Near-Optimal Algorithms for Autonomous Exploration and Multi-Goal Stochastic Shortest Path

Haoyuan Cai 1  Tengyu Ma 2  Simon Du 3

Abstract
We revisit the incremental autonomous exploration problem proposed by Lim & Auer (2012). In this setting, the agent aims to learn a set of near-optimal goal-conditioned policies to reach the $L$-controllable states: states that are incrementally reachable from an initial state $s_0$ within $L$ steps in expectation. We introduce a new algorithm with stronger sample complexity bounds than existing ones. Furthermore, we also prove the first lower bound for the autonomous exploration problem. In particular, the lower bound implies that our proposed algorithm, Value-Aware Autonomous Exploration, is nearly minimax-optimal when the number of $L$-controllable states grows polynomially with respect to $L$. Key in our algorithm design is a connection between autonomous exploration and multi-goal stochastic shortest path, a new problem that naturally generalizes the classical stochastic shortest path problem. This new problem and its connection to autonomous exploration can be of independent interest.

1. Introduction
Reinforcement learning (RL) with a known state space has been studied in a wide range of settings (e.g., Schmidhuber, 1991; Oudeyer et al., 2007; Oudeyer & Kaplan, 2009; Baranes & Oudeyer, 2009). When the state space is large, it is difficult for a learning agent to discover the whole environment. Instead, the agent can only explore a small portion of the environment. At a high level, we hope that the agent can discover states near the initial state, expand the range of known states by exploration, and learn near-optimal goal-conditioned policies for the known states. Because the agent discovers its known states of the environment incrementally, this learning problem was named Autonomous Exploration (AX) (Lim & Auer, 2012; Tarbouriech et al., 2020).

The autonomous exploration problem generalizes the Stochastic Shortest Path (SSP) problem (Bertsekas et al., 2000) where the agent aims to reach a predefined goal state while minimizing its total expected cost. However, in the autonomous exploration setting, the agent aims to discover a set of reachable states in a large environment and find the optimal policies to reach them. The autonomous exploration formulation is applicable to an increasing number of real-world RL problems, ranging from navigation in mazes (Devo et al., 2020) to game playing (Mnih et al., 2013). For example, in the maze navigation problem, a robot aims to follow a predefined path in an unknown environment, and the robot has to discover and expand the size of regions known to itself autonomously without prior knowledge of the environment. See (Lim & Auer, 2012) for more discussions.

Related Work. The setting of autonomous exploration (AX) was introduced by Lim & Auer (2012), who gave the first algorithm, UcbExplore, with sample complexity $\tilde{O}(L^3S^2A/\varepsilon^3)$. Here $L$ denotes the distance within which we hope the learning agent to discover, $S$ denotes the number of states we need to explore, $A$ denotes the size of the action space, and $\varepsilon$ denotes the error that we can tolerate. Recent work by Tarbouriech et al. (2020) designed the DisCo algorithm with a sample complexity bound $O(L^3S^2A/\varepsilon^2)$, which improves the $1/\varepsilon$ dependency. We will briefly discuss the two algorithms in Sect. 2.1. In this paper, we present a new algorithm, VALAE (Alg. 2), to further improve the sample complexity, and we also derive the first lower bound.

1.1. Contributions
In this paper, we take important steps toward resolving the autonomous exploration problem. We compare our results

In AX, $S$ is often significantly smaller than the size of the entire state space.

We translate their absolute error $\varepsilon_{abs}$ to the relative error $\varepsilon_{rel}$, and $\varepsilon_{abs} = \varepsilon_{rel}L$. We will explain the difference of two definitions of $\varepsilon$ in Sect. 2.1.
Sample Complexity \( \Omega(\tilde{\epsilon}^2) \)

While our work borrows ideas from prior work on autonomous exploration (Lim & Auer, 2012; Tarbouriech et al., 2020) and recent advances in SSP (Tarbouriech et al., 2021), we develop new techniques to overcome additional difficulties that are unique in autonomous exploration.

### 1.2. Main Difficulties and Technique Overview

While our work borrows ideas from prior work on autonomous exploration (Lim & Auer, 2012; Tarbouriech et al., 2020) and recent advances in SSP (Tarbouriech et al., 2021), we develop new techniques to overcome additional difficulties that are unique in autonomous exploration.

### Connection between Autonomous Exploration and Multi-Goal SSP

In standard RL setting, it is known that in order to obtain a tight dependency on \( L \), one needs to exploit the variance information in the value function (Azar et al., 2017). However, in autonomous exploration, it is unclear how to exploit the variance information because even which state is reachable is unknown.

To this end, we first consider a simpler problem, multi-goal SSP, and extend the technique for single-goal SSP (Tarbouriech et al., 2021) to this new problem (cf. Alg. 2). We also present a reduction from autonomous exploration to multi-goal SSP (cf. Alg. 1). These two techniques together yield the first tight dependency on \( L \) for autonomous exploration.

### Using Regret to Bound the Sample Complexity

To estimate the sample complexity of VALAE, we need to bound the total number of rounds \( r \). Inspired by (Lim & Auer, 2012), we classify each round into three categories: failure round, success round, and skipped round. Moreover, we adopt the idea of using regret bound.

A failure round has regret larger than \( \tilde{O}(L/\varepsilon) \), but the number of failure rounds \( r_f \) is hard to estimate. The number of success rounds and skipped rounds are bounded by \( \tilde{O}(S A) \), but the regret in a success round or skipped round can be negative. Hence, to bound the total number of failure rounds \( r_f \), careful analyses of both the upper bound and the lower bound of regret are required.

For the upper bound, we extend the techniques of variance analysis from classical SSP (cf. (Tarbouriech et al., 2021)) to this problem, and we obtain the upper bound of regret scaling as \( \tilde{O}(\sqrt{T}) \). For the lower bound, the total regret in all the failure rounds grows linearly with respect to \( r_f \), and we use concentration inequalities to lower bound the total regret in success rounds and skipped rounds (cf. Lem. D.3.) By solving the inequality that the lower bound of regret is no more than the upper bound, we can obtain an upper bound of \( r_f \), and we can finally bound the total number of rounds \( r \).

### 2. Preliminaries

#### Notations.

For any two vectors \( X, Y \in \mathbb{R}^S \), we write their inner product as \( XY := \sum_{s \in S} X(s)Y(s) \). We denote \( \|X\|_\infty := \max_{s \in S} |X(s)| \), and if \( X \) is a probability distribution on \( S \), we define \( \mathbb{V}(X, Y) := \sum_{s \in S} X(s)Y(s)^2 - (\sum_{s \in S} X(s)Y(s))^2 \), i.e. the variance of random variable \( Y \) over distribution \( X \).

#### Markov Decision Process.

We consider an MDP \( M := (S, A, P, c, s_0) \), where \( S \) is the state space with size \( S \), \( A \) is the action space with size \( A \), and \( s_0 \in S \) is the initial state. In state \( s \), taking action \( a \) has a cost drawn i.i.d. from a distribution on \([c_{\min}, 1]\) (where \( c_{\min} > 0 \)) with expectation \( c(s, a) \),

| Algorithm          | Sample Complexity |
|--------------------|------------------|
| UcbExplore         | \( \tilde{O}(L^3 S^2 A/\varepsilon^3) \) |
| DisCo              | \( \tilde{O}(L^3 S^2 A/\varepsilon^2) \) |
| VALAE              | \( \tilde{O}(LSA/\varepsilon^2) \) |
| Lower Bound        | \( \Omega(LSA/\varepsilon^2) \) |

Table 1: Comparisons between our results and prior results. Algorithms and results in this paper are in grey cells. \( L \) is the exploration radius, \( A \) is the action number, \( S \) is the number of states we need to explore, and \( \varepsilon \) is the target accuracy. We will define them in Sect. 2. For simplicity, we only display the leading term in terms of the scaling in \( 1/\varepsilon \).

1 In (Lim & Auer, 2012), the cost is 1 uniformly for all state-action pairs. In this paper, we allow non-uniform costs. In Table 1, we consider uniform costs for fair comparisons.
and transits to the next state \( s' \) with probability \( P(s'|s,a) \).
For convenience, we use \( P_{s,a} \) and \( P_{s,a,s'} \) to denote \( P(s'|s,a) \) and \( P(s'|s,a,s') \), respectively. A deterministic and stationary policy \( \pi : S \rightarrow A \) is a mapping, and the agent following the policy \( \pi \) will take action \( \pi(s) \) at state \( s \).

For a fixed state \( g \in S \) we define the random variable \( t_g^s(g) \) as the number of steps it takes to reach state \( g \) starting from state \( s \) when executing policy \( \pi \), i.e., \( t_g^s(g) := \inf\{ t \geq 0 : s_{t+1} = g | s_1 = s_0, \pi \} \). A policy \( \pi \) is a proper policy if for any state \( s \in S \), \( t_g^s(s) < +\infty \) with probability 1. Then we define the value function of a proper policy \( \pi \) with respect to the goal state \( g \) and its corresponding \( Q \)-function as follows:

\[
V_g^\pi(s) = \mathbb{E} \left[ \sum_{t=1}^{t_g^s(s)} c_t(s_t, \pi(s_t)) \mid s_1 = s \right],
\]
\[
Q_g^\pi(s, a) = \mathbb{E} \left[ \sum_{t=1}^{t_g^s(s)} c_t(s_t, \pi(s_t)) \mid s_1 = s, \pi(s_1) = a \right],
\]

where \( c_t \in [c_{\min}, 1] \) is the instantaneous cost at step \( t \) incurred by the state-action pair \((s_t, \pi(s_t))\), and the expectation is taken over the random sequence of states generated by executing \( \pi \) starting from state \( s \in S \). Here we have \( V_g^\pi(g) = 0 \). We use \( \pi_Q \) to denote the greedy policy over a vector \( Q \in \mathbb{R}^{S \times A} \), i.e., \( \pi_Q(s) := \arg \min_{a \in A} Q(s,a) \).

For a fixed state \( g \in S \), we denote \( V_g^* \) as the value function of the optimal policy on MDP with respect to goal state \( g \), and here we list some important properties of \( V_g^* \): there exists a stationary, deterministic and proper policy \( \pi^* \), such that its value function \( V_g^* := V_g^{\pi^*} \) and its corresponding \( Q \)-function \( Q_g^* := Q_g^{\pi^*} \) satisfies the following Bellman optimality equations (cf. Lem. A.1):

\[
Q_g^*(s, a) = c(s, a) + P_{s,a} V_g^* , \quad V_g^*(s) = \min_{a \in A} Q_g^*(s, a).
\]

We stress that in our setting, given an MDP \( M \), the agent knows the state space \( S \), the action space \( A \), the constant \( c_{\min} \), but the agent has no prior knowledge of the transition model \( P \) or the cost function \( c \). In each step \( t \), the agent knows its current state \( s_t \in S \), and taking an action \( a_t \in A \) will transit to another state \( s' \) with some cost \( c_t \).

Incrementally \( L \)-controllable States. Before we introduce the Autonomous Exploration problem, we need to define incrementally \( L \)-controllable states, which are the states we need to explore. To formally discuss the setting, we need the following assumption on our MDP \( M \).

Assumption 2.1. The action space contains a RESET action s.t. \( P(s|s, \text{RESET}) = 1 \) for any \( s \in S \). Moreover, taking RESET in any state \( s \) will incur a cost \( c_{\text{RESET}} \) with probability 1, where \( c_{\text{RESET}} \) is a constant in \([c_{\min}, 1]\).

Given any fixed length \( L \geq 1 \), the agent needs to learn the set of incrementally controllable states \( S_L^+ \). To introduce the concept of \( S_L^+ \), we first give the definition of policies restricted on a subset:

Definition 2.2 (Policy restricted on a subset). For any \( S' \subseteq S \), a policy \( \pi \) is restricted on the set \( S' \) if \( \pi(s) = \text{RESET} \) for all \( s \notin S' \).

Now we discuss the optimal policy restricted on a set of states \( K \subseteq S \) with respect to goal state \( g \). We denote \( V_{K,g}^* \in \mathbb{R}^S \) as the value function of the optimal policy restricted on \( K \) with goal \( g \in S \), and \( Q_{K,g}^* \) as the \( Q \)-function corresponding to \( V_{K,g}^* \). We consider the case that there exists at least one proper policy restricted on \( K \) with the goal state \( g \). Then, \( V_{K,g}^* \) and \( Q_{K,g}^* \) are finite, and they satisfy the following Bellman equations:

\[
Q_{K,g}^*(s, a) = c(s, a) + P_{s,a} V_{K,g}^* , \quad \forall (s, a) \in S \times A , \quad V_{K,g}^*(s) = \min_{a \in A} Q_{K,g}^*(s, a) , \quad \forall s \in K, s \neq g , \quad V_{K,g}^*(s) = 0 .
\]

We note that when \( K_1 \subseteq K_2 \), for any \( g \in S \), if \( V_{K_1,g}^* \) is finite, then \( V_{K_2,g}^* \) is also finite, and we have \( V_{K_2,g}^* \leq V_{K_1,g}^* \) component-wise. Also, we have \( V_g^* = V_{S,g}^* \) component-wise.

Now we introduce the definition of incrementally controllable states \( S_L^+ \) (see [Tarbouriech et al., 2020] for more intuitions on this definition.):

Definition 2.3 (Incrementally \( L \)-controllable states \( S_L^+ \)). Let \( < \) be any partial order on \( S \). We denote \( S_L^+ \) as the set of states reachable from \( s_0 \) with expected cost no more than \( L \) w.r.t. \( < \), which is defined as follows:

\[
\text{• } s_0 \in S_L^+ , \quad \text{• if there is a policy } \pi \text{ restricted on } \{s' \in S_L^+ : s' < s\} \text{ such that } V_g^*(s_0) \leq L , \text{ then } s \in S_L^+ .
\]

The set of incrementally \( L \)-controllable states \( S_L^+ \) is given by \( S_L^+ = \bigcup_{<} S_L^+ \). And we denote \( S_L = |S_L^+| \).

Multi-Goal Stochastic Shortest Path. Now we define the multi-goal SSP problem, a natural generalization of the classical SSP problem. In multi-goal SSP, we consider an MDP \( M \) that satisfies Assmp. 2.1, and all of its states are incrementally \( L \)-controllable, i.e. \( S_L^+ = S \). Also, we assume that the agent knows \( L \).

A learning algorithm for multi-goal SSP takes the error parameter \( \epsilon \in (0, 1) \), confidence \( \delta \in (0, 1) \), and the goal space \( G \subseteq S \) as input, and with probability over \( 1 - \delta \), the algorithm outputs a set of policies \( \{\pi_s\}_{s \in G} \), such that

\[
\forall s \in G, V_g^\pi(s_0) \leq V_g^* (s_0) + \epsilon L ,
\]

i.e., the algorithm learns near-optimal policies to reach each
We want to find an algorithm with a probably approximately correct (PAC) bound of \( C_T \), i.e., with probability over \( 1 - \delta \), \( C_T \) is bounded by some polynomial of \( L, S, A, \varepsilon^{-1}, \varepsilon_{\text{min}}^{-1}, \) and \( \log(1/\delta) \).

Here we explain the reason why we need the RESET action (Asmp. 2.1). The classical SSP problem uses an episodic learning protocol, i.e. when the agent reaches the goal state \( g \), the agent can "reset" to initial state \( s_0 \) and start a new episode. But in multi-goal SSP, we do not have episode learning protocol because we need to ensure that for each goal \( g \in \mathcal{G} \), the agent learns a near-optimal policy to reach \( g \). Therefore, each time when the agent arrives at any of the goal, the agent has to "reset" to \( s_0 \). Hence the RESET action is necessary, and the previous works (Lim & Auer, 2012) and (Tarbouriech et al., 2020) also assume the existence of the RESET action.

We also remark that multi-goal SSP is fundamentally different from reward-free RL (Jin et al., 2020). Reward-free RL contains two phases: exploration phase and planning phase. In exploration phase we have no knowledge of reward, \( r \), and in planning phase we cannot interact with MDP. But in multi-goal SSP, we can estimate the cost function \( c \), and the agent does not need to separate into two phases.

**Autonomous Exploration.** Now we introduce the autonomous exploration (AX) problem, which generalizes multi-goal SSP. AX problem was first introduced by (Lim & Auer, 2012), and we use their definition of AX problem.

In AX, we consider an MDP \( M \) that satisfies Asmp. 2.1. A learning algorithm of AX problem inputs the exploration radius \( L \geq 1 \), the error parameter \( \varepsilon \in (0, 1) \) and confidence \( \delta \in (0, 1) \), and with probability over \( 1 - \delta \), the algorithm should output a set of "known" states \( \mathcal{K} \subseteq \mathcal{S} \) such that \( \mathcal{S}_{r}^{\mathcal{K}} \subseteq \mathcal{K} \), i.e., the algorithm discovers all the states that we want to explore. And the algorithm should also output a set of policies \( \{\pi_s\}_{s \in \mathcal{K}} \), such that

\[
\forall s \in \mathcal{S}_{r}^{\mathcal{K}}, V_{\pi_s}(s_0) \leq (1 + \varepsilon)L,
\]

i.e., the algorithm learns a policy to reach each \( s \in \mathcal{S}_{r}^{\mathcal{K}} \) and the expected cost is no more than \((1 + \varepsilon)L\). In AX, we also use cumulative cost \( C_T \) to measure the performance, but we hope \( C_T \) depends on \( |\mathcal{S}_{r}^{\mathcal{K}}| \) instead of the global size \( |\mathcal{S}| \).

We note that different complexity bounds of \( C_T \) may depend on \( S_L, S_{2L}, S_{(1+\varepsilon)L} \). But if we assume that \( S_L \) grows polynomially with respect to \( L \), i.e., there exist constants \( C, d \) independent of \( L \), such that \( S_L \leq CL^d \) for all \( L \geq 1 \), we will have \( S_{2L} \leq C^2L^d = O(L^d) \), and \( S_{(1+\varepsilon)L} = O(L^d) \).

Under this assumption, \( S_L, S_{2L}, S_{(1+\varepsilon)L} \) are of the same order \( O(L^d) \), thus we use \( S \) as the abbreviation for all these quantities in Table 1. This assumption is implicitly considered in the literature, because otherwise one may need to consider the logarithmic dependency on \( S_L \).

In AX, the learning agent does not know the set \( \mathcal{S}_{r}^{\mathcal{K}} \) or the size of \( \mathcal{S}_{r}^{\mathcal{K}} \), and it needs to discover and explore \( \mathcal{S}_{r}^{\mathcal{K}} \) by itself and find policies to reach each state in \( \mathcal{S}_{r}^{\mathcal{K}} \). This is why the problem is called "autonomous exploration".

We remark that in Sect. 3, we will prove that our Alg. 2 outputs a set \( \mathcal{K} \supseteq \mathcal{S}_{r}^{\mathcal{K}} \) and a set of policies \( \{\pi_s\}_{s \in \mathcal{K}} \) restricted on \( \mathcal{K} \), such that

\[
\forall s \in \mathcal{K}, V_{\pi_s}(s_0) \leq V_{K,s}(s_0) + \varepsilon L.
\]

This implies \( \forall s \in \mathcal{S}_{r}^{\mathcal{K}}, V_{\pi_s}(s_0) \leq (1 + \varepsilon)L \), because when \( \mathcal{S}_{r}^{\mathcal{K}} \subseteq \mathcal{K} \), we have \( V_{K,s}(s_0) \leq V_{\pi_s}(s_0) \), and for any \( s \in \mathcal{S}_{r}^{\mathcal{K}} \), we have \( V_{K,s}(s_0) \leq L \).

In the special case that \( \mathcal{S}_{r}^{\mathcal{K}} = \mathcal{S} \) (i.e., in the setting of multi-goal SSP), our Alg. 2 will output \( \mathcal{K} = \mathcal{S} \), and the inequality above will be reduced to \( \forall s \in \mathcal{S}, V_{\pi_s}(s_0) \leq V_{\pi_s}(s_0) + \varepsilon L \). Hence our Alg. 2 for AX problem also solves multi-goal SSP problem with goal space \( \mathcal{G} = \mathcal{S} \).

### 2.1. Review of Prior Algorithms

We review prior algorithms because our algorithm also relies on some components from prior algorithms.

**DisCo Algorithm for Autonomous Exploration.**

DisCo algorithm was introduced in (Tarbouriech et al., 2020), and we use DisCo algorithm as a burn-in step for Alg. 2. Here we give the lemma of the sample complexity of DisCo algorithm for autonomous exploration.

**Lemma 2.4** (Corollary 1, (Tarbouriech et al., 2020)). Assume that \( L \geq 1 \), \( 0 < \varepsilon \leq 1 \) and \( 0 < \delta < 1 \). For any
Near-Optimal Algorithms for Autonomous Exploration and Multi-Goal Stochastic Shortest Path

\[ M = (S, A, P, c, s_0) \text{ satisfying Asmp. 2.1, with probability at least } 1 - \delta, \text{ DisCo algorithm will terminate and output a set of states } K \text{ such that } S^{\text{ref}}_L \subseteq K \subseteq S^{\text{ref}}_{1+\varepsilon}L, \text{ and a set of policies } \{\pi_s\}_{s \in K} \text{ restricted on } K, \text{ such that } \forall s \in K, V^{\pi_s}_L(s_0) \leq V^{\pi^*}_L(s_0) + \varepsilon L, \text{ and the cumulative cost } C_T = \bar{O}(L^3 S^2_0(1+\varepsilon)L A_{\text{min}}^{-2}). \]

Here we clarify that the definitions of \( \varepsilon \) in our work and in (Tarbouriech et al., 2020) are different. Tarbouriech et al. (2020) denotes absolute error as \( \varepsilon \) (i.e., they require that the output policies satisfy \( V^{\pi^*_L}_L(s_0) \leq L + \varepsilon \) and \( V^{\pi^*_L}_L(s_0) \leq V^{\pi^*_L}_L(s_0) + \varepsilon L \)), and our paper denotes relative error as \( \varepsilon \) (i.e., we require \( V^{\pi^*_L}_L(s_0) \leq (1 + \varepsilon)L \) and \( V^{\pi^*_L}_L(s_0) \leq V^{\pi^*_L}_L(s_0) + \varepsilon L \)). And their absolute error \( \varepsilon_{\text{abs}} \) and our relative error \( \varepsilon_{\text{rel}} \) satisfies the following equation: \( \varepsilon_{\text{abs}} = \varepsilon_{\text{rel}} L \).

We also remark that when \( c_{\text{min}} = 1 \), the original form of sample complexity in Theorem 1, (Tarbouriech et al., 2020) was \( \bar{O}(L^3 S_{L+c_0}^2 S_{L+c_0}^{\text{ref}} A_{\text{abs}}^{-2} + L^3 S_{L+c_0}^{\text{ref}} A_{\text{abs}}^{-2}) \), where \( \Gamma_L := \max_{(s,a) \in S_{L+c_0}^2 \times A} \| (P(s' \mid s, a))_{s' \in S_{L+c_0}^2} \|_0 \) and \( \Gamma_{L+c_0} = S_{L+c_0}^L \) in the worst case. By setting \( \varepsilon_{\text{abs}} = \varepsilon_{\text{rel}} L \) and \( \Gamma_{L+c_0} = S_{L+c_0}^L \), we can obtain the sample complexity bound \( \bar{O}(L^3 S_0^2(1+c_0)L A_{\text{min}}^{-2}) \) in Lem. 2.4 when \( c_{\text{min}} = 1 \). And in Corollary 1, (Tarbouriech et al., 2020), they discussed the case when \( c_{\text{min}} \in (0, 1) \), which incurs an additional \( c_{\text{min}}^2 \) in their sample complexity.

3. Algorithms and Sample Complexity Bounds

Now we are ready to describe our main algorithm VALAE (cf. Alg. 2), and currently we focus on autonomous exploration problem. There are three key components in Alg. 2. The first component is running DisCo algorithm (cf. (Tarbouriech et al., 2020)) with \( \varepsilon = 1 \). Our aim is to discover a set of states \( K \) such that \( S^{\text{ref}}_L \subseteq K \subseteq S^{\text{ref}}_{2L} \), and compute a set of policies \( \{\pi_s\}_{s \in K} \) to reach each state \( s \in K \) with expected cost \( V^{\pi^*_L}_L(s_0) \) no more than \( 2L \). After the first component, we will fix our set \( K \), and to solve the AX problem, we need only learn a set of policies \( \{\pi_s\}_{s \in K} \) such that \( V^{\pi^*_L}_L(s_0) \leq V^{\pi^*}_L(s_0) + \varepsilon L \) for all \( s \in K \).

The second component reduces the autonomous exploration problem to multi-goal SSP (cf. Alg. 1) using the set \( K \) computed from the first component. Alg. 1 first constructs a new MDP \( M^1 \) by “merging” all the states \( s \notin K \) to a single artificial state \( x \), and to solve AX problem, we need only solve multi-goal SSP problem on MDP \( M^1 \) with goal space \( G = K \). Then the algorithm collects fresh samples of the form \((s, a, s', c)\) for all state-action pairs \((s, a) \in K \times A\), and the aim is to compute the empirical probability \( \hat{P}(s' \mid s, a) \) and the average cost \( \hat{c}(s, a) \) with small error.

In the third component, inspired by recent advances in stochastic shortest path (Tarbouriech et al., 2021), we design a policy evaluation step to obtain near-optimal estimates of the costs of getting to each \( s \in S^{\text{ref}}_L \) (cf. Alg. 2).

Below we give detailed descriptions for each component.

3.1. Running DisCo Algorithm with \( \varepsilon = 1 \)

In the first component of our main algorithm VALAE (cf. Alg. 2), we use DisCo algorithm with (relative) error \( \varepsilon = 1 \) as a subroutine. By Lem. 2.4, we can obtain a set \( K \) such that \( S^{\text{ref}}_L \subseteq K \subseteq S^{\text{ref}}_{2L} \), and a set of policies \( \{\pi_s\}_{s \in K} \) such that \( \forall s \in K, V^{\pi^*_L}_L(s_0) \leq 2L \), and the total cost is bounded by \( \bar{O}(L^3 S_0^2(1+c_0)L A_{\text{min}}^{-2}) \). In the next subsection, we will focus on a fixed set \( K \), and reduce the autonomous exploration problem to multi-goal SSP problem.

3.2. Connection between Autonomous Exploration and Multi-Goal SSP

In our main algorithm VALAE (Alg. 2), after running DisCo with \( \varepsilon = 1 \), we have obtained a set of known states \( K \supseteq S^{\text{ref}}_L \) and discovered all the states that we want to explore, and we denote \( K = |K| \). Now we focus on the second component of VALAE (cf. Alg. 1). We will fix our set of known states \( K \), and focus only on the policies restricted on \( K \). Therefore, for all the states \( s \notin K \), we can regard them as one artificial state \( x \), and the only action at state \( x \) is RESET. To this purpose, we will construct an MDP \( M^1 := (\langle K^1, A, P^1, c^1, s_0 \rangle) \) where we first define the artificial state \( x \), and we set \( K^1 = K \cup \{x\} \).
and we denote $K' = |K^\dagger| = K + 1$. For any $(s, a) \in K \times A$, we define $P^{\dagger}_{s,a,s'}$ as follows:

$$P^{\dagger}_{s,a,s'} = P_{s,a,s'} \forall s' \in K,$$

and $P^{\dagger}_{s,a,s'} = \sum_{s' \notin K} P_{s,a,s'}$.

We also define $P_{x,a,s'} = \mathbb{I}[s' = s_0]$ for any $a \in A$, $s' \in K^\dagger$. Finally, we define $c'(s, a) = c(s, a)$ for all $(s, a) \in S \times A$, and $c'(x, a) = \epsilon_{\text{reset}}$ for all $a \in A$. In this way, the AX problem reduces to multi-goal SSP problem on MDP $M^\dagger$ with the set of states being $K^\dagger$ and goal space $G = K$, and all states in $K$ are incrementally $2L$-controllable from $s_0$.

Next, we collect $\hat{P}(s'|s, a)$ and the average cost $\hat{c}(s, a)$, so that our estimation $\hat{P}(s'|s, a)$ and $\hat{c}(s, a)$ are close enough to $P^\dagger(s'|s, a)$ and $c^\dagger(s, a)$, respectively. In DisCo algorithm, we have computed a policy $\pi_s$ for each state $s$, such that we can execute $\pi_s$ to reach state $s$ from $s_0$ with expected cost no more than $2L$. Hence, to obtain a sample $(s, a, s', c)$ at any state-action pair $(s, a) \in K \times A$, we need only first execute $\pi_s$ to arrive at state $s$, then execute action $a$.

We remark that using fresh samples is essential for Alg. 2 to ensure these samples are independent of $K$, and we cannot use the samples collected in DisCo algorithm because they are dependent of $K$. Also, we note that in Alg. 1 and Alg. 2, the estimated transition probability $\hat{P}(s'|s, a)$ and the estimated cost $\hat{c}(s'|s, a)$ are only evaluated for all $(s, a) \in K^\dagger \times A$ on MDP $M^\dagger$, rather than for all $(s, a) \in S \times A$ on MDP $M$, hence the computational complexity of Alg. 1 and Alg. 2 does not depend on “global” $|S|$.

We note that the idea of uniformly connecting $\phi$ samples for each state-action pair $(s, a) \in K \times A$ is similar with DisCo algorithm. The difference is that DisCo algorithm collects $\tilde{\Omega}(L^2|K|c_{\text{min}}^{-2})$ samples for each state-action pair $(s, a) \in K \times A$, but in Alg. 1 our $\phi = \tilde{\Omega}(L^2|K|c_{\text{min}}^{-2})$ and is smaller than that in DisCo.

### 3.3. Value-Aware Algorithms for Autonomous Exploration and Multi-Goal SSP

Finally we describe our main algorithm, Value-Aware Autonomous Exploration (VALAE, cf. Alg. 2). First, VALAE uses DisCo algorithm with $\epsilon = 1$ as a subroutine, and DisCo algorithm computes a set $K$ such that $S^*_K \subset K$. We discard all the samples collected in DisCo algorithm, in order to ensure the independence of $K$ and $\hat{P}_{s,a}$. Second, we use Alg. 1 as a burn-in step to collect $\tilde{\Omega}(L^2|K|c_{\text{min}}^{-2})$ samples for each of the state-action pair $(s, a)$ so that the empirical model $\hat{P}$ and the true model $P^\dagger$ are close enough. This guarantees that with high probability, in any round $r$, the expected cost of the greedy policy $\pi_\hat{\sigma}$ in Phase (a) on model $P^\dagger$ is no more than $O(L)$, which is proved in Lem. C.5.

From now on we work on the MDP $M^\dagger$, and we will solve the multi-goal SSP problem on $M^\dagger$ and compute near-optimal policies $\pi_\sigma$ for all the goal states $g \in K$. We choose the goal state $g \in K$ one by one, and we move to another goal state $g$ if the average performance of the policy $\pi_\sigma$ is close to our estimation of the optimal policy. In each round, we have two phases. In the first phase, we use $\text{VISGO}$ (cf.

---

**Algorithm 2 Value-Aware Autonomous Exploration (VALAE)**

1. **Input:** Confidence $\delta \in (0, 1)$, error $\epsilon \in (0, 1]$, and $L \geq 1$.
2. **Input (for multi-goal SSP only):** Goal Space $G \subseteq S$.
3. **(For autonomous exploration, set $G = \emptyset$)**.
4. **Specify:** Trigger set $N \leftarrow \{2^{i-1} : j = 1, 2, \ldots\}$. We run DisCo algorithm with $\epsilon = 1$ and get a set $K$ such that $S^*_K \subset K \subseteq S^*_1$.
5. Run DisCo algorithm with input $(\delta, \epsilon = 1, L)$ and we get a set $K$ and a set of policies $\{\pi_s\}_{s \in K}$.
6. Run Alg. 1 with input $(\delta, L, K, \{\pi_s\}_{s \in K})$, and obtain the variables $N(\cdot), n(\cdot), \hat{P}, \hat{\theta}(\cdot), \tilde{c}$.
7. Set time step $t \leftarrow 1$ and trigger index $j \leftarrow 5 + \log_2 \frac{1}{\epsilon_{\text{min}}}$.
8. Set $\epsilon \leftarrow \epsilon/3$, $B \leftarrow 10L$, $\lambda = \tilde{O}(1/\epsilon^2)$, and $g \leftarrow s_0$.
9. Initialize $G \leftarrow K$ if $G = \emptyset$.
10. Solve multi-goal SSP problem on $M^\dagger$ with goal space $G$.
11. **for** round $r = 1, 2, \ldots$ **do**
12. **Phase (a): Compute Optimal Policy**
13. Compute $(Q, V) := \text{VISGO}(g, 2^{-j}/(|K^\dagger|A))$.
14. Set the policy $\pi_\sigma$ as the greedy policy over $Q$, and $\tilde{\pi} \leftarrow 0$.
15. **Phase (b): Policy Evaluation**
16. **for episode $k = 1, 2, \ldots, \lambda$ do**
17. Set $s_t \leftarrow s_0$ and reset to the initial state $s_0$, and $\tilde{\tau}_k \leftarrow 0$.
18. **while** $s_t \neq g$ **do**
19. Take action $a_t := \arg\min_{a \in A} Q(s_t, a_t) \in M^\dagger$, incur cost $c_t$ and observe next state $s_{t+1} \sim P^\dagger(\cdot | s_t, a_t)$.
20. Set $(s, a, s', c) \leftarrow (s_t, a_t, s_{t+1}, c_t)$ and $t \leftarrow t + 1$.
21. Set $N(s, a) \leftarrow N(s, a) + 1, \theta(s, a) \leftarrow \theta(s, a) + c, N(s, a) \leftarrow \tilde{N}(s, a) + 1$.
22. **if** $(s, a) \in N$ **then**
23. Set $j \leftarrow j + 1, \tilde{c}(s, a) \leftarrow 2\tilde{c}(s, a) / N(s, a)$ and $\theta(s, a) \leftarrow 0$.
24. **for** all $s' \in K^\dagger$ **do**
25. Return to line 11, start a new round (the current round has been a skipped round).
26. **end if**
27. Set $\tilde{\tau} \leftarrow \tau + \frac{c}{q}, \tilde{\tau}_k \leftarrow \tilde{\tau}_k + c$.
28. **end while**
29. **if** $\tilde{\tau} > V(s_0) + \epsilon L$ **then**
30. Return to line 11, start a new round. (the current round has been a failure round).
31. **end if**
32. **end for**
33. Set $\pi_\sigma \leftarrow \pi_{\tilde{\tau}}$. Remove $g$ from $G$. (The current round has been a success round.)
34. Choose another state $g \in G$.
35. **Stop the algorithm** if $G$ is empty.
36. **end for**

**Output:** The states $s$ in $K$ and their corresponding policy $\pi_s$.
Alg. 3 in Appendix C) to estimate the value function of the optimal policy with goal state \( g \) (denoted as \( V \)), and we set the policy \( \pi \) as the greedy policy over its output \( Q \). We note that \( V \) is optimistic, i.e., \( V(s) \leq V_{\pi}^*(s) \leq 2L + 1 \). Since we do not know whether the policy \( \pi \) is close enough to the optimal policy, in the second phase, we will execute \( \tilde{\pi} \) for \( \lambda = \tilde{O}(1/\epsilon^2) \) times and check whether the average performance is close enough to our estimation of the optimal cost (i.e., check whether \( \bar{\tau} \leq V(s_0) + \epsilon L \)). By setting
\[
\lambda = \left[ \frac{2048}{\epsilon^2} \ln^2 \left( \frac{256}{\epsilon} \right) \ln \left( \frac{2|K|}{\delta} \right) \right]
\]
and using concentration inequalities (Lem. D.1), we can prove that the average performance \( \bar{\tau} \) in \( \lambda \) episodes is close enough to the expected cost of \( \tilde{\pi} \). In this process, we also collect samples, and use them to help us estimate the value function of the optimal policy.

In the second phase, the current round will be classified into three cases: failure round, skipped round, and success round. This borrows the idea from (Lim & Auer, 2012). If the average performance of the policy \( \tilde{\pi} \) is too bad (i.e., \( \bar{\tau} \) is larger than \( V(s_0) + \epsilon L \)), we will consider the current round as a failure round. If the number of samples \( N(s,a) \) meets the trigger set (i.e. is a power of 2), we will consider the current round as a skipped round, following the idea in (Jaksch et al., 2010). Otherwise, the current round is a success round. In the case of a failure round or a skipped round, we will not change the goal state \( g \), and in the next round, we compute a new policy by VISGO using the samples collected in this round. In the case of a success round, as the average performance of the policy \( \tilde{\pi} \) is close to optimal, we can set the \( \tilde{\pi} \) as the policy \( \pi_g \) for the goal state \( g \), and choose another goal state \( g \).

**Theorem 3.1 (Cumulative Cost for AX),** Assume that \( L \geq 1 \), \( 0 < \epsilon \leq 1 \) and \( 0 < \delta < 1 \). For any MDP \( M = (S,A,P,c,s_0) \) satisfying Asmp. 2.1, with probability at least \( 1 - \delta \), our Alg. 2 will terminate and output a set of states \( K \) such that \( S^-_L \subseteq K \subseteq S^+_2 \), and a set of policies \( \{\pi_s\} \subseteq K \) restricted on \( K \), such that \( \forall s \in K, V_{\pi_s}^*(s_0) \leq V_{\pi,s}^*(s_0) + \epsilon L \), and the cumulative cost \( C_T = \tilde{O}(LS_2A\epsilon^{-2} + LS_2^2A\epsilon^{-1} + L^3S_2^2A\epsilon^{-2}) \). And when \( \epsilon \leq \min(S_2^{-1},1/\epsilon \min) \), we have \( C_T = \tilde{O}(LS_2A\epsilon^{-2}) \).

Thm.3.1 shows that Alg.2 solves autonomous exploration problem. Note that in Thm.3.1, the dependency on \( L \) is tight when \( \epsilon \rightarrow 0 \), because we leverage the variance information in the policy-evaluation phase, which is necessary in RL problems generally. DisCo algorithm does not use the variance information because it collects equal number of samples on each state-action pair \( (s,a) \), i.e., the sample collection in DisCo algorithm does not use the estimated value function as the guidance.

We highlight that the leading term of \( C_T \) does not have \( c_{\min} \). This is because the variance fundamentally does not scale with \( c_{\min} \) (cf. Lem. D.2 and Lem. D.3). While we discover a larger set \( K \subseteq S^+_{2L} \) compared with (Lim & Auer, 2012) and (Tarbouriech et al., 2020), we note that if the number of the \( L \)-controllable states grows polynomially with respect to \( L \), \( S_L \) and \( S^+_{2L} \) will be of the same order. Hence under this assumption, our sample complexity bound strictly improves the existing ones and is nearly minimax optimal.

Lastly, we note that Alg. 2 also solves the multi-goal SSP problem, and it enjoys a near-optimal sample complexity for multi-goal SSP:

**Theorem 3.2 (Cumulative Cost for Multi-Goal SSP),** Assume that \( L \geq 1 \), \( 0 < \epsilon \leq 1 \), \( 0 < \delta < 1 \) and goal space \( \mathcal{G} \subseteq S \). For any MDP \( M = (S,A,P,c,s_0) \) satisfying Asmp. 2.1 and \( S^+_c = S \), with probability at least \( 1 - \delta \), our Alg. 2 will terminate and output a set of policies \( \{\pi_s\} \subseteq K \) such that \( \forall s \in \mathcal{G}, V_{\pi,s}^*(s_0) \leq V_{\pi,s}^*(s_0) + \epsilon L \), and the cumulative cost \( C_T = \tilde{O}(LS_A\epsilon^{-2} + LS_A\epsilon^{-1} + L^3S_2^2A\epsilon^{-2}) \). And when \( \epsilon \leq \min(S^{-1},1/\epsilon \min) \), we have \( C_T = \tilde{O}(LS_A\epsilon^{-2}) \).

4. A Minimax Lower Bound for Autonomous Exploration

![Figure 1: Illustration of our construction of the hard MDP. Here we present our lower bound of sample complexity for the autonomous exploration problem, and we follow the definitions in (Domingues et al., 2021).](image)

We define a learning algorithm as a history-dependent policy \( \pi \) used to interact with an MDP \( M \), and the rigorous definition of \( \pi \) is in Appendix E. We recall that in AX setting, the algorithm eventually stops and output a set \( K \subseteq S \) and a set of policies \( \{\pi_s\} \subseteq K \). Hence we define an algorithm for the AX problem as a tuple \( (\pi, \tau, K, \{\pi_s\}) \), where \( \tau \) is the stopping time (total number of steps) chosen by the algorithm, \( K \) and \( \{\pi_s\} \subseteq K \) are the output of the algorithm. Now we formally write the definition of an algorithm for AX problem.

**Definition 4.1.** An algorithm \( (\pi, \tau, K, \{\pi_s\}) \) is \( (\epsilon, \delta, L) \)-PAC for AX problem on MDP \( M \), if with probability over \( 1 - \delta \), the algorithm returns a set of states \( K \) and a set of policies \( \{\pi_s\} \subseteq K \) after \( \tau \) steps, such that \( K \supseteq S_L^+ \) and \( \forall s \in S_L^+, V_{\pi,s}^*(s_0) \leq (1 + \epsilon)L. \)
We note that $\tau$ is a random variable over the probability distribution $P_{\pi, M}$, where $P_{\pi, M}$ is determined by algorithm $\pi$ and $MDP$ $M$, and $P_{\pi, M}$ is defined in Appendix E. Also, we denote the operator $E_{\pi, M}$ as the expectation under $P_{\pi, M}$.

Then for any real numbers $L, c_{\min}$ and positive integers $S, A, S_L$, we define a class of MDPs $\mathcal{M}(L, S_L)$ as follows: $\mathcal{M}(L, S_L)$ contains all the MDPs $M = (S, A, P, c, s_0)$, such that $|S| \leq S$, $|A| \leq A$, $c(s, a) \in [c_{\min}, 1]$ for all $(s, a) \in S \times A$, and $M$ satisfies Asmp. 2.1 and $|S_L| \leq S_L$. Finally, the following theorem states the lower bound for the autonomous exploration problem.

**Theorem 4.2.** Assume that $L > 4$, $S > 8$, $A > 4$, $4 \leq S_L \leq \min\{(A - 1)(\frac{1}{2}), S\}$, $0 < \varepsilon < \frac{1}{4}$, $0 < \delta < \frac{1}{10}$, and $0 < c_{\min} \leq 1$. Then for any algorithm $(\pi, \tau, K, \{\pi_s\}_{s \in K})$ that is $(\varepsilon, \delta, L)$-PAC for AX problem on any MDP $M \in \mathcal{M}(L, S_L)$, there exists an MDP $M \in \mathcal{M}(L, S_L)$ such that

$$E_{\pi, M}[\tau] = \Omega\left(\frac{LS_L A}{c_{\min}^2 \log \frac{1}{\delta}}\right).$$

As $\tau$ is the total number of steps used in the algorithm $\pi$, the lower bound of cumulative cost $C_T$ is $c_{\min}$ multiplies the lower bound of $\tau$, i.e., $\Omega(LS_LAe^{-2}\log \frac{1}{\delta})$. This lower bound further implies our upper bound (Theorem 3.1) is nearly minimax-optimal when $S_L$ and $S_{2L}$ are of the same order. We also have a lower bound for multi-goal SSP (cf. Appendix F).

### 4.1. Proof Sketch

We briefly sketch our proof of the lower bound. We consider the case $c_{\min} = 1$ and $L > 2$ for convenience, and we first construct our family of hard MDPs for $S = 3$ states (cf. Fig. 1), where $s_0$ is initial state, $s_1$ is middle state and $g$ is goal state.

Let $s_0$, $s_1$, and $s_2$ be the three states, and each middle state $s_i$ has a transition probability $1 - \frac{2}{L}$. In state $s_0$, the only one optimal action is $a^*$. When we take the action $a^*$ in $s_1$ (the blue edges), the agent will transit to the goal state $g$ with probability $\frac{1}{L}$ and stay at state $s_1$ with probability $1 - \frac{2}{L}$. When we take an action $a \neq a^*$ in $s_1$ (the dashed edges), the agent will transit to the goal state $g$ with smaller probability $\frac{2}{(1 + 6\varepsilon)L}$. We note that the algorithm does not act in $s_1$.

We can verify that $V_g^\pi(s_0) = \frac{L}{2} + \frac{1}{L} = L$, and $g \in S_{Lg}^*$ (hence $g$ should be contained in the learning algorithm’s output $K$). Let $\pi_g$ be the output policy of the learning algorithm with respect to goal state $g$. If $\pi_g(s_1) = a^*$, we have $V_g^\pi(s_0) = L$. Otherwise, we have $V_g^\pi(s_0) = \frac{L}{2} + \frac{1}{(1 + 6\varepsilon)L}$, $\pi_g(s_1) \neq a^*$. Hence, if the algorithm solves AX problem on this MDP, it has to discriminate between two Bernoulli distributions with $p_1 = \frac{2}{L}$ and $p_2 = \frac{2}{(1 + 6\varepsilon)L}$ among all the $A$ actions, and the KL divergence of the two distributions is $\Omega(2/L)$. Hence we can prove that we need at least $\Omega(LA/e^2)$ to solve AX on this MDP. The technique of KL divergence is similar with (Domingues et al., 2021).

Then we can generalize our hard MDP to larger $S_L$. We first construct an MDP $M_0$ with $S_L - 1$ states, and each middle state $s_i$ can be reached from $s_0$ in $L/2$ steps in expectation. Then we add a goal states $g$, and we choose one optimal state-action pair $(s_i, a^*)$ among all the middle states and actions. Finally, we set the transition probability $P(g|s_i, a^*) = \frac{2}{L}$, and $P(g|s_i, a) = \frac{2}{(1 + 6\varepsilon)L}$ for other pair of middle state and action $(s_i, a)$. In intuition, this extends the construction in Fig. 1 from $A$ actions to $O(S_L A)$ actions. The full construction is in Appendix E. Under this construction, we can prove that the lower bound scales as $\Omega(LS_LA/e^2)$.

### 5. Conclusion

We introduced a new algorithm for the autonomous exploration problem, which improves existing ones. Along the way, we also introduced a new problem, multi-goal SSP problem, which can be of independent interest. The natural future directions include designing an algorithm with $\tilde{O}(\frac{L^2A}{e^2})$ sample complexity instead of $\tilde{O}(\frac{L^2S_A}{e^2})$, and improving the lower order terms in existing bounds.

### Acknowledgements

The work is supported by JD.com. Simon S. Du gratefully acknowledges the funding from NSF Award’s IIS-2110170 and DMS-2134106.

### References

Azar, M. G., Osband, I., and Munos, R. Minimax regret bounds for reinforcement learning. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pp. 263–272. JMLR. org, 2017.

Baranes, A. and Oudeyer, P-Y. R-iac: Robust intrinsically motivated exploration and active learning. IEEE Transactions on Autonomous Mental Development, 1(3):155–169, 2009.

Bertsekas, D. P. and Tsitsiklis, J. N. An analysis of stochastic shortest path problems. Mathematics of Operations Research, 16(3):580–595, 1991.

Bertsekas, D. P. et al. Dynamic programming and optimal control: Vol. 1. Athena scientific Belmont, 2000.

Cohen, A., Kaplan, H., Mansour, Y., and Rosenberg, A. Near-optimal regret bounds for stochastic shortest path. arXiv preprint arXiv:2002.09869, 2020.

Devo, A., Costante, G., and Valigi, P. Deep reinforcement learning for instruction following visual navigation in 3d maze-like environments. IEEE Robotics and Automation Letters, 5(2):1175–1182, 2020.

Domingues, O. D., Ménard, P., Kaufmann, E., and Valko, M. Episodic reinforcement learning in finite mdps: Minimax
lower bounds revisited. In *Algorithmic Learning Theory*, pp. 578–598. PMLR, 2021.

Garivier, A., Ménard, P., and Stoltz, G. Explore first, exploit next: The true shape of regret in bandit problems. *Mathematics of Operations Research*, 44(2):377–399, 2019.

Jaksch, T., Ortner, R., and Auer, P. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(Apr):1563–1600, 2010.

Jin, C., Krishnamurthy, A., Simchowitz, M., and Yu, T. Reward-free exploration for reinforcement learning. *arXiv preprint arXiv:2002.02794*, 2020.

Lim, S. H. and Auer, P. Autonomous exploration for navigating in mdps. In *Conference on Learning Theory*, pp. 40–1. JMLR Workshop and Conference Proceedings, 2012.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Oudeyer, P.-Y. and Kaplan, F. What is intrinsic motivation? a typology of computational approaches. *Frontiers in neurorobotics*, 1:6, 2009.

Oudeyer, P.-Y., Kaplan, F., and Hafner, V. V. Intrinsic motivation systems for autonomous mental development. *IEEE transactions on evolutionary computation*, 11(2):265–286, 2007.

Schmidhuber, J. A possibility for implementing curiosity and boredom in model-building neural controllers. In *Proc. of the international conference on simulation of adaptive behavior: From animals to animats*, pp. 222–227, 1991.

Tarbouriech, J., Pirotta, M., Valko, M., and Lazaric, A. Improved sample complexity for incremental autonomous exploration in mdps. *arXiv preprint arXiv:2012.14755*, 2020.

Tarbouriech, J., Zhou, R., Du, S. S., Pirotta, M., Valko, M., and Lazaric, A. Stochastic shortest path: Minimax, parameter-free and towards horizon-free regret. *arXiv preprint arXiv:2104.11186*, 2021.

Yu, H. and Bertsekas, D. P. On boundedness of q-learning it-erates for stochastic shortest path problems. *Mathematics of Operations Research*, 38(2):209–227, 2013.
A. Basic Property of the Optimal Policy

Lemma A.1 (Bertsekas & Tsitsiklis, 1991; Yu & Bertsekas, 2013). Suppose that there exists a proper policy with respect to the goal state $g$ and that for every improper policy $\pi'$ there exists at least one state $s \in S$ such that $V^*_{g}(s) = +\infty$. Then the optimal policy $\pi^*$ is stationary, deterministic, and proper. Moreover, $V^*_{g} = L V^*_{g}$ is the unique solution of the optimality equations $V^*_{g} = L V^*_{g}$ and $V^*_{g}(s) < +\infty$ for any $s \in S$, where for any vector $V \in \mathbb{R}^S$ the optimal Bellman operator $L$ is defined as

$$L V(s) := \min_{a \in \mathcal{A}} \left\{ c(s, a) + P_{s,a} V \right\}.$$ 

Furthermore, the optimal $Q$-value, denoted by $Q^*_{g} = Q^*_{g}$, is related to the optimal value function as follows

$$Q^*_{g}(s, a) = c(s, a) + P_{s,a} V^*_{g}, \quad V^*_{g}(s) = \min_{a \in \mathcal{A}} Q^*_{g}(s, a), \quad \forall(s, a) \in S \times \mathcal{A}.$$ 

B. High-Probability Event

First we define the high-probability event $\mathcal{E}$ to do concentration on all the samples collected in Alg. 1 and Alg. 2. We note that in Alg. 2, after running DisCo algorithm, the set of known states $\mathcal{K}$ is fixed, and our algorithm focuses on the new MDP $\mathcal{M}^{\dagger} = (\mathcal{K}^{\dagger}, \mathcal{A}, P^{\dagger}, c^{\dagger}, s_0)$, where $\mathcal{M}^{\dagger}$ is defined in Sect. 3.2.

We recall that for any two vectors $X, Y \in \mathbb{R}^{K'}$ ($K' = |\mathcal{K}|$), we write their inner product as $X Y := \sum_{s \in \mathcal{K}} X(s) Y(s)$, and we denote $\|X\| := \max_{s \in \mathcal{K}^{\dagger}} |X(s)|$. If $X$ is a probability distribution on $\mathcal{K}^{\dagger}$, we denote $\mathbb{V}(X, Y) := \sum_{s \in \mathcal{K}^{\dagger}} X(s) Y(s)^2 - (\sum_{s \in \mathcal{K}^{\dagger}} X(s) Y(s))^2$, i.e. the variance of random variable $Y$ over distribution $X$. And we use $P^{\dagger}_{s,a}$ and $P^{\dagger}_{s,a,s'}$ to denote $P^{\dagger}(|s, a)$ and $P^{\dagger}(|s'| s, a)$, respectively.

Here for any $g \in \mathcal{K}$, we denote the vector $V^*_{g} \in \mathbb{R}^{K'}$ as the value function of the optimal policy on MDP $\mathcal{M}^{\dagger}$ with respect to goal $g$, and we denote $V^*_{g}(s)$ as the expected cost of the optimal policy to reach state $g$ from $s$ on MDP $\mathcal{M}^{\dagger}$. Also, we define $B_{s} := \max_{(s,g) \in \mathcal{K}^{\dagger} \times \mathcal{K}} V^*_{g}(s)$, and we denote $Q^*_{g} \in \mathbb{R}^{K' \times \mathcal{A}}$ as the Q-function corresponding to $V^*_{g}$.

In Alg. 2, we set $B = 10L$. Thus when $\mathcal{K} \subseteq S_{2L}^T$, we have $\forall(s, g) \in \mathcal{K}^{\dagger} \times \mathcal{K}, V^*_{g}(s) \leq 2L + 1 \leq B$, and $B_{s} \leq B$.

Then we define the high-probability event $\mathcal{E}$. We note that our definition of $\mathcal{E}$ is similar with Definition 12 in Sect. D.1, (Tarbouriech et al., 2021).

Definition B.1 (High-probability event $\mathcal{E}$). We define the event $\mathcal{E} := \mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3$, where

$$\mathcal{E}_1 := \left\{ (s, a) \in \mathcal{K}^{\dagger} \times \mathcal{A}, \forall n(s, a) \geq 1 : |\overline{c}(s, a) - c^\dagger(s, a)| \leq 2 \sqrt{\frac{2\overline{c}(s, a)}{n(s, a)^{3/2}}} + \frac{28\ell_{s,a}}{3n(s, a)} \right\},$$

$$\mathcal{E}_2 := \left\{ (s, a, s') \in \mathcal{K}^{\dagger} \times \mathcal{A} \times \mathcal{K}^{\dagger}, \forall n(s, a) \geq 1 : |P^\dagger_{s,a,s'} - \overline{P}_{s,a,s'}| \leq 2 \sqrt{\frac{2P^\dagger_{s,a,s'}}{n(s, a)^{3/2}}} + \frac{\ell_{s,a}}{n(s, a)} \right\},$$

$$\mathcal{E}_3 := \left\{ (s, a, g) \in \mathcal{K}^{\dagger} \times \mathcal{A} \times \mathcal{K}, \forall n(s, a) \geq 1 : |(\overline{P}_{s,a} - P^\dagger_{s,a}) V^*_{g}(s)| \leq 2 \sqrt{\frac{\overline{V}(P_{s,a}, V^*_{g})(s,a)}{n(s, a)}} + \frac{14\overline{B}_{s,a}}{3n(s, a)} \right\},$$

where $\ell_{s,a} := 4 \ln \left( \frac{12K'\overline{A}(n(s,a))}{3} \right)$.

Lemma B.2. It holds that $\mathbb{P}(\mathcal{E}) \geq 1 - \delta$.

Proof. The proof is the same with Lemma 13 in Sect. D.1, (Tarbouriech et al., 2021).

The event $\mathcal{E}_1$ holds with probability $1 - \delta/3$ by Lem. 27 in Sect. F, (Tarbouriech et al., 2021) and by union bound over all $(s, a) \in \mathcal{K}^{\dagger} \times \mathcal{A}$. The event $\mathcal{E}_2$ holds with probability $1 - \delta/3$ by Lem. 26 and Lem. 33 in Sect. F, (Tarbouriech et al., 2021) and by union bound over all $(s, a, s') \in \mathcal{K}^{\dagger} \times \mathcal{A} \times \mathcal{K}^{\dagger}$. The event $\mathcal{E}_3$ holds with probability $1 - \delta/3$ by Lem. 27 in Sect. F, (Tarbouriech et al., 2021) and by union bound over all $(s, a, g) \in \mathcal{K}^{\dagger} \times \mathcal{A} \times \mathcal{K}$. The lemma above is a direct consequence of concentration inequalities. We note that we do not use the samples collected in DisCo algorithm, and the set $\mathcal{K}$ is fixed after running DisCo algorithm. Hence for any $g \in \mathcal{K}$, the vector $V^*_{g}$ depends only on...
the set $\mathcal{K}$, and $V^*_g$ is fixed after running DisCo algorithm and does not depend on the samples collected in Alg. 1 and Alg. 2. Thus $\tilde{P}_{s,a}$ and $V^*_{\tilde{g}}$ are independent for any $(s, a, g) \in \mathcal{K} \times \mathcal{A} \times \mathcal{K}$ and $n(s, a) \geq 1$.

**Lemma B.3** (Cohen et al., 2020, Lem. B.5). Let $\pi$ be a proper policy such that for some $d > 0$, the expected cost $V^*_g(s) \leq d$ for every non-goal state $s \neq g$. Then the probability that the cumulative cost of $\pi$ to reach the goal state from any state $s$ is more than $m$, is at most $2e^{-m/4d}$ for all $m \geq 0$.

**Lemma B.4.** Let $\tau$ be a random variable on $[0, +\infty)$ such that $\Pr(\tau > m) \leq 2e^{-m/4d}$ for any $m \geq m$, where $d > 0$ is a constant. We define the random variable $\hat{\tau} = \frac{1}{n} \sum_{k=1}^{n} \hat{\tau}_k$, where each $\hat{\tau}_k$ is i.i.d. and has the same distribution with $\tau$. Then for any $\epsilon > 0$, we have $\Pr(\hat{\tau} > \hat{\tau} + \epsilon) \leq \exp(-\frac{ne^2 \epsilon^2}{20 \ln^2(64/\epsilon)}).

**Proof.** We set the constant $\Gamma = \lfloor 8d \ln(64/\epsilon) \rfloor$. Then we define the random variables $\tau_I = \min(\tau, \Gamma)$, $\hat{\tau}_k = \min(\hat{\tau}_k, \Gamma)$, and $\hat{\tau} = \frac{1}{n} \sum_{k=1}^{n} \hat{\tau}_k$.

As each $\hat{\tau}_k$ is a random variable on $[0, \Gamma]$, by Hoeffding’s inequality, we have

$$\Pr(\hat{\tau}_k > \hat{\tau}_k + \frac{1}{2} \epsilon d) \leq \exp(-\frac{ne^2 d^2}{20 \ln^2(64/\epsilon)}).$$

Moreover, we have

$$E(\tau) \leq E(\tau_I) + \sum_{i=1}^{\infty} i \cdot \Pr(\Gamma + i - 1 < \tau \leq \Gamma + i) = E(\tau_I) + \sum_{m=\Gamma}^{\infty} \Pr(\tau > m) \leq E(\tau_I) + \sum_{m=\Gamma}^{\infty} \exp(-m/4d) \leq E(\tau_I) + \frac{1}{2} \epsilon d.$$  

Therefore, we obtain

$$\Pr(\hat{\tau} > \hat{\tau} + \epsilon d) \leq \Pr(\hat{\tau} > \hat{\tau} + \epsilon d) \leq \Pr(E(\tau_I) > \hat{\tau} + \frac{1}{2} \epsilon d) \leq \exp(-\frac{ne^2 \epsilon^2}{20 \ln^2(64/\epsilon)}).$$

\[\square\]

### C. Analysis of a VISGO Procedure

In this section, we fix the known states $\mathcal{K}$ and the goal state $g$ and we analysis an execution of the VISGO procedure in Alg. 3. We use the value iteration of the form $V^{(i+1)} = \hat{L} V^{(i)}$ to estimate the value funtion of the optimal policy. Here, we define the operator $\hat{L}$ in the following way. For any $U \in \mathbb{R}^{K'}$ such that $U \geq 0$, $U(g) = 0$, and $\|U\|_\infty \leq B$, we first define

$$\hat{L} U(s, a) := \hat{c}(s, a) + \tilde{P}_{s,a} U - b(U, s, a),$$

for any $s \in \mathcal{K} \setminus \{g\}$ and $a \in \mathcal{A}$, where we define

$$b(U, s, a) := \max \left\{ c_1 \sqrt{\frac{\mathbb{V}(\tilde{P}_{s,a} U)_s}{n(s,a)}}, \frac{c_2 B_{s,a}}{n(s,a)} \right\} + c_3 \sqrt{\frac{\hat{c}(s,a)_s}{n(s,a)}},$$

for any $s \in \mathcal{K} \setminus \{g\}$ and $a \in \mathcal{A}$. Here we recall that $B = 10L$, $\iota_{s,a} = 4 \ln(\frac{12K^2 \text{An}(s,a)}{a})$ (cf. Def. B.1), and $\mathbb{V}(X, Y) := \sum_{s \in \mathcal{K}} X(s) Y^2(s) - (\sum_{s \in \mathcal{K}} X(s) Y(s))^2$ is the variance of random variable $Y$ over distribution $X$. And we define the transition probability $\tilde{P}_{s,a,s'} = \frac{n(s,a)}{n(s,a)+1} \tilde{P}_{s,a,s'} + \frac{1}{n(s,a)+1}$ that slightly increases the probability to reach the goal $g$ at each state-action pair.

Then, we set $\hat{L} U(s) := \min_{a \in \mathcal{A}} \hat{L} U(s, a)$ for $s \in \mathcal{K}$ and $s \neq g$, and we set $\hat{L} U(x) := c_{\text{RESET}} + U(s_0)$. Finally, we set $\hat{L} U(g) := 0$.

We note that in Alg. 2, before we executed VISGO procedure, we have collected $\phi = \tilde{O}(L^2 |\mathcal{K}| / \iota_{\text{min}}^2)$ samples for each state-action pair $(s, a) \in \mathcal{K} \times \mathcal{A}$ in Alg. 1. Thus we have $n(s, a) \geq \phi$ for each $(s, a) \in \mathcal{K} \times \mathcal{A}$. We stress that the lemmas of this section are based on the conditions that $n(s, a) \geq \phi$ for all $(s, a) \in \mathcal{K} \times \mathcal{A}$.
Algorithm 3 Subroutine VISGO

1: **Input:** Goal state \( g \) and \( c v t \).
2: **Global variables:** \( B, L, N(), n(), \tilde{P}, \theta(), \tilde{c}() \).
3: **Specify:** Constants \( c_1 = 6, c_2 = 72, c_3 = 2\sqrt{2} \).
4: For all \( (s,a,s') \in K \times A \times K' \), set

\[
\tilde{P}_{s,a,s'} \leftarrow \frac{n(s,a)}{n(s,a) + 1}\tilde{P}_{s,a,s'} + \frac{E[s' = g]}{n(s,a) + 1}.
\]

5: For all \( (s,a) \in K \times A \), set \( t_{s,a} \leftarrow 4 \ln \left( \frac{12K'An(s,a)}{\varepsilon} \right) \).
6: Set \( i \leftarrow 0, V^{(0)} \leftarrow 0, V^{(-1)} \leftarrow +\infty \).
7: **while** \( \|V^{(i)} - V^{(i-1)}\|_\infty > cvt \) **do**
8: For all \( s \in K \setminus \{g\} \) and \( a \in A \), set

\[
b^{(i+1)}(s,a) \leftarrow \max \left\{ c_1 \left( \frac{\sqrt{\tilde{V}(\tilde{P}_{s,a},V^{(i)}(s,a))}}{n(s,a)} + c_2 \frac{B_{t_{s,a}}}{n(s,a)} \right)^{1/2}, c_3 \frac{\sqrt{\tilde{c}(s,a)t_{s,a}}}{n(s,a)} \right\},
\]

\[
Q^{(i+1)}(s,a) \leftarrow \tilde{c}(s,a) + \tilde{P}_{s,a}V^{(i)} - b^{(i+1)}(s,a),
\]

\[
V^{(i+1)}(s) \leftarrow \min_{a \in A} Q^{(i+1)}(s,a).
\]

9: Set \( V^{(i+1)}(x) \leftarrow \text{reset} + V^{(i)}(s_0) \).
10: Set \( V^{(i+1)}(g) \leftarrow 0, i \leftarrow i + 1 \).
11: **end while**
12: **return** \( Q^{(i)}, V^{(i)} \).

We note that the variance \( \forall(\tilde{P}_{s,a}, U) \leq O(L^2) \) when \( \|U\|_\infty \leq B = 10L \), and \( t_{s,a} \) contains only logarithmic terms, thus \( b(U, s, a) = O(L/\sqrt{n(s,a)}) \). As \( n(s,a) \geq \phi = O(L^2Kc_{\text{min}}^{-2}) \), we have \( b(U, s, a) \leq c_{\text{min}}/18 \leq c_{\text{min}} \) for any \((s,a) \in K \times A \). Therefore, if \( U(g) = 0 \), \( \|U\|_\infty \leq B \), and \( U \geq 0 \) component-wise, when \( n(s,a) \geq \phi \) for any \((s,a) \in K \times A \), we have \( L^\phi(s,a) \geq 0 \) for any \((s,a) \in K \times A \). Hence the output of \textsc{VISGO} \((Q, V)\) satisfies \( Q \geq 0 \) and \( V \geq 0 \) component-wise.

For convenience, we define \( b(U, x, a) := 0 \) and \( b(U, g, a) := 0 \) for any \( a \in A \).

**Lemma C.1** ([Tarbourieh et al., 2021], Lemma 12). For any non-negative vector \( U \in \mathbb{R}^{K'} \) such that \( U(g) = 0 \), for any \((s,a) \in K \times A \), it holds that

\[
\tilde{P}_{s,a}U \leq \tilde{P}_{s,a}U \leq \tilde{P}_{s,a}U + \frac{\|U\|_\infty}{n(s,a) + 1}.
\]

The proof of the following Lem. C.2 is similar with Lem. 16 in (Tarbourieh et al., 2021), but here we have two distributions \( \tilde{p} \) and \( p \). We give the whole proof for completeness.

**Lemma C.2.** Let \( \Upsilon := \{v \in \mathbb{R}^{K'}: v \geq 0, v(g) = 0, \|v\|_\infty \leq B \} \). Let \( f : \Delta K' \times \Delta K' \times \Upsilon \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R} \) with

\[
f(\tilde{p}, p, v, n, B, \iota) := \tilde{p}v - \max \left\{ c_1 \sqrt{\frac{\sqrt{v(p,v)\iota}}{n}}, c_2 \frac{B_{\iota}}{n} \right\},
\]

with constants \( c_1 = 6 \) and \( c_2 \geq 2c_1^2 \). Then \( f \) satisfies, for all \( v \in \Upsilon, n, \iota > 0, \tilde{p}, p \in \Delta K' \) s.t. \( \tilde{p}(s) - \frac{1}{2}p(s) \geq 0 \) for all \( s \neq g \),

1. \( f(\tilde{p}, p, v, n, B, \iota) \) is non-decreasing in \( v(s) \), i.e.

\[
\forall(v, v') \in \Upsilon^2, v \leq v' \implies f(\tilde{p}, p, v, n, B, \iota) \leq f(\tilde{p}, p, v', n, B, \iota);
\]

2. \( f(\tilde{p}, p, v, n, B, \iota) \leq \tilde{p}v - \frac{c_1}{2} \sqrt{\frac{\sqrt{v(p,v)\iota}}{n}} - \frac{c_2}{2} \frac{B_{\iota}}{n} \leq \tilde{p}v - 2\sqrt{\frac{v(p,v)\iota}{n}} - 14 \frac{B_{\iota}}{n} \);

3. If \( \tilde{p}(g) > 0 \), then \( f(\tilde{p}, p, v, n, B, \iota) \) is \( \rho_{\tilde{p}} \)-contractive in \( v(s) \), with \( \rho_{\tilde{p}} := 1 - p(g) < 1 \), i.e.

\[
\forall(v, v') \in \Upsilon^2, |f(\tilde{p}, p, v, n, B, \iota) - f(\tilde{p}, p, v', n, B, \iota)| \leq \rho_{\tilde{p}} \|v - v'\|_\infty.
\]

**Proof.** We use the idea in (Tarbourieh et al., 2021), Lemma 14 to finish the proof.

The second claim holds by \( \max \{x, y\} \geq (x + y)/2, \forall x, y \), by the choices of \( c_1, c_2 \) and because both \( \sqrt{\frac{\sqrt{v(p,v)\iota}}{n}} \) and \( \frac{B_{\iota}}{n} \) are non-negative. To verify the first and third claims, we fix all other variables but \( v(s) \) and view \( f \) as a function in \( v(s) \).
Because the derivative of \( f \) in \( v(s) \) does not exist only when \( c_1 \sqrt{\frac{\mathbb{V}(p,v)}{n}} = c_2 \frac{B_1}{n} \), where the condition has at most two solutions, it suffices to prove \( \frac{\partial f}{\partial v(s)} \geq 0 \) when \( c_1 \sqrt{\frac{\mathbb{V}(p,v)}{n}} \neq c_2 \frac{B_1}{n} \). Direct computation gives that for any \( s \in K^1 \) and \( s \neq g \),

\[
\frac{\partial f}{\partial v(s)} = \bar{p}(s) - c_1 \mathbb{I} \left[ c_1 \sqrt{\frac{\mathbb{V}(p,v)}{n}} \geq c_2 \frac{B_1}{n} \right] \frac{p(s)(v(s) - pv)}{\sqrt{n \mathbb{V}(p,v)}}
\]

\[
\geq \min \left\{ \bar{p}(s), \bar{p}(s) - \frac{c_1^2}{c_2 B} p(s)(v(s) - pv) \right\}
\]

\[
\geq \min \left\{ \bar{p}(s), \bar{p}(s) - \frac{c_1^2}{c_2} p(s) \right\}
\]

\[
\geq 0.
\]

Here (i) is by \( v(s) - pv \leq v(s) \leq B \). In addition, we have

\[
\sum_{s \neq g} \left| \frac{\partial f}{\partial v(s)} \right| = \sum_{s \neq g} \left( \bar{p}(s) - c_1 \mathbb{I} \left[ c_1 \sqrt{\frac{\mathbb{V}(p,v)}{n}} \geq c_2 \frac{B_1}{n} \right] \frac{p(s)(v(s) - pv)}{\sqrt{n \mathbb{V}(p,v)}} \right)
\]

\[
= 1 - \bar{p}(g) - c_1 \mathbb{I} \left[ c_1 \sqrt{\frac{\mathbb{V}(p,v)}{n}} \geq c_2 \frac{B_1}{n} \right] \sqrt{\frac{\mathbb{V}(p,v)}{n}} \left[ pv - (1 - p(g)) \cdot pv \right]
\]

\[
\leq 1 - \bar{p}(g).
\]

Therefore, we obtain that \( f \) is \( \rho \)-contractive. \( \square \)

We note that by definition of \( \bar{P}_{s,a} \), we have \( \bar{P}_{s,a,s'} - \frac{1}{2} \bar{P}_{s,a,s'} \geq 0 \) for all \( (s,a,s') \in K \times A \times K^1 \).

The following two lemmas follow the same proof with Lem.18, Lem.19 in (Tarbouriech et al., 2021), respectively.

**Lemma C.3.** The sequence \((\mathbb{V}^{(i)})_{i \geq 0}\) is non-decreasing.

**Lemma C.4.** \( \tilde{L} \) is a \( \rho \)-contractive operator with modulus \( \rho := 1 - \nu < 1 \), where \( \nu = \min_{(s,a) \in K \times A} \bar{P}_{s,a,g} \), i.e. for any two vectors \( U_1, U_2 \in \mathcal{Y} \) (where \( \mathcal{Y} \) is defined in Lemma C.2), \( \| \tilde{L} U_1 - \tilde{L} U_2 \| \leq \rho \| U_1 - U_2 \| \). Hence, the VISGO procedure will terminate after at most \( \left\lceil \frac{\log(1/\epsilon_{vi})}{\log(1/\rho)} \right\rceil \) iterations.

**C.1. The Bounded Error Property of VISGO**

Now we focus on Alg. 2. We give the following lemma of the bounded error property (Lem. C.5), which indicates that the value function of the policy \( \pi_\gamma \) is close to our estimation. The proof of Lem. C.5 uses the techniques of Lem. 2 in (Tarbouriech et al., 2020). Our Lem. C.5 focuses on a more general operator \( \tilde{L} \). In our \( \tilde{L} \), we involve the bonus function \( b(U,s,a) \), which is not contained in (Tarbouriech et al., 2020). And we note that in our proof of the following Lem. C.5, we use the condition that \( n(s,a) \geq 0 = \Omega(L^2 K c_{\min}^{-2}) \) for each \( (s,a) \in K \times A \). Also, we have \( \epsilon_{vi} \leq c_{\min}/18 \) because we set the initial trigger index \( j = 5 + \log_2 c_{\min}^{-1} \) and \( \epsilon_{vi} = 2^{-j} \).

We note that by optimism property (Lem. C.6), when \( K \subseteq S_{2L}^1 \), we have \( V(s) \leq 2L + 1 \) for all \( s \in K^1 \). Hence the following bounded error property (Lem. C.5) implies that in any round, the expected cost of the greedy policy \( \hat{\pi} \) on model \( P^1 \) is no more than \( 2(2L + 1) = O(L) \).

**Lemma C.5 (Bounded Error Property).** In Alg. 2, under the event \( \mathcal{E} \), for any output \( (Q,V) \) of the VISGO procedure in any round, let \( \pi \) be the greedy policy with respect to \( Q \). Then \( \pi \) is proper on the model \( P^1_{s,a,s'} \) and for all \( s \in K^1 \), we have \( V^\pi_g(s) \leq 2V(s) \), where \( g \) is the goal state in that round.

**Proof.** We define \( \tilde{V}^\pi_g(s) \) as the value function of \( \pi \) with goal state \( g \) on the model \( \bar{P}_{s,a,s'} \). We will first prove that \( \tilde{V}^\pi_g(s) \leq \frac{1}{4} V(s) \), and then prove that \( V^\pi_g(s) \leq \frac{1}{4} \tilde{V}^\pi_g(s) \) using the simulation lemma on the two models \( \bar{P}_{s,a,s'} \) and \( P^1_{s,a,s'} \).

Combining them together yields \( V^\pi_g(s) \leq 2V(s) \).

First we focus on model \( \bar{P}_{s,a,s'} \). We recall that for any \( s \in K \) and \( s \neq g \),

\[
\tilde{U}(s) := \min_{a \in A} \left\{ \tilde{c}(s,a) - b(u,s,a) + \bar{P}_{s,a} u \right\}
\]
where \( b(u, s, a) \) is defined in Eq. 4, i.e., for any \( s \in \mathcal{K} \setminus \{g\} \) and \( a \in \mathcal{A} \),

\[
b(u, s, a) = \max \left\{ c_1 \sqrt{V(\tilde{\pi}_{s,a}, u) n(s,a) + c_2 B l_{s,a} n(s,a)} + c_3 \sqrt{\frac{\ell_{s,a}}{n(s,a)}} \right\},
\]

and we define \( b(u, x, a) = 0 \) and \( b(u, g, a) = 0 \).

We observe that when \( \|u\|_{\infty} \leq B = 10L \), the variance \( V(\tilde{\pi}_{s,a}, u) \leq B^2 \), and \( \ell_{s,a} \) contains only logarithmic terms. Thus we have \( b(u, s, a) = \tilde{O}(L/\sqrt{n(s,a)}) \).

As we set \( \phi = \Theta(\frac{L^2|K|}{c_{\min} \ln(\frac{|K|}{A})}) \),

and \( n(s, a) \geq \phi \), we can obtain \( b(u, s, a) \leq c_{\min}/18 \) when \( \|u\|_{\infty} \leq B \).

In addition, under the event \( \mathcal{E}_1 \), we have \( |\tilde{c}(s, a) - c(s, a)| \leq O(1/\sqrt{n(s,a)}) \).

Thus when \( n(s, a) \geq \phi \), we have \( |\tilde{c}(s, a) - c(s, a)| \leq c_{\min}/18 \) for all \( (s, a) \in \mathcal{K}^{1} \times \mathcal{A} \).

We denote \( l \) as the final iteration index of VISGO, and \( V = V^{(l)} \). In VISGO, we have \( V^{(i)} = \tilde{E}V^{(i-1)} \) for all \( i = 1, 2, \ldots, l \).

As \( V^{(l-1)} \leq V^{(l)} \) component-wise, we have for any \( s \in \mathcal{K}^{1} \), \( V(s) \leq V^{(l-1)}(s_0) + 1 \leq 2L + 1 \). Thus, \( \|V\|_{\infty} \leq 2L + 1 \leq B \).

We set \( \gamma = c_{\min}/6 \). As \( c_{\min}/18 \), we have \( b(u, s, a) \) and \( |\tilde{c}(s, a) - c(s, a)| + c_{\min}/6 \leq \gamma \) when \( \|u\|_{\infty} \leq B \).

Given the policy \( \pi \) restricted on \( \mathcal{K} \), we introduce the following operators on \( \mathbb{R}^{|K|} \):

\[
\mathcal{L}\pi u(s) = \tilde{c}(s, \pi(s)) - b(u, s, \pi(s)) + \tilde{P}_{s,\pi(s)} u,
\]

\[
\mathcal{T}\pi u(s) := c(s, \pi(s)) - \gamma + \tilde{P}_{s,\pi(s)} u.
\]

We can write component-wise

\[
\mathcal{T}\pi \gamma V \leq \mathcal{L}\pi V - \epsilon_{\gamma} (a) \leq \tilde{E}V - \epsilon_{\gamma} \leq V,
\]

where \( (a) \) uses that \( \pi \) is the greedy policy with respect to \( V \). To prove \( (b) \), we recall that \( V = V^{(l)} = \tilde{E}V^{(l-1)} \). By contraction property of \( \tilde{E} \) (Lem. C.4), we have \( \|\tilde{E}V - V\|_{\infty} \leq \|V^{(l)} - V^{(l-1)}\|_{\infty} \leq \epsilon_{\gamma} \), thus \( (b) \) is proved. By monotonicity of the Bellman operator \( \mathcal{T}\pi \gamma \), we have for all \( m > 0 \),

\[
(T_{\pi}^{m})^{1} V \leq (T_{\pi}^{m})^{0} V \leq \cdots \leq V.
\]

We observe that the vector \( (T_{\pi}^{m})^{1} V \) does not increase element-wise when \( m \) increases, and \( (T_{\pi}^{m})^{0} V \geq 0 \) element-wise because \( V \geq 0 \) element-wise. Hence when \( m \to \infty \), it will converge to some vector \( W_{\gamma} \), where \( W_{\gamma} \) is the value function of policy \( \pi \) in the model \( \tilde{P} \) with \( \gamma \) subtracted to all the costs, and we have \( W_{\gamma} \leq V \) component-wise. We define the random variable \( \hat{\tau}_g(s) \) as the number of steps it takes to reach \( g \) starting from \( s \) on model \( \tilde{P} \) when executing policy \( \pi \). Thus

\[
W_{\gamma} s := \mathbb{E}_{\tilde{P}} \left[ \sum_{t=1}^{\hat{\tau}_g(s)} c(s_t, \pi(s_t)) - \gamma \mid s_1 = s \right] = \tilde{V}_{\gamma} s - \gamma \mathbb{E}_{\tilde{P}} [\hat{\tau}_g(s)].
\]

Moreover, we have \( c_{\min} \mathbb{E}[\hat{\tau}_g(s)] \leq \tilde{V}_{\gamma} s \). Therefore, we get

\[
\tilde{V}_{\gamma} s \leq \frac{W_{\gamma} s}{1 - \gamma/c_{\min}} \leq \frac{V(s)}{1 - \gamma/c_{\min}} \leq \frac{4}{3} V(s).
\]

Under the event \( \mathcal{E}_2 \), we have \( |P_{s,a,s'}^{\dagger} - \tilde{P}_{s,a,s'}| = \tilde{O}(\sqrt{P_{s,a,s'} n(s,a) + n(s,a)^{-1}}) \). As \( n(s,a) \geq \phi = \tilde{O}(L^2|K|) \), and \( B = 10L \), we can obtain \( \forall (s, a, s') \in \mathcal{K}^{1} \times \mathcal{A} \times \mathcal{K}^{1} \),

\[
|P_{s,a,s'}^{\dagger} - \tilde{P}_{s,a,s'}| \leq \frac{c_{\min}}{24B} \sqrt{\frac{P_{s,a,s'}^{1}}{|K|}} + \frac{c_{\min}}{24B|K|^1}.
\]
By the Cauchy–Schwarz’s inequality, \( \sum_{s' \in \mathcal{K}} \left| P_{s,a,s'}^i - \tilde{P}_{s,a,s'} \right| \leq c_{\min} \frac{12B}{n}, \quad \forall (s,a) \in \mathcal{K} \times A. \)

Also, as \( |\tilde{P}_{s,a,s'} - \tilde{P}_{s,a,s'}| \leq 1/n, (s,a) \leq c_{\min}/(12B|\mathcal{K}|), \) and \( \bar{V}_g^\pi(s) \leq \frac{2}{3}(2L + 1) \leq B \) for all \( s \in \mathcal{K} \), we can obtain that

\[
\sum_{s' \in \mathcal{K}} \left| P_{s,a,s'}^i - \tilde{P}_{s,a,s'} \right| \leq c_{\min} \frac{12B}{n}, \quad \forall (s,a) \in \mathcal{K} \times A.
\]

Thus by simulation lemma for SSP (Lemma 3 in (Tarbouriech et al., 2020)), \( \pi \) is proper on true model \( P_{s,a,s'}^i \), and for all \( s \in \mathcal{K} \), \( V_g^\pi(s) \leq (1 + \frac{1}{2})\bar{V}_g^\pi(s) = \frac{2}{3}V_g^\pi(s) \leq 2V(s) \). The proof is completed.

C.2. Optimistic Property of VISGO

Now we will give the optimistic property. We still focus on Alg. 2, and we will prove that the output of the VISGO procedure \((Q, V)\) is optimistic. And we recall that we denote \( V_g^* \) as the value function of the optimal policy on MDP \( \mathcal{M} \) to reach \( g \), and \( Q_g^* \) as the Q-function corresponding to \( V_g^* \).

Lemma C.6 (Optimistic Property). In Alg. 2, under the event \( \mathcal{E} \), for any output \((Q, V)\) of the VISGO procedure, it holds that

\[
Q(s, a) \leq Q_g^*(s, a), \quad \forall s \in \mathcal{K} \setminus \{g\}, a \in A, \\
V(s) \leq V_g^*(s), \quad \forall s \in \mathcal{K},
\]

where \( g \) is the goal state in VISGO procedure.

Proof. We prove by induction that for any inner iteration \( i \) of VISGO, \( Q^{(i)}(s, a) \leq Q_g^*(s, a) \) for any \( (s, a) \in \mathcal{K} \times A \), and \( V^{(i)}(s) \leq V_g^*(s) \) for any \( s \in \mathcal{K} \). By definition we have \( Q^{(0)} = 0 \leq Q_g^* \), and \( V^{(0)} = 0 \leq V_g^* \). Assume that the optimistic property holds for iteration \( i \), then for any \( (s, a) \in \mathcal{K} \times A \) and \( s \neq g \),

\[
Q^{(i+1)}(s, a) = \tilde{c}(s, a) + \tilde{P}_{s,a} V^{(i)} - b^{(i+1)}(s, a),
\]

where

\[
\tilde{c}(s, a) + \tilde{P}_{s,a} V^{(i)} - b^{(i+1)}(s, a) \\
= \tilde{c}(s, a) + \tilde{P}_{s,a} V^{(i)} - \max \left\{ c_1 \sqrt{\frac{\mathcal{V}(\tilde{P}_{s,a} V^{(i)} \iota_{s,a}, n(s,a))}{n(s,a)}}, c_2 \frac{B_{t,s,a}}{n(s,a)} \right\} - c_3 \sqrt{\frac{\tilde{c}(s, a) \iota_{s,a}}{n(s,a)}} \\
\leq c(s, a) + \tilde{P}_{s,a} V^{(i)} - \max \left\{ c_1 \sqrt{\frac{\mathcal{V}(\tilde{P}_{s,a} V^{(i)} \iota_{s,a}, n(s,a))}{n(s,a)}}, c_2 \frac{B_{t,s,a}}{n(s,a)} \right\} + \frac{288_{s,a}}{3n(s,a)} \\
= c(s, a) + f(\tilde{P}_{s,a}, \tilde{P}_{s,a}, V^{(i)}, n(s,a), B_{t,s,a}) + \frac{288_{s,a}}{3n(s,a)} \\
\leq c(s, a) + f(\tilde{P}_{s,a}, \tilde{P}_{s,a}, V_g^*, n(s,a), B_{t,s,a}) + \frac{288_{s,a}}{3n(s,a)} \\
\leq c(s, a) + \tilde{P}_{s,a} V_g^* - 2 \sqrt{\frac{\mathcal{V}(\tilde{P}_{s,a} V_g^* \iota_{s,a}, n(s,a))}{n(s,a)}} - \frac{14B_{t,s,a}}{3n(s,a)} \\
\geq c(s, a) + \tilde{P}_{s,a} V_g^* - 2 \sqrt{\frac{\mathcal{V}(\tilde{P}_{s,a} V_g^* \iota_{s,a}, n(s,a))}{n(s,a)}} - \frac{14B_{t,s,a}}{3n(s,a)} \\
\leq Q_g^*(s, a) \\
Q_g^*(s, a).
\]
where (i) is by definition of $\mathcal{G}_1$ and choice of $c_3$, (ii) uses the first property of Lem. C.2 and the induction hypothesis that $V^{(i)} \leq V_g^*$, (iii) uses the second property of Lem. C.2 and assumption $B \geq \max\{B_s, 1\}$, (iv) uses Lem. C.1, (v) is by definition of $\mathcal{G}_3$. Ultimately, for any $(s, a) \in \mathcal{K} \times \mathcal{A}$ and $s \neq g$, 

$$Q^{(i+1)}(s, a) \leq Q^*_g(s, a).$$

Then for any $s \in \mathcal{K}$ and $s \neq g$, we have $V^{(i+1)}(s) = \min_{a \in \mathcal{A}} Q^{(i+1)}(s, a) \leq \min_{a \in \mathcal{A}} Q^*_g(s, a) = V_g^*(s)$.

In addition, $V^{(i+1)}(g) = 0 = V_g^*(g)$, and we have

$$V^{(i+1)}(x) = c_{\text{reset}} + V^{(i)}(s_0) \leq c_{\text{reset}} + V^*_g(s_0) = V_g^*(x).$$

This completes the proof of this lemma.

\[\square\]

**D. Proof of Thm. 3.1**

Here we give a proof of Thm. 3.1 and we focus on the fixed set $\mathcal{K}^+$ and our constructed MDP $M^+ = (\mathcal{K}^+, \mathcal{A}, P^+, c^+, s_0)$. We denote $K = |\mathcal{K}|$, and $K^+ = |\mathcal{K}^+| = K + 1$.

**Proof idea.** First we prove that Alg. 2 solves AX problem (cf. Lem. D.1), i.e., $\mathcal{K} \supseteq S^*_L$, and $s \in \mathcal{K}, V^*_{\pi}(s_0) \leq V_{\pi,\pi}(s_0) + \varepsilon L$. By running DisCo algorithm with $\varepsilon = 1$, we have $S^*_L \subseteq \mathcal{K} \subseteq S^*_{2L}$. To prove that each output policy $\pi$ is near-optimal, we observe that in the success round with respect to goal $s$, the average cost of executing $\pi_s$ (denoted as $\bar{\pi}$) in $\lambda$ episodes is no more than $V(s_0) + \varepsilon L$. We set $\lambda = O(1/\varepsilon^2)$, by concentration inequalities (cf. Lem. B.4), we can obtain that the expected cost of $\bar{\pi}$ is close to the average cost $\hat{\pi}$, i.e., $V^*_\pi(s_0) \leq \hat{\pi} + \varepsilon L \leq V(s_0) + 2\varepsilon L$. By optimistic property of $\text{VISO}$ (Lem. C.6), we have $V(s_0) \leq V^*_\pi(s_0)$, i.e., our estimation of the value function $V(s_0)$ is no more than the optimal cost. Hence we obtain $V^*_\pi(s_0) \leq V^*_\pi(s_0) + 2\varepsilon L \leq V^*_\pi(s_0) + \varepsilon L$ for all $s \in \mathcal{K}$.

Then we bound the cumulative cost $C_T$. We first bound the total cost in DisCo algorithm and Alg. 1. By Lem. 2.4, with probability over $1 - \delta$, DisCo algorithm with $\varepsilon = 1$ uses no more than $O(LS^2_{2L}A/c^2_{\text{min}})$ samples. In Alg. 1, for each state-action pair $(s, a) \in \mathcal{K} \times \mathcal{A}$, we collected $O(L^2/K/c^2_{\text{min}})$ samples. And to reach each $s \in \mathcal{K}$, we executed the policy $\pi_s$, and the cost to reach $s$ is no larger than $\hat{O}(L)$ with high probability. Thus the total cost in Alg. 1 can be bounded by $O(LS^2_{2L}A/c^2_{\text{min}})$.

Now we will bound the cumulative cost of Alg. 2 after running DisCo algorithm and Alg. 1. It’s straightforward to show that the total cost in each round is bounded by $O(L\lambda) = O(L/\varepsilon^2)$. Hence to bound the cumulative cost $C_T$, we need only to bound the total number of rounds $r$. The number of success rounds is at most $K$. As the trigger condition holds for at most $\log_2(2T)$ times for each state-action pair $(s, a)$, the number of skipped rounds can be bounded by $O^\prime \lambda A \log_2(2T)$ (where $T$ is the total number of samples we collected in Alg. 2). Now we need only to bound the number of failure rounds $r_f$.

To bound $r_f$, we borrow the idea from (Lim & Auer, 2012). We first define the regret of an episode as the total cost in this episode minus our estimation of the optimal cost $V(s_0)$, and define the total regret as the sum of the regret in each episode (cf. Eq. 5). Then we will give both the upper bound and lower bound of the regret, where the upper bound scales as $\bar{O}(\sqrt{T})$ and the lower bound scales as $\Omega(r_f)$. For the upper bound, we extend the techniques in (Tarbouriech et al., 2021) from classical SSP to multi-goal SSP, and we obtain an upper bound that scales as $\bar{O}(L^{-1}\sqrt{KA}r_f + LKAe^{-1} + L^2K^2A)$ (cf. Eq. 6).

For the lower bound, as the regret in each failure round is at least $\lambda e L = \bar{O}(L/e)$, we need only to give a lower bound for the regret of all the success rounds and skipped rounds (which can be negative). We observe that the regret in any round is larger than the total cost of executing policy $\bar{\pi}$ in this round minus the expected cost of $\bar{\pi}$, hence we can use concentration inequalities to bound the regret in all the success rounds and skipped rounds (which scales as $-\bar{O}(LKAe^{-1})$). The lower bound of the total regret scales as $\Omega(L^{-1}e^{-1}r_f - LKAe^{-1})$ (cf. Lem. D.3). Hence we can prove that the number of failure rounds $r_f = \bar{O}(KA + eK^2A)$, and the total number of rounds $r = \bar{O}(S_{2L}A + eS^2_{2L}A)$ (here we use $K \leq S_{2L}$).

As the cost in each round is bounded by $\bar{O}(L/e^2)$, the cumulative cost after running Alg. 1 is bounded by $O(LS^2_{2L}Ae^{-2} + LS^2_{2L}Ae^{-1})$. And the cumulative cost in DisCo algorithm and Alg. 1 is bounded by $O(L^3S^2_{2L}Ae^{-2} + L^2S^2_{2L}Ae^{-1})$, hence we complete the proof of Thm. 3.1.

\[\square\]

Now we give the full proof. We recall that we denote $V^*_g(s)$ as the expected cost of the optimal policy on MDP $M^+$ to reach goal $g$ from state $s$, which equals to $V^*_{\pi,s}(s)$ for all $s \in \mathcal{K}$.

First we prove the correctness of Alg. 2, i.e., with probability over $1 - \delta$, Alg. 2 outputs a set $\mathcal{K} \supseteq S^*_L$ and a set of policies $\{\pi_s\}_{s \in \mathcal{K}}$, such that

$$\forall s \in \mathcal{K}, V^*_\pi(s_0) \leq V^*_{\pi,s}(s_0) + \varepsilon L.$$
The main intuition is that each policy $\pi_s$ has been tested for $\lambda$ times in a success round, and the average cost it takes to reach $s$ from $s_0$ is less than our estimate for optimal cost $V(s_0)$ plus $\epsilon L$. Thus by concentration inequalities, the expected cost of $\pi_s$ is close to optimal.

**Lemma D.1 (ValueE Solves AX Problem).** Let $\{\pi_s\}_{s \in K}$ be the set of policies output by Alg.2. With probability at least $1 - \delta$, $V^*_{\pi_s}(s_0) \leq V^*_s(s_0) + \epsilon L$ for all $s \in K$.

**Proof.** We fix any state $s \in K$. In any given round where the chosen target is $s$, let $\hat{r}_h$ be the total cost in the $k$-th episode of that round. Recall that for the algorithm to output a policy $\pi_s$, its empirical performance after $\lambda$ episodes must satisfy that $\hat{r} \leq V(s_0) + \epsilon L$, where $\hat{r} = \frac{\sum_{k=1}^{K} \hat{r}_k}{\lambda}$ and $V$ is the output of VISGO in that round. By optimism property (Lem. C.6), when $K \subseteq S^*_\pi$, we have $V(s) \leq 2L + 1$ for all $s \in K^1$.

We define the random variable $\tau$ as the total cost it takes to reach the goal state $s$ from the start state $s_0$ when executing policy $\pi_s$, and we have $E(\tau) = V^*_{\pi_s}(s_0)$ by definition. We note that we have collected $\phi = \Omega(L^2K/c_{\min}^2)$ samples for each of the state-action pair $(s, a)$ (cf. Alg. 1). By Lem. C.5, under event $\mathcal{E}$, the policy $\pi_s$ is proper, and we have $E(\tau) \leq 2V(s_0) \leq 4L + 2$. Moreover, we have $d := \|V^*_{\pi_s}\|_\infty \leq 4L + 2$. By Lem B.3, we obtain $Pr(\tau > m) \leq 2exp(m/4d)$ for any $m > 0$. As we set

$$\lambda = \left[\frac{2048}{\epsilon^2} \ln^2(2) \ln(\frac{2|K|}{\delta})\right],$$

by Lem. B.4, we obtain that with probability at least $1 - \delta/(2K')$, we have

$$V^*_{\pi_s}(s_0) = E(\tau) \leq \hat{r} + \epsilon L.$$

We note that $\hat{r} \leq V(s_0) + \epsilon L \leq V^*_s(s_0) + \epsilon L$ by the optimistic property (Lem. C.6). Hence, as we set $\epsilon = \epsilon/3$ in initial, we obtain that with probability at least $1 - \delta/(2K)$,

$$V^*_{\pi_s}(s_0) \leq V^*_{\pi_s}(s_0) + \epsilon L.$$

Finally, as there are at most $K$ states in total, and the event $\mathcal{E}$ holds with probability at least $1 - \delta$, by the union bound and setting $\delta \rightarrow \delta/C$ in initial ($C$ is a large constant), the total success probability is at least $1 - \delta$.

Now we focus on bounding the cumulative cost of Alg.2.

First, we bound the total cost in DisCo algorithm and Alg.1. Disco algorithm has sample complexity $O(L^3S_0^2/\varepsilon^2 \lambda)A/(c_{\min}^2 \varepsilon^2)$, and when $\varepsilon = 1$, the total cost is bounded by $O(L^3S_0^2A/c_{\min}^2)$. In Alg. 1, we collect $\phi = O(L^2K/c_{\min}^2)$ samples for each state-action pair $(s, a) \in K \times A$. To collect each sample $(s, a, s', c)$, we execute a policy $\pi_s$ to reach the state $s$, and the expected cost $V^*_{\pi_s}(s_0) \leq 2L$. By Lem. B.3, we obtain that with probability at least $1 - \delta$, for any state $s$, each time when the policy $\pi_s$ is executed, the total cost to reach $s$ from $s_0$ is no larger than $O(L \log(K/\delta))$. Therefore, the total cost of Alg.1 can be bounded by $O(KA\phi L \log(K/\delta)) = O(L^3S_0^2A/c_{\min}^2)$. We note that we used Lem. B.3 no more than $\phi KA$ times, the total failure probability is no more than $\phi KA \delta$. Substituting $\delta$ by $\delta/(2\phi KA)$ in the proof, we can obtain that the total cost of DisCo and Alg.1 is bounded by $O(L^3S_0^2A/c_{\min}^2)$ with probability 1 - $\delta$.

Then we bound the total cost of Alg.2 after running Alg. 1. The key idea lies in bounding the "regret". We will use the regret to bound the total number of rounds. We first define the regret in the $k$-th episode of the $j$-th round. We denote $H^{j,k}$ as the number of steps it takes in the $k$-th episode of the $j$-th round. The regret in an episode $k$ is defined as

$$H^{j,k} \quad \sum_{h=1}^{H^{j,k}} c^{j,k}_h - V^j(s_0),$$

where $c^{j,k}_h$ is the empirical cost in the $h$-th step in the $k$-th episode in the $j$-th round, and $V^j(s_0)$ is the value of $V(s_0)$ in the $j$-th round. Let $n_j$ be the total number of episodes executed in the $j$-th round, we define the regret in the $j$-th round as follows:

$$\sum_{k=1}^{n_j} (\sum_{h=1}^{H^{j,k}} c^{j,k}_h - V^j(s_0)).$$

Then we will define the total regret of Alg.2. Let $r$ be the total number of rounds, $n_j$ be the total number of episodes executed in the $j$-th round, and $0 \leq n_j \leq \lambda$. Then we know that the total number of episodes in the whole process of Alg.2...
is \( M = \sum_{j=1}^{r} n_j \). For notation convenience, we define \( H^m \) as the number of steps it takes in the \( m \)-th episode of the whole process of Alg. 2, and denote \( c^m_h \) as the empirical cost in the \( h \)-step of episode \( m \). Finally we define the total regret of all the rounds as

\[
R := \sum_{j=1}^{r} \sum_{k=1}^{n_j} ((\sum_{h=1}^{H^{j,k}} c^j_h) - V^j(s_0)) = \sum_{m=1}^{M} ((\sum_{h=1}^{H^m} c^m_h) - V^m(s_0)).
\]  

(5)

We will give both the upper bound and the lower bound of the regret. Here we give the upper bound.

Lemma D.2 (Upper Bound of Regret). Under event \( \mathcal{E} \), the total regret in \( M \) episodes is at most

\[
R = \tilde{O}(L\sqrt{\lambda K M} + LK^2 A).
\]

This upper bound comes from the regret bound of the EB-SSP algorithm (cf. (Tarbouriech et al., 2021)), which solves the classical SSP problem with a single goal state \( g \). To extend their result to multi-goal SSP, instead of only concentrating on \( \tilde{P}_{s,a}V^*_g \) for a single goal \( g \) and one vector \( V^*_g \), in our high probability event \( \mathcal{E}_3 \), we use concentration over \( (\tilde{P}_{s,a} - P^\dagger_{s,a})V^*_g \) for all the goal states \( g \in \mathcal{G} \). Then following similar proof with Thm. 3, (Tarbouriech et al., 2021), we can obtain the regret upper bound in Lem. D.2.

We note that the original form of the regret upper bound in Thm. 3, (Tarbouriech et al., 2021) was \( \tilde{O}(B \sqrt{SAM} + BS^2 A) \), where \( B_s := \max_{s \in \mathcal{S}} V^*_g(s) \) in their work, \( B \) is an upper bound of \( B_s \) which is used in VISGO, and \( M \) is the number of episodes. In our Alg. 2, we work on MDP \( \mathcal{M}^1 \), and all the states in \( \mathcal{G} \) are incrementally 2L-controllable from \( s_0 \). Hence in our settings, \( B_s := \max_{(s,g) \in \mathcal{K}^1 \times \mathcal{G}} V^*_g(s) \leq 2L + 1 \), and the number of states in MDP \( \mathcal{M}^1 \) is \( K' = K + 1 \). And in our Alg. 2, we set \( B = 10L \). Therefore, by setting \( B_s = O(L) \), \( B = O(L) \), \( S = K + 1 \) in their regret bound \( \tilde{O}(B \sqrt{SAM} + BS^2 A) \), we can obtain the regret bound \( \tilde{O}(L\sqrt{\lambda K M} + LK^2 A) \).

We observe that there are at most \( O(KA) \) skipped rounds and \( K \) success rounds. We denote by \( r_f \) the number of failure rounds, and we have the total number of episodes \( M = \tilde{O}((KA + r_f)\lambda) = \tilde{O}((KA + r_f)/\epsilon^2) \). Thus the total regret in \( r \) rounds can be bounded by \( r_f \) sublinearly:

\[
R = \tilde{O}(\frac{L}{\epsilon} \sqrt{\lambda K Ar_f} + \frac{LKA}{\epsilon} + LK^2 A).
\]  

(6)

Then we gives the lower bound of the total regret in terms of the number of failure rounds \( r_f \).

Lemma D.3 (Lower Bound of Regret). With probability \( 1 - \delta \), when \( r = \tilde{O}((KA)^2) \), the total regret in the first \( r \) rounds is at least

\[
R = \Omega(\frac{Lr_f}{\epsilon} - \frac{LKA}{\epsilon}),
\]

where \( r_f \) is the number of failure rounds in the \( r \) rounds.

Proof. By the criterion of our performance check, in any failure round, we have \( \hat{\tau} > V(s_0) + \epsilon L \), and in round \( j \), we have \( \hat{\tau} = \frac{1}{n_j} \sum_{k=1}^{H^{j,k}} (\sum_{h=1}^{H^{j,k}} c^j_h) \) by definition. Hence, in any failure round \( j \), the regret is \( \lambda \hat{\tau} - n_jV^j(s_0) \geq \lambda(\hat{\tau} - V^j(s_0)) \geq \lambda \epsilon L = \tilde{O}(\frac{L}{\epsilon}) \).

Then we focus on skipped rounds and success rounds. We denote \( q^j \) as the goal state in the \( j \)-th round, and \( \pi_j \) as the policy \( \hat{\pi} \) in the \( j \)-th round, which is the greedy policy over the \( Q \)-function in the \( j \)-th round. We observe that the regret in any round \( j \) satisfies

\[
\sum_{k=1}^{n_j} ((\sum_{h=1}^{H^{j,k}} c^j_h) - V^j(s_0)) \geq -L + \sum_{k=1}^{n_j-1} ((\sum_{h=1}^{H^{j,k}} c^j_h) - V^{q^j}(s_0)) \geq -L + \sum_{k=1}^{n_j-1} ((\sum_{h=1}^{H^{j,k}} c^j_h) - V^{p^{q^j}}(s_0)),
\]

where we used the optimism property in Lem. C.6. We note that \( \sum_{h=1}^{H^{j,k}} c^j_h \) is the empirical cost of policy \( \pi_j \) in episode \( k \), and we will use the concentration inequality to give a lower bound of the regret in round \( j \). As the last episode in a skipped
round can terminate before reaching the goal, we should take special considerations the last episode of each round. We directly use \(-L\) to lower bound the regret of the last episode in round \(j\). Then we denote \(n = n_j - 1\), and focus on the previous \(n\) episodes in round \(j\).

Now we fix the round index \(j\). We denote the random variable \(\tau\) as the cost to reach \(g^j\) from \(s_0\), and we recall that \(\hat{\tau}_k = \sum_{h=1}^{H^j} c_h^{j,k}\). By Lem. B.4 with \(d = 4L\), with probability at least \(1 - \frac{\delta}{(KA)^{2\gamma}}\), we have

\[
\sum_{k=1}^{n}(\hat{\tau}_k - E(\tau)) \geq -2\Gamma \sqrt{n \ln(\frac{KA}{\delta})} \geq -2\Gamma \sqrt{\lambda \ln(\frac{KA}{\delta})} \geq -\tilde{O}(\frac{L}{\epsilon}),
\]

where \(\Gamma = |8d \ln(64/\epsilon)|\). Thus the regret in any round \(j\) is larger than \(-\tilde{O}(\frac{L}{\epsilon})\). As there are at most \(\tilde{O}(KA)\) skipped rounds and \(K\) success rounds, we obtain that the total regret \(R\) has the lower bound

\[
R = \tilde{O}(\frac{Lrf}{\epsilon} - \frac{LKA}{\epsilon}).
\]

Now we bound the total failure probability. The number of rounds \(r = \tilde{O}((KA)^2)\), in each round the failure probability is at most \(\frac{\delta}{(KA)^{2\gamma}}\), and the events \(E\) fails with probability \(\delta\). By replacing \(\delta\) by \(\delta/C\) throughout the proof (\(C\) is a large constant), we obtain that the total failure probability is at most \(\delta\).

As the lower bound is linear in \(rf\), and the upper bound is sublinear in \(rf\), we can solve it and obtain that \(rf = \tilde{O}(KA + \epsilon K^2A)\), thus the total number of rounds can be bounded by \(\tilde{O}(KA + \epsilon K^2A)\).

To get the cumulative cost bound in Thm. 3.1, we need only to bound the cost in a round. In any round, we observe that except for the last episode, the average cost \(\tilde{r}\) for all the other episodes is no larger than \(V(s_0) + \epsilon L \leq 2L\), thus the cost in these episodes is no larger than \(2L\lambda = \tilde{O}(L/\epsilon^2)\). Also, we know that in the any episode, the expected cost of the policy \(\hat{\tau}\) to reach the goal from \(s_0\) is no larger than \(2L\). Thus by Lem. B.3, in any round, with probability at least \(1 - \frac{\delta}{(KA)^{2\gamma}}\), the cost in the last episode is no larger than \(\tilde{O}(L)\). Hence, the total cost in each round is no larger than \(\tilde{O}(L/\epsilon^2)\). By multiplying it with \(\tilde{O}(KA + \epsilon K^2A)\) and using \(K \leq S_2L\), the cumulative cost in Alg.2 can be bounded by \(\tilde{O}(LS_2L/A/\epsilon^2 + Ls_2L^3A/\epsilon + L^3S_2LA/\epsilon^2)\), where the term \(L^3S_2LA/\epsilon^2\) comes from the subroutine of DisCo algorithm and Alg. 1. Hence we obtain the bound in Thm. 3.1.

Now we count the total failure probability. First, DisCo algorithm fails with probability \(\delta\), the event \(E\) fails with probability \(\delta\), and the lower bound of the total regret \(R\) fails with probability \(\delta\). And in the previous paragraph, to bound the cost in the last episode of each round using Lem. B.3, the failure probability in each episode is at most \(\tilde{O}(\frac{L}{\epsilon})\). We observe that the total number of these failures is no larger than the total number of rounds, and the total number of rounds can be bounded by \(\tilde{O}(KA + \epsilon K^2A)\), where we omit the logarithmic factors. Thus by setting \(\delta \rightarrow \delta/C\) in the proof (\(C\) is a large constant), we can bound the total failure probability by \(\delta\), and the proof of Thm. 3.1 is completed.

Here we briefly discuss the time complexity and space complexity of our algorithm VALAE. The time complexity scales as \(\tilde{O}(TK^3A^2(KA + \epsilon K^2A))\), where \(T\) is the total number of samples collected, and \(\tilde{O}(KA + \epsilon K^2A)\) is the total number of rounds. The bottleneck is on the \textsc{Visgo} procedure, and the time complexity of a \textsc{Visgo} procedure is analyzed in Appendix G, (Turbouriech et al., 2021), which scales as \(\tilde{O}(TK^3A^2)\). The space complexity scales as \(\tilde{O}(T + K^2A) = \tilde{O}(T)\), and the bottleneck is on storing the samples and the empirical model \(\hat{P}\).

E. Analysis of the Lower Bound

Here we discuss the lower bound of the autonomous exploration problem. We recall that our algorithm needs output a set \(K \supseteq S^*_L\) and a set of policies \(\{\pi_s\}_{s \in K}\), and when \(s \in S_L^+,\) the policy \(\pi_s\) satisfies \(V^\pi_s(s_0) \leq (1 + \epsilon)L\). Moreover, we note that in our proof of the lower bound, we allow the algorithm to output Markov policies, i.e. non-stationary and non-deterministic policies, which is defined in the next paragraph.

We recall some basic concepts about the definition of a learning algorithm, and we use the notations in (Domingues et al., 2021). Let \(I^t = (S \times A)^{t-1} \times S\) be the set of all possible histories up to \(t\) steps, i.e., be the set of tuples of the form \((s_1, a_1, s_2, a_2, \ldots, s_t) \in I^t\). Let \(\Delta(A)\) be the set of probability distributions over the action space \(A\), and \(\mathbb{N}^+\) be the set of positive integers. A Markov policy is a function \(\pi : S \times \mathbb{N}^+ \rightarrow \Delta(A)\) such that \(\pi(a \mid s, h)\) denotes the probability of taking action \(a\) in state \(s\) at step \(h\). And we note that the Markov policy \(\pi\) is history-independent.

A history-dependent policy is a family of functions denoted as \(\pi \triangleq (\pi^t)_{t \geq 1}\), where \(\pi^t : I^t \rightarrow \Delta(A)\) describes the probability of taking action \(a \in A\) after observing some history \(i^t \in I^t\).
Given an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, c, s_0)$, a policy $\pi$ interacting with the MDP $\mathcal{M}$ defines a stochastic process denoted by $(S^t, A^t)_{t \geq 1}$, where $(S^t, A^t)$ is the state-action pair at time $t$. The Ionescu-Tulcea theorem ensures the existence of the probability space $(\Omega, \mathcal{F}, P_M)$ such that

$$P_M[S^1 = s] = \mathbb{I}[s = s_0]$$

where $\pi = (\pi^t)_{t \geq 1}$ and for any $t$, $I^t \equiv (S^1, A^1, S^2, A^2, \ldots S^t)$ is the random vector in $I^t$ containing all state-action pairs observed up to step $t$. We denote the $\sigma$-algebra generated by $I^t$ as $\mathcal{F}^t$. And we denote by $P_M^{I^t}$ the measure of $I^t$ under $P_M$ as follows:

$$P_M^{I^t}[s^t] \triangleq P_M[I^t = i^t] = \mathbb{I}(s^1 = s_0) \prod_{t=1}^{T-1} \pi^t(a^t | i^t)p(s^{t+1} | s^t, a^t).$$

Then we denote $E_M$ as the expectation under $P_M$. Note that the dependence of $P_M$ and $E_M$ on the policy $\pi$ is denoted implicitly in the definition of $P_M$. We will denote them explicitly as $P_{\pi, M}$ and $E_{\pi, M}$ respectively when we need to stress $\pi$.

We recall that we define an algorithm for the AX problem as a tuple $(\pi, \tau, \mathcal{K}, \{\pi_s\}_{s \in \mathcal{K}})$, where $\pi$ is a history-dependent policy, $\tau$ is the stopping time chosen by the algorithm, $\mathcal{K}$ and $\{\pi_s\}_{s \in \mathcal{K}}$ are the output of the algorithm. And given the algorithm $\pi$ and the MDP $\mathcal{M}$ for AX problem, we can regard the number $\tau$, the set of states $\mathcal{K}$, and the set of policies $\{\pi_s\}_{s \in \mathcal{K}}$ as random variables on distribution $P_{\pi, \mathcal{M}}$.

Moreover, for any the MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, c, s_0)$, given a Markov policy $\pi$ with goal state $g$, we denote $V^*_M, g(s)$ as the expected cost of policy $\pi$ to reach state $g$ from state $s$ in MDP $\mathcal{M}$. Formally, $V^*_M, g(s) = \mathbb{E}_{\mathcal{M}, \pi} \left[ \sum_{t=1}^{t^*_\pi(s)} c_t(s_t, \pi(s_t)) \right]$, where $t^*_\pi(s) \equiv \inf\{t \geq 0 : s_{t+1} = g\}$. And we denote $V^*_M, g(s)$ as the expected cost of the optimal policy $\pi^*$ to reach the goal state $g$ from the state $s$ on MDP $\mathcal{M}$.

Here we introduce the basic definitions and the technical lemmas used in our proof.

**Definition E.1 (KL divergence).** The Kullback-Leibler divergence between two distributions $P_1$ and $P_2$ on a measurable space $(\Omega, \mathcal{G})$ is defined as

$$KL(P_1, P_2) \triangleq \int \log \left( \frac{dP_1}{dP_2}(\omega) \right) dP_1(\omega),$$

if $P_1 \ll P_2$ and $+\infty$ otherwise. For Bernoulli distributions, we define $\forall (p, q) \in [0, 1]^2$,

$$kl(p, q) \triangleq KL(B(p), B(q)) = p \log \left( \frac{p}{q} \right) + (1 - p) \log \left( \frac{1 - p}{1 - q} \right).$$

**Lemma E.2 (Lemma 5, (Domingues et al., 2021), modified).** Let $\mathcal{M}$ and $\mathcal{M}'$ be two MDPs that are identical except for their transition probabilities, denoted by $p$ and $p'$, respectively. Assume that we have $\forall (s, a), p(\cdot | s, a) \ll p'(\cdot | s, a)$. Then, for any stopping time $\tau$ with respect to $(I^t)_{t \geq 1}$ that satisfies $P_M[\tau < \infty] = 1$,

$$KL(P^{I^t}_{\mathcal{M}}, P^{I^t}_{\mathcal{M}'}) = \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} E_M[N^\tau_{s,a}] KL(p(\cdot | s, a), p'(\cdot | s, a)),$$

where $N^\tau_{s,a} \triangleq \sum_{t=1}^{\tau} \mathbb{I}\{ (S^t, A^t) = (s, a) \}$ and $I^\tau$ is the random vector representing the history of $\tau$ samples.

**Lemma E.3 (Lemma 1, (Garivier et al., 2019)).** Consider a measurable space $(\Omega, \mathcal{F})$ equipped with two distributions $P_1$ and $P_2$. For any $\mathcal{F}$-measurable function $Z : \Omega \rightarrow [0, 1]$, we have

$$KL(P_1, P_2) \geq kl(E_1[Z], E_2[Z]),$$

where $E_1$ and $E_2$ are the expectations under $P_1$ and $P_2$ respectively.

**Lemma E.4.** For any $p, q \in (0, \frac{1}{2}]$, $kl(p, q) \leq 2(p - q)^2/q$.

**Lemma E.5 (Lemma 15, (Domingues et al., 2021)).** For any $p, q \in [0, 1]$, $kl(p, q) \geq -(1 - p) \log(1 - q) - \log(2)$.

Now we construct a family of adversarial MDPs to obtain the lower bound of sample complexity.

**The construction of hard MDPs with general $S_L$.** Now we fix $L, S, A, \varepsilon, c_{\min}$ such that $L > 4$, $S > 8$, $A > 4$, $4 \leq S_L \leq \min\{((A - 1)\lfloor \frac{B}{4} \rfloor, S), 0 < \varepsilon < \frac{1}{4} \}$, and $0 < c_{\min} \leq 1$. 

---

**Near-Optimal Algorithms for Autonomous Exploration and Multi-Goal Stochastic Shortest Path**

---

**Recent Advances in Autonomous Exploration for Stochastic Shortest Path.** The stochastic shortest path (SSP) problem is a fundamental model for autonomous exploration in uncertain environments. It is widely used in various fields, such as robotics, reinforcement learning, and decision-making. The goal of an agent in an SSP is to reach a goal state while minimizing the expected cost or the expected time to reach the goal.

---

**Theoretical Foundations.** The theoretical foundations of SSP have been extensively studied in the past decades. One of the key results is the optimality gap, which measures the difference between the optimal and the sub-optimal policies. The optimality gap has been shown to be exponential in the size of the state-action space.

---

**Algorithmic Developments.** Various algorithms have been proposed to tackle the SSP problem. Among them, the Expected Upper Confidence Bound (EUCB) algorithm has been widely used due to its simplicity and effectiveness.

---

**Future Directions.** While significant progress has been made in the field of SSP, there are still many open challenges that need to be addressed. For example, developing algorithms that can handle large-scale or continuous SSP problems remains an open problem.

---

**Conclusion.** In conclusion, the SSP problem is a rich and challenging area of research that has the potential to impact various fields. Theoretical and algorithmic advancements continue to be made, and the future is promising for further exploration in this area.
We first construct an MDP $\mathcal{M}_0' = (S, A, p_0', c', s_0)$ with $|S| = S_L - 1$ states and $|A| = A$ actions ($\mathcal{M}_0'$ does not contain the goal state $g$ in Fig. 2). As is illustrated in Fig. 2, the construction on $\mathcal{M}_0'$ follows a tree structure. This is inspired from (Domingues et al., 2021). We denote $S'$ as all the leaf states, and for any $s \notin S'$ and $a \in A (a \neq \text{RESET})$, we set $c(s, a) = 1$ with probability 1, and the transition $p_0'(|s, a)$ is deterministic, i.e., taking any action $a$ at a non-leaf node $s$ will transit to one of its son $s' \in S$ with probability 1.

As $S_L \leq (A - 1)\frac{1}{2}L$, there exists a tree structure with depth $d_0 \leq L/2 (d_0 \in \mathbb{N})$, such that the number of leaves $|S'| \geq \frac{S_L}{2}$, and $V_{\mathcal{M}_0',c}(s_0) = d_0$ for all $s \in S'$, i.e., all the leaf nodes can be reached within $d_0$ steps from $s_0$. Hence in MDP $\mathcal{M}_0'$, all the states in $S$ are incrementally $d_0$-controllable. And we denote $d_1 = L - d_0$.

![Figure 2: Illustration of the hard MDP with general $S_L$.](image)

Then we construct the MDP $\mathcal{M}_0 = (S \cup \{g\}, A, p_0, c, s_0)$ based on $\mathcal{M}_0'$ by adding a new state $g$. The state $g$ can only be reached from all the leaf nodes $s \in S'$. And for any leaf state $s \in S'$ and any action $a \in A (a \neq \text{RESET})$, we set $c(s, a) = c_{\text{min}}$, and

$$p_0(g|s, a) = \frac{c_{\text{min}}}{(1 + 6\varepsilon)d_1}, \quad p_0(s|s, a) = 1 - \frac{c_{\text{min}}}{(1 + 6\varepsilon)d_1}.$$

Finally, we set $p_0(g|s, a) = 1$ for any action $a \neq \text{RESET}$.

In this way, we have $V_{\mathcal{M}_0,c}(s_0) = d_0$ for any $s \in S'$, and $V_{\mathcal{M}_0,g}(s_0) = d_0 + (1 + 6\varepsilon)d_1 > (1 + \varepsilon)L$. Hence $g \notin S_L^-$ for MDP $\mathcal{M}_0$. Also, we note that in MDP $\mathcal{M}_0$, with probability 1, the goal state $g$ cannot be reached from $s_0$ within $d_0$ steps.

Now we will construct other adversarial MDPs based on $\mathcal{M}_0$. We choose any $(s^*, a^*) \in S' \times A (a^* \neq \text{RESET})$, and we define the MDP $\mathcal{M}(s^*, a^*) = (S \cup \{g\}, A, p(s^*, a^*), c, s_0)$ by slightly increasing $p_0(g|s^*, a^*)$, i.e., we set

$$p(s^*, a^*)(s|s^*, a^*) = \frac{c_{\text{min}}}{d_1}, \quad p(s^*, a^*)(s^|s^*, a^*) = 1 - \frac{c_{\text{min}}}{d_1}.$$

In this way, we have $V_{\mathcal{M}_0,g}(s_0) = d_0 + d_1 = L$. Hence $g \in S_L^-$ for MDP $\mathcal{M}(s^*, a^*)$.

Finally, we define the family of our adversarial MDPs as $\{\mathcal{M}_0\} \cup \{\mathcal{M}(s, a)\}_{(s, a) \in S' \times A}$. We note that for each MDP $\mathcal{M}(s, a)$, its $|S_L^-| = S_L$, and it satisfies Asmp. 2.1. Also, for the MDP $\mathcal{M}_0$, its $|S_L^-| = S_L - 1$, and it also satisfies Asmp. 2.1. Thus the family is valid for the AX problem.

We note that in MDP $\mathcal{M}(s^*, a^*)$, for any Markov policy $\pi$,

$$V^\pi_{\mathcal{M}(s^*, a^*)}(s_0) = \mathbb{E}_{\pi, \mathcal{M}(s, a)} \left[ \sum_{t=1}^{T_\pi(s)} c_t(s_t, \pi(s_t)) \right].$$
Also, when Thm. 4.2. and the event \(Z\) number of samples that the algorithm takes at the state-action pair \((s_t, a_t)\) we define the event \(\pi\), \(\tau\), \(\sum\)} \(P\) \(g\) \(\{s_t, \pi(s_t)\} | (s_0, a_0) \neq (s^*, a^*)\) \(\sum\)} \(P\) \(g\) \(\{s_t, \pi(s_t)\} | (s_0, a_0) = (s^*, a^*)\) \(d\)} \(A\) \(\epsilon\)} \(\delta\)} \(\Omega\)} \(\epsilon\)} \(\delta\)} \(\Omega\)} \(\epsilon\)} \(\delta\)} \(\Omega\)} \(\epsilon\)} \(\delta\)} \(\Omega\)} \(\epsilon\)} \(\delta\)} 2/3. We will use it in our proof of Thm. 4.2.

Now we give our proof of Thm. 4.2 through the adversarial family of MDPs. Here we use the techniques of Thm. 7 in (Tarbouriech et al., 2020).

**Proof.** We denote by \(P_{(s^*, a^*)} \equiv P_{\pi^* M_{(s^*, a^*)}}\) and \(E_{(s^*, a^*)} \equiv E_{\pi^* M_{(s^*, a^*)}}\) the probability measure and expectation in the MDP \(M_{(s^*, a^*)}\) by following \(\pi\) and by \(P_0\) and \(E_0\) the corresponding operators in the MDP \(M_0\). We fix any algorithm \((\pi, \tau, K, \{\pi_s\}_{s \in K})\) that solves the AX problem. We will prove that when working on the MDP \(M_0\), the algorithm will cost at least \(\Omega(\frac{LSL A}{\epsilon_{\min} \epsilon^2 \log \frac{1}{\delta}})\) samples in expectation, i.e.

\[
E_0[\tau] = \Omega(\frac{LSL A}{\epsilon_{\min} \epsilon^2 \log \frac{1}{\delta}}),
\]

which yields that the lower bound of the total cost is \(\Omega(\frac{LSL A}{\epsilon_{\min} \epsilon^2 \log \frac{1}{\delta}})\).

Now we fix the state-action pair \((s^*, a^*) \in S^* \times A\) \((a^* \neq \text{RESET})\). Also, we denote the random variable \(N_{(s,a)}^\tau\) as the number of samples that the algorithm takes at the state-action pair \((s, a) \in S \times A\). For any \(\mathcal{F}_\tau\)-measurable random variable \(Z\) taking values in \([0, 1]\), we have

\[
E_0\left[N_{(s^*, a^*)}^\tau\right] \geq \frac{144\epsilon_{\min} \epsilon^2}{L} \geq \epsilon_0\left[N_{(s^*, a^*)}^\tau\right] \geq \text{KL}\left(P_0^\tau, P_{(s^*, a^*)}^\tau\right) \geq \text{KL}(E_0[Z], E_{(s^*, a^*)}[Z]),
\]

where (a) uses Lemma E.4 and \(d_1 \geq L/2\); (b) uses Lemma E.2; (c) uses Lemma E.3.

For any \((s, a) \in S \times A\), we define the event \(Z_{s,a} = 1\) \{The algorithm’s output satisfies \(g \in K\) and \(V_{M_{(s,a)}}^\tau g(s_0) \leq (1 + \epsilon) L\}\}. And we set the event \(Z = Z_{s^*, a^*}\). We note that \(Z_{s,a}\) can be viewed as a random event on distribution \(P_{(s,a)}\), and can also be viewed as a random event on distribution \(P_{(s,a)}\).

First we focus on distribution \(P_{(s^*, a^*)}\). We observe that as the algorithm \((\pi, \tau, K, \{\pi_s\}_{s \in K})\) solves the AX problem, when working on the MDP \(M_{(s^*, a^*)}\), with probability at least \(1 - \delta\), its output should satisfy \(g \in K\) and the expected cost of the policy \(\pi_{g}\) to reach \(g\) from state \(s_0\) is no more than \((1 + \epsilon) L\). Therefore, for any \((s^*, a^*) \in S^* \times A\) \((a^* \neq \text{RESET})\), we have

\[
P_{(s^*, a^*)}[Z_{s^*, a^*}] \geq 1 - \delta.
\]

Then we focus on probability distribution \(P_0\) (i.e., \(P_{(s^*, a^*)}\)). We recall that the event \(Z_{s,a}\) implies \(P_{\pi_{s,M_0}}[(s_0, a_0) = (s, a)] \geq 2/3\). And for any two distinct state-action pairs \((s, a)\) and \((s', a')\), the event \(P_{\pi_{s,M_0}}[(s_0, a_0) = (s, a)] \geq 2/3\) and the event \(P_{\pi_{s,M_0}}[(s_0, a_0) = (s', a')] \geq 2/3\) are mutually exclusive. Hence \(Z_{s,a}\) and \(Z_{s',a'}\) are mutually exclusive on \(P_0\), and we have

\[
\sum_{(s,a) \in S^* \times A} P_0[Z_{s,a}] \leq 1.
\]
We recall that we set $Z = Z_{s^*, a^*}$, and we can obtain
\[
\text{kl}(E_0[Z], E_{(s^*, a^*)}[Z]) = \text{kl}(P_0[Z_{s^*, a*}], P_{(s^*, a*)}[Z_{s^*, a*}]) \\
\geq (a) (1 - P_0[Z_{s^*, a*}]) \log \left(\frac{1}{1 - P_{(s^*, a*)}[Z_{s^*, a*}]\log(2)}\right) \\
\geq (b) (1 - P_0[Z_{s^*, a*}]) \log \left(\frac{1}{\delta}\right) - \log(2),
\]
where (a) uses Lem. E.5; (b) uses that $P_{(s^*, a*)}[Z_{s^*, a*}] \geq 1 - \delta$. Therefore, we have
\[
E_0[N^*_{s^*, a*}] \geq \frac{L}{144\min\epsilon^2}((1 - P_0[Z_{s^*, a*}]) \log \left(\frac{1}{\delta}\right) - \log(2)).
\]
We recall that $\sum_{(s, a) \in S' \times A} P_0[Z_{s, a}] \leq 1$. Thus summing up all the state-action pairs $(s^*, a^*) \in S' \times A$, we can obtain that
\[
\sum_{(s, a) \in S' \times A} E_0[N^*_{s^*, a*}] \geq \frac{L}{144\min\epsilon^2}((|S'||A| - 1) \log \left(\frac{1}{\delta}\right) - \log(2)|S'||A|).
\]
Hence provided that $|S'| \geq \frac{L}{4\delta^2}L > 4, S > 8, A > 4, 4 \leq S_L \leq \min\{(A - 1)\frac{1}{2}, S\}$, $0 < \epsilon < \frac{1}{2}$, and $0 < \delta < \frac{1}{10}$, we can eventually obtain the lower bound of the total number of steps $\tau$,
\[
E_0[\tau] = \sum_{(s, a) \in S \times A} E_0[N^*_{s, a}] \geq \sum_{(s^*, a^*) \in S' \times A} E_0[N^*_{s^*, a^*}] \geq \Omega\left(\frac{L S_L^2 A}{\min\epsilon^2} \log \frac{1}{\delta}\right).
\]

\section*{F. Lower Bounds for Multi-goal SSP}

Here we formulate the lower bound for the multi-goal SSP problem. First we define an algorithm for the multi-goal SSP problem with goal space $G$ as a triple $(\pi, \tau, \{\pi_s\}_{s \in G})$, which means the algorithm executes a history-dependent policy $\pi$, and returns a set of policies $\{\pi_s\}_{s \in G}$ after sampling $\tau$ times. Also, we allow $\pi_s$ to be Markov policies. And we release the multi-goal SSP problem in this way: we only require the algorithm output policies $\pi_s$ such that $V^*_s(\pi_s(0)) \leq (1 + \epsilon)L$.

\begin{definition}
An algorithm $(\pi, \tau, \{\pi_s\}_{s \in G})$ is $(\epsilon, \delta, L)$-PAC for multi-goal SSP problem on MDP $M$ with goal space $G \subseteq S$, if with probability over $1 - \delta$, the algorithm returns a set of policies $\{\pi_s\}_{s \in G}$ after $\tau$ steps, such that $\forall s \in G, V^*_s(\pi_s(0)) \leq (1 + \epsilon)L$.
\end{definition}

Then for any real numbers $L, c_{\min}$ and positive integers $S, A$, we define a class of MDPs $\mathcal{M}_{\text{MSSP}}(L, S)$ as follows: $\mathcal{M}_{\text{MSSP}}(L, S)$ contains all the MDPs $M = (S, A, P, c, s_0)$, such that $|S| \leq S, |A| \leq A, c(s, a) \in [c_{\min}, 1]$ for all $(s, a) \in S \times A$, and $M$ satisfies Assm. 2.1 and $S_L^1 = S$.

We remark that our constructed adversarial examples (cf. Fig. 2) for the autonomous exploration problem can also be applied to multi-goal SSP using the similar proof with Thm. 4.2. Thus we obtain the following lower bound for multi-goal SSP, which implies that our Alg. 2 is also minmax for multi-goal SSP problem. See Appendix E for more details.

\begin{theorem}
Assume that $L > 4, A > 4, 8 < S \leq (A - 1)\frac{1}{2}, 0 < \epsilon < \frac{1}{2}, 0 < \delta < \frac{1}{10}$, and $0 < c_{\min} \leq 1$. Then for any algorithm $(\pi, \tau, K, \{\pi_s\}_{s \in K})$ that is $(\epsilon, \delta, L)$-PAC for multi-goal SSP problem on any MDP $M \in \mathcal{M}_{\text{MSSP}}(L, S)$ with any goal space $G \subseteq S$, there exists an MDP $M \in \mathcal{M}_{\text{MSSP}}(L, S)$ such that
\[
E_{\pi, M}[\tau] = \Omega\left(\frac{L S_L A}{c_{\min}^2} \log \frac{1}{\delta}\right).
\]
\end{theorem}

We note that in our construction of adversarial examples (cf. Fig. 2) and in our proof of Thm. 4.2, we only involved one goal state $g$. Hence we can also prove that for the classical single-goal SSP problem with $G = \{g\}$, learning a policy $\pi_g$ such that $V^*_g(\pi_g(0)) \leq (1 + \epsilon)L$ also requires $\Omega\left(\frac{L S_L A}{c_{\min}^2} \log \frac{1}{\delta}\right)$ samples, and the lower bound of cumulative cost scales as $\Omega(L S L A c_{\min}^{-2} \log \frac{1}{\delta})$. 

\section*{Near-Optimal Algorithms for Autonomous Exploration and Multi-Goal Stochastic Shortest Path}