a useful reference for some of the notation used throughout the proofs.

True MDP	$M = \langle S, A, T, R, \gamma \rangle$
True safe, optimal goal policy	π_{safe}^*
Empirical transition probability	$\hat{T}$
Empirical $L1$ confidence interval width	$\hat{\epsilon}_T$
Arbitrary MDP	$M^\dagger = \langle S, A, T^\dagger, R^\dagger, \gamma^\dagger \rangle$
Optimal Q-function on the optimistic MDP $\overline{M}_{\mathrm{goal}}$	$\overline{Q}_{\mathrm{goal}}$
Optimal Q-function on the optimistic MDP $\overline{M}_{\mathrm{explore}}$	$\overline{Q}_{\mathrm{explore}}$
Optimal Q-function on the optimistic MDP $\overline{M}_{\mathrm{switch}}$	$\overline{Q}_{\mathrm{switch}}$
Optimistic goal policy (defined by $\overline{Q}_{\mathrm{goal}}$)	$\overline{\pi}_{\mathrm{goal}}$
Optimistic explore policy (defined by $\overline{Q}_{\mathrm{goal}}$)	$\overline{\pi}_{\mathrm{explore}}$
Optimistic switching policy (defined by $\overline{Q}_{\mathrm{switch}}$)	$\overline{\pi}_{\mathrm{switch}}$
Analogy-based empirical probabilities and confidence intervals	$\overset{\triangle}{T}, \overset{\triangle}{\epsilon}_T$
Optimal value function for some MDP $M^\dagger$	$V_{M^\dagger}^*$

B.1 Algorithms

We first restate Algorithm 1, the full overview of our algorithm, with a bit more detail (most notably including $\hat{Z}_{\mathrm{unsafe}}$). Next, Algorithm 2 details how the agent computes the current estimate of the safe set while ensuring

---

Algorithm 1 Analogous Safe-state Exploration $(\alpha, \Delta, m, \delta_T, R, \gamma, \gamma_{\text{explore}}, \gamma_{\text{switch}}, \tau)$

Initialize: $\hat{Z}_{\text{safe}} \leftarrow Z_0$; $n(s,a), n(s,a,s') \leftarrow 0$; $\overline{Z}_{\text{goal}} \leftarrow S \times A$; $s_0 \leftarrow s_{\text{init}}$.

Initialize: $\hat{Z}_{\text{unsafe}} \leftarrow \{(s,a) \in S \times A : R(s,a) < 0\}$.

Compute confidence intervals, $\overset{\triangle}{T}$ and $\overset{\triangle}{\epsilon}_T$, using Alg 6 (Appendix B) with
state-action-state counts n, parameter δ_T, and analogy function α, Δ.

Compute $\overline{\pi}_{\text{goal}}, \overline{Z}_{\text{goal}}, Z_{\text{explore}}, \hat{Z}_{\text{unsafe}}$ using Alg 3 and 4 with parameters γ, τ
and reward function R.

Compute $\overline{\pi}_{\text{explore}}, \overline{\pi}_{\text{switch}}$ using value iteration (Appendix B.4) with parameters $\gamma_{\text{explore}}, \gamma_{\text{switch}}$.

for $t = 1, 2, 3, \ldots$ **do**

$$a_t \leftarrow \begin{cases} \overline{\pi}_{\text{goal}}(s_t) & \textbf{if } s_t \in \overline{Z}_{\text{goal}} \ \& \ \overline{Z}_{\text{goal}} \subset \hat{Z}_{\text{safe}} \\ \overline{\pi}_{\text{explore}}(s_t) & \textbf{if } \overline{Z}_{\text{goal}} \not\subset \hat{Z}_{\text{safe}} \\ \overline{\pi}_{\text{switch}}(s_t) & \textbf{otherwise}. \end{cases}$$

Take action a_t and observe next state s_{t+1}.

if $n(s_t, a_t) < m$ **then**

$n(s_t, a_t) \mathrel{+}= 1$, $n(s_t, a_t, s_{t+1}) \mathrel{+}= 1$.

Recompute confidence intervals, then expand $\hat{Z}_{\text{safe}}$ using Alg 2 with parameter τ.

Recompute $\overline{\pi}_{\text{goal}}, \overline{Z}_{\text{goal}}, Z_{\text{explore}}, \hat{Z}_{\text{unsafe}}, \overline{\pi}_{\text{explore}}, \overline{\pi}_{\text{switch}}$ as above.

---

reachability, returnability, and closedness. The correctness and efficiency of this algorithm is proven in Section D.1. Algorithms 3, 4, and 5 together provide an overview of how ASE computes the goal and explore policies.

B.2 Confidence intervals

We let $\hat{T}$ denote the empirical transition probabilities. We then let $\hat{\epsilon}_T(s,a)$ denote the width of the $L1$ confidence interval for the empirical transition probability $\hat{T}(s,a)$. As shown in Strehl and Littman (2008), by using the Hoeffding bound, we can ensure that if

$$\hat{\epsilon}_T(s,a) = \sqrt{\frac{2[\ln(2^{|S|} - 2) - \ln(\delta_T)]}{n(s,a)}} \tag{1}$$

where $n(s,a)$ is the number of times we have experienced state-action (s,a), our $L1$ confidence interval hold with probability δ_T. Using the given analogies, we can then derive tighter confidence intervals of width $\overset{\triangle}{\epsilon}_T$ (centered around an estimated $\overset{\triangle}{T}$), especially for state-action pairs we have not experienced, as in Algorithm 6. The algorithm essentially transfers the confidence interval from a sufficiently similar, well-explored state-action pair to the under-explored state-action pair, using analogies.

Given these analogy-based $L1$ confidence intervals, we now define a slightly narrower space of candidate transition probabilities than the space defined by these confidence intervals in order to fully establish the support of certain transitions. Specifically, we take into account Assumption 4, to rule out candidates which do not have sufficiently large transition probabilities. We also make sure that a transition probability is a candidate only if Z_0 is closed under it, as assumed in Assumption 1.

Definition 9. *Given the transition probabilities $\overset{\triangle}{T}$ and confidence interval widths $\overset{\triangle}{\epsilon}_T \colon S \times A \to \mathbb{R}$, we say that $T^\dagger$ is a **candidate** transition if it satisfies the following for all $(s,a) \in S \times A$:*

1. *$\|T^\dagger(s,a) - \overset{\triangle}{T}(s,a)\|_1 \leq \overset{\triangle}{\epsilon}_T(s,a)$.*

2. *if for some s', $\overset{\triangle}{T}(s,a,s') = 0$ and $\overset{\triangle}{\epsilon}_T(s,a) < \tau$, then $T^\dagger(s,a,s') = 0$.*

3. *if $(s,a) \in Z_0$, then $\forall s' \notin Z_0$, $T^\dagger(s,a,s') = 0$*

Furthermore, we let $CI(\overset{\triangle}{T})$ denote the space of all candidate transition probabilities.

Algorithm 2 Compute Safe Set (R, τ)

Require: Estimated safe set $\hat{Z}_{\text{safe}}$ and confidence intervals $\overset{\Delta}{\hat{T}}, \overset{\Delta}{\hat{\epsilon}}_T$.

   $Z_{\text{candidate}} \leftarrow \{(s, a) \in (S \times A) \setminus \hat{Z}_{\text{safe}} \ s.t. \ \overset{\Delta}{\hat{\epsilon}}_T (s, a) < \tau/2, R(s, a) \geq 0\}$.
   while $Z_{\text{candidate}} \neq Z_{\text{closed}}$ in the last iteration **do**
      $Z_{\text{reachable}} \leftarrow \{(s, a) \in Z_{\text{candidate}} : s \in \hat{Z}_{\text{safe}}\}$.
      while $Z_{\text{reachable}}$ changed in the last iteration **do**
         for $(s, a) \in Z_{\text{reachable}} \cup \hat{Z}_{\text{safe}}$ **do**
            Add $\{(s', a') \in Z_{\text{candidate}} \ s.t. \ \overset{\Delta}{\hat{T}} (s, a, s') > 0\}$ to $Z_{\text{reachable}}$.
      $Z_{\text{returnable}} \leftarrow \emptyset$.
      while $Z_{\text{returnable}}$ changed in the last iteration **do**
         for $(s, a) \in Z_{\text{reachable}}$ **do**
            if $\exists \ (s', a') \in Z_{\text{returnable}} \cup \hat{Z}_{\text{safe}} \ s.t. \ \overset{\Delta}{\hat{T}} (s, a, s') > 0$ **then**
               Add (s, a) to $Z_{\text{returnable}}$.
      $Z_{\text{closed}} \leftarrow Z_{\text{returnable}}$.
      while Z_{closed} changed in the last iteration **do**
         for $(s, a) \in Z_{\text{closed}}$ **do**
            if $\exists \ s' \in S \ s.t. \ \overset{\Delta}{\hat{T}} (s, a, s') > 0$ and $\forall \ a' \in A, \ (s', a') \notin Z_{\text{closed}} \cup \hat{Z}_{\text{safe}}$ **then**
               Remove (s, a) from Z_{closed}.
      $Z_{\text{candidate}} \leftarrow Z_{\text{closed}}$.
   $\hat{Z}_{\text{safe}} \leftarrow Z_{\text{closed}} \cup \hat{Z}_{\text{safe}}$.

Algorithm 3 Compute $\overline{\pi}_{\text{goal}}, \overline{Z}_{\text{goal}}$ and $Z_{\text{explore}} \ (\alpha, \Delta, \tau)$

Require: Estimated safe and unsafe sets $\hat{Z}_{\text{safe}}, \hat{Z}_{\text{unsafe}}$ and confidence intervals $\overset{\Delta}{\hat{T}}, \overset{\Delta}{\hat{\epsilon}}_T$.

   Initialize: $Z_{\text{explore}} \leftarrow \emptyset$.
   for $i = 1, 2, \ldots$ **do**
      Compute $\overline{\pi}_{\text{goal}}$ using Eq 5.
      Compute $\overline{Z}_{\text{goal}}$ using Alg 5.
      if $\overline{Z}_{\text{goal}} \subset \hat{Z}_{\text{safe}}$ **then**
         Break.
      $Z_{\text{edge}} \leftarrow \{(s, a) \in \overline{Z}_{\text{goal}} \setminus \hat{Z}_{\text{safe}} \ |s \in \hat{Z}_{\text{safe}}\}$.
      Compute Z_{explore} using Alg 4.
      if $Z_{\text{explore}} = \emptyset$ **then**
         Add Z_{edge} to $\hat{Z}_{\text{unsafe}}$.
      else
         Break.

B.3 Discounted future state distribution.

Below we define the notion of a discounted future state distribution (originally defined in Sutton et al. (2000)), and then describe how we compute it in practice. We will need this notion in order to compute $\overline{Z}_{\text{goal}}$ (discussed in the section titled "Goal MDP").

Given an MDP $M^{\dagger}$, policy π, and state s, the discounted future state distribution is defined as follows:

$$\rho_{\pi,s}^{M^{\dagger}}(s', a') = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t \mathrm{P}\left(s_t = s', a_t = a' | \pi, s_0 = s\right) \tag{2}$$

In words, for any state action pair (s', a'), $\rho_{\pi,s}(s', a')$ denotes the sum of discounted probabilities that (s', a') is taken at any $t \geq 0$ following policy π from s in $M^{\dagger}$.

Computing discounted future state distributions. We use a dynamic programming approach to approximate the discounted future state distribution. Note that we are assuming that the policy π is deterministic.

Algorithm 4 Compute Z_{explore} (α, Δ, τ)

Require: Sets of state-action pairs on the edge of the safe set, Z_{edge}, and on the goal path, $\overline{Z}_{\text{goal}}$,
   along with $\hat{Z}_{\text{safe}}, \hat{Z}_{\text{unsafe}}$ and $\overset{\Delta}{\hat{T}}, \overset{\Delta}{\hat{\epsilon}}_T$.
   Initialize: $Z_{\text{explore}}, Z_{\text{return}} \leftarrow \emptyset, Z_{\text{candidate}} \leftarrow \{(s, a) \in \overline{Z}_{\text{goal}} : s \in \hat{Z}_{\text{safe}}\}, L \leftarrow 0, Z_{\text{next}}^0 \leftarrow Z_{\text{edge}}$.
   while $Z_{\text{explore}} = \emptyset$ and $Z_{\text{next}}^L \neq \emptyset$ **do**
     $Z_{\text{next}}^{L+1} \leftarrow \emptyset$.
     for $(s, a) \in Z_{\text{next}}^L$ **do**
       Add (s, a) to Z_{return}
       if $\overset{\Delta}{\hat{\epsilon}}_T (s, a) > \tau/2$ **then**
         Add $\{\tilde{s}, \tilde{a} \in \hat{Z}_{\text{safe}} : \Delta((s, a), (\tilde{s}, \tilde{a})) < \tau/4\}$ to Z_{explore}.
       else
         Add $\{s', a' \in S \times A : \overset{\Delta}{\hat{T}} (s, a, s') > 0\} \setminus (Z_{\text{return}} \cup \hat{Z}_{\text{safe}} \cup \hat{Z}_{\text{unsafe}})$ to Z_{next}^{L+1}.
     $L \leftarrow L + 1$.

Algorithm 5 Compute $\overline{Z}_{\text{goal}}$ (α, Δ, τ)

Require: Confidence intervals $\overset{\Delta}{\hat{T}}, \overset{\Delta}{\hat{\epsilon}}_T$.
   For convenience, denote $\rho_{\overline{\pi}_{\text{goal}}, s_{\text{init}}}^{\overline{M}_{\text{goal}}}$ as $\overline{\rho}_{\text{goal}}$.
   Initialize: $\overline{Z}_{\text{goal}} \leftarrow \emptyset, \overline{\rho}_{\text{goal}}(\cdot, \cdot, 0)$ as in Eq 3.
   Using dynamic programming, compute $\overline{\rho}_{\text{goal}}(\cdot, \cdot, t)$ as in Eq 4 for $t = 1, 2, \ldots, |S|$.
   Add all $s, a \in S \times A$ where $\overline{\rho}_{\text{goal}}(s, a, |S|) > 0$ to $\overline{Z}_{\text{goal}}$

First, for all $\tilde{s} \in S$ and $\tilde{a} \in A$, we set:

$$\rho_{\pi,s}^{M^\dagger}(\tilde{s}, \tilde{a}, 0) = \begin{cases} 1 - \gamma & \tilde{s} = s \text{ and } \tilde{a} = \pi(s) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Then, at each step, $t + 1$, we will set:

$$\rho_{\pi,s}^{M^\dagger}(\tilde{s}, \tilde{a}, t + 1) = \rho_{\pi,s}^{M^\dagger}(\tilde{s}, \tilde{a}, t) + \gamma \sum_{\tilde{s}' \in S} T(\tilde{s}', \pi(\tilde{s}'), \tilde{s}) \rho_{\pi,s}^{M^\dagger}(\tilde{s}', \pi(\tilde{s}'), t) \tag{4}$$

B.4   Computing optimistic policies

$\overline{\pi}_{\text{goal}}, \overline{\pi}_{\text{explore}}$, and $\overline{\pi}_{\text{switch}}$ are the optimisitc policies for three different MDPs, $M_{\text{goal}}, M_{\text{explore}}$, and M_{switch} (described below). For the theory, we assume that these policies are the true optimistic policy, but in practice this is computed using finite-horizon optimistic form of value iteration introduced in Strehl and Littman (2008). Here we describe this optimistic value iteration procedure.

Optimistic Value Iteration   Let $M^\dagger$ be an MDP that is the same as M but with an arbitrary reward function $R^\dagger$ and discount factor $\gamma^\dagger$. Then, the optimistic state-action value function is computed as follows.

$$\begin{aligned} \overline{Q}^\dagger(s, a, 0) &= 0 \\ \overline{Q}^\dagger(s, a, 1) &= R^\dagger(s, a) \\ \overline{Q}^\dagger(s, a, t) &= R^\dagger(s, a) + \gamma^\dagger \max_{T^\dagger \in CI(\overset{\Delta}{\hat{T}})} \sum_{s' \in S} T^\dagger(s, a, s') \max_{a' \in A} \overline{Q}^\dagger(s', a', t - 1), \; \forall t > 0. \end{aligned} \tag{5}$$

As $t \to \infty$, $\overline{Q}^\dagger(s, a, t)$ converges to a value $\overline{Q}^\dagger(s, a)$ since the above mapping is a contraction mapping. For our theoretical discussion, we assume that we compute these values for an infinite horizon i.e., we compute $\overline{Q}^\dagger(s, a)$.

We then let $\overline{T}^\dagger$ denote the transition probability from $CI(\overset{\Delta}{\hat{T}})$ that corresponds to the optimistic transitions that maximize $\overline{Q}^\dagger$ in Equation 5. Also, we let $\overline{M}^\dagger$ denote the 'optimistic' MDP, $\langle S, A, \overline{T}^\dagger, R^\dagger, \gamma^\dagger \rangle$.

Algorithm 6 Compute Analogy-based Confidence Intervals (α, Δ)

Require: State-action and state-action-state counts $n(\cdot, \cdot)$, $n(\cdot, \cdot, \cdot)$.
  Construct $\hat{T}$ (the empirical transition probabilities) using $n(\cdot, \cdot)$, $n(\cdot, \cdot, \cdot)$.
  Compute $\hat{\epsilon}_T(s, a)$ with Eq 1 using δ_T.
  for $(s, a) \in S \times A$ **do**
    $(\tilde{s}, \tilde{a}) \leftarrow \arg\min\{\hat{\epsilon}_T(s, a), \ \min_{\tilde{s}, \tilde{a}} \epsilon_T(\tilde{s}, \tilde{a}) + \Delta((s, a), (\tilde{s}, \tilde{a}))\}$.
    $\overset{\Delta}{\epsilon}_T(s, a) := \hat{\epsilon}_T(\tilde{s}, \tilde{a})$.
    for $s' \in S$ **do**
      $\tilde{s}' \leftarrow \alpha((s, a, s'), (\tilde{s}, \tilde{a}))$.
      $\overset{\Delta}{\hat{T}}(s, a, s') := \hat{T}(\tilde{s}, \tilde{a}, \tilde{s}')$.

Goal MDP. We define M_{goal} to be an MDP that is the same as M, but without the state-action pairs from $\hat{Z}_{\text{unsafe}}$ (which is a set of state-action pairs that we will mark as unsafe). More concretely, $M_{\text{goal}} = \langle S, A, T, R_{\text{goal}}, \gamma_{\text{goal}} \rangle$, where:

$$R_{\text{goal}}(s, a) = \begin{cases} -\infty & (s, a) \in \hat{Z}_{\text{unsafe}} \\ R(s, a) & \text{otherwise.} \end{cases}$$

We then define $\overline{Q}_{\text{goal}}$ to be the finite-horizon optimistic Q-value computed on M_{goal}, and $\overline{\pi}_{\text{goal}}$ the policy dictated by the estimate of $\overline{Q}_{\text{goal}}$. Also, let $\overline{T}_{\text{goal}}$ denote the optimistic transition probability and $\overline{M}_{\text{goal}}$ the optimistic MDP.

Using the above quantities, we now describe **how to compute $\overline{Z}_{\text{goal}}$** (which we also summarize in Alg 5). Recall that we want $\overline{Z}_{\text{goal}}$ to be the set of all state-actions that would be visited with some non-zero probability by following $\overline{\pi}_{\text{goal}}$ under the optimistic MDP $\overline{M}_{\text{goal}}$. More concretely, for convenience, first define $\overline{\rho}_{\text{goal}} := \rho_{\overline{\pi}_{\text{goal}}, s_{\text{init}}}^{\overline{M}_{\text{goal}}}$, where $\rho_{\overline{\pi}_{\text{goal}}, s_{\text{init}}}^{\overline{M}_{\text{goal}}}$ is as defined in Equation 2. Then, we would like $\overline{Z}_{\text{goal}}$ to be the set $\{(s, a) \in S \times A : \overline{\rho}_{\text{goal}}(s, a) > 0\}$.

However, directly computing this infinite-horizon estimate in practice is impractical. Instead, here we make use of Lemma 11 and Corollary 3, which allow us to exactly compute $\overline{Z}_{\text{goal}}$ through a finite-horizon estimate of $\overline{\rho}_{\text{goal}}$. Specifically, consider the finite-horizon estimate of $\overline{\rho}_{\text{goal}}$ (i.e., the finite horizon estimate of $\rho_{\overline{\pi}_{\text{goal}}, s_{\text{init}}}^{\overline{M}_{\text{goal}}}$ as defined in Equation 3 and Equation 4) which can be computed using dynamic programming. In Lemma 11, we show that if we set the horizon $H \geq |S|$, then $\overline{\rho}_{\text{goal}}(s, a, H) > 0$ if and only if $\overline{\rho}_{\text{goal}}(s, a) > 0$. Hence, we fix H to be any value greater than or equal to $|S|$ and then compute

$$\overline{Z}_{\text{goal}} := \{(s, a) \in S \times A : \overline{\rho}_{\text{goal}}(s, a, H) > 0.\} \tag{6}$$

As stated in Corollary 3, this will guarantee what we need, namely that $\overline{Z}_{\text{goal}} = \{(s, a) \in S \times A : \overline{\rho}_{\text{goal}}(s, a) > 0\}$. We summarize this algorithm in Algorithm 5

Explore MDP. We define $M_{\text{explore}} = \langle S, A, T, R_{\text{explore}}, \gamma_{\text{explore}} \rangle$ to be an MDP with the same states, actions, and transition function as M, but with a different reward function, R_{explore} (computed in Algorithm 4), and discount factor, γ_{explore}. R_{explore} is defined as follows:

$$R_{\text{explore}}(s, a) = \begin{cases} 1 & (s, a) \in Z_{\text{explore}} \\ 0 & (s, a) \in \hat{Z}_{\text{safe}} \setminus Z_{\text{explore}} \\ -\infty & \text{otherwise.} \end{cases}$$

Switch MDP. We define $M_{\text{switch}} = \langle S, A, T, R_{\text{switch}}, \gamma_{\text{switch}} \rangle$ to be an MDP with the same states, actions, and transition function as M, but with a different reward function, R_{switch}, and discount factor, γ_{switch}. More specifically, R_{switch} is defined as follows:

$$R_{\mathrm{switch}}(s,a) = \begin{cases} 1 & (s,a) \in \overline{Z}_{\mathrm{goal}} \\ 0 & (s,a) \in \hat{Z}_{\mathrm{safe}} \setminus \overline{Z}_{\mathrm{goal}} \\ -\infty & \text{otherwise.} \end{cases}$$

B.5   Regarding Safe Islands

Here, we elaborate on the motivation behind involving the notion of returnability (a) in Assumption 5 and (b) in the definition of Z_{safe} in Definition 5.

Returnability in Assumption 5. Recall that a key motivation behind Assumption 5 was that in order to add a safe subset of state-actions to the current safe set, it is necessary for the agent to establish that subset's returnability i.e., establish a return path from the to-be-added subset to the current safe set. Here we explain why this is necessary. Consider a hypothetical agent that tries to expand its safe set without ensuring that whatever it adds to the safe set is returnable. Such an agent might venture into a safe island: although the agent knows that the subset of state-action pairs it has entered into is safe, the agent does not know of any safe path from that subset back to the original safe set. There are two distinct kinds of such safe islands. The first is where there is truly no safe return path; the second is where there does, in fact, exist some safe return path, but the agent has not yet established that this path is indeed safe. We will refer to these islands as True Safe Islands and False Safe Islands.

Although entering into a True Safe Island is not a problem for ensuring optimality in the PAC-MDP sense, entering into a False Safe Island creates trouble. More concretely, in a True Safe Island, since there is no safe way to leave such an island, even the safe-optimal policy *must* remain on this True Safe Island. Thus, the agent that has ventured into a True Safe Island, can potentially find the ϵ-optimal policy, even though it may be forever stuck in this island. However, in a False Safe Island, since there is indeed a safe path to leave this island, it can be the case that the safe-optimal policy from this island will leave the island (and then achieve far higher future reward, than a policy confined to the False Safe Island). Hence, for the agent to be PAC-MDP optimal, it must first establish safety of this path. However, for an agent stuck inside this island, there may be no means to establish safety of that path simply by exploring that island – unless the island is rich enough with analogous states like Z_0 is (which may not be the case if this happens to be a tiny island). Thus, the agent could be forever stuck in the False Safe Island and even worse, it might act ϵ-sub-optimally forever (by choosing to remain instead of exiting). Hence, it's necessary for the agent to establish returnability of any state-action pair before adding it (and Assumption 5 enables us to do this).

Returnability in Definition 5. Next, we explain the motivation behind defining Z_{safe} in Definition 5 to be a "returnable" set. Specifically, recall that Z_{safe} is a safe subset of state-actions, and we would like to compete against the optimal policy on this subset; more importantly, we defined this in a way that any state-action pair in this set is returnable, meaning that it has a return path to Z_0.

Consider the hypothetical scenario where Z_{safe} is defined to allow non-returnable state-actions. Here, we argue that the agent will have to navigate some impractical complications. To begin with, this alternative definition of Z_{safe} could mean that the safe-optimal policy may lead one into True Safe Islands i.e., safe subsets of state-actions from which there is no safe path back to Z_0. This in turn could potentially require the agent to enter into a True Safe Island in order to be PAC-MDP-optimal. Therefore, when the agent expands its safe set, it is necessary for it to find True Safe Islands and add them to the safe set; while doing so, crucially, as discussed earlier, the agent must also avoid adding False Safe Islands to the safe set. Then, in order to meet these two objectives, the agent should consider every possible safe island and consider all its possible return paths, and establish their safety. If it can be established that no safe return paths exist for a particular safe island, the agent can label the island as a True Safe Island and add it to its safe set.

Thus, in theory, the above fairly exhaustive algorithm can address the more liberal definition of Z_{safe}; however, in many practical settings, it may be expensive to fully determine the safety of every state-action pair. Hence, we choose to ignore this situation by enforcing that Z_{safe} is returnable. With this framework, our algorithm can grow the safe set by establishing return paths from the edges of the safe set (as against having to also look for safe islands and establish safety of all their possible return paths).

B.6 MDP metrics and our analogy function

State-action similarities have been used outside of the safety literature in order to improve computation time of planning and sample complexity of exploration. Bisimulation seeks to aggregate states into groupings of states that have similar dynamics or similar Q-values (Givan et al., 2003; Taylor et al., 2009; Abel et al., 2017; Kakade et al., 2003). These state aggregations allow for more efficient planning and exploration (Kakade et al., 2003). Other work has used pseudo-counts to learn approximate state aggregations (Taïga et al., 2018). The reason we do not use state-aggregation methods for transferring dynamics knowledge is that we want to include environments where similarities cannot easily partition the state space, such as situations where the similarity between two states is proportional to their distance.

C Proof Outline

The following subsections describe the overall techniques and intuition, and serve as a rough sketch of the proof of Theorem 1.

C.1 Establishing Safety

We now highlight the key algorithmic aspects which ensure provably safe learning, in other words, that (w.h.p) the agent always experiences only non-negative rewards. Recall that our agent maintains a safe set $\hat{Z}_{\text{safe}}$, and in order to add new state-action pairs to $\hat{Z}_{\text{safe}}$ while ensuring that $\hat{Z}_{\text{safe}}$ is closed, we must be able to determine a "safe return policy" to $\hat{Z}_{\text{safe}}$. However, doing this in a setting with unknown stochastic dynamics poses a significant challenge: we must be able to find a return policy where, for every state-action pair in the return path, we know the exact support of its next state; furthermore, all these state-action pairs should return to $\hat{Z}_{\text{safe}}$ with probability 1. Below, we lay out the key aspects of our approach to tackling this.

"Transfer" of confidence intervals. As a first step, we start by establishing confidence intervals on the transition distributions of all state-action pairs as described below. Let $\hat{T}$ denote the empirical transition probabilities. Just as in Strehl and Littman (2008), we can compute $L1$ confidence intervals of these estimates using the Hoeffding bound (details in Appendix B.2). Let $\hat{\epsilon}_T(s, a)$ denote the $L1$ confidence interval for the empirical transition probability $\hat{T}(s, a)$. Using the provided distance and analogy function Δ and α, and using simple triangle inequalities, we can then derive tighter confidence intervals $\overset{\Delta}{\hat{\epsilon}}_T$ (centered around an estimated $\overset{\Delta}{\hat{T}}$) as in Alg 6. The idea here is to "transfer" the confidence interval from a sufficiently similar, well-explored state-action pair to an under-explored state-action pair, using analogies.

Learning the next-state support. Crucially, we can use these transferred confidence intervals to infer the support of state-action pairs we have not experienced. More concretely, in Lemma 2 we show that, when a confidence interval is sufficiently tight, specifically when $\overset{\Delta}{\hat{\epsilon}}(s, a) \leq \tau/2$ for some (s, a) (where τ is the smallest non-zero transition probability defined in Assumption 4), we can exactly recover the support of the next state distribution of (s, a). This fact is then exploited by Algorithm 2 to expand the safe set whenever the confidence intervals are updated.

Correctness of $\hat{Z}_{\text{safe}}$. To expand $\hat{Z}_{\text{safe}}$ while ensuring that it is safe and communicating, Algorithm 2 first creates a candidate set, $Z_{\text{candidate}}$, of all state-action pairs (s, a) with sufficiently tight confidence intervals and non-negative rewards (and so, we know their next state supports). The algorithm then executes three (inner) loops each of which prunes this candidate set. To ensure communicatingness, the first loop eliminates candidates that have no probability of reachability from $\hat{Z}_{\text{safe}}$, and the second loop eliminates those from which there is no probability of return to $\hat{Z}_{\text{safe}}$. In order to ensure closedness, the third loop eliminates those that potentially lead us outside of $\hat{Z}_{\text{safe}}$ or the remaining candidates. We repeat these three loops until convergence. We prove in Lemmas 4 and 3 that Algorithm 2 correctly maintains the safety and communicatingness of $\hat{Z}_{\text{safe}}$ and in Lemma 1 that the algorithm terminates in polynomial time. Note that in Theorem 1, we prove that (w.h.p.) our agent always picks actions only from $\hat{Z}_{\text{safe}}$.

Completeness of $\hat{Z}_{\text{safe}}$. While the above aspects ensure correctness of Algorithm 2, these would be satisfied even by a trivial algorithm that always only returns Z_0. Hence, it is important to establish that for given set of confidence intervals, $\hat{Z}_{\text{safe}}$ is "as large as it can be". More concretely, consider any state on the edge of $\hat{Z}_{\text{safe}}$ for which there exists a return policy to $\hat{Z}_{\text{safe}}$ which passes only through (non-negative reward) state-action pairs

with confidence intervals at most $\tau/2$; this means that we know all possible trajectories in this policy, and all of these lead to $\hat{Z}_{\text{safe}}$. In such a case, we show in Lemma 6 that Algorithm 2 does indeed add this edge action *and all of the actions in every possible return trajectory* to $\hat{Z}_{\text{safe}}$.

C.2 Guided Exploration

To be able to add a state-action pair to our conservative estimate of the safe-set, $\hat{Z}_{\text{safe}}$, we not only need to tighten the confidence intervals of that state-action pair but also that of every state-action in all its return trajectories to $\hat{Z}_{\text{safe}}$. Observe that this can be accomplished by exploring state-action pairs inside $\hat{Z}_{\text{safe}}$ that are similar to this return path, and using analogies to transfer their confidence intervals. However, this raises two main algorithmic challenges.

Selecting unexplored actions for establishing safety. First, for which unexplored state-action pairs outside $\hat{Z}_{\text{safe}}$ do we want to establish safety? Instead of expanding $\hat{Z}_{\text{safe}}$ arbitrarily, we will keep in mind the objective mentioned in our outline of ASE: we want to expand $\hat{Z}_{\text{safe}}$ so that we can get to a stage where every possible trajectory when following the optimistic goal policy, $\overline{\pi}_{\text{goal}}$, is guaranteed to be safe, allowing the agent to safely follow $\overline{\pi}_{\text{goal}}$. By letting $\overline{Z}_{\text{goal}}$ denote the set of all state-action pairs on any path following $\overline{\pi}_{\text{goal}}$ from the initial state s_{init}, this condition can be equivalently stated as $\overline{Z}_{\text{goal}} \subset \hat{Z}_{\text{safe}}$.

So, to carefully select such unexplored state-action pairs, ASE calls Algorithm 3, which is an iterative procedure: in each iteration, it first (re)computes the optimistic goal policy $\overline{\pi}_{\text{goal}}$ and the set $\overline{Z}_{\text{goal}}$. Using this, it then creates a set Z_{edge}, which is the intersection of $\overline{Z}_{\text{goal}}$ and the set of all edge state-action pairs of $\hat{Z}_{\text{safe}}$. We then hope to establish safety of Z_{edge}, so that, intuitively, we can *expand the frontier of our safe set only along the direction of the optimistic path*. To this end, Algorithm 3 calls Algorithm 4 to compute a corresponding $Z_{\text{explore}} \subset \hat{Z}_{\text{safe}}$ to explore (we will describe Algorithm 4 shortly).

Now, in the case Z_{explore} is non-empty, Algorithm 3 returns control back to ASE, for it to pursue $\overline{\pi}_{\text{explore}}$ – and Lemma 12 shows that $\overline{\pi}_{\text{explore}}$ indeed explores Z_{explore} in poly-time. But if Z_{explore} is empty, Algorithm 3 adds all of Z_{edge} to $\hat{Z}_{\text{unsafe}}$; in the next iteration, $\overline{\pi}_{\text{goal}}$ is updated to ignore $\hat{Z}_{\text{unsafe}}$. In Lemma 9, we use Assumption 5, to prove that the elements added to $\hat{Z}_{\text{unsafe}}$ are indeed elements that do not belong to Z_{safe} (and so we can confidently ignore $\hat{Z}_{\text{unsafe}}$ while computing $\overline{\pi}_{\text{goal}}$). In Lemma 10, we show that this iterative approach terminates in poly-time and either returns a non-empty Z_{explore} that can be explored by $\overline{\pi}_{\text{explore}}$, or updates $\overline{\pi}_{\text{goal}}$ in a way that $\overline{Z}_{\text{goal}} \subset \hat{Z}_{\text{safe}}$. In the case that $\overline{Z}_{\text{goal}} \subset \hat{Z}_{\text{safe}}$, using Lemma 12, we show that the agent first takes $\overline{\pi}_{\text{switch}}$ to enter into $\overline{Z}_{\text{goal}}$ in finite time, so that the agent can pursue $\overline{\pi}_{\text{goal}}$.

Selecting safe actions for exploration. To establish safety of an unexplored $(s, a) \in Z_{\text{edge}}$, we must explore state-action pairs from $\hat{Z}_{\text{safe}}$ that are similar to state-actions along an *unknown* return policy from (s, a) in order to learn that *unknown* policy. While such a policy does exist if $(s, a) \in Z_{\text{safe}}$ (according to Assumption 5), the challenge is to resolve this circularity, without exploring $\hat{Z}_{\text{safe}}$ exhaustively.

Instead, Algorithm 4 uses a breadth-first-search (BFS) from (s, a) which essentially enumerates a superset of trajectories that contains the true return trajectories. Specifically, it first enumerates a list of state-action pairs that are a 1-hop distance away and if any of them have a loose confidence interval, it adds to Z_{explore} a corresponding similar state-action pair from $\hat{Z}_{\text{safe}}$ (if any exist). If Z_{explore} is empty at this point, Algorithm 4 repeats this process for 2-hop distance, 3-hop distance and so on, until either Z_{explore} is non-empty or the BFS tree cannot be grown any further. Lemma 9 argues that this procedure does populate Z_{explore} with all the state-action pairs necessary to establish the required return paths; Lemma 7 demonstrates its polynomial run-time. Although we cannot guarantee that this method does not explore all of $\hat{Z}_{\text{safe}}$, we do see this empirically, as we show in our experiments (see Section 6).

D Proofs

This section details our proof of Theorem 1, that ASE is guaranteed to be safe with high probability and is optimal in the PAC-MDP sense. We start by restating Theorem 1 and proving it. We then examine proofs for the correctness and polynomial computation time of Algorithm 2. Specifically, we show that the computed $\hat{Z}_{\text{safe}}$ is closed and communicating. Next we show that Algorithms 4 and 3 correctly compute the desired Z_{explore} and an estimate of $\overline{Z}_{\text{goal}}$ in polynomial time. The following section shows that the computed estimate of $\overline{Z}_{\text{goal}}$ is in

fact correct, under certain conditions. Subsection D.4 provides the key lemmas for proving PAC-MDP, namely that that our agent, following $\overline{\pi}_{\text{goal}}$, $\overline{\pi}_{\text{explore}}$, or $\overline{\pi}_{\text{switch}}$ either performs the desired behavior (acting optimally or reaching certain state-action pairs) or learns something new about the transition function. By bounding the number of times our agent learns something new, we can show that the agent follows the optimal after a polynomial number of steps. The final subsection provides and proves additional supporting lemmas.

Theorem 1. *For any constant $c \in (0, 1/4]$, $\epsilon, \delta \in (0, 1]$, MDP $M = \langle S, A, T, \mathcal{R}, \gamma \rangle$, for $\delta_T = \delta/(2|S||A|m)$, $\gamma_{explore} = \gamma_{switch} = c^{1/H}$, and $m = O\left((|S|/\tilde{\epsilon}^2) + (1/\tilde{\epsilon}^2)\ln(|S||A|/\tilde{\epsilon})\right)$ where $\tilde{\epsilon} = \min\left(\tau, \epsilon(1-\gamma)^2, 1/H^2\right)$ and $H = O\left(\max\left\{H_{com}\log H_{com}, (1/(1-\gamma))\ln(1/\epsilon(1-\gamma))\right\}\right)$ ASE is Safe-PAC-MDP with a sample complexity bounded by $O\left(Hm|S||A|(1/(\epsilon(1-\gamma)))\ln(1/\delta)\right)$.*

Proof.

Proof of admissibility. We first establish the probability with which our confidence intervals remain admissible throughout the entire execution of the algorithm. Note that we only calculate each confidence interval m times for every state-action pair. Thus, by the union bound and our choice of δ_T, the confidence intervals defined by $\hat{T}$ and $\hat{\epsilon}_T$ hold with probability $1 - \delta/2$. Then, by the triangle inequality, even the tighter confidence intervals computed by Algorithm 6 – defined by $\overset{\triangle}{\hat{T}}$ and $\overset{\triangle}{\hat{\epsilon}}_T$ – are admissible.

Proof of safety. Next we will show that, given that the confidence intervals are admissible, the algorithm never takes a state-action pair outside of Z_{safe}. Corollary 2, Lemma 3 and 4 together show that $\hat{Z}_{\text{safe}}$ is a safe (which also implies, closed), communicating subset of Z_{safe}. Using this, we will inductively show that the agent is always safe under our algorithm. Specifically, assume that at any time instant, starting from s_0, the agent has so far only taken actions from $\hat{Z}_{\text{safe}}$. Since $\hat{Z}_{\text{safe}}$ is closed and safe, this means that the agent has so far been safe, and is currently at $s_t \in \hat{Z}_{\text{safe}}$. We must establish that even now the agent takes an action a_t such that $(s_t, a_t) \in \hat{Z}_{\text{safe}}$.

Now, at each step, recall that according to Algorithm 1, the agent follows either $\overline{\pi}_{\text{explore}}$, $\overline{\pi}_{\text{switch}}$, or $\overline{\pi}_{\text{goal}}$. Consider the case when the agent follows either $\overline{\pi}_{\text{explore}}$ or $\overline{\pi}_{\text{switch}}$.

In this case, for all $(s, a) \notin \hat{Z}_{\text{safe}}$ the rewards are set to be $-\infty$, and for all $(s, a) \in \hat{Z}_{\text{safe}}$ the rewards are set to be non-negative. To address both $\overline{\pi}_{\text{explore}}$ and $\overline{\pi}_{\text{switch}}$ together, let the assigned rewards be $R^{\dagger}$.

Now, for any $t \geq 0$, recall that $\overline{Q}^{\dagger}(s, a, t)$ denotes the estimate of $\overline{Q}^{\dagger}(s, a)$ after t iterations of dynamic programming (not to be confused with the finite-horizon value of the optimistic policy). We first claim that, for all $(s, a) \in \hat{Z}_{\text{safe}}$ and for all iterations $t \geq 1$, the resulting optimistic Q-values are such that $\overline{Q}^{\dagger}(s, a, t) \geq 0$ if $(s, a) \in \hat{Z}_{\text{safe}}$ and $\overline{Q}^{\dagger}(s, a, t) = -\infty$ otherwise.

We prove this claim by induction on t, assuming that $\overline{Q}^{\dagger}(s, a, t)$ is initialized to some non-negative value for $t = 0$. For $t = 1$, our claim is satisfied because the Q-values equal to the sum of the reward function and some positive quantity.

Consider any $t > 1$ and $(s, a) \in \hat{Z}_{\text{safe}}$. We know from Equation 5 that the Q-value for this horizon can be decomposed into a sum of the reward and the maximum Q-value of the next states (with a positive, multiplicative discount factor). For $(s, a) \in \hat{Z}_{\text{safe}}$, in Equation 5, we will have that the first term, which is the reward function, is non-negative. The second term is an expectation over the maximum Q-values (for a horizon of $t - 1$), where the expectation corresponds to the probability distribution of $\overline{T}^{\dagger}(s, a, \cdot)$ over the next states. Since $(s, a) \in \hat{Z}_{\text{safe}}$ and since $\overline{T}^{\dagger}$ is a candidate transition function, by Corollary 1, all the next states according to this transition function, belong to $\hat{Z}_{\text{safe}}$. Now, for any $s' \in \hat{Z}_{\text{safe}}$, there exists a' such that $(s', a') \in \hat{Z}_{\text{safe}}$. By induction, we have $\overline{Q}^{\dagger}(s', a', t - 1) \geq 0$, and hence $\max_{a''} \overline{Q}^{\dagger}(s', a'', t - 1) \geq 0$. Hence, even the second term in the expansion of $\overline{Q}^{\dagger}(s, a, t)$ is non-negative, implying that $\overline{Q}^{\dagger}(s, a, t) \geq 0$. Now, for any $(s, a) \notin \hat{Z}_{\text{safe}}$, it follows trivially from Equation 5 that $\overline{Q}^{\dagger}(s, a, t) = -\infty$ since the reward is set to be $-\infty$.

Thus, for any state $s \in \hat{Z}_{\text{safe}}$, we have established that there exists a such that $(s, a) \in \hat{Z}_{\text{safe}}$ and $\overline{Q}^{\dagger}(s, a) \geq 0$. Furthermore, for any action a' such that $(s, a') \notin \hat{Z}_{\text{safe}}$, $\overline{Q}^{\dagger}(s, a') = -\infty$. Therefore, $\overline{\pi}^{\dagger}(s)$ must be an action a such that $(s, a) \in \hat{Z}_{\text{safe}}$. This proves that $\overline{\pi}^{\dagger} \in \Pi(\hat{Z}_{\text{safe}})$. In other words, this means that at $s \in \hat{Z}_{\text{safe}}$ the agent takes an action a such that $(s, a) \in \hat{Z}_{\text{safe}}$. This completes our argument for $\overline{\pi}_{\text{explore}}$ and $\overline{\pi}_{\text{switch}}$.

As the final case, consider a time instant when the agent follows π_{goal}. By design of Algorithm 1, we know that this happens only if $\overline{Z}_{\text{goal}} \subset \hat{Z}_{\text{safe}}$ and $s \in \overline{Z}_{\text{goal}}$. Now, by definition of $\overline{Z}_{\text{goal}}$, since $s \in \overline{Z}_{\text{goal}}$, we know that $(s, \overline{\pi}_{\text{goal}}(s)) \in \overline{Z}_{\text{goal}}$. Then, since $\overline{Z}_{\text{goal}} \subset \hat{Z}_{\text{safe}}$, $(s, \overline{\pi}_{\text{goal}}(s)) \in \hat{Z}_{\text{safe}}$, implying that the algorithm picks only safe actions, even when it follows $\overline{\pi}_{\text{goal}}$.

Proof of PAC-MDP. Now we will show that ASE is PAC-MDP. To do this, we will show that at any step of the algorithm, assuming our confidence intervals are admissible, the agent will either act ϵ-optimally or reach a state outside of the known set $K = \{(s, a) \in S \times A : n(s, a) \geq m\}$ in some polynomial number of steps with some positive polynomial probability. To prove this, recall the agent follows either $\overline{\pi}_{\text{explore}}$, $\overline{\pi}_{\text{switch}}$, or $\overline{\pi}_{\text{goal}}$ in three mutually exclusive cases; let us examine each of these three cases.

Case 1: $\overline{Z}_{\text{goal}} \not\subset \hat{Z}_{\text{safe}}$. If $\overline{Z}_{\text{goal}} \not\subset \hat{Z}_{\text{safe}}$, then the agent follows $\overline{\pi}_{\text{explore}}$. In this case, we will show that the agent will experience a state-action pair from K^c (the complement of K) in the first H_1 steps, where $H_1 = O(H^2/c)$.

First note that the condition $\overline{Z}_{\text{goal}} \not\subset \hat{Z}_{\text{safe}}$ can only change if $\overline{Z}_{\text{goal}}$ or $\hat{Z}_{\text{safe}}$ are modified, which can only happen if the agent experiences a state action pair outside K; so, if before the agent takes its H_1th step, this condition changes, we know that the agent has experienced a state-action pair outside K, and hence, we are done.

Consider the case when the agent does not experience any element of K^c in the first $H_1 - 1$ steps; hence the agent follows a fixed $\overline{\pi}_{\text{explore}}$ for these steps. By Lemma 10, since $\overline{Z}_{\text{goal}} \not\subset \hat{Z}_{\text{safe}}$, there must exist some element in Z_{explore} (where $Z_{\text{explore}} \subset \hat{Z}_{\text{safe}}$ by design of Algorithm 4). Now, recall that $\overline{\pi}_{\text{explore}}$ is computed using rewards R_{explore}, which are set to 1 on Z_{explore}, and either 0 or $-\infty$ otherwise, depending on whether the state-action pair is in $\hat{Z}_{\text{safe}}$ or not. If we define $M_{\text{explore}} = \langle S, A, T, R_{\text{explore}}, \gamma_{\text{explore}} \rangle$, then we can invoke Lemma 12 for M_{explore} to establish that the agent reaches Z_{explore} or K^c.

To do this, we must establish that all requirements of Lemma 12 hold. In particular, we have $Z_{\text{explore}} \subset \hat{Z}_{\text{safe}}$ by design of Algorithm 4. We also have $\hat{Z}_{\text{safe}}$ is communicating (Lemma 4) and closed (Lemma 3) and that $\overline{\pi}_{\text{explore}} \in \Pi(\hat{Z}_{\text{safe}})$ (from our proof for safety of Algorithm 1). Finally, since m is sufficiently large, by Lemma 12, the agent will reach Z_{explore} or K^c in $H_1 = O(H^2/c)$ steps with probability at least $1/2$ as long as $H \geq H_{\text{com}} \log \frac{16 H_{\text{com}}}{c} / \log \frac{1}{c}$. Note that although Lemma 12 guarantees this for the behavior of the agent on M_{explore}, the same would apply for M as well, since both these MDPs share the same transition. Finally, note that since $Z_{\text{explore}} \notin K$ (by Lemma 8), this means that the agent escapes K in H_1 steps.

Case 2: $\overline{Z}_{\text{goal}} \subset \hat{Z}_{\text{safe}}$, $s_t \notin \overline{Z}_{\text{goal}}$. Now consider the next mutually exclusive case where $\overline{Z}_{\text{goal}} \subset \hat{Z}_{\text{safe}}$ but the current state $s \notin \overline{Z}_{\text{goal}}$. In this case our agent will attempt to return to $\overline{Z}_{\text{goal}}$ by following $\overline{\pi}_{\text{switch}}$. In this case, we argue that, in the next $H_2 = O(H^2/c)$ steps, the agent either does reach $\overline{Z}_{\text{goal}}$ or experiences a state-action pair in K^c. To see why, note that the current condition can only change if $\overline{Z}_{\text{goal}}$ or $\hat{Z}_{\text{safe}}$ change or if the agent reaches a state $s \in \overline{Z}_{\text{goal}}$. As noted before, $\overline{Z}_{\text{goal}}$ or $\hat{Z}_{\text{safe}}$ are modified only if the agent experiences a state action pair outside K; so, if before the agent takes its H_2th step, this condition changes, we know that the agent has either experienced a state-action pair outside K or has reached $\overline{Z}_{\text{goal}}$, and hence, we are done.

Consider the case when the agent does not experience any element of K^c in the first $H_2 - 1$ steps, and hence follows a fixed $\overline{\pi}_{\text{switch}}$ for these steps. Since $\overline{Z}_{\text{goal}} \neq \emptyset$ (which is trivially true since $s_0 \in \overline{Z}_{\text{goal}}$ always), using the same reasoning as the previous case, we can again use Lemma 12 to show that the agent will reach a state-action pair in $\overline{Z}_{\text{goal}}$ or outside of K in $H_2 = O(H^2/c)$ steps with probability at least $1/2$, since m is sufficiently large and $H \geq H_{\text{com}} \log \frac{16 H_{\text{com}}}{c} / \log \frac{1}{c}$.

Case 3: $\overline{Z}_{\text{goal}} \subset \hat{Z}_{\text{safe}}$, $s_t \in \overline{Z}_{\text{goal}}$. Finally, we consider the last case where $\overline{Z}_{\text{goal}} \subset \hat{Z}_{\text{safe}}$ and the current state $s \in \overline{Z}_{\text{goal}}$. In this case, we argue that the agent either takes an action that is near-optimal, or in the next H steps, it reaches K^c with sufficiently large probability.

Let $P(A_M)$ be the probability that starting at this step, the Algorithm 1 leads the agent out of K in H steps, conditioned on the history p_t. Now, if $P(A_M) \geq \epsilon(1 - \gamma)/4$, the agent will escape K in H steps with sufficient probability. Hence, consider the case when $P(A_M) \leq \epsilon(1 - \gamma)/4$.

Then, assuming $H \geq O\left(\frac{1}{1-\gamma} \ln \frac{1}{\epsilon(1-\gamma)}\right)$ and sufficiently large m, we can use the above probability bounds and

Lemma 14 to show that in this case the state s_t that the agent currently is in satisfies:

$$V_M^{\mathcal{A}}(p_t) \geq V_{M_{\text{goal}}}^*(s_t) - \epsilon. \tag{7}$$

To complete our discussion of this case, we need to lower bound the right hand side in terms of the value of the safe-optimal policy π_{safe}^* on the true MDP M. Recall that π_{safe}^* is a policy that maximizes $V_M^{\pi_{\text{safe}}^*}$ subject to the constraint that $\pi_{\text{safe}}^* \in \Pi(Z_{\text{safe}})$. Now, consider an MDP M_{goal}^* with the same transitions as M. However it has rewards R_{goal}^* such that for all $(s, a) \notin Z_{\text{safe}}$, $R_{\text{goal}}^*(s, a) = -\infty$ and everywhere else $R_{\text{goal}}^*(s, a) = R(s, a)$. Now, since $\pi_{\text{safe}}^* \in \Pi(Z_{\text{safe}})$, from Fact 1, we have that following π_{safe}^* from any $s \in Z_{\text{safe}}$, the agent would never exit Z_{safe}. Hence, for any $s \in Z_{\text{safe}}$, $V_{M_{\text{goal}}^*}^{\pi_{\text{safe}}^*}(s) = V_M^{\pi_{\text{safe}}^*}(s)$. Note that this equality applies to the current state s_t since it is inside $\overline{Z}_{\text{goal}}$, and we know that $\overline{Z}_{\text{goal}} \subset \hat{Z}_{\text{safe}}$ and $\hat{Z}_{\text{safe}} \subset Z_{\text{safe}}$ (by Corollary 2).

Now, let us compare M_{goal}^* and M_{goal}. Recall that M_{goal} has its rewards set to $-\infty$ only on $\hat{Z}_{\text{unsafe}}$ (and everywhere else, it equals $R(s, a)$). By Lemma 9, we know that $\hat{Z}_{\text{unsafe}} \cap Z_{\text{safe}} = \emptyset$, and therefore $\hat{Z}_{\text{unsafe}} \subset Z_{\text{safe}}^c$. Thus, both M_{goal} and M_{goal}^* have the same rewards as M, except M_{goal} has the rewards set to $-\infty$ on a set $\hat{Z}_{\text{unsafe}}$, while M_{goal}^* has rewards set to $-\infty$ on a superset of $\hat{Z}_{\text{unsafe}}$, Z_{safe}^c. In other words, the rewards of M_{goal} are greater than or equal to the rewards of M_{goal}^*. Thus, the value of the optimal policy on M_{goal} cannot be less than that of M_{goal}^*. Formally, for all s, $V_{M_{\text{goal}}}^*(s) \geq V_{M_{\text{goal}}^*}^{\pi_{\text{safe}}^*}(s)$. Since we also have $V_{M_{\text{goal}}^*}^{\pi_{\text{safe}}^*}(s_t) = V_M^{\pi_{\text{safe}}^*}(s_t)$, we get:

$$V_{M_{\text{goal}}}^*(s_t) \geq V_M^{\pi_{\text{safe}}^*}(s_t). \tag{8}$$

Thus, from Equations 7 and 8, we get that at this state, $V_{M_{\text{goal}}}^{\mathcal{A}}(p_t) \geq V_M^{\pi_{\text{safe}}^*}(s_t) - \epsilon$.

In summary, the agent does at least one of the following at any timestep:

1. reach K^c in H_1 steps (starting from Case 1) with probability at least $1/2$

2. reach K^c in H_2 steps (starting from Case 2), with probability at least $1/4$

3. reach K^c in H steps (starting from Case 3) with probability at least $\Omega(\epsilon(1 - \gamma))$.

4. reach $\overline{Z}_{\text{goal}}$ in H_2 steps (starting from Case 2) with probability at least $1/4$.

5. take a nearly-optimal action (in Case 3).

Note that in the above list, we have slightly modified the guarantee from Case 2. In particular, Case 2 guaranteed that with a probability of $1/2$ the agent would reach either K^c or $\overline{Z}_{\text{goal}}$; from this we have concluded that at least one of these events would have a probability of at least $1/4$.

We first upper bound the number of sub-optimal steps corresponding to the first three events. Observe that the agent can only experience a state-action pair outside of K a total of $m|S||A|$ times. Now, by the Hoeffding bound, we have that it takes at most $O\left(m|S||A|\frac{1}{\epsilon(1-\gamma)} \ln \frac{1}{\delta}\right)$ independent trials to see $m|S||A|$ heads in a coin that has a probability of at least $\Omega(\min(1/4, \epsilon(1 - \gamma))) = \Omega(\epsilon(1 - \gamma))$ as turning out to be heads. Hence, these trajectories would correspond to at most $O\left(\max(H_1, H_2, H) \cdot m|S||A| \cdot \frac{1}{\epsilon(1-\gamma)} \ln \frac{1}{\delta}\right)$ many sub-optimal steps.

Next, we upper bound the number of sub-optimal steps corresponding to the fourth event. Consider two successful occurrences of this event. That is, in both instances, the agent did indeed reach $\overline{Z}_{\text{goal}}$. In such a case, there must be at least one time instant between these two occurrences when the agent experienced K^c; if not, after the first occurrence, by design of Algorithm 1, the agent would have remained in $\overline{Z}_{\text{goal}}$, thereby precluding the second occurrence of the event from happening. Thus, there can be at most as many successful occurrences of this event as $m|S||A| + 1$. Again, by a Hoeffding bound, this corresponds to $O\left(H_2 \times m|S||A|\frac{1}{\epsilon(1-\gamma)} \ln \frac{1}{\delta}\right)$ many sub-optimal steps, with probability $1 - \delta/2$.

Combining the above two bounds and plugging in the bound on m gives our final bound on the number of sub-optimal steps.

$\square$

D.1   Proofs about Algorithm 2

Recall that Algorithm 2 expands the set of safe state-action pairs $\hat{Z}_{\text{safe}}$ (by making use of an updated set of confidence intervals) by first creating a candidate set and then iteratively pruning the set until it stops changing.

First, in Lemma 1, we establish this procedure terminates in polynomial time. In the lemmas that follow after that, we prove soundness and completeness. In particular, in Lemma 3, we establish that the updated $\hat{Z}_{\text{safe}}$ is indeed safe. In Lemma 4, we establish that $\hat{Z}_{\text{safe}}$ is communicating and as a result of which in Corollary 2, we establish that $\hat{Z}_{\text{safe}} \subset Z_{\text{safe}}$. Finally, in Lemma 6 we prove completeness in that, if there exists a state-action pair on the edge of $\hat{Z}_{\text{safe}}$, and if there exists a return path from that edge that only takes state-action pairs whose confidence intervals are sufficiently small, then that state-action pair and all of that return path is added to $\hat{Z}_{\text{safe}}$.

Lemma 1. *Algorithm 2 terminates in poly($|S|, |A|$) time.*

Proof. The algorithm begins with a set $Z_{\text{candidate}}$ with $|S| \cdot |A|$ many elements. Now in each outer iteration, $Z_{\text{candidate}}$ either ends up losing some elements, or remains the same. If it does remain the same, then we break out of the outer loop. Thus, there can be at most $\mathcal{O}(|S| \cdot |A|)$ many iterations of the outer loop.

Next, consider the first inner while block. In each iteration of the while loop, we either add an element to $Z_{\text{reachable}}$ or break out of the loop if $Z_{\text{reachable}}$ does not change. Thus, there can be at most $\mathcal{O}(|S| \cdot |A|)$ many iterations of this while loop. As for the time complexity of the inner while loops, since $\hat{Z}_{\text{safe}} \cup Z_{\text{reachable}}$ is a finite set, it is easy to see that these iterations terminate in polynomial time. A similar argument holds for the next while block too. For the third while block, we must apply a slightly different version of this argument where we make use of the fact that in each run of the while loop, we either remove a state-action pair from Z_{closed} or break out of the loop. From the above arguments, it follows that the algorithm terminates in polynomial time. $\square$

Note: In the following discussions, unless otherwise specified, Z_{closed} denotes the set as it is in the last step of Algorithm 2.

Before we prove our other lemmas about Algorithm 2, we first establish a result about Algorithm 6 that computes the tighter confidence intervals. Specifically we show that these confidence intervals are computed in a way that if a particular interval is sufficiently tight, then every candidate transition probability in that interval has the same support of next state-action pairs as the true support.

Lemma 2. *Assume the confidence intervals are admissible. Then, for any (s, a) such that $\overset{\triangle}{\hat{\epsilon}}_T (s, a) < \tau/2$, for all $s' \in S$, $\overset{\triangle}{\hat{T}} (s, a, s') > 0$ if and only if $T(s, a, s') > 0$.*

Proof. First consider the case when $T(s, a, s') = 0$; we will show that $\overset{\triangle}{\hat{T}} (s, a, s') = 0$. To see why, consider the $(\tilde{s}, \tilde{a})$ that contributed to this confidence interval as computed in the step of Algorithm 6. Since $\overset{\triangle}{\hat{\epsilon}}_T (s, a) < \tau/2$, from Algorithm 6, we have that $\Delta((s, a), (\tilde{s}, \tilde{a})) \leq \tau/2$. This implies that if $\tilde{s}' := \alpha((s, a, s'), (\tilde{s}, \tilde{a}))$, then $|T(\tilde{s}, \tilde{a}, \tilde{s}') - T(s, a, s')| \leq \tau/2$. Since we assumed $T(s, a, s') = 0$, we have that $|T(\tilde{s}, \tilde{a}, \tilde{s}')| \leq \tau/2$. However, by Assumption 4, we have that the range of T lies in $[0] \cup [\tau, 1]$, therefore, to satisfy the above inequality, we must have that $T(\tilde{s}, \tilde{a}, \tilde{s}') = 0$. This would imply that the empirical probability $\hat{T}(\tilde{s}, \tilde{a}, \tilde{s}')$ too is zero, which then is assigned to $\overset{\triangle}{\hat{T}} (s, a, s')$ in Algorithm 6. Thus, $\overset{\triangle}{\hat{T}} (s, a, s') = 0$.

Now consider the case when $T(s, a, s') > 0$. We will show that $\overset{\triangle}{\hat{T}} (s, a, s') > 0$. First, since $T(s, a, s') > 0$, by Assumption 4, it means that $T(s, a, s') \geq \tau$. As argued in the previous case, since $|T(\tilde{s}, \tilde{a}, \tilde{s}') - T(s, a, s')| \leq \tau/2$, we will also have $T(\tilde{s}, \tilde{a}, \tilde{s}') > \tau/2$. Again, by Assumption 4, this would imply $T(\tilde{s}, \tilde{a}, \tilde{s}') \geq \tau$. Note that, from Algorithm 6, we have that, since $\overset{\triangle}{\hat{\epsilon}}_T (s, a) < \tau/2$, $\hat{\epsilon}_T(\tilde{s}, \tilde{a}) < \tau/2$. And since confidence intervals are admissible, this means that $|T(\tilde{s}, \tilde{a}, \tilde{s}') - \hat{T}(\tilde{s}, \tilde{a}, \tilde{s}')| \leq \tau/2$. This, together with the fact that $T(\tilde{s}, \tilde{a}, \tilde{s}') > \tau/2$, means that $\hat{T}(\tilde{s}, \tilde{a}, \tilde{s}') > \tau/2$. In Algorithm 6, we would assign this value to $\overset{\triangle}{\hat{T}} (\tilde{s}, \tilde{a}, \tilde{s}')$, thus, resulting in $\overset{\triangle}{\hat{T}} (s, a, s') > 0$. $\square$

This lemma allows us to compute the support of state-action pairs we have never experienced. Algorithm 2 uses this idea to expand $\hat{Z}_{\text{safe}}$ in a way that retains closedness, thus, safety.

Lemma 3. *Assume our confidence intervals are admissible. Whenever Algorithm 1 calls Algorithm 2, in the final step of Algorithm 2, $\hat{Z}_{safe} \cup Z_{closed}$ is a safe set.*

Proof. We will prove this statement using induction. That is we assume that, before every call to Algorithm 2, $\hat{Z}_{\text{safe}}$ is a safe set. As the base case, this is satisfied in the first call because then, $\hat{Z}_{\text{safe}} = Z_0$ and we have assumed Z_0 to be a safe set in Assumption 1.

Since we populate $Z_{\text{candidate}}$ with only those state-action pairs with non-negative rewards, it follows directly from the run of the algorithm that all state-action pairs that are eventually found in Z_{closed} have non-negative rewards. Thus, to show that $Z_{\text{closed}} \cup \hat{Z}_{\text{safe}}$ is safe, we only need to show that $Z_{\text{closed}} \cup \hat{Z}_{\text{safe}}$ is a closed set. That is, we need to show that for any $(s, a) \in Z_{\text{closed}} \cup \hat{Z}_{\text{safe}}$, every possible next state has an action in $\hat{Z}_{\text{safe}} \cup Z_{\text{closed}}$ i.e., for all $s' \in \{s' \in S : T(s, a, s') > 0\}$, there exists an $a' \in A$ where $(s', a') \in \hat{Z}_{\text{safe}} \cup Z_{\text{closed}}$. From the induction assumption (that $\hat{Z}_{\text{safe}}$ is closed), this trivially holds for all $(s, a) \in \hat{Z}_{\text{safe}}$.

Hence, consider any $(s, a) \in Z_{\text{closed}}$. From our choice of $Z_{\text{candidate}}$ in the first step of the algorithm, we know that $\hat{\overset{\Delta}{\epsilon}}_T (s, a) < \tau/2$. Then from Lemma 2, we know that the set $\overset{\Delta}{\hat{S}}' = \{s' \in S : \overset{\Delta}{\hat{T}} (s, a, s') > 0\}$ is identical to the true support of the next state-action pairs of (s, a). But from the third inner while loop of our algorithm, we have that for all $s' \in \overset{\Delta}{\hat{S}}'$, we ensure that there exists an $a' \in A$ where $(s', a') \in Z_{\text{closed}} \cup \hat{Z}_{\text{safe}}$, implying that all possible next states of (s, a) have a corresponding action in $Z_{\text{closed}} \cup \hat{Z}_{\text{safe}}$. Thus, $Z_{\text{closed}} \cup \hat{Z}_{\text{safe}}$ is closed, and also, safe. $\qquad\square$

As a corollary of the above result, we can also show that $\hat{Z}_{\text{safe}}$ is closed even if we replaced the true transitions by some candidate transition.

Corollary 1. *Assume the confidence intervals are admissible. During the run of Algorithm 1, we always have that, for any $T^\dagger \in CI(\overset{\Delta}{\hat{T}})$ and for any $(s, a) \in \hat{Z}_{safe}$, if there exists $s' \in S$ such that $T^\dagger(s, a, s') > 0$, then $s' \in \hat{Z}_{safe}$.*

Proof. By design of Algorithm 2, we know that for any $(s, a) \in \hat{Z}_{\text{safe}}$, either $(s, a) \in Z_0$ or $\overset{\Delta}{\hat{\epsilon}} (s, a) < \tau/2$.

Consider the case where $(s, a) \in Z_0$. Since the confidence intervals are admissible, by the third requirement in the definition of the candidate transition set (Definition 9), we have that if $T^\dagger(s, a, s') > 0$, then $s' \in Z_0$. Since $Z_0 \subset \hat{Z}_{\text{safe}}$, $s' \in \hat{Z}_{\text{safe}}$.

Consider the case where $\overset{\Delta}{\hat{\epsilon}} (s, a) < \tau/2$. Since the confidence intervals are admissible, we have from Lemma 2 that if $T^\dagger(s, a, s') > 0$, then $T(s, a, s') > 0$. Since we have established in Lemma 3 that $\hat{Z}_{\text{safe}}$ is closed, this means that $s' \in \hat{Z}_{\text{safe}}$. $\qquad\square$

In order to make sure that our agent can continue exploring without ever getting stuck, we must ensure that whenever Algorithm 2 expands $\hat{Z}_{\text{safe}}$, it remains communicating.

Lemma 4. *Assume our confidence intervals are admissible. Whenever Algorithm 1 calls Algorithm 2, in the final step of Algorithm 2, $\hat{Z}_{safe} \cup Z_{closed}$ is communicating.*

Proof. We will prove this statement using induction. We first assume that, before every call to Algorithm 2, $\hat{Z}_{\text{safe}}$ is an communicating set, and using this, prove that the updated safe set, namely $\hat{Z}_{\text{safe}} \cup Z_{\text{closed}}$ computed at the end of Algorithm 2, is also communicating. As the base case, this is satisfied in the first call because $\hat{Z}_{\text{safe}} = Z_0$ and we have assumed Z_0 to be an communicating set in Assumption 1.

Informally, to show that $\hat{Z}_{\text{safe}}$ is communicating, we will first show that for every state in $\hat{Z}_{\text{safe}}$, the agent has a return policy which ensures that from anywhere in $\hat{Z}_{\text{safe}} \cup Z_{\text{closed}}$, it can reach that state with non-zero probability. As a second step, we will show that for every state in Z_{closed}, the agent has a 'reach' policy which ensures that from anywhere in $\hat{Z}_{\text{safe}} \cup Z_{\text{closed}}$ it can reach that state with non-zero probability. Finally, we will put these together to establish communicatingness.

Proof that Z_{closed} is returnable. As the first part of the proof, we will show that,

$$\forall \tilde{s} \in \hat{Z}_{\text{safe}} \ \exists \pi_{\text{return}} \in \Pi(\hat{Z}_{\text{safe}} \cup Z_{\text{closed}}), \quad s.t. \ \forall s \in \hat{Z}_{\text{safe}} \cup Z_{\text{closed}} \ \text{P}\left[\exists t, \ s_t = \tilde{s} \middle| \pi_{\text{return}}, s_0 = s\right] > 0. \quad (9)$$

Fix an $\tilde{s} \in \hat{Z}_{\text{safe}}$. We will prove the existence of a suitable π_{return} by induction. Consider the last outer iteration of our algorithm during which we know that $Z_{\text{returnable}}$ at the end of the second inner while block is identical to

Z_{closed} at the end of the third inner while block. In this round, consider some (s, a) that is about to be added to $Z_{\text{returnable}}$. For the induction hypothesis, we will consider a hypothesis that is stronger than the one above. In particular, assume that there exists π_{return} such that for every initial state s' currently in $\hat{Z}_{\text{safe}} \cup Z_{\text{returnable}}$, there is non-zero probability of returning to $\tilde{s}$, *while visiting only those states in $\hat{Z}_{safe} \cup Z_{returnable}$ on the way to $\tilde{s}$.* Formally, assume that $\exists \pi_{\text{return}} \in \Pi(\hat{Z}_{\text{safe}} \cup Z_{\text{closed}})$ such that:

$$\forall s' \in \hat{Z}_{\text{safe}} \cup Z_{\text{returnable}}, \ \mathrm{P}\left[\exists t, \ \substack{s_t = \tilde{s}, \\ \forall t' < t \ s_{t'} \in \hat{Z}_{\text{safe}} \cup Z_{\text{returnable}}} \ \middle| \ \pi_{\text{return}}, s_0 = s'\right] > 0. \tag{10}$$

This assumption is of course true initially when $\hat{Z}_{\text{safe}} \cup Z_{\text{returnable}} = \hat{Z}_{\text{safe}}$, by communicatingness of $\hat{Z}_{\text{safe}}$.

Now, by the manner in which the second while block works, we know that there exists an s' such that $\overset{\triangle}{\hat{T}}(s, a, s') > 0$ and there exists a' such that $(s', a') \in \hat{Z}_{\text{safe}} \cup Z_{\text{returnable}}$. Note that since $\overset{\triangle}{\hat{\epsilon}}_T(s, a) < \tau/2$ (by our initial choice of $Z_{\text{candidate}}$) and since $\overset{\triangle}{\hat{T}}(s, a, s') > 0$, it follows from Lemma 2 that $T(s, a, s') > 0$.

Next, consider π'_{return} that is identical to π everywhere, except $\pi'_{\text{return}}(s) = a$. Note that since $(s, a) \in \hat{Z}_{\text{safe}} \cup Z_{\text{closed}}$, $\pi'_{\text{return}} \in \Pi(\hat{Z}_{\text{safe}} \cup Z_{\text{closed}})$.

First, we have that the induction assumption still holds for every $s' \in \hat{Z}_{\text{safe}} \cup Z_{\text{returnable}}$. That is, for every $s' \in \hat{Z}_{\text{safe}} \cup Z_{\text{returnable}}$, with non-zero probability, π'_{return} starts from s' to return to $\tilde{s}$, without visiting any state outside $\hat{Z}_{\text{safe}} \cup Z_{\text{returnable}}$. This is because, for a given random seed, if the agent were to follow π_{return} from s to return to s'', without visiting any states outside of $\hat{Z}_{\text{safe}} \cup Z_{\text{returnable}}$, the agent would do the same under π'_{return} since the two policies would agree on all the visited states.

Next, we have the following when starting from s:

$$\mathrm{P}\left[\exists t, \ \substack{s_t = \tilde{s} \\ \forall t' < t \ s_{t'} \in \hat{Z}_{\text{safe}} \cup Z_{\text{returnable}}} \ \middle| \ \pi'_{\text{return}}, s_0 = s\right] \geq T(s, a, s')\mathrm{P}\left[\exists t, \ \substack{s_t = \tilde{s} \\ \forall t' < t \ s_{t'} \in \hat{Z}_{\text{safe}} \cup Z_{\text{returnable}}} \ \middle| \ \pi'_{\text{return}}, s_0 = s'\right]$$
$$\geq T(s, a, s')\mathrm{P}\left[\exists t, \ \substack{s_t = \tilde{s} \\ \forall t' < t \ s_{t'} \in \hat{Z}_{\text{safe}} \cup Z_{\text{returnable}}} \ \middle| \ \pi_{\text{return}}, s_0 = s'\right]$$
$$> 0.$$

Here, the first inequality simply follows from the fact that one possible way to reach $\tilde{s}$ from s, is by first taking a step to s'. In the second step, we were able to replace π'_{return} with π_{return} by a similar logic as before. Specifically, for a given random seed, if an agent starts from s' to reach $\tilde{s}$ following π_{return} without ever visiting s, it should do the same under π'_{return} too.

Finally, we have that both the above terms are strictly positive. Hence, we establish that π'_{return}, with non-zero probability, allows the agent to return to $\tilde{s}$, while never visiting any state outside $\hat{Z}_{\text{safe}} \cup Z_{\text{returnable}} \cup \{(s, a)\}$.

Proof for $Z_{\textbf{closed}}$ is reachable. For the second part of the proof, we will show that $\forall s \in Z_{\text{closed}}$,

$$\exists \pi_{\text{reach}} \in \Pi(Z_{\text{closed}} \cup \hat{Z}_{\text{safe}}), \ s.t. \forall \tilde{s} \in \hat{Z}_{\text{safe}} \cup Z_{\text{closed}} \ \mathrm{P}[\exists t, s_t = s | s_0 = \tilde{s}, \pi_{\text{reach}}] > 0. \tag{11}$$

We will prove this by induction. Consider the last outer iteration of the algorithm and consider the first inner while loop where we populate $Z_{\text{reachable}}$. Note that since this is the last outer iteration, at the end of this while block, $Z_{\text{reachable}}$ is exactly equal to Z_{closed} that is output at the end of the algorithm. (Thus, during the run of this while loop, we always have that $Z_{\text{reachable}} \subset Z_{\text{closed}}$.) In this while loop, consider the instant at which some (s', a') is about to be added to $Z_{\text{reachable}}$. We will assume by induction that for all $\tilde{s}$ that are currently in $\hat{Z}_{\text{safe}} \cup Z_{\text{reachable}}$, Equation 11 holds i.e., $\exists \pi_{\text{reach}}$ such that from anywhere in $\hat{Z}_{\text{safe}} \cup Z_{\text{reachable}}$, we can reach s with non-zero probability.

As the base case, because of how we have initialized $Z_{\text{reachable}}$, we have that for all $s \in \hat{Z}_{\text{safe}} \cup Z_{\text{reachable}}$, $s \in \hat{Z}_{\text{safe}}$. Thus, for this case, the induction assumption holds from the fact that we have proven returnability to $\hat{Z}_{\text{safe}}$.

Now, let us turn to the point when the algorithm is about to add (s', a') to $Z_{\text{reachable}}$. Then, note that this means that there exists $(s, a) \in \hat{Z}_{\text{safe}} \cup Z_{\text{reachable}}$ such that $\overset{\triangle}{\hat{T}}(s, a, s') > 0$. Since $\overset{\triangle}{\hat{\epsilon}}_T(s, a) < \tau/2$ (by our initial choice of $Z_{\text{candidate}}$) and since $\overset{\triangle}{\hat{T}}(s, a, s') > 0$, by Lemma 2, we have that $T(s, a, s') > 0$.

Next, consider the policy π_{reach} guaranteed by our induction assumption, to reach s from anywhere in $\hat{Z}_{\text{safe}} \cup Z_{\text{closed}}$. Then, define a policy π'_{reach} which is identical to π_{reach} on all states, except that $\pi'_{\text{reach}}(s) = a$. Note that since $(s, a) \in \hat{Z}_{\text{safe}} \cup Z_{\text{reachable}} \subset \hat{Z}_{\text{safe}} \cup Z_{\text{closed}}$, and since $\pi_{\text{reach}} \in \Pi(\hat{Z}_{\text{safe}} \cup Z_{\text{closed}})$, we have $\pi'_{\text{reach}} \in \Pi(\hat{Z}_{\text{safe}} \cup Z_{\text{closed}})$.

Now, we can show that π'_{reach} reaches s' from any $\tilde{s} \in \hat{Z}_{\text{safe}} \cup Z_{\text{closed}}$ because of the following:

$$\begin{aligned}
\mathrm{P}\left[\exists t, s_t = s' \middle| \pi'_{\text{reach}}, s_0 = \tilde{s}\right] &\geq T(s, a, s') Pr\left[\exists t, s_{t-1} = s \middle| \pi'_{\text{reach}}, s_0 = \tilde{s}\right] \\
&\geq T(s, a, s') Pr\left[\exists t, s_{t-1} = s \middle| \pi_{\text{reach}}, s_0 = \tilde{s}\right] \\
&> 0.
\end{aligned}$$

Here, the first inequality comes from the fact that one way to reach s' is by traveling to s and then taking the action a. In the next inequality, we make use of the fact that, for a given random seed, if the agent follows π_{reach} from $\tilde{s}$ to visit s for the first time, it would follow the same steps to reach s even under π'_{reach}, since the policies would agree until then. Finally, we have from our induction assumption that the probability term in the penultimate line is strictly positive; since the transition probability is strictly positive too, the last inequality holds.

This proves that $\pi'_{\text{reach}} \in \Pi(\hat{Z}_{\text{safe}} \cup Z_{\text{closed}})$ reaches s' from anywhere in $\hat{Z}_{\text{safe}} \cup Z_{\text{closed}}$ with non-zero probability.

Proof for communicatingness. Finally, we will wrap the above results to establish communicatingness. From the above results, we have that $\forall s' \in \hat{Z}_{\text{safe}} \cup Z_{\text{closed}}$:

$$\exists \pi_{\text{visit}} \in \Pi(\hat{Z}_{\text{safe}} \cup Z_{\text{closed}}), \ s.t. \forall s \in \hat{Z}_{\text{safe}} \cup Z_{\text{closed}} \ \ \mathrm{P}\left[\exists t, \ s_t = s' \middle| \pi_{\text{visit}}, s_0 = s\right] > 0.$$

To establish communicatingness, we need to show that this probability is in fact 1. To do this, we will first note that $\pi_{\text{visit}} \in \Pi(\hat{Z}_{\text{safe}} \cup Z_{\text{closed}})$ and since $\hat{Z}_{\text{safe}} \cup Z_{\text{closed}}$ is closed (as proven in Lemma 3), then use Lemma 5 to establish communicatingness.

$\square$

Corollary 2. *Assume our confidence intervals are admissible. Whenever Algorithm 1 calls Algorithm 2, in the final step of Algorithm 2, $\hat{Z}_{safe} \cup Z_{closed} \subset Z_{safe}$.*

Proof. Note that $Z_0 \subset \hat{Z}_{\text{safe}} \cup Z_{\text{closed}}$ by construction. Then, since we have established communicatingness of $\hat{Z}_{\text{safe}} \cup Z_{\text{closed}}$ in Lemma 4, for every $(s, a) \in \hat{Z}_{\text{safe}} \cup Z_{\text{closed}}$, there should exist a policy π_{return} that travels from (s, a) and goes to any states in Z_0 with probability 1. Furthermore since we established safety in Lemma 3, this also means that every state-action pair visited in this path has non-negative reward. Thus, by definition of Z_{safe}, the claim follows. $\square$

Although our definition of communicating, Def 2, seems strict since its guarantee must hold with probability 1, we show here that this is no stronger than having the guarantee simply hold with positive probability. This lemma helps us prove that our definition of communicating is equivalent to that of the standard definition (see Fact 2) as well as help prove that the safe set we construct is indeed communicating (see Lemma 4).

Lemma 5. *If there exists a closed set of state-action pairs Z such that $\forall s' \in Z$:*

$$\exists \ \pi_{visit} \in \Pi(Z), \ s.t. \forall s \in Z \ \ \mathrm{P}\left[\exists t, \ s_t = s' \middle| \pi_{visit}, s_0 = s\right] > 0$$

then it must be the case that this probability is, in fact, equal to 1; specifically, $\forall s' \in Z$:

$$\exists \ \pi_{visit} \in \Pi(Z), \ s.t. \forall s \in Z \ \ \mathrm{P}\left[\exists t, \ s_t = s' \middle| \pi_{visit}, s_0 = s\right] = 1$$

Proof. From the above, we have that for a given $s' \in Z$, there must exists a constant $H \geq 0$ and a constant $p > 0$ such that:

$$\forall s \in Z \ \ \mathrm{P}\left[\exists t \leq H, \ s_t = s' \middle| \pi_{\text{visit}}, s_0 = s\right] \geq p. \tag{12}$$

We will start by equating the probability of never visiting s' by decomposing the trajectory into a prefix of H steps and the rest, and then applying the Markov property, as follows:

$$\mathrm{P}[\forall t, s_t \neq s' \mid \pi_{\mathrm{visit}}, s_0 = s] = \sum_{s'' \in S} \mathrm{P}[\forall t, s_t \neq s' \mid \pi_{\mathrm{visit}}, s_0 = s''] \mathrm{P}\left[\begin{smallmatrix} \forall t \leq H, s_t \neq s' \\ s_H = s'' \end{smallmatrix} \middle| \pi_{\mathrm{visit}}, s_0 = s \right].$$

Note that since $\pi_{\mathrm{visit}} \in \Pi(Z)$ and since Z is closed, we have from Fact 1 that s'', which is the Hth state in the trajectory, satisfies $s'' \in Z$. Therefore, we can restrict the summation to Z (the remaining terms would zero out). Hence,

$$\begin{aligned}
&\mathrm{P}[\forall t, s_t \neq s' \mid \pi_{\mathrm{visit}}, s_0 = s] \\
&= \sum_{s'' \in Z} \mathrm{P}[\forall t, s_t \neq s' \mid \pi_{\mathrm{visit}}, s_0 = s''] \mathrm{P}\left[\begin{smallmatrix} \forall t \leq H, s_t \neq s' \\ s_H = s'' \end{smallmatrix} \middle| \pi_{\mathrm{visit}}, s_0 = s \right] \\
&\leq \left(\max_{s'' \in Z} \mathrm{P}[\forall t, s_t \neq s' \mid \pi_{\mathrm{visit}}, s_0 = s''] \right) \left(\sum_{s'' \in Z} \mathrm{P}\left[\begin{smallmatrix} \forall t \leq H, s_t \neq s' \\ s_H = s'' \end{smallmatrix} \middle| \pi_{\mathrm{visit}}, s_0 = s \right] \right) \\
&= \left(\max_{s'' \in Z} \mathrm{P}[\forall t, s_t \neq s' \mid \pi_{\mathrm{visit}}, s_0 = s''] \right) \left(\mathrm{P}\left[\forall t \leq H, s_t \neq s' \mid \pi_{\mathrm{visit}}, s_0 = s \right] \right) \\
&\leq \left(\max_{s'' \in Z} \mathrm{P}[\forall t, s_t \neq s' \mid \pi_{\mathrm{visit}}, s_0 = s''] \right) (1 - p)
\end{aligned}$$

Since the above inequality holds for any $s \in Z$, we can apply a $\max_{s \in Z}$ on the left hand side and rearrange to get:

$$\max_{s'' \in Z} \mathrm{P}[\forall t, s_t \neq s' \mid \pi_{\mathrm{visit}}, s_0 = s''] \cdot p \leq 0$$

Since $p > 0$ (see Equation 12), this means that the first term here is equal to zero. In other words, with probability 1, $\exists t$ such that $s_t = s'$ when the agent starts from any s and follows π_{visit}, as claimed. $\qquad \square$

Finally, we want to show that Algorithm 2 computes the largest possible $\hat{Z}_{\mathrm{safe}}$ that still retains safety and communicatingness. This is necessary since, if this were not true, we could get into a situation where we know enough to ensure we can perform the optimal policy, but our agent remains trapped in $\hat{Z}_{\mathrm{safe}}$ forever.

Lemma 6. *Assume that the confidence intervals are admissible. Consider some call of Algorithm 2 while executing Algorithm 1. Consider $(\tilde{s}, \tilde{a}) \notin \hat{Z}_{safe}$ such that $\tilde{s} \in \hat{Z}_{safe}$. Let $\exists \pi_{return}$ such that starting at $(\tilde{s}, \tilde{a})$, π_{return} reaches a state in $\hat{Z}_{safe}$ with probability 1. Let $\tilde{Z}$ be the set of state-action pairs visited by the agent starting $(\tilde{s}, \tilde{a})$ following π_{return} before reaching $\hat{Z}_{safe}$. Formally, let:*

$$\tilde{Z} = \{(s, a) \in S \times A : \mathrm{P}\left[\exists t \geq 0 \begin{smallmatrix} (s_t, a_t) = (s, a) \\ \forall t' < t \ s_{t'} \notin \hat{Z}_{safe} \end{smallmatrix} \middle| (s_0, a_0) = (\tilde{s}, \tilde{a}), \pi_{return} \right] > 0\}.$$

In the final step of Algorithm 2, if $\forall (s, a) \in \tilde{Z} \setminus \hat{Z}_{safe}, \overset{\Delta}{\epsilon}_T (s, a) < \tau/2$, then $\tilde{Z} \subset \hat{Z}_{safe} \cup Z_{closed}$.

Proof. Informally, we will show that in each iteration of the algorithm, after the first inner while block, we have $\tilde{Z} \subset \hat{Z}_{\mathrm{safe}} \cup Z_{\mathrm{reachable}}$; and using this, we will show that after the second inner while block, we have $\tilde{Z} \subset \hat{Z}_{\mathrm{safe}} \cup Z_{\mathrm{returnable}}$; and finally, because of this, after the third inner while block, $\tilde{Z} \subset \hat{Z}_{\mathrm{safe}} \cup Z_{\mathrm{closed}}$. Below, we prove these three statements, and finally wrap them up to prove the main claim.

Proof for $\tilde{Z} \subset \hat{Z}_{\mathbf{safe}} \cup Z_{\mathbf{reachable}}$. Assume there exists $(\tilde{s}, \tilde{a})$ as specified in the lemma statement. Then, formally, we will show that if $\tilde{Z} \subset \hat{Z}_{\mathrm{safe}} \cup Z_{\mathrm{candidate}}$ before the beginning of the first inner while loop, then $\tilde{Z} \subset \hat{Z}_{\mathrm{safe}} \cup Z_{\mathrm{reachable}}$ at the end of the loop.

We will prove this by contradiction. Assume for the sake of contradiction that there exists a non-empty $\tilde{Z}_{\mathrm{bad}} \subset \tilde{Z}$ such that $\tilde{Z}_{\mathrm{bad}}$ has no intersection with $\hat{Z}_{\mathrm{safe}} \cup Z_{\mathrm{reachable}}$. Let $\tilde{Z}_{\mathrm{good}} = \tilde{Z} \setminus \tilde{Z}_{\mathrm{bad}}$. First we note why $\tilde{Z}_{\mathrm{good}}$ is non-empty. Since $\tilde{Z} \setminus \hat{Z}_{\mathrm{safe}} \subset Z_{\mathrm{candidate}}$ (from our initial assumption), since $(\tilde{s}, \tilde{a}) \in \tilde{Z} \setminus \hat{Z}_{\mathrm{safe}}$, and since $\tilde{s} \in \hat{Z}_{\mathrm{safe}}$ (from lemma statement), when initializing $Z_{\mathrm{reachable}}$ with $\{(s, a) \in Z_{\mathrm{candidate}} : s \in \hat{Z}_{\mathrm{safe}}\}$, we would add $(\tilde{s}, \tilde{a})$ to $Z_{\mathrm{reachable}}$. Thus, $\tilde{Z}_{\mathrm{good}}$ must contain at least $(\tilde{s}, \tilde{a})$.

Next, we argue that there must exist some $(s', a') \in \tilde{Z}_{\text{bad}}$ such that there exists an $(s, a) \in \tilde{Z}_{\text{good}}$ for which $T(s, a, s') > 0$ and $s \notin \hat{Z}_{\text{safe}}$. If not, then starting from $(\tilde{s}, \tilde{a}) \in \tilde{Z}_{\text{good}}$, the agent can never hope to reach $\tilde{Z}_{\text{bad}}$ before visiting a state $\hat{Z}_{\text{safe}}$, contradicting the fact that $\tilde{Z}_{\text{bad}}$ is a subset of state-action pairs that it visits before reaching $\hat{Z}_{\text{safe}}$.

Now, consider such an (s', a') (which was not added to $Z_{\text{reachable}}$) and its predecessor (s, a), which belongs to $\hat{Z}_{\text{safe}} \cup Z_{\text{reachable}}$ because it belongs to $\tilde{Z}_{\text{good}}$. In some iteration of this while loop, we must have examined $(s, a) \in \hat{Z}_{\text{safe}} \cup Z_{\text{reachable}}$. Since $s \notin \hat{Z}_{\text{safe}}$, we also have $(s, a) \notin \hat{Z}_{\text{safe}}$. Since $(s, a) \in \tilde{Z}_{\text{good}}$, and $\tilde{Z}_{\text{good}} \subset \tilde{Z}$, this further means that $(s, a) \in \tilde{Z} \setminus \hat{Z}_{\text{safe}}$. From our lemma statement, we then have that the confidence interval of (s, a) is less than $\tau/2$. Hence $\overset{\Delta}{\epsilon}_T (s, a) < \tau/2$. Then, from Lemma 2, since $T(s, a, s') > 0$, we have that $\overset{\Delta}{T} (s, a, s') > 0$. Then, since $(s', a') \in Z_{\text{candidate}}$, we would have added (s', a') to $Z_{\text{reachable}}$ in this iteration, contradicting the fact that we never added it to $Z_{\text{reachable}}$ in the first place. Thus, $\tilde{Z}_{\text{bad}}$ must be empty, implying that $\tilde{Z} \subset \hat{Z}_{\text{safe}} \cup Z_{\text{reachable}}$.

Proof for $\tilde{Z} \subset \hat{Z}_{\text{safe}} \cup Z_{\text{returnable}}$. Formally, we will show that if $\tilde{Z} \subset \hat{Z}_{\text{safe}} \cup Z_{\text{reachable}}$ before the beginning of the second while block, then $\tilde{Z} \subset \hat{Z}_{\text{safe}} \cup Z_{\text{returnable}}$ at the end of the block.

At the end of the second block, let us define $\tilde{Z}_{\text{bad}} := \tilde{Z} \setminus (\hat{Z}_{\text{safe}} \cup Z_{\text{returnable}})$. We want to show that $\tilde{Z}_{\text{bad}}$ is empty, but assume on the contrary it is not. First we argue that there must exist $(s, a) \in \tilde{Z}_{\text{bad}}$ such that one of its next states belongs to $\hat{Z}_{\text{safe}} \cup Z_{\text{returnable}}$. If this was not the case, then whenever the agent enters $\tilde{Z}_{\text{bad}}$, it will never be able to return to $\hat{Z}_{\text{safe}}$. This is however in contradiction to the definition of $\tilde{Z}$.

For the rest of the discussion, consider such an $(s, a) \in \tilde{Z}_{\text{bad}}$ such that $\exists (s', a') \in \hat{Z}_{\text{safe}} \cup Z_{\text{returnable}}$ for which $T(s, a, s') > 0$. Since $(s, a) \in \tilde{Z}_{\text{bad}}$ and $Z_{\text{bad}} \subset \tilde{Z} \setminus \hat{Z}_{\text{safe}}$, it means that $(s, a) \in \tilde{Z} \setminus \hat{Z}_{\text{safe}}$. Then, from our lemma statement we have $\overset{\Delta}{\epsilon}_T (s, a) \leq \tau/2$. Now, from Lemma 2, we have that since the confidence intervals are admissible and since $T(s, a, s') > 0$, it must be the case that $\overset{\Delta}{T} (s, a, s') > 0$. Then, in the iteration of the while loop during which (s', a') is present in $\hat{Z}_{\text{safe}} \cup Z_{\text{returnable}}$, (s, a) would in fact be added to $Z_{\text{returnable}}$. This, however, contradicts our assumption that $(s, a) \in \tilde{Z}_{\text{bad}}$. Therefore, $\tilde{Z}_{\text{bad}}$ should in fact be empty. Thus, $\tilde{Z} \subset \hat{Z}_{\text{safe}} \cup Z_{\text{returnable}}$ at the end of this while block.

Proof for $\tilde{Z} \subset \hat{Z}_{\text{safe}} \cup Z_{\text{closed}}$. Formally, we will show that if $\tilde{Z} \subset \hat{Z}_{\text{safe}} \cup Z_{\text{returnable}}$ before the beginning of the third while block, then $\tilde{Z} \subset \hat{Z}_{\text{safe}} \cup Z_{\text{closed}}$ at the end of the block.

Assume on the contrary that there exists $(s, a) \in \tilde{Z}$ such that $(s, a) \notin \hat{Z}_{\text{safe}} \cup Z_{\text{closed}}$ at the end of the third block. Since Z_{closed} is initialized with all of $\tilde{Z}$ contained in it, consider the *first* such (s, a) that is removed from Z_{closed} during the course of this second inner iteration. Now, just before the moment at which (s, a) is removed, by design of the algorithm, we would have that there exists s' such that $\overset{\Delta}{T} (s, a, s') > 0$ and $s' \notin \hat{Z}_{\text{safe}} \cup Z_{\text{closed}}$. Note that at this point, we also still have $\tilde{Z} \subset \hat{Z}_{\text{safe}} \cup Z_{\text{closed}}$. Therefore, this means that $s' \notin Z$. Now, we have that $(s, a) \in \tilde{Z} \setminus \hat{Z}_{\text{safe}}$, which means, by the lemma statement, $\overset{\Delta}{\epsilon}_T (s, a) \leq \tau/2$. Then, since $\overset{\Delta}{T} (s, a, s') > 0$, from Lemma 2, we have $T(s, a, s') > 0$. However, this contradicts the fact, if $(s, a) \in \tilde{Z}$, then every next state of (s, a) must lie in $\tilde{Z}$. Thus, no (s, a) belonging to $\tilde{Z}$ must have been removed from $\hat{Z}_{\text{safe}} \cup Z_{\text{closed}}$ during this while block, implying that $\tilde{Z} \subset \hat{Z}_{\text{safe}} \cup Z_{\text{closed}}$ at the end of this block.

Proof of main claim. From the above arguments, we have that whenever the outer iteration begins with $\tilde{Z} \subset Z_{\text{candidate}} \cup \hat{Z}_{\text{safe}}$ it ends with $\tilde{Z} \subset Z_{\text{candidate}} \cup Z_{\text{closed}}$. Now, at the beginning of Algorithm 2, we must have $\tilde{Z} \setminus \hat{Z}_{\text{safe}} \subset Z_{\text{candidate}}$ due to the fact that all elements of $\tilde{Z}$ have confidence intervals at most $\tau/2$. In other words, $\tilde{Z} \subset Z_{\text{candidate}} \cup \hat{Z}_{\text{safe}}$. Then, from the above arguments, we have that $\tilde{Z} \subset \hat{Z}_{\text{safe}} \cup Z_{\text{closed}}$ at the end of the first iteration. Since $Z_{\text{candidate}}$ at the beginning of the second outer iteration is equal to Z_{closed} from the end of the previous outer iteration, we again have $\tilde{Z} \subset Z_{\text{candidate}} \cup \hat{Z}_{\text{safe}}$ at the beginning of the second iteration. Thus, by a similar argument we have that the algorithm preserves the condition that $\tilde{Z} \subset Z_{\text{closed}} \cup \hat{Z}_{\text{safe}}$ at the end of every outer iteration, proving our main claim.

$\square$

D.2 Proofs about Algorithm 4 and Algorithm 3

In the next few lemmas, we prove results about Algorithm 4 and Algorithm 3. Recall that Algorithm 4 takes as input a set of edge state-action pairs (which are state-action pairs that do not belong to $\hat{Z}_{\text{safe}}$ but whose state belongs to $\hat{Z}_{\text{safe}}$) and outputs a set of elements from $\hat{Z}_{\text{safe}}$ which need to be explored in order to learn the return paths from the edges. Also recall that the idea of Algorithm 3 is to return an updated $\overline{Z}_{\text{goal}}$ (the set of states in the optimistic goal path) and Z_{explore} (by making a call to Algorithm 4).

In Lemma 7 we demonstrate that Algorithm 4 terminates in polynomial time.

Lemma 8 helps establish that running Algorithm 4 allows the agent progress without getting stuck. More concretely, in Lemma 8 we argue that the elements of Z_{explore} do not belong to K. That is, the elements of Z_{explore} are those that have not already been explored (as otherwise, the agent may be stuck perpetually in exploring what has already been explored).

In Lemma 9, we establish that Algorithm 4 works correctly (and hence, so does Algorithm 3). In particular, we show that Algorithm 4 does not terminate with an empty Z_{explore} when some of the edge state-action pairs of $\hat{Z}_{\text{safe}}$ are indeed safe. If we did not have this guarantee, then it's possible that even though there are some edge state-action pairs are safe, our agent may be stuck without exploring any state-action pair within $\hat{Z}_{\text{safe}}$. On the other hand, with this guarantee, we can be confident that our agent will explore to learn the return path of such edge states, and then establish their safety, using which it can then expand $\hat{Z}_{\text{safe}}$ in the future. As a corollary of this guarantee, we show that Algorithm 3 always ensures that $\hat{Z}_{\text{unsafe}}$ contains no element that actually belongs to Z_{safe}.

Finally, in Lemma 10, we prove that Algorithm 3 terminates in polynomial time, guaranteeing that either Z_{explore} is non-empty (which means the agent can explore $\hat{Z}_{\text{safe}}$ to learn a return path and expand $\hat{Z}_{\text{safe}}$) or that $\overline{Z}_{\text{goal}}$ has been updated in a way that $\overline{Z}_{\text{goal}} \subset \hat{Z}_{\text{safe}}$ (which means that the agent can stop exploring, and instead, start exploiting).

Below, we prove our result about the run-time complexity of Algorithm 4.

Lemma 7. *Algorithm 4 terminates in poly($|S|, |A|$) time.*

Proof. Recall that Algorithm 4 executes an iteration of a while loop whenever $Z^L_{\text{next}} \neq \emptyset$ and $Z_{\text{explore}} = \emptyset$. Also recall that inside the while loop, the algorithm executes a for loop that iterates over all elements of Z^L_{next}. Hence, during the while loop, since Z^L_{next} is non-empty, we must also execute at least one iteration of the inner for loop. Now, note that, by design of the algorithm, Z^L_{next} is populated only with elements that do not belong to Z_{return}. Since the for loop adds all these elements to Z_{return}, every call to the for loop corresponds to increasing the cardinality of Z_{return} by at least one. Thus, there can be at most as many executions of the for loop as there are state-action pairs, $|S| \times |A|$. By extension, the while loop can be executed at most $|S| \times |A|$ times, after which it should terminate. $\square$

Next, we show that the state-action pairs marked for exploration by Algorithm 4 have not already been explored well before.

Lemma 8. *Algorithm 4 returns $Z_{explore}$ such that for every $(\tilde{s}, \tilde{a}) \in Z_{explore}$, $(\tilde{s}, \tilde{a}) \in K^c$ where $K = \{(s, a) \in S \times A : n(s, a) \geq m\}$ when $m \geq O\left(\frac{|S|}{\tau^2} + \frac{1}{\tau^2} \ln \frac{|S||A|}{\tau^2 \delta}\right)$.*

Proof. Assume on the contrary that there exists $(\tilde{s}, \tilde{a}) \in Z_{\text{explore}}$ such that $(\tilde{s}, \tilde{a}) \in K$. By design of Algorithm 4, we have that there exists $(s, a) \notin \hat{Z}_{\text{safe}}$ which resulted in the addition of $(\tilde{s}, \tilde{a})$ to Z_{explore}. In particular, we would have that $\overset{\Delta}{\hat{\epsilon}}_T (s, a) > \tau/2$ and $\Delta((s, a), (\tilde{s}, \tilde{a})) < \tau/2$. Since $(\tilde{s}, \tilde{a}) \in K$, we also have $\hat{\epsilon}_T(\tilde{s}, \tilde{a}) \leq \tau/4$ (and this follows from Lemma 16). Then, we would have $\overset{\Delta}{\hat{\epsilon}}_T (s, a) \leq \hat{\epsilon}_T(\tilde{s}, \tilde{a}) + \Delta((s, a), (\tilde{s}, \tilde{a}))$, implying $\overset{\Delta}{\hat{\epsilon}}_T (s, a) \leq \tau/4$ which is a contradiction. Thus, the above claim is correct. $\square$

Below, we show that as long as there is a edge to $\hat{Z}_{\text{safe}}$ with a safe return path to $\hat{Z}_{\text{safe}}$, Algorithm 4 will return with a non-empty Z_{explore}.

Lemma 9. *Assume our confidence intervals are admissible. During a run of Algorithm 1, at the end of every call to Algorithm 4, we will have $Z_{explore} = \emptyset$ only if $\forall\ (\tilde{s}, \tilde{a}) \in Z_{edge}$, $(\tilde{s}, \tilde{a}) \notin Z_{safe}$. As a corollary of this, we always have that $\hat{Z}_{unsafe} \cap Z_{safe} = \emptyset$.*

Proof. We will prove this by induction. Specifically, since Algorithm 3 is the only block where we call Algorithm 4 and modify $\hat{Z}_{\text{unsafe}}$, we will consider a particular iteration of the while loop in Algorithm 3. Then, we will assume that in all the previous iterations of this while loop, when Algorithm 4 was called, it satisfied the above guarantee. Additionally, we will assume that $\hat{Z}_{\text{unsafe}}$ satisfies $\hat{Z}_{\text{unsafe}} \cap Z_{\text{safe}} = \emptyset$ in the beginning of this loop. Then, we will show that the guarantee about Algorithm 4 is satisfied even when we call it in this loop, and that, by the end of this loop, $\hat{Z}_{\text{unsafe}}$ continues to satisfy $\hat{Z}_{\text{unsafe}} \cap Z_{\text{safe}} = \emptyset$.

As the base case, in the first iteration, since we have never called Algorithm 4 before, the induction hypothesis about Algorithm 4 is trivially satisfied. Furthermore, since $\hat{Z}_{\text{unsafe}}$ is initialized to be empty set, we again trivially have $\hat{Z}_{\text{unsafe}} \cap Z_{\text{safe}} = \emptyset$ in the beginning of this loop.

Now, consider any arbitrary iteration of the while loop of Algorithm 3. Through the next few paragraphs below, we will argue why the call to Algorithm 4 in this loop, satisfies the above specified guarantee. In the final paragraph, we provide a simple argument showing why $\hat{Z}_{\text{unsafe}} \cap Z_{\text{safe}} = \emptyset$ at the end of the loop.

Proof for claim about Algorithm 4. Assume that during this particular call to Algorithm 4, there exists $(\tilde{s}, \tilde{a}) \in Z_{\text{edge}}$ such that $(\tilde{s}, \tilde{a}) \in Z_{\text{safe}}$. To prove the above claim, we only need to show that in this case Algorithm 4 will result in a non-empty Z_{explore}. So, for the sake of contradiction, we will assume that Z_{explore} is empty after the execution of Algorithm 4.

The outline of our idea is to make use of the fact that by Assumption 5 we are guaranteed a return path from $(\tilde{s}, \tilde{a})$ that is sufficiently similar to state-action pairs in $\hat{Z}_{\text{safe}}$. We will then show that we can pick a particular trajectory from this return path which visits a 'bad' state-action pair – a state-action pair whose counterpart in $\hat{Z}_{\text{safe}}$ has not been explored sufficiently. Then, under the assumption that Z_{explore} remains empty, we will argue that Algorithm 4 will visit all the state-action pairs in this trajectory, and finally, when it encounters the bad state-action pair, the algorithm will add the counterpart of this bad pair to Z_{explore}, reaching a contradiction.

First, let us apply Assumption 5 to $(\tilde{s}, \tilde{a})$ which guarantees a return path from $(\tilde{s}, \tilde{a})$ because it is an edge state-action i.e., $\tilde{s} \in \hat{Z}_{\text{safe}}$ and $(\tilde{s}, \tilde{a}) \notin \hat{Z}_{\text{safe}}$. But to apply this assumption, we must establish that $Z_0 \subset \hat{Z}_{\text{safe}}$. This is indeed true as it follows from how Algorithm 1 initializes $\hat{Z}_{\text{safe}}$ with Z_0 and every call to Algorithm 2 only adds elements to $\hat{Z}_{\text{safe}}$.

Now, consider the π_{return} guaranteed by Assumption 5. Starting from (s, a), and following π_{return}, the agent returns to Z_0 with probability 1. Furthermore, if we define the set of state-action pairs visited by π_{return} on its way to $\hat{Z}_{\text{safe}}$ as:

$$\tilde{Z} = \{(s, a) \in S \times A : \mathrm{P}\left[\exists t \geq 0 \ _{\forall t' < t\ s_{t'} \notin \hat{Z}_{\text{safe}}}^{(s_t, a_t) = (s, a)} \ \middle|\ (s_0, a_0) = (\tilde{s}, \tilde{a}), \pi_{\text{return}}\right] > 0\},$$

then we are given that every element of $\tilde{Z} \setminus \hat{Z}_{\text{safe}}$ corresponds to an element $(s', a') \in \hat{Z}_{\text{safe}}$ such that $\Delta((s, a), (s', a')) \leq \tau/4$.

To make our discussion easier, let us partition the elements of $\tilde{Z} \setminus \hat{Z}_{\text{safe}}$ into two sets $\tilde{Z}_{\text{good}}$ and $\tilde{Z}_{\text{bad}}$ as follows, depending on whether or not the corresponding (s', a') has been explored sufficiently well or not:

$$\tilde{Z}_{\text{good}} = \{(s, a) \in \tilde{Z} \setminus \hat{Z}_{\text{safe}} \mid \exists(s', a') \in S \times A \text{ s.t. } \Delta((s, a), (s', a')) + \hat{\epsilon}_T(s', a') \leq \tau/2\}$$

and

$$\tilde{Z}_{\text{bad}} = \{(s, a) \in \tilde{Z} \setminus \hat{Z}_{\text{safe}} \mid \forall(s', a') \in S \times A \text{ s.t. } \Delta((s, a), (s', a')) + \hat{\epsilon}_T(s', a') > \tau/2\}.$$

Note that every element of $\tilde{Z} \setminus \hat{Z}_{\text{safe}}$ belongs to exactly one of $\tilde{Z}_{\text{good}}$ and $\tilde{Z}_{\text{bad}}$. Also note that for all $(s, a) \in \tilde{Z}_{\text{good}}$, $\overset{\Delta}{\hat{\epsilon}}_T(s, a) \leq \tau/2$ since $\Delta((s, a), (s', a')) + \hat{\epsilon}_T(s', a') \leq \tau/2$. However, for all $(s, a) \in \tilde{Z}_{\text{bad}}$, since there exists no sufficiently explored (s', a') that is also sufficiently smaller, $\overset{\Delta}{\hat{\epsilon}}_T(s, a) > \tau/2$.

Next, we argue that there must exist at least one element in $\tilde{Z}_{\text{bad}}$. If this was not the case, then we would have that all elements in $\tilde{Z} \setminus \hat{Z}_{\text{safe}}$ belong to $\tilde{Z}_{\text{good}}$ and therefore have a confidence interval at most $\tau/2$. Then, from Lemma 6, we will have that when Algorithm 1 executed Algorithm 2 just before calling Algorithm 3, $\hat{Z}_{\text{safe}}$ is

updated in a way that $\tilde{Z} \subset \hat{Z}_{\text{safe}}$, which would imply that $(\tilde{s}, \tilde{a}) \in \hat{Z}_{\text{safe}}$. However, this contradicts our assumption in the beginning that that $(\tilde{s}, \tilde{a})$ is an edge state-action pair that does not belong to $\hat{Z}_{\text{safe}}$.

Since $\tilde{Z}_{\text{bad}}$ is non-empty, and since $\tilde{Z}_{\text{bad}} \subset \tilde{Z}$, there should exist a trajectory of π_{return} starting from $(\tilde{s}, \tilde{a})$ that passes through an element of $\tilde{Z}_{\text{bad}}$ before visiting $\hat{Z}_{\text{safe}}$. Let $(s_0, a_0), (s_1, a_1), \ldots, (s_n, a_n)$ be one such trajectory, where $(s_0, a_0) = (\tilde{s}, \tilde{a})$. Let (s_n, a_n) be the first element in this trajectory that belongs to $\tilde{Z}_{\text{bad}}$. Since all the elements in this trajectory preceding (s_n, a_n) belong to $\tilde{Z}$ but not $\hat{Z}_{\text{safe}}$ or $\tilde{Z}_{\text{bad}}$, all these elements must belong to $\tilde{Z}_{\text{good}}$.

We will now use the fact that, under our initial assumption, Algorithm 4 returns with an empty Z_{explore} to argue by induction that, during that run of Algorithm 4, all state-action pairs in this trajectory up until and including (s_n, a_n) are added to Z_{return}. (After this, we will reach a contradiction).

For the base case, consider (s, a). When the while loop condition is executed the first time, $Z_{\text{explore}} = \emptyset$ by initialization, and $Z_{\text{next}}^L \neq \emptyset$ because it is equal to Z_{edge} which has at least one state-action pair, namely (s, a). Thus, the while loop will be executed, and every element in Z_{next}^L will be added to Z_{return}. Since $Z_{\text{next}}^L = Z_{\text{edge}}$ at this point, this implies that (s, a) will be added to Z_{return}.

Next, for some $i \in [1, n]$, assume by induction that all state-action pairs preceding (s_k, a_k), where $k < i$, have been added to Z_{return}; we must prove the same happens to (s_i, a_i). Consider the loop when (s_{i-1}, a_{i-1}) is examined and added to Z_{return}. Since $(s_{i-1}, a_{i-1}) \in \tilde{Z}_{\text{good}}$, its confidence interval is at most as large as $\tau/2$; thus, in this loop, we would execute the else branch of the if-condition. As a result of this, we can argue that (s_i, a_i) is added to Z_{next}^{L+1} from the following four observations.

First, since the considered trajectory has non-zero probability, we have $T(s_{i-1}, a_{i-1}, s_i) > 0$. Furthermore, since $\hat{\tilde{\epsilon}}_T(s_{i-1}, a_{i-1}) \leq \tau/2$, and since the confidence intervals are admissible, by Lemma 2, we have $\hat{\tilde{T}}(s_{i-1}, a_{i-1}, s_i) > 0$. Second, if (s_i, a_i) was part of Z_{return} at this point, we are already done; so let us consider the case that currently $(s_i, a_i) \notin Z_{\text{return}}$. Thirdly, since $(s_i, a_i) \in \tilde{Z} \setminus \hat{Z}_{\text{safe}}$, $(s_i, a_i) \notin \hat{Z}_{\text{safe}}$. Finally, from our induction assumption, we have that $\hat{Z}_{\text{unsafe}} \cap Z_{\text{safe}} = \emptyset$; and since π_{return} is a safe policy, it follows that $(s_i, a_i) \notin \hat{Z}_{\text{unsafe}}$. As a result of these four observations, in this else branch, we would add (s_i, a_i) to Z_{next}^L.

Now, consider the instant when the algorithm evaluates the while condition after exiting the for loop that examined (s_{i-1}, a_{i-1}). At this point, by our initial assumption, Z_{explore} is still empty while Z_{next}^L is not as it contains (s_i, a_i). Thus the algorithm would proceed with executing this while loop (as against exiting from it then). Now since $(s_i, a_i) \in Z_{\text{next}}^L$, inside the inner for loop, there must be an iteration when (s_i, a_i) is examined and added to Z_{return}, proving our induction statement.

Thus, consider the for loop iteration when (s_n, a_n) is added to Z_{return}. Since $(s_n, a_n) \in \tilde{Z}_{\text{bad}}$, $\hat{\epsilon}_T(s_n, a_n) > \tau/2$. Hence, we would enter the if-branch of the if-else condition. Again, since $(s_n, a_n) \in \tilde{Z} \setminus \hat{Z}_{\text{safe}}$, from Assumption 5, we know there must exist $(s, a) \in \hat{Z}_{\text{safe}}$ such that $\Delta((s, a), (s_n, a_n)) \leq \tau/4$. As a result, (s, a) will be added to Z_{explore} contradicting the fact that Algorithm 4 exited the while loop without adding any element to Z_{explore}. Thus our initial assumption must be wrong, proving the main claim about Algorithm 4.

Proof for $\hat{Z}_{\textbf{unsafe}} \cap Z_{\textbf{safe}} = \emptyset$. From the induction assumption, we have that $\hat{Z}_{\text{unsafe}} \cap Z_{\text{safe}} = \emptyset$ in the beginning of this while loop. During this loop, $\hat{Z}_{\text{unsafe}}$ is modified by Algorithm 3 only when the call to Algorithm 4 returns with an empty Z_{explore}. In such a case, $\hat{Z}_{\text{unsafe}}$ is modified by adding Z_{edge} to it. Fortunately, from the above discussion, we know that when Z_{explore} is empty, Z_{edge} contains no element from Z_{safe}. Thus, $\hat{Z}_{\text{unsafe}} \cap Z_{\text{safe}} = \emptyset$ even at the end of the while loop.

$\square$

Finally, we show that Algorithm 3 terminates in finite time ensuring that $\overline{Z}_{\text{goal}}$ has been updated in a way that all of it has been established to be safe, or Z_{explore} is non-empty.

Lemma 10. *Algorithm 3 terminates in* $poly(|S|, |A|)$ *time, after which either* $\overline{Z}_{goal} \subset \hat{Z}_{safe}$ *or* $Z_{explore} \neq \emptyset$.

Proof. First, we show that, in Algorithm 3, whenever the condition $\overline{Z}_{\text{goal}} \subset \hat{Z}_{\text{safe}}$ fails, the subsequently computed Z_{edge} is non-empty. Assume for the sake of contradiction that even though $\overline{Z}_{\text{goal}} \not\subset \hat{Z}_{\text{safe}}$, Z_{edge} is empty. Now, recall that $\overline{Z}_{\text{goal}}$ is the set of all state-action pairs visited starting from s_0 following $\overline{\pi}_{\text{goal}}$ in the MDP $\overline{M}_{\text{goal}}$, with transition probabilities $\overline{T}_{\text{goal}}$.

Then, consider any trajectory $(s_0, a_0), (s_1, a_1), \ldots$ of non-zero probability under this corresponding policy and transition function. Since $s_0 \in \hat{Z}_{\text{safe}}$ and Z_{edge} is empty, $(s_0, a_0) \in \hat{Z}_{\text{safe}}$. Then, since $\overline{T}_{\text{goal}} \in CI(\hat{\tilde{T}})$, and since the confidence intervals are admissible, we have from Corollary 1 that $s_1 \in \hat{Z}_{\text{safe}}$. Since Z_{edge} is empty, by a similar argument, we can establish that $(s_1, a_1) \in \hat{Z}_{\text{safe}}$, and so on for all (s_t, a_t). Since this holds for any trajectory under this policy and transition function, it would mean that $\overline{Z}_{\text{goal}} \subset \hat{Z}_{\text{safe}}$, which is a contradiction. Thus, Z_{edge} is indeed non-empty.

Next, we show that Algorithm 3 terminates in polynomial time. Whenever Algorithm 3 does not break out of a loop, then by the design of the algorithm, $\overline{Z}_{\text{goal}} \not\subset \hat{Z}_{\text{safe}}$ and $Z_{\text{explore}} = \emptyset$. In addition to this, Z_{edge} must have been added to $\hat{Z}_{\text{unsafe}}$. Furthermore, from the above argument, since $\overline{Z}_{\text{goal}} \not\subset \hat{Z}_{\text{safe}}$, Z_{edge} must be non-empty. In other words, in each loop that does not break, we take a non-empty subset of $\overline{Z}_{\text{goal}}$, namely Z_{edge} and add it to $\hat{Z}_{\text{unsafe}}$. Note that since we computed $\overline{Z}_{\text{goal}}$ in a way that it does not include any of $\hat{Z}_{\text{unsafe}}$, this also means that $Z_{\text{edge}} \cap \hat{Z}_{\text{unsafe}} \neq \emptyset$. Thus, by the end of this loop, we increase the cardinality of $\hat{Z}_{\text{unsafe}}$. Since $\hat{Z}_{\text{unsafe}}$ cannot be any larger than the finite quantity $|S| \times |A|$, we are guaranteed that no more than $O(|S| \times |A|)$ for loops are run when Algorithm 3 is executed. (In fact, we can say something stronger: no more than $O(|S| \times |A|)$ for loops are run, across multiple calls to Algorithm 3 during the whole run of Algorithm 1).

Finally, observe that, by design of the algorithm, whenever the algorithm terminates, it must have broken out of the for loop. This is possible only if either $\overline{Z}_{\text{goal}} \subset \hat{Z}_{\text{safe}}$ or $Z_{\text{explore}} \neq \emptyset$, thus proving all of our claim. $\qquad\square$

D.3 Proofs about computing the set of state-actions along the goal path

Recall that $\overline{Z}_{\text{goal}}$ is intended to be the set of state-action pairs visited by the optimistic goal policy under optimistic transitions. To compute this set, we need to know which states have a positive probability of being reached from s_{init} following $\overline{\pi}_{\text{goal}}$, or equivalently the state-actions (s, a) for which $\overline{\rho}_{\text{goal}}(s, a) > 0$. Formally, we must enumerate the set

$$\{(s, a) \in S \times A : \overline{\rho}_{\text{goal}}(s, a) > 0\}.$$

However, computing this as defined is not feasible in finite time (as we must enumerate infinite length trajectories). Instead, recall from Equation 6 that we can approximate the above set by only computing the finite-horizon estimate $\overline{\rho}_{\text{goal}}(\cdot, \cdot, H)$ for some horizon H. Fortunately, computing $\overline{Z}_{\text{goal}}$ does not require a good estimate of $\overline{\rho}_{\text{goal}}(\cdot, \cdot)$, but only requires knowing when the $\overline{\rho}_{\text{goal}}(\cdot, \cdot)$ is positive or 0. Lemma 11 shows that as long as $H \geq |S|$, $\overline{\rho}_{\text{goal}}(\cdot, \cdot, H) > 0$ if and only if $\overline{\rho}_{\text{goal}}(s, a) > 0$; as a corollary of which we have that $\overline{Z}_{\text{goal}}$ is exactly what we intend it to be.

Lemma 11. *For any policy π, starting state $s \in S$, and state-action pair $(s', a') \in S \times A$, $\rho_{\pi,s}^M(s', a', H) > 0$ if and only if $\rho_{\pi,s}^M(s', a') > 0$ as long as $H \geq |S|$.*

Proof. First we establish sufficiency. That is, if the finite-horizon estimate is positive, then, so is the infinite horizon estimate. Consider the following inequality relating these two quantities:

$$\rho_{\pi,s}^M(s', a', H) = (1 - \gamma) \sum_{t=0}^{H} \gamma^t \mathrm{P}\left(s_t = s', a_t = a' | \pi, s_0 = s\right)$$

$$\leq \lim_{H \to \infty} (1 - \gamma) \sum_{t=0}^{H} \gamma^t \mathrm{P}\left(s_t = s', a_t = a' | \pi, s_0 = s\right)$$

$$= \rho_{\pi,s}^M(s', a')$$

The second step uses the fact that $\gamma > 0$ and $\mathrm{P}\left(s_t = s', a_t = a' | \pi, s_0 = s\right) \geq 0$ for all t. Thus, if $\rho_{\pi,s}^M(s', a', H) > 0$, then $\rho_{\pi,s}^M(s', a') > 0$, establishing sufficiency.

Now we will establish necessity i.e., if the infinite horizon estimate was positive, then the same must hold for the finite-horizon estimate. If $\rho_{\pi,s}^M(s', a') > 0$, then there exists at least one sequence of states $(s_0, s_1, \ldots s_n)$ where $s_0 = s$, $s_n = s'$, and $T(s_i, \pi(s_i), s_{i+1}) > 0$ for all $0 \leq i < n$. Without loss of generality, consider the shortest such sequence. Now, for the sake of contradiction, assume that $n > |S|$. By the pigeonhole principle, there exists at least one state that is repeated at least twice. That is, for two indices j, k $(j < k)$, we have $s_j = s_k$. Since

$s_j = s_k$ and $T(s_k, \pi(s_k), s_{k+1}) > 0$, then $T(s_j, \pi(s_j), s_{k+1}) > 0$. Thus, we can construct a shorter sequence by removing all indices i such that $j < i \leq k$ and this sequence still satisfies the fact that every transition observed has non-zero probability. This contradicts our assertion that this is the shortest such sequence. Thus, $n \leq |S|$. Given $H \geq |S| \geq n$, we have $\rho^M_{\pi,s}(s', a', H) > 0$, establishing necessity. $\qquad\square$

As a straightforward corollary of the above, we have:

Corollary 3. $\overline{Z}_{goal}$ *as computed in Equation 6 satisfies:*

$$\overline{Z}_{goal} = \{(s, a) \in S \times A : \overline{\rho}_{goal}(s, a) > 0\}.$$

when $H \geq |S|$.

D.4 Proofs about goal, explore, and switching policies

This subsection details the key lemmas for proving that ASE is PAC-MDP. The main idea is to show that, under these policies, our agent either will perform the desired behavior, i.e. act ϵ-optimally or reach a desired state-action set, or reach an insufficiently explored state-action pair (i.e., not in K). Since a state-action pair that is experienced m times is added to K, we can bound the number of times we reach a state-action pair outside of K. With this bound and the following lemmas, we can bound the number of times the agent performs undesired behaviors, e.g. acting sub-optimally. We start by proving this claim for $\overline{\pi}_{\text{explore}}$ and $\overline{\pi}_{\text{switch}}$ (Lemma 12), then for $\overline{\pi}_{\text{goal}}$ (Lemma 13 and Lemma 14).

Recall that the $\overline{\pi}_{\text{explore}}$ is based on a reward system where the rewards are non-zero only on state-action pairs that are in $\hat{Z}_{\text{safe}}$ and are not sufficiently explored (i.e., not in K). We first show that if we were to follow the $\overline{\pi}_{\text{explore}}$ policy, in polynomially many steps, we are guaranteed to obtain a non-zero reward i.e., we are guaranteed to reach a state-action pair not in K. In other words, if we were to follow $\overline{\pi}_{\text{explore}}$, we will definitely obtain a useful sample and learn something new. Similarly, if we were to follow $\overline{\pi}_{\text{switch}}$, we will definitely return to $\overline{Z}_{\text{goal}}$ or learn something new.

Lemma 12. *Assume that the confidence intervals are admissible. Consider an MDP $M^\dagger = \langle S, A, T, R^\dagger, \gamma^\dagger \rangle$, which is the same as the true MDP but with different rewards and discount factor. Let Z be a closed, communicating subset of $S \times A$ and $Z^\dagger$ a non-empty subset of Z. Let $R^\dagger$ be defined such that:*

$$R^\dagger(s, a) = \begin{cases} 1 & (s, a) \in Z^\dagger \\ 0 & (s, a) \in Z \setminus Z^\dagger \\ -\infty & (s, a) \notin Z \end{cases}$$

Let $H = H_{com} \log \frac{16 H_{com}}{c} / \log \frac{1}{c}$ and let $\gamma^\dagger = c^{1/H}$ where $c \in (0, 1/4]$ is a constant. Let $\tilde{H} = \max\left(H \frac{1}{\sqrt{8c}}, \frac{1}{\sqrt{\tau}}\right)$. Let $\overline{Q}^\dagger$ denote the optimistic value function of this MDP, and $\overline{\pi}^\dagger$ be the optimistic policy i.e., $\overline{\pi}^\dagger(s, a) = \arg\max_{a \in A} \overline{Q}^\dagger(s, a)$.

Then, for any $\delta > 0$ and $\epsilon \in (0, c/8]$, starting from any state in Z and following $\overline{\pi}^\dagger$, the agent will reach a state-action pair either in $Z^\dagger$ or outside of $K = \{(s, a) \in S \times A : n(s, a) \geq m\}$, where $m \geq O\left(\tilde{H}^4 |S| + \tilde{H}^4 \ln \frac{|S||A|\tilde{H}^2}{\delta}\right)$, in at most $O(\frac{H^2}{c})$ time steps, with probability at least $1/2$, provided $\overline{\pi}^\dagger \in \Pi(Z)$.

Proof. Consider any $s \in Z$. We will first upper bound the optimistic value $\overline{V}^\dagger(s)$ and then derive a lower bound on it, and then relate these two bounds together to prove our claim. Note that if we let $\overline{M}^\dagger$ be the same MDP as $M^\dagger$, but with the optimistic transitions (the transitions $\overline{T}^\dagger \in CI(\hat{T})$ which maximize the optimistic Q-values), then $\overline{V}^\dagger(\cdot) = \overline{V}^{\overline{\pi}^\dagger}_{\overline{M}^\dagger}(\cdot)$.

We will begin by upper bounding the finite-horizon value of $\overline{\pi}^\dagger$ on $M^\dagger$ (and then relate it to its value on $\overline{M}^\dagger$). Let $s_0, s_1, s_2, \ldots$, denote the random sequence of states visited by the agent by following $\overline{\pi}^\dagger$ from $s_0 = s$ on $M^\dagger$.

Then, we have:

$$V_{M^\dagger}^{\overline{\pi}^\dagger}(s, H) = \mathbb{E}\left[\sum_{i=0}^{H} (\gamma^\dagger)^i R^\dagger(s_i, \overline{\pi}^\dagger(s_i))\right]$$

$$\leq \mathbb{E}\left[\sum_{i=0}^{H} R^\dagger(s_i, \overline{\pi}^\dagger(s_i))\right]$$

$$\leq \mathrm{P}(\exists\, i \leq H : (s_i, \overline{\pi}^\dagger(s_i)) \in Z^\dagger) \cdot H$$

The first inequality follows from the fact that $\gamma^\dagger < 1$. The second inequality follows from the fact that every trajectory of $\overline{\pi}^\dagger$ that experiences a positive cumulative reward, must experience some state-action pair in $Z^\dagger$; and such a trajectory can at best experience a reward of 1 at each timestep.

In the next step we will upper bound the finite-horizon optmistic value of following $\overline{\pi}^\dagger$ in $\overline{M}^\dagger$. To do this, define M' to be an MDP that is identical to $M^\dagger$ on $(s, a) \in K$, and identical to $\overline{M}^\dagger$ everywhere else. Then,

$$V_{\overline{M}^\dagger}^{\overline{\pi}^\dagger}(s, H) \leq V_{M'}^{\overline{\pi}^\dagger}(s, H) + \frac{c}{8}$$

$$\leq V_{M^\dagger}^{\overline{\pi}^\dagger}(s, H) + H\mathrm{P}(\exists\, i \leq H : (s_i, \overline{\pi}^\dagger(s_i)) \notin K) + \frac{c}{8}$$

$$\leq H\mathrm{P}(\exists\, i \leq H : (s_i, \overline{\pi}^\dagger(s_i)) \in Z^\dagger) + H\mathrm{P}(\exists\, i \leq H : (s_i, \overline{\pi}^\dagger(s_i)) \notin K) + \frac{c}{8}$$

$$\leq 2H\mathrm{P}(\exists\, i \leq H : (s_i, \overline{\pi}^\dagger(s_i)) \in Z^\dagger \cup K^c) + \frac{c}{8}.$$

Here, the first inequality follows from Lemma 15 and 16. Specifically, from Lemma 16, we have when $m \geq O\left(\tilde{H}^4|S| + \tilde{H}^4 \ln \frac{|S||A|\tilde{H}^2}{\delta}\right)$, the width of the confidence interval of $(s, a) \in K$ is at most $1/\tilde{H}^2 \leq c/(8H^2)$. Then, Lemma 15 can be used to bound the difference in their value functions by $c/8$.

Note that to apply Lemma 15 we must also ensure that for all $(s, a) \in K$, the support of the next state distribution is the same under M' and $\overline{M}^\dagger$. To see why this is true, observe that for all $(s, a) \in K$, the confidence interval is at most $1/\tilde{H}^2 \leq \tau/2$. Then, since the transition probabilities for $(s, a) \in K$ in $\overline{M}^\dagger$ and M' correspond to $\overline{T}^\dagger \in CI(\hat{T})$ and T respectively, Lemma 2 implies that the support of these state-action pairs are indeed the same for these two transition functions. Also note that Lemma 15 implies that these value functions are either close to each other or both equal to $-\infty$; even in the latter case, the above inequalities would hold (although, we will show in the remaining part of the proof that these quantities are lower bounded by some positive value).

The second inequality follows from Lemma 17. Note that in order to apply Lemma 17, we must establish that, with probability 1, the agent experiences only non-negative rewards, when it starts from s and follows $\overline{\pi}^\dagger$ for H steps. This is indeed true because we know that Z is closed and $\overline{\pi}^\dagger \in \Pi(Z)$. Then, we know from Fact 1 that the agent always remains in Z, which means it experiences only non-negative rewards.

The third inequality above uses the upper bound on $V_{M^\dagger}^\pi(s, H)$ that we derived previously.

Having established the above inequalities, we are now ready to upper bound the optimistic value using Lemma 18 as follows:

$$\overline{V}_{\overline{M}^\dagger}^{\overline{\pi}^\dagger}(s) \leq \overline{V}_{\overline{M}^\dagger}^{\overline{\pi}^\dagger}(s, H) + \frac{(\gamma^\dagger)^{H+1}}{1 - \gamma^\dagger}$$

$$\leq 2H\mathrm{P}(\exists\, i \leq H : (s_i, \overline{\pi}^\dagger(s_i)) \in Z^\dagger \cup K^c) + \frac{c}{8} + \frac{(\gamma^\dagger)^{H+1}}{1 - \gamma^\dagger} \tag{13}$$

Next, as the second part of our proof, we will derive a lower bound on the optimistic value using Assumption 3. Choose some $(s', a') \in Z^\dagger$ (which is given to be non-empty). Since Z is given to be communicating, by Assumption 3, we know that there exists a policy π_{com} which has a probability of at least $\frac{1}{2}$ of reaching s' from s in H_{com} steps, while visiting only state-action pairs in Z. Without loss of generality, let us assume that $\pi_{\mathrm{com}}(s') = a'$ (since, regardless of what the action at s' is, it guarantees reachability of s' from everywhere else.).

Let us lower bound the value of this policy. Let $s_0, s_1, s_2, \ldots$, denote the random sequence of states visited by the agent by following π_{com} from $s_0 = s$ on $M^\dagger$. Then:

$$V_{M^\dagger}^{\pi_{\text{com}}}(s) \geq V_{M^\dagger}^{\pi_{\text{com}}}(s, H_{\text{com}})$$

$$= \mathbb{E}\left[\sum_{i=0}^{H_{\text{com}}} (\gamma^\dagger)^i R^\dagger(s_i, \pi_{\text{com}}(s_i)) \right]$$

$$\geq (\gamma^\dagger)^{H_{\text{com}}} \mathbb{E}\left[\sum_{i=0}^{H_{\text{com}}} R^\dagger(s_i, \pi_{\text{com}}(s_i)) \right]$$

$$\geq (\gamma^\dagger)^{H_{\text{com}}} \mathrm{P}(\exists\, i : (s_i, \pi_{\text{com}}(s_i)) \in Z^\dagger)$$

$$\geq (\gamma^\dagger)^{H_{\text{com}}} \frac{1}{2}.$$

Here, the first step follows from Fact 1 which says that, since $\pi_{\text{com}} \in \Pi(Z)$ and Z is closed, π_{com} only visits state-action pairs in Z, all of which have non-negative $R^\dagger$ reward; as a result, truncating the value function to H_{com} steps only maintains/decreases the value.

The third step follows from the fact that $\gamma^\dagger < 1$. The fourth step comes from the fact that, since π_{com} takes only state-action pairs in Z, $R^\dagger(s_i, \pi_{\text{com}}(s_i)) \in \{0, 1\}$; then, every trajectory with a total non-zero reward has a reward of at least 1. The last step follows from the guarantee of Assumption 3.

Now, as the final step in our proof we note that if our confidence intervals are admissible, by Lemma 19, we know that $V_{\overline{M}^\dagger}^{\overline{\pi}^\dagger}(s) \geq V_{M^\dagger}^*(s)$. Furthermore, $V_{M^\dagger}^*(s)$ must be lower bounded by the value of π_{com} on $M^\dagger$, which we just lower bounded. Now, equating this with the upper bound from Equation 13, we get:

$$2H\mathrm{P}(\exists\, i \leq H : (s_i, \overline{\pi}^\dagger(s_i)) \in Z^\dagger \cup K^c) + \frac{c}{8} + \frac{(\gamma^\dagger)^{H+1}}{1 - \gamma^\dagger} \geq (\gamma^\dagger)^{H_{\text{com}}} \frac{1}{2}$$

$$\mathrm{P}(\exists\, i \leq H : (s_i, \overline{\pi}^\dagger(s_i)) \in Z^\dagger \cup K^c) \geq \frac{1}{2H}\left((\gamma^\dagger)^{H_{\text{com}}} \frac{1}{2} - \frac{c}{8} - \frac{(\gamma^\dagger)^{H+1}}{1 - \gamma^\dagger} \right)$$

$$\geq \frac{1}{2H}\left(\frac{3c}{8} - \frac{(\gamma^\dagger)^{H+1}}{1 - \gamma^\dagger} \right) \qquad (14)$$

Here, in the last step, we make use of the fact that $\gamma^\dagger = c^{1/H_{\text{com}}}$. Next, we will upper bound the last term by making use of the inequality: if $c < 1$, then $\forall x > 0, c^x \leq 1 - x(1-c)$. Then, we get:

$$\frac{1}{1 - \gamma^\dagger} = \frac{1}{1 - c^{1/H_{\text{com}}}} \leq \frac{H_{\text{com}}}{1 - c}$$

Furthermore, since $\log(16H_{\text{com}}/c) = (H/H_{\text{com}})\log(1/c)$, by applying $\exp(\cdot)$ on both sides, we have:

$$\frac{c}{16H_{\text{com}}} = c^{H/H_{\text{com}}} = (\gamma^\dagger)^H$$

From the above two inequalities, we have:

$$\frac{(\gamma^\dagger)^{H+1}}{1 - \gamma^\dagger} \leq \frac{(\gamma^\dagger)^H}{1 - \gamma^\dagger} \leq \frac{c}{16(1 - c)} \leq \frac{c}{8}$$

In the first step above, we make use of $\gamma^\dagger < 1$ and in the second step, $c < 1/2$. Plugging this back in Equation 14, we get:

$$\mathrm{P}(\exists\, i \leq H : (s_i, \overline{\pi}^\dagger(s_i)) \in Z^\dagger \cup K^c) \geq \frac{c}{8H}$$

In other words, following policy $\overline{\pi}^{\dagger}$ for H steps in $M^{\dagger}$, the agent will reach a state-action pair either in $Z^{\dagger}$ or outside K with probability at least $c/(8H)$. Then, by the Hoeffding bound applied to multiple subsequent trajectories each of H timesteps, we would have that with probability at least $1/2$, following policy $\overline{\pi}_{\text{goal}}$ for $O(H \cdot \frac{H}{c})$ timesteps, the agent will reach an element either in Z or outside K.

$\square$

In Lemma 13, we show that once our algorithm begins following the optimistic goal policy $\overline{\pi}_{\text{goal}}$, it will continue to do so until it learns something new. (Since we can bound the number of times the agent learns something new, observe that this also means that, eventually, the agent will follow $\overline{\pi}_{\text{goal}}$ for all time.)

Lemma 13. *Assume the confidence intervals are admissible. Then, during the run of Algorithm 1, if the agent is currently following $\overline{\pi}_{goal}$, then it will continue to do so until it experiences a state-action pair outside* $K = \{(s,a) \in S \times A : n(s,a) \geq m\}$ *when* $m \geq O\left(\frac{|S|}{\tau^2} + \frac{1}{\tau^2}\ln\frac{|S||A|}{\tau^2\delta}\right)$.

Proof. Assume that during a run of Algorithm 1, the agent is currently at s and takes the action $\overline{\pi}_{\text{goal}}(s)$. By design of Algorithm 1, we have $s \in \overline{Z}_{\text{goal}}$. Recall that $\overline{Z}_{\text{goal}}$ is the set of all state-action pairs that can be visited by the agent if it were to following $\overline{\pi}_{\text{goal}}$ starting from s_0 under the optimistic transitions $\overline{T}_{\text{goal}}$. More formally, we have from Corollary 3 that $\overline{Z}_{\text{goal}} = \{(s,a) \in S \times A : \overline{\rho}_{\text{goal}}(s,a) > 0\}$.

Now, to prove our claim, we only need to argue that if $(s, \overline{\pi}_{\text{goal}}(s)) \in K$, then the next state s' belongs to $\overline{Z}_{\text{goal}}$. Then, by design of Algorithm 1, the agent will take $\overline{\pi}_{\text{goal}}$ even in the next state, proving our claim. To argue this, observe that since $(s, \overline{\pi}_{\text{goal}}(s)) \in K$ and $m \geq O\left(\frac{|S|}{\tau^2} + \frac{1}{\tau^2}\ln\frac{|S||A|}{\tau^2\delta}\right)$, by Lemma 16, the confidence interval of $(s, \overline{\pi}_{\text{goal}}(s))$ has width at most $\tau/2$. Then, from Lemma 2, since $T(s, \overline{\pi}_{\text{goal}}(s), s') > 0$ and since $\overline{T}_{\text{goal}} \in CI(\hat{T})$, we have $\overline{T}_{\text{goal}}(s, \overline{\pi}_{\text{goal}}(s), s') > 0$. Thus, since we know that $s \in \overline{Z}_{\text{goal}}$, by definition of $\overline{Z}_{\text{goal}}$, s' should also belong to $\overline{Z}_{\text{goal}}$. $\square$

In the following lemma, we show that in the MDP M_{goal} (which is the same as the original MDP but with the unsafe state-action pairs set to $-\infty$ rewards), when we follow the optimistic goal policy $\overline{\pi}_{\text{goal}}$, we either take a near-optimal action (with respect to M_{goal}) or we experience an action outside of K with sufficient probability in the next H steps.

Lemma 14. *Assume the confidence intervals are admissible. Consider any instant when the agent has taken a trajectory p_t and is at state s_t, and the Algorithm 1 instructs the agent to follow $\overline{\pi}_{goal}$. Let $\mathrm{P}(A_M)$ be the probability that starting at this step, the Algorithm 1 leads the agent out of $K = \{(s,a) \in S \times A : n(s,a) \geq m\}$ in H steps, conditioned on p_t. Then, for any $\epsilon, \delta \in (0,1)$, and for $H = O\left(\frac{1}{1-\gamma}\ln\frac{1}{\epsilon(1-\gamma)}\right)$ and $m \geq O\left(\frac{1}{\tilde{\epsilon}^2(1-\gamma)^4}\left(\frac{|S|}{\tilde{\epsilon}} + \frac{1}{\tilde{\epsilon}^2}\ln\frac{|S||A|}{\tilde{\epsilon}^2\delta}\right)\right)$, where $\tilde{\epsilon} = O\left(\min\left(\frac{\tau}{2}, \frac{\epsilon(1-\gamma)^2}{3}\right)\right)$, we have:*

$$V_M^{\mathcal{A}}(p_t) \geq V_{M_{goal}}^*(s_t) - \frac{\epsilon}{2} - 2\frac{\mathrm{P}(A_M)}{1-\gamma}.$$

Proof. While we follow the general outline of the proof of Theorem 1 from Strehl and Littman (2008), we note that there are crucial differences for incorporating safety (such as dealing with rewards of $-\infty$).

At the outset, we establish two useful inequalities. First, since $H = O\left(\frac{1}{1-\gamma}\ln\frac{1}{\epsilon(1-\gamma)}\right)$, by Lemma 18, we have that:

$$V_{\overline{M}_{\text{goal}}}^{\overline{\pi}_{\text{goal}}}(s_t, H) \geq V_{\overline{M}_{\text{goal}}}^{\overline{\pi}_{\text{goal}}}(s_t) - \frac{\epsilon}{3} \tag{15}$$

Secondly, let M'_{goal} be an MDP that is equivalent to M_{goal} for all state-action pairs in K and equal to $\overline{M}_{\text{goal}}$ otherwise. (Note that all these MDPs have the same reward function, namely R_{goal}.) We claim that for all a:

$$Q_{M'_{\text{goal}}}^{\overline{\pi}_{\text{goal}}}(s_t, a, H) \geq Q_{\overline{M}_{\text{goal}}}^{\overline{\pi}_{\text{goal}}}(s_t, a, H) - \frac{\epsilon}{3} \tag{16}$$

Let us see why this inequality holds. From Lemma 16, we have when $m \geq O\left(\frac{1}{\tilde{\epsilon}^2(1-\gamma)^4}\left(\frac{|S|}{\tilde{\epsilon}} + \frac{1}{\tilde{\epsilon}^2}\ln\frac{|S||A|}{\tilde{\epsilon}^2\delta}\right)\right)$, the width of the confidence interval of $(s,a) \in K$ is at most $O(\tilde{\epsilon}) = \frac{\epsilon}{3}\frac{(1-\gamma)^2}{\gamma}$. Then, Lemma 15 can be used to bound the difference in the value functions by $\epsilon/3$ as above.

Note that to apply Lemma 15, we must also ensure that for all $(s,a) \in K$, the support of the next state distribution is the same under M'_{goal} and $\overline{M}_{\text{goal}}$. To see why this is true, observe that for all $(s,a) \in K$, the width of the confidence interval is at most $O(\tilde{\epsilon}) = \tau/2$. Then, since the transition probability for $(s,a) \in K$ in $\overline{M}^\dagger$ and M' correspond to $\overline{T}^\dagger \in CI(\hat{T})$ and T respectively, Lemma 2 implies that the support of these state-action pairs are indeed the same for these two transition functions. Also note that Lemma 15 implies that these value functions are either ϵ_1-close to each other or both equal to $-\infty$; even in the latter case, the above inequality would hold (although, to be precise, this latter case does not really matter since Theorem 1 eventually shows that these quantities are lower bounded by a positive quantity).

Having established the above inequalities, we now begin lower bounding the value of the algorithm. First, since the rewards $R(\cdot,\cdot)$ are bounded below by -1, and since $H = O\left(\frac{1}{1-\gamma}\ln\frac{1}{\epsilon(1-\gamma)}\right)$, by Lemma 18, we can lower bound the infinite-horizon value of the algorithm by its finite-horizon value as

$$V_M^{\mathcal{A}}(p_t) \geq V_M^{\mathcal{A}}(p_t, H) - \frac{\epsilon}{3}. \tag{17}$$

Next, we claim to bound the value of following the algorithm for the next H steps as follows:

$$V_M^{\mathcal{A}}(p_t, H) \geq V_{M'_{\text{goal}}}^{\overline{\pi}_{\text{goal}}}(s, H) - 2\frac{P(A_M)}{1-\gamma}. \tag{18}$$

In other words, we have lower bounded the value of following the algorithm on M, in terms of following $\overline{\pi}_{\text{goal}}$ on M'_{goal}. Let us see why this is true. For the sake of convenience, let us define two cases corresponding to the above inequality: Case A, where the agent follows Algorithm 1 to take actions for H steps starting from s_t in MDP M and Case B, where the agent follows the fixed policy $\overline{\pi}_{\text{goal}}$ to take actions for H steps starting from s_t in MDP M'_{goal}.

Now, recall that we are considering a state s_t where Algorithm 1 currently follows $\overline{\pi}_{\text{goal}}$, and will continue to follow it until it takes an action in K^c. Then, consider a particular random seed for which, in Case A, the agent does not reach K^c.

We argue that for this random seed, the agent will see the same cumulative discounted reward over H steps in both Case A and Case B. To see why, recall that M'_{goal} and M share the same transition functions on K. Next, since in Case A, the agent does not escape K, by Lemma 13, we know that the agent follows only $\overline{\pi}_{\text{goal}}$ for H steps. Thus in both these cases, the agent experiences the same sequence of state-action pairs for H steps. It only remains to argue that these state-action pairs have the same rewards in both cases. To see why, recall that by design of Algorithm 1, since the agent does not escape K, all these state-action pairs would belong to $\overline{Z}_{\text{goal}}$ which in turn is a subset of $\hat{Z}_{\text{safe}}$. Now the MDP M and M'_{goal} share the same rewards on $\hat{Z}_{\text{safe}}$; this is because, their rewards differ only in $\hat{Z}_{\text{unsafe}}$, and we know $\hat{Z}_{\text{unsafe}} \cup Z_{\text{safe}} = \emptyset$ (Lemma 9). Thus, in both cases, the agent experiences the same sequence of rewards.

As a result, the value functions in Case A and B differ only due to trajectories where the algorithm either leads the agent to K^c in H steps or leads the agent to negative rewards any time in the future. Hence, we can upper bound the difference $V_{M'_{\text{goal}}}^{\overline{\pi}_{\text{goal}}}(s_t, H) - V_M^{\mathcal{A}}(p_t, H)$ in terms of $P(A_M)$ multiplied by the maximum difference between the respective cumulative rewards. We know that the cumulative reward in Case A is at least $-1/(1-\gamma)$, because the rewards $R(\cdot,\cdot)$ are bounded below by -1. On the other hand, in Case B, the cumulative reward experienced is at most $1/(1-\gamma)$, assuming the agent receives a reward of 1 for each step. From this, we establish Equation 18.

Subsequently, we further lower bound $V_{M'_{\text{goal}}}^{\overline{\pi}_{\text{goal}}}(s_t, H)$ as follows:

$$
\begin{aligned}
V_{M'_{\text{goal}}}^{\overline{\pi}_{\text{goal}}}(s_t, H) &\geq V_{\overline{M}_{\text{goal}}}^{\overline{\pi}_{\text{goal}}}(s_t, H) - \frac{\epsilon}{3} \\
&\geq V_{\overline{M}_{\text{goal}}}^{\overline{\pi}_{\text{goal}}}(s_t) - 2\frac{\epsilon}{3} \\
&\geq V_{M_{\text{goal}}}^{*}(s_t) - \epsilon
\end{aligned}
$$

Here, the first inequality comes from Equation 16. The second inequality comes from Equation 15. The last step comes from Lemma 19 (given admissibility). Then, by combining the above inequality with Equations 18 and 17, we get our final result. $\qquad\square$

D.5 Supporting lemmas for showing PAC-MDP

The following lemmas are necessary for proving our algorithm is PAC-MDP. Note that most of these lemmas are similar to lemmas from Strehl and Littman (2008). However, because we construct MDPs with infinitely negative rewards, additional care must be taken to ensure that these properties still hold.

Here we provide a quick description of the lemmas detailed in this section. We start with Lemma 15, which shows that if two MDPs have sufficiently similar transition functions, the optimal Q-values on these two MDPs must also be similar. This, together with Lemma 16, allows us to show that, if we have sufficiently explored the state-space, we can accurately estimate the optimal policy on any MDP. Next we show, in Lemma 17, that the difference between the value functions of the true and an estimated MDP for a given policy is proportional to the probability of reaching an under-explored state-action pair, i.e. a state-action pair outside of K. This allows us to claim that either the probability of reaching an element outside of K is sufficiently large, or our estimated value function is sufficiently accurate. Lemma 18 bounds the difference between the finite horizon and infinite horizon value functions, allowing us to consider only finite length trajectories. Lemma 19 simply shows that our optimistic value function always over-estimates the true value function (given that our confidence intervals are admissible).

Lemma 15. *Let $M_1 = \langle S, A, T_1, R^\dagger, \gamma^\dagger \rangle$ and $M_2 = \langle S, A, T_2, R^\dagger, \gamma^\dagger \rangle$ be two MDPs with identical rewards that either belong to $[0, 1]$ or equal $-\infty$ and $\gamma^\dagger < 1$. Let K be a subset of state-action pairs such that*

1. *for all $(s, a) \notin K$, $T_1(s, a, \cdot) = T_2(s, a, \cdot)$,*

2. *for all $(s, a) \in K$, $\|T_1(s, a, \cdot) - T_2(s, a, \cdot)\|_1 \leq \beta$ and*

3. *for all $(s, a) \in K$, the next state distribution (s, a) has identical support under both T_1 and T_2.*

Then, for any (stationary, deterministic) policy π, and for any (s, a) and any $H \geq 0$, we have that, either:

$$
\left| Q_{M_1}^{\pi}(s, a, H) - Q_{M_2}^{\pi}(s, a, H) \right| \leq \min\left(\frac{\gamma^\dagger \beta}{(1 - \gamma^\dagger)^2}, \beta H^2 \right).
$$

or

$$
Q_{M_1}^{\pi}(s, a, H) = Q_{M_2}^{\pi}(s, a, H) = -\infty.
$$

Proof. First, we note that for any (s, a) such that $R^\dagger(s, a) = -\infty$, $Q_{M_1}^{\pi}(s, a, H) = Q_{M_2}^{\pi}(s, a, H) = -\infty$ for all H. Hence, for the rest of the discussion, we will consider (s, a) such that $R^\dagger(s, a) \neq -\infty$.

We prove our claim by induction on H. For $H = 1$, for all (s, a), $Q_{M_1}^{\pi}(s, a, H) = Q_{M_2}^{\pi}(s, a, H) = R^\dagger(s, a)$.

Consider any arbitrary H. First, we show that, if there exists a next state s' in the support of $T_1(s, a, \cdot)$ such that $Q_{M_1}^{\pi}(s', \pi(s'), H - 1) = -\infty$, then $Q_{M_1}^{\pi}(s, a, H) = Q_{M_2}^{\pi}(s, a, H) = -\infty$. Note that, by conditions 1 and 3 of the Lemma statement, we have that for all (s, a), regardless of whether in K or not, the support of the next state distribution is identical between T_1 and T_2. Hence, if there exists a next state s' in the support of T_1 such that $Q_{M_1}^{\pi}(s', \pi(s'), H - 1) = -\infty$, then s' would belong even to the support of T_2 and, thus, we would have that $Q_{M_2}^{\pi}(s', \pi(s'), H - 1) = -\infty$. Hence, by definition of Q-values, we would have $Q_{M_1}^{\pi}(s, a, H) = Q_{M_2}^{\pi}(s, a, H) = -\infty$.

Now consider a case where none of the next states s' in the support of T_1 (and T_2, without loss of generality) have Q-value $Q^\pi_{M_1}(s', \pi(s'), H-1) = -\infty$. We will prove by induction that in this case,

$$\left|Q^\pi_{M_1}(s,a,H) - Q^\pi_{M_2}(s,a,H)\right| \le \frac{\gamma^\dagger \beta}{1-\gamma^\dagger} \cdot \frac{1-\gamma^{\dagger H}}{1-\gamma^\dagger}.$$

Note that when $H = 0$, the right hand side above resolves to zero, which is indeed true.

Consider any $H > 0$. Now, observe that for all (s,a), regardless of whether in K or not, we have $\|T_1(s,a,\cdot) - T_2(s,a,\cdot)\|_1 \le \beta$. Besides, since $R^\dagger(s,a) \ne -\infty$, we can upper bound the difference in the Q values as follows:

$$
\begin{aligned}
&|Q^\pi_{M_1}(s,a,H) - Q^\pi_{M_2}(s,a,H)| \\
&= \gamma^\dagger \left| \sum_{s'} T_1(s,\pi(s'),s') Q^\pi_{M_1}(s',\pi(s'),H-1) - \sum_{s'} T_2(s,\pi(s'),s') Q^\pi_{M_2}(s',\pi(s'),H-1) \right| \\
&\le \gamma^\dagger \left| \sum_{s'} T_1(s,\pi(s'),s') \left(Q^\pi_{M_1}(s',\pi(s'),H-1) - Q^\pi_{M_2}(s',\pi(s'),H-1)\right) \right| \\
&\quad + \gamma^\dagger \left| \sum_{s'} \left(T_1(s,\pi(s'),s') - T_2(s,\pi(s'),s')\right) Q^\pi_{M_2}(s',\pi(s'),H-1) \right| \\
&\le \gamma^\dagger \left(\frac{\gamma^\dagger \beta}{1-\gamma^\dagger} \cdot \frac{1-\gamma^{\dagger H-1}}{1-\gamma^\dagger} \right) + \gamma^\dagger \frac{\beta}{1-\gamma^\dagger} \\
&= \frac{\gamma^\dagger \beta}{1-\gamma^\dagger} \left(\frac{\gamma^\dagger(1-\gamma^{\dagger H-1})}{1-\gamma^\dagger} + 1 \right) \\
&= \frac{\gamma^\dagger \beta}{1-\gamma^\dagger} \cdot \frac{1-\gamma^{\dagger H}}{1-\gamma^\dagger}
\end{aligned}
$$

Here, the second step follows by a simple algebraic rearrangement that decomposes the difference in the Q-values, in terms of the difference in the transitions and the difference in the next-state Q-values. In the third step, for the first term, we make use of the fact that in this case, the next state s' has a Q-value that is not $-\infty$; this means that $(s',\pi(s'))$ does not have any next states with Q-value $-\infty$ and therefore the induction assumption holds. By applying the induction assumption for $H-1$, we get the first term. For the second term, we make use of the fact that the total sum of transition probabilities equals 1. Furthermore, we also make use of the fact the maximum magnitude of the Q-value is at most $\frac{1}{1-\gamma^\dagger}$ if it is not $-\infty$; this is because, if the Q-value is not $-\infty$, it is a discounted summation of expected rewards that lie between 0 and 1.

Hence, our induction hypothesis is true. Our main upper bound can then be established by noting that $1-\gamma^{\dagger H} < 1$.

To prove our other upper bound, we consider the induction hypothesis:

$$\left|Q^\pi_{M_1}(s,a,H) - Q^\pi_{M_2}(s,a,H)\right| \le \beta H^2.$$

Then, in the third step above, we would instead have:

$$
\begin{aligned}
|Q^\pi_{M_1}(s,a,H) - Q^\pi_{M_2}(s,a,H)| &\le \gamma^\dagger \left(\beta(H-1)^2\right) + \gamma^\dagger \beta H \\
&\le \beta(H-1)^2 + \beta H \\
&\le \beta H^2.
\end{aligned}
$$

To get the first term on the right hand side, we again make use of the induction assumption. For the second term, we simply upper bound the sum of the maximum discounted rewards to be H. Finally, we make use of the fact that $\gamma^\dagger < 1$ and $H^2 - (H-1)^2 \ge 2H - 1 \ge H$ when $H > 0$. $\qquad\square$

Lemma 16. *Suppose that as input to Algorithm 1, we set $\delta_T = \delta/(2|S||A|m)$ and all confidence intervals computed by our algorithm are admissible. Then, for any $\beta > 0$, there exists an $m = O\left(\frac{|S|}{\beta^2} + \frac{1}{\beta^2}\ln\frac{|S||A|}{\beta\delta}\right)$ such that $\|\hat{T}(s,a,\cdot) = T(s,a,\cdot)\|_1 \le \beta$ holds for all state-action pairs (s,a) that have been experienced at least m times.*

Proof. See Lemma 5 from Strehl and Littman (2008). □

Lemma 17. *Let $M^\dagger = \langle S, A, R^\dagger, T, \gamma^\dagger \rangle$ be an MDP that is the same as the true MDP M, but with arbitrary rewards $R^\dagger$ (bounded above by 1) and discount factor $\gamma^\dagger < 1$. Let K be some set of state-action pairs. Let $M' = \langle S, A, R^\dagger, T', \gamma^\dagger \rangle$ be an MDP such that T' is identical to T on all elements inside K. Consider a policy π. Let $A_{M^\dagger}$ be the event that a state-action pair not in K is encountered in a trial generated by starting from state s_1 and following π for H steps in $M^\dagger$ (where H is a positive constant). If, with probability 1, the agent starting at s_1 and following π in $M^\dagger$ will receive only non-negative rewards then,*

$$V_M^\pi(s_1, H) \geq V_{M'}^\pi(s_1, H) - \min\left(\frac{1}{(1 - \gamma^\dagger)}, H\right) \mathrm{P}(A_M).$$

Proof. This proof is similar to that of Lemma 3 from Strehl and Littman (2008). The only aspect we need to be careful about is the magnitude of the rewards.

The two MDPs $M^\dagger$ and M' differ only in their transition functions, and moreover, only outside the set K. Then, observe that, for a fixed random seed, if the agent were to follow π starting from s_1, it would receive the same cumulative reward for H steps in both $M^\dagger$ and M', if it remained in K for all those H steps. In other words, the value of π in these two MDPs differs only because of those random seeds which led the agent out of K in $M^\dagger$. Thus, the difference $V_{M'}^\pi(s_1, H) - V_{M^\dagger}^\pi(s_1, H)$ cannot be any larger than the respective cumulative rewards. Our claim then follows by lower bounding the cumulative reward in $M^\dagger$ and upper bounding the cumulative reward in M'. More concretely, note that cumulative reward in $M^\dagger$ can not be any lower than zero, since we assume that the agent receives rewards bounded in $[0, 1]$. On the other hand, in M', the agent can receive a reward of 1 in all H steps, so the value function can be upper bounded by $\min\left(\frac{1}{(1-\gamma^\dagger)}, H\right)$.

□

Lemma 18. *Consider an MDP $M^\dagger = \langle S, A, T^\dagger, R^\dagger, \gamma^\dagger \rangle$ with rewards bounded above by 1, and a stationary or non-stationary policy π and state s. Then, for any $H \geq 0$, we have:*

$$V_{M^\dagger}^\pi(s, H) \geq V_{M^\dagger}^\pi(s) - \frac{(\gamma^\dagger)^{H+1}}{1 - \gamma^\dagger}.$$

As a corollary of this, for $H \geq \frac{1}{1-\gamma^\dagger} \ln \frac{1}{\epsilon(1-\gamma^\dagger)}$, we have:

$$V_{M^\dagger}^\pi(s, H) \geq V_{M^\dagger}^\pi(s) - \epsilon.$$

By the same argument, in the case that the rewards are bounded below by -1,

$$V_{M^\dagger}^\pi(s, H) \leq V_{M^\dagger}^\pi(s) + \epsilon.$$

Proof. This proof follows the proof of Lemma 2 from Kearns and Singh (2002).

Observe that by truncating any trajectory to H steps, the cumulative discounted reward for this trajectory can drop by a value of at most:

$$\sum_{t=H+1}^\infty (\gamma^\dagger)^t = \frac{(\gamma^\dagger)^{H+1}}{1 - \gamma^\dagger},$$

which happens when it receives a reward of 1 at every time step after H. Thus for any H such that the above quantity is lesser than or equal to ϵ, we will have $V^\pi(s, H) \geq V^\pi(s) - \epsilon$. This is indeed true for $H \geq \frac{1}{1-\gamma^\dagger} \ln \frac{1}{\epsilon(1-\gamma^\dagger)}$. □

Lemma 19. *Suppose that all confidence intervals are admissible. Let $M^\dagger = \langle S, A, T, R^\dagger, \gamma^\dagger \rangle$ be the same MDP as M except with arbitrary rewards that are upper bounded by 1 and discount factor $\gamma^\dagger < 1$. Let $\overline{\pi}^\dagger$ denote the optimal policy i.e., $\forall s, \overline{\pi}^\dagger(s) = \arg\max_{a \in A} \overline{Q}^\dagger(s, a)$. Let $\overline{M}^\dagger$ denote the optimal MDP. Then, for all $H \geq 0$ and for all (s, a), we have:*

$$V_{\overline{M}^\dagger}^{\overline{\pi}^\dagger}(s) \geq V_{M^\dagger}^*(s)$$
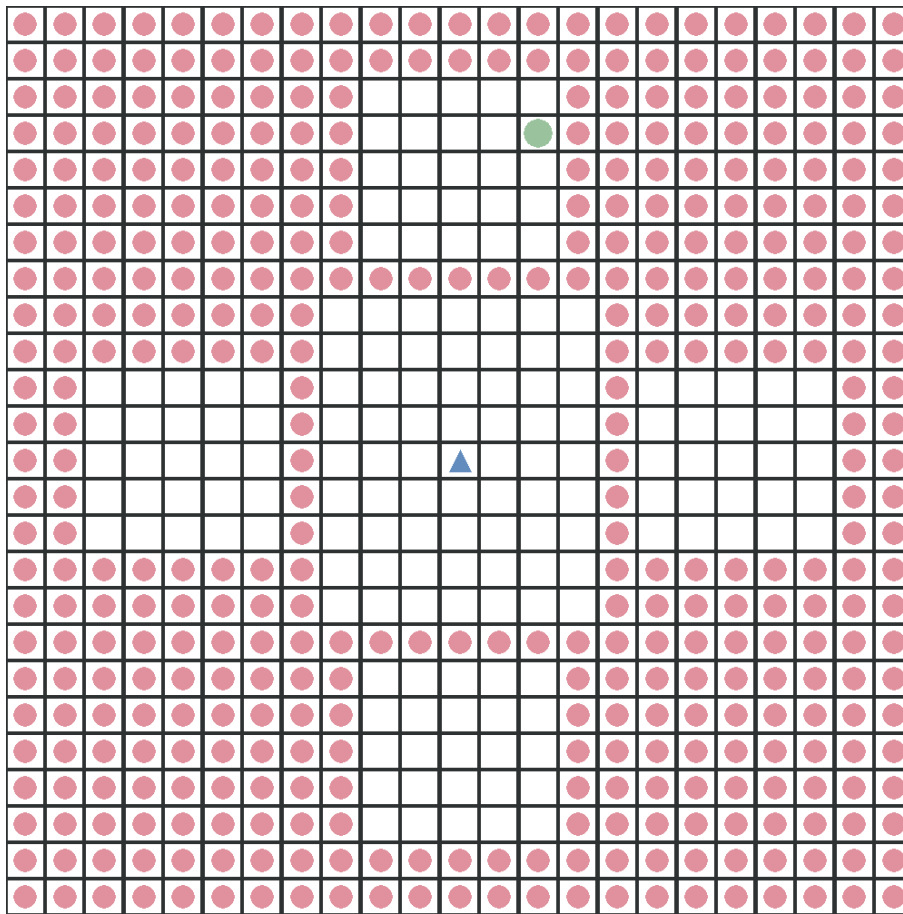
Figure 3: Full map of the Unsafe Grid World environment. The green circle marks the goal, the blue triangle marks the initial location of the agent s_{init}, and red circles correspond to dangerous states.

Proof. See Lemma 6 from Strehl and Littman (2008). Note that this proof also applies to MDPs with negative rewards.

$\square$

E   Experiment Details

E.1   Unsafe Grid World

The first domain we consider is a grid world domain with dangerous states, where the agent receives a reward of -1 for any action and the episode terminates. The agent starts on a 7×7 island of safe states and is surrounded by four 5×5 islands of safe states in all four directions, separated from the center island by a one-state-thick line of dangerous states (see Figure 3). The goal is placed on one of the surrounding islands. The agent can take actions up, down, left, or right to move in those directions one step, or can take actions jump up, jump down, jump left, or jump right to move two steps, allowing the agent to jump over dangerous states. There is a slipping probability of 60%, which causes the agent to fall left or right of the intended target (30% for either side).

The initial safe set provided to the agent is the whole center island (except for the corners) and all actions that with probability 1 will keep the agent on the center island. The distance function Δ provided to the agent is $\Delta((s, a), (\tilde{s}, \tilde{a})) = 0$ if $a = \tilde{a}$ and s and $\tilde{s}$ are within 5 steps from each other (in L_∞ norm) and $\Delta((s, a), (\tilde{s}, \tilde{a})) = 1$ otherwise. The analogous state function α is simply $\alpha((s, \cdot, s'), (\tilde{s}, \cdot)) = (x_{s'} + (x_{\tilde{s}} - x_s), y_{s'} + (y_{\tilde{s}} - y_s))$, where the subscripts denote the state to which the attribute belongs.

E.2   Discrete Platformer

We also consider a more complicated discrete Platformer domain. The states space consists of tuples $(x, y, \dot{x}, \dot{y})$ where x, y are the coordinates of the agent and $\dot{x}, \dot{y}$ are the directional velocities of the agent. The actions provided to the agent are the tuple $(\dot{x}_{\text{desired}}, j)$ where $\dot{x}_{\text{desired}}$ is the desired $\dot{x}$ and ranges from -2 to 2, and j is a boolean indicating whether or not the agent should jump. While on the ground, at every step $\dot{x}$ changes by at most 1 in the direction of $\dot{x}_{\text{desired}}$ and $\dot{y} \in \{1, 2\}$ if $j = 1$ (otherwise $\dot{y}$ remains unchanged). While in the air, however, the agent's actions have no effect and gravity decreases $\dot{y}$ by one at every step. When the agent returns to the ground, $\dot{y}$ is set to 0.

There are three types of surfaces in the environment: 1) concrete, 2) ice, and 3) sand. These surfaces change how high the agent can jump. On concrete, when the agent jumps, $\dot{y} = 2$ with probability 1; on ice $\dot{y} = 2$ with probability 0.5 and $\dot{y} = 1$ with probability 0.5; and on sand $\dot{y} = 1$ with probability 1.

The environment is arranged into three islands. The first island has all three surface materials from left to right: sand, ice, then concrete. The next two islands are just concrete, with the last one containing the goal state (where the reward is 1). The regions surrounding these island are unsafe, meaning they produce rewards of -1 and are terminal. The islands are spaced apart such that the agent must be on concrete to make the full jump to the next islands (and visa versa).

The initial safe set provided to the agent is the whole first island and all actions that with probability 1 will keep the agent on the center island. The distance function Δ provided to the agent is $\Delta((s, a), (\tilde{s}, \tilde{a})) = 0$ if $a = \tilde{a}$ and s and $\tilde{s}$ are either both in the air or both on the same type of surface and $\Delta((s, a), (\tilde{s}, \tilde{a})) = 1$ otherwise. The analogous state function α is simply $\alpha((s, \cdot, s'), (\tilde{s}, \cdot)) = \tilde{s}'$ where $\tilde{s}'$ has the same y, $\dot{x}$, and $\dot{y}$ values as s' with the x value shifted by the x difference between s and $\tilde{s}$.

E.3   Baselines

Here we describe the details for the baselines we compare against. We note that all these baselines make use of the distance metric and analogous state function to transfer information between different states, just like our algorithm. For all of our "unsafe" algorithms, we set all negative rewards to be very large to ensure that they converged to the safe-optimal policy. To improve the runtime of the experiments, the value functions and safe sets are only re-computed every 100 time steps.

MBIE   MBIE (Strehl and Littman, 2008) is a guided exploration algorithm that always follows a policy that maximizes an optimistic estimate of the optimal value function. As noted above, one of the motivations of our method was to construct a safe version of MBIE.

R-Max and Safe R-Max   The next algorithm we compare against is R-Max (Brafman and Tennenholtz, 2002). This algorithm sets the value function for all state-action pairs that have been seen fewer than m times (for some integer m) to be equal to $V_{\max}$, the maximum value the agent can obtain. In order to ensure that all states are sufficiently explored and still make use of the analogous state function, we set the value of any state-action pair, (s, a), to $V_{\max} = 1$ (since all goal states are terminal) only if there is a state-action pair similar to (s, a) with a transferred confidence interval length greater than some $\epsilon' > 0$. In mathematical terms, all state-action pairs in $\{(s, a) \in S \times A : \exists (\tilde{s}, \tilde{a}) \in S \times A \text{ where } \overset{\scriptscriptstyle\Delta}{\hat{\epsilon}}_T(s, a) < \epsilon' \text{ and } \Delta((s, a), (\tilde{s}, \tilde{a})) < \tau/2\}$ are set to $V_{\max}$. Clearly, this requires at most every state to be explored m times, but in most cases decreases the number of times each state-action pair needs to be explored. In our experiments we set $\epsilon' = \tau/2$ to give R-Max the most generous comparison against ASE, since ASE requires that a state-action only have a confidence interval of $\tau/2$ before it can be marked as safe. However, note that in many problems $\tau/2$ may be much larger than the desired confidence interval.

Our safe modification of this algorithm, "Safe R-Max," simply restricts the allowable set of actions the agent can take to $\hat{Z}_{\text{safe}}$.

ϵ-greedy and Safe ϵ-greedy   Another classic algorithm we compare against is ϵ-greedy. This algorithm acts according to the optimal policy over its internal model at every time step with probability $1 - \epsilon$ and with probability ϵ the agent takes a random action. For our experiments we anneal ϵ between 1 and 0.1 for the first N number of steps ($N = 5{,}000$ for the unsafe grid world and $N = 20{,}000$ for the discrete platformer game). Our

safe modification of this algorithm, "Safe ϵ-greedy," simply restricts the allowable set of actions the agent can take to $\hat{Z}_{\text{safe}}$.

Undirected ASE We also compare against a modified version of our algorithm "Undirected ASE." This modification changes Algorithm 3 such that $Z_{\text{edge}} \leftarrow \{(s,a) \in \hat{Z}_{\text{safe}}^c \mid s \in \hat{Z}_{\text{safe}}\}$, removing the use of $\overline{Z}_{\text{goal}}$. With this change, "Undirected ASE" simply tries to expand the safe set in all directions, instead of only along the direction of the optimistic goal policy. This baseline is to illustrate the efficacy of using our directed exploration method.