Improved Sample Complexity for Incremental Autonomous Exploration in MDPs

Jean Tarbouriech
Facebook AI Research Paris & Inria Lille
jean.tarbouriech@gmail.com

Matteo Pirotta
Facebook AI Research Paris
pirotta@fb.com

Michal Valko
DeepMind Paris
valkom@deepmind.com

Alessandro Lazaric
Facebook AI Research Paris
lazaric@fb.com

Abstract

We investigate the exploration of an unknown environment when no reward function is provided. Building on the incremental exploration setting introduced by Lim and Auer [1], we define the objective of learning the set of ε-optimal goal-conditioned policies attaining all states that are incrementally reachable within L steps (in expectation) from a reference state s_0. In this paper, we introduce a novel model-based approach that interleaves discovering new states from s_0 and improving the accuracy of a model estimate that is used to compute goal-conditioned policies to reach newly discovered states. The resulting algorithm, DisCo, achieves a sample complexity scaling as $\widetilde{O}(L^5 S_{L+\varepsilon}\Gamma_{L+\varepsilon}A\,\varepsilon^{-2})$, where A is the number of actions, $S_{L+\varepsilon}$ is the number of states that are incrementally reachable from s_0 in $L+\varepsilon$ steps, and $\Gamma_{L+\varepsilon}$ is the branching factor of the dynamics over such states. This improves over the algorithm proposed in [1] in both ε and L at the cost of an extra $\Gamma_{L+\varepsilon}$ factor, which is small in most environments of interest. Furthermore, DisCo is the first algorithm that can return an $\varepsilon/c_{\min}$-optimal policy for any cost-sensitive shortest-path problem defined on the L-reachable states with minimum cost $c_{\min}$. Finally, we report preliminary empirical results confirming our theoretical findings.

1 Introduction

In cases where the reward signal is not informative enough — e.g., too sparse, time-varying or even absent — a reinforcement learning (RL) agent needs to explore the environment driven by objectives other than reward maximization, see [e.g., 2, 3, 4, 5, 6]. This can be performed by designing intrinsic rewards to drive the learning process, for instance via state visitation counts [7, 8], novelty or prediction errors [9, 10, 11]. Other recent methods perform information-theoretic skill discovery to learn a set of diverse and task-agnostic behaviors [12, 13, 14]. Alternatively, goal-conditioned policies learned by carefully designing the sequence of goals during the learning process are often used to solve sparse reward problems [15] and a variety of goal-reaching tasks [16, 17, 18, 19].

While the approaches reviewed above effectively leverage deep RL techniques and are able to achieve impressive results in complex domains (e.g., Montezuma's Revenge [15] or real-world robotic manipulation tasks [19]), they often lack substantial theoretical understanding and guarantees. Recently, some *unsupervised RL* objectives were analyzed rigorously. Some of them quantify how well the agent visits the states under a sought-after frequency, e.g., to induce a maximally entropic state distribution [20, 21, 22, 23]. While such strategies provably mimic their desired behavior via a Frank-Wolfe algorithmic scheme, they may not learn how to effectively reach any state of the environment and thus may not be sufficient to efficiently solve downstream tasks. Another relevant take is the reward-free RL paradigm of [24]: following its exploration phase, the agent is able to compute a near-optimal policy for any reward function at test time. While this framework yields strong end-to-end guarantees, it is limited to the finite-horizon setting and the agent is thus unable to tackle tasks beyond finite-horizon, e.g., goal-conditioned tasks.

In this paper, we build on and refine the setting of incremental exploration of [1]: the agent starts at an initial state s_0 in an unknown, possibly large environment, and it is provided with a RESET action to restart at s_0. At a high level, in this setting the agent should explore the environment and stop when it has identified the *tasks* within its *reach* and learned to *master* each of them sufficiently well. More specifically, the objective of the agent is to learn a goal-conditioned policy for *any* state that can be reached from s_0 within L steps in expectation; such a state is said to be L-controllable. Lim and Auer [1] address this setting with the `UcbExplore` method for which they bound the number of exploration steps that are required to identify in an incremental way all L-controllable states (i.e., the algorithm needs to define a suitable stopping condition) and to return a set of policies that are able to reach each of them in *at most* $L + \varepsilon$ steps. A key aspect of `UcbExplore` is to first focus on simple states (i.e., states that can be reached within a few steps), learn policies to efficiently reach them, and leverage them to identify and tackle states that are increasingly more difficult to reach. This approach aims to avoid wasting exploration in the attempt of reaching states that are further than L steps from s_0 or that are too difficult to reach given the limited knowledge available at earlier stages of the exploration process. Our main contributions are:

- We strengthen the objective of incremental exploration and require the agent to learn ε-optimal goal-conditioned policies for any L-controllable state. Formally, let $V^\star(s)$ be the length of the shortest path from s_0 to s, then the agent needs to learn a policy to navigate from s_0 to s in at most $V^\star(s) + \varepsilon$ steps, while in [1] any policy reaching s in *at most* $L + \varepsilon$ steps is acceptable.
- We design `DisCo`, a novel algorithm for incremental exploration. `DisCo` relies on an estimate of the transition model to compute goal-conditioned policies to the states observed so far and then use those policies to improve the accuracy of the model and incrementally discover new states.
- We derive a sample complexity bound for `DisCo` scaling as[1] $\widetilde{O}(L^5 S_{L+\varepsilon}\Gamma_{L+\varepsilon} A\,\varepsilon^{-2})$, where A is the number of actions, $S_{L+\varepsilon}$ is the number of states that are *incrementally* controllable from s_0 in $L+\varepsilon$ steps, and $\Gamma_{L+\varepsilon}$ is the branching factor of the dynamics over such incrementally controllable states. Not only is this sample complexity obtained for a more challenging objective than `UcbExplore`, but it also improves in both ε and L at the cost of an extra $\Gamma_{L+\varepsilon}$ factor, which is small in most environments of interest.
- Leveraging the model-based nature of `DisCo`, we can also readily compute an $\varepsilon/c_{\min}$-optimal policy for *any* cost-sensitive shortest-path problem defined on the L-controllable states with minimum cost $c_{\min}$. This result serves as a goal-conditioned counterpart to the reward-free exploration framework defined by Jin et al. [24] for the finite-horizon setting.

2 Incremental Exploration to Discover and Control

In this section we expand [1], with a more challenging objective for autonomous exploration.

2.1 L-Controllable States

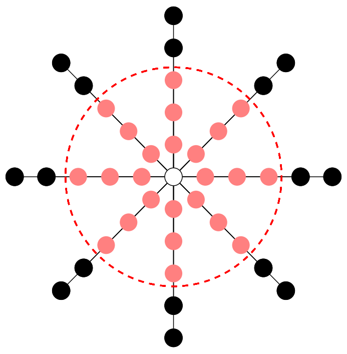
We consider a *reward-free* Markov decision process [25, Sect. 8.3] $M := \langle \mathcal{S}, \mathcal{A}, p, s_0 \rangle$. We assume a finite action space $\mathcal{A}$ with $A = |\mathcal{A}|$ actions, and a finite, possibly large state space $\mathcal{S}$ for which an upper bound S on its cardinality is known, i.e., $|\mathcal{S}| \leq S$.[2] Each state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$ is characterized by an unknown transition probability distribution $p(\cdot|s,a)$ over next states. We denote by $\Gamma_{\mathcal{S}'} := \max_{s\in\mathcal{S}',a} \lVert \{p(s'|s,a)\}_{s'\in\mathcal{S}'} \rVert_0$ the largest branching factor of the dynamics over states in any subset $\mathcal{S}' \subseteq \mathcal{S}$. The environment has no extrinsic reward, and $s_0 \in \mathcal{S}$ is a designated initial state.

A deterministic stationary policy $\pi : \mathcal{S} \to \mathcal{A}$ is a mapping between states to actions and we denote by Π the set of all possible policies. Since in environments with arbitrary dynamics the learner may get stuck in a state without being able to return to s_0, we introduce the following assumption.[3]

[1]We say that $f(\varepsilon) = \widetilde{O}(\varepsilon^\alpha)$ if there are constants a, b, such that $f(\varepsilon) \leq a \cdot \varepsilon^\alpha \log^b(\varepsilon)$.

[2]Lim and Auer [1] originally considered a countable, possibly infinite state space; however this leads to a technical issue in the analysis of `UcbExplore` (acknowledged by the authors via personal communication and explained in App. E.3), which disappears by considering only finite state spaces.

[3]This assumption should be contrasted with the finite-horizon setting, where each policy resets automatically after H steps, or assumptions on the MDP dynamics such as ergodicity or bounded diameter, which guarantee that it is always possible to find a policy navigating between any two states.



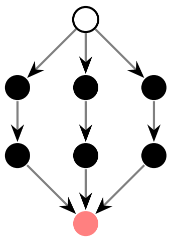


Figure 1: Two environments where the starting state s_0 is in white. *Left:* Each transition between states is deterministic and depicted with an edge. *Right:* Each transition from s_0 to the first layer is *equiprobable* and the transitions in the successive layers are deterministic. If we set $L = 3$, then the states belonging to $\mathcal{S}_L$ are colored in red. As the right figure illustrates, L-controllability is not necessarily linked to a notion of distance between states and an L-controllable state may be achieved by traversing states that are not L-controllable themselves.

Assumption 1. *The action space contains a* RESET *action s.t.* $p(s_0|s, \text{RESET}) = 1$ *for any* $s \in \mathcal{S}$.

We make explicit the states where a policy π takes action RESET in the following definition.

Definition 1 (Policy restricted on a subset)**.** *For any* $\mathcal{S}' \subseteq \mathcal{S}$*, a policy* π *is restricted on* $\mathcal{S}'$ *if* $\pi(s) = \text{RESET}$ *for any* $s \notin \mathcal{S}'$*. We denote by* $\Pi(\mathcal{S}')$ *the set of policies restricted on* $\mathcal{S}'$.

We measure the performance of a policy in navigating the MDP as follows.

Definition 2. *For any policy* π *and a pair of states* $(s, s') \in \mathcal{S}^2$*, let* $\tau_\pi(s \rightarrow s')$ *be the (random) number of steps it takes to reach* s' *starting from* s *when executing policy* π*, i.e.,* $\tau_\pi(s \rightarrow s') := \inf\{t \geq 0 : s_{t+1} = s' \,|\, s_1 = s, \pi\}$*. We also set* $v_\pi(s \rightarrow s') := \mathbb{E}[\tau_\pi(s \rightarrow s')]$ *as the expected traveling time, which corresponds to the value function of policy* π *in a stochastic shortest-path setting (SSP, [26, Sect. 3]) with initial state* s*, goal state* s' *and unit cost function. Note that we have* $v_\pi(s \rightarrow s') = +\infty$ *when the policy* π *does not reach* s' *from* s *with probability 1. Furthermore, for any subset* $\mathcal{S}' \subseteq \mathcal{S}$ *and any state* s*, we denote by*

$$V^\star_{\mathcal{S}'}(s_0 \rightarrow s) := \min_{\pi \in \Pi(\mathcal{S}')} v_\pi(s_0 \rightarrow s),$$

the length of the shortest path to s*, restricted to policies resetting to* s_0 *from any state outside* $\mathcal{S}'$.

The objective of the learning agent is to *control efficiently* the environment in the *vicinity* of s_0. We say that a state s is controlled if the agent can reliably navigate to it from s_0, that is, there exists an effective *goal-conditioned policy* — i.e., a *shortest-path policy* — from s_0 to s.

Definition 3 (L-controllable states)**.** *Given a reference state* s_0*, we say that a state* s *is* L*-controllable if there exists a policy* π *such that* $v_\pi(s_0 \rightarrow s) \leq L$*. The set of* L*-controllable states is then*

$$\mathcal{S}_L := \{s \in \mathcal{S} : \min_{\pi \in \Pi} v_\pi(s_0 \rightarrow s) \leq L\}. \tag{1}$$

We illustrate the concept of controllable states in Fig. 1 for $L = 3$. Interestingly, in the right figure, the black states are not L-controllable. In fact, there is no policy that can directly choose which one of the black states to reach. On the other hand, the red state, despite being in some sense *further* from s_0 than the black states, *does* belong to $\mathcal{S}_L$. In general, there is a crucial difference between the existence of a *random* realization where a state s is reached from s_0 in less than L steps (i.e., black states) and the notion of L*-controllability*, which means that there exists a policy that consistently reaches the state in a number of steps less or equal than L on average (i.e., red state). This explains the choice of the term *controllable* over *reachable*, since a state s is often said to be reachable if there is a policy π with a non-zero probability to eventually reach it, which is a weaker requirement.

Unfortunately, Lim and Auer [1] showed that in order to discover all the states in $\mathcal{S}_L$, the learner may require a number of exploration steps that is *exponential* in L or $|\mathcal{S}_L|$. Intuitively, this negative result is due to the fact that the minimum in Eq. 1 is over the set of all possible policies, including those that may traverse states that are not in $\mathcal{S}_L$.[4] Hence, we similarly constrain the learner to focus on the set of *incrementally controllable* states.

Definition 4 (Incrementally controllable states $\mathcal{S}_L^{\rightarrow}$)**.** *Let* $\prec$ *be some partial order on* $\mathcal{S}$*. The set* $\mathcal{S}_L^{\prec}$ *of states controllable in* L *steps w.r.t.* $\prec$ *is defined inductively as follows. The initial state* s_0

[4]We refer the reader to [1, Sect. 2.1] for a more formal and complete characterization of this negative result.

belongs to $\mathcal{S}_L^{\prec}$ by definition and if there exists a policy π restricted on $\{s' \in \mathcal{S}_L^{\prec} : s' \prec s\}$ with $v_\pi(s_0 \to s) \leq L$, then $s \in \mathcal{S}_L^{\prec}$. The set $\mathcal{S}_L^{\to}$ of incrementally L-controllable states is defined as $\mathcal{S}_L^{\to} := \cup_{\prec} \mathcal{S}_L^{\prec}$, where the union is over all possible partial orders.

By way of illustration, in Fig. 1 for $L = 3$, it holds that $\mathcal{S}_L^{\to} = \mathcal{S}_L$ in the left figure, whereas $\mathcal{S}_L^{\to} = \{s_0\} \neq \mathcal{S}_L$ in the right figure. Indeed, while the red state is L-controllable, it requires traversing the black states, which are not L-controllable.

2.2 AX Objectives

We are now ready to formalize two alternative objectives for *Autonomous eXploration* (AX) in MDPs.

Definition 5 (AX sample complexity)**.** *Fix any length $L \geq 1$, error threshold $\varepsilon > 0$ and confidence level $\delta \in (0, 1)$. The sample complexities $\mathcal{C}_{\mathrm{AX}_L}(\mathfrak{A}, L, \varepsilon, \delta)$ and $\mathcal{C}_{\mathrm{AX}^\star}(\mathfrak{A}, L, \varepsilon, \delta)$ are defined as the number of time steps required by a learning algorithm $\mathfrak{A}$ to identify a set $\mathcal{K} \supseteq \mathcal{S}_L^{\to}$ such that with probability at least $1 - \delta$, it has learned a set of policies $\{\pi_s\}_{s \in \mathcal{K}}$ that respectively verifies the following AX requirement*

$$(\mathrm{AX_L}) \quad \forall s \in \mathcal{K}, v_{\pi_s}(s_0 \to s) \leq L + \varepsilon,$$
$$(\mathrm{AX^\star}) \quad \forall s \in \mathcal{K}, v_{\pi_s}(s_0 \to s) \leq V^\star_{\mathcal{S}_L^{\to}}(s_0 \to s) + \varepsilon.$$

Designing agents satisfying the objectives defined above introduces critical difficulties w.r.t. standard goal-directed learning in RL. First, the agent has to find accurate policies for a set of goals (i.e., all incrementally L-controllable states) and not just for one specific goal. On top of this, the set of desired goals itself (i.e., the set $\mathcal{S}_L^{\to}$) is *unknown* in advance and has to be estimated online. Specifically, $\mathrm{AX_L}$ is the original objective introduced in [1] and it requires the agent to discover all the incrementally L-controllable states as fast as possible.[5] At the end of the learning process, for each state $s \in \mathcal{S}_L^{\to}$ the agent should return a policy that can reach s from s_0 in at most L steps (in expectation). Unfortunately, this may correspond to a rather poor performance in practice. Consider a state $s \in \mathcal{S}_L^{\to}$ such that $V^\star_{\mathcal{S}_L^{\to}}(s_0 \to s) \ll L$, i.e., the shortest path between s_0 to s following policies restricted on $\mathcal{S}_L^{\to}$ is much smaller than L. Satisfying $\mathrm{AX_L}$ only guarantees that a policy reaching s in L steps is found. On the other hand, objective $\mathrm{AX^\star}$ is more demanding, as it requires learning a near-optimal shortest-path policy for each state in $\mathcal{S}_L^{\to}$. Since $V^\star_{\mathcal{S}_L^{\to}}(s_0 \to s) \leq L$ and the gap between the two quantities may be arbitrarily large, especially for states close to s_0 and far from the fringe of $\mathcal{S}_L^{\to}$, $\mathrm{AX^\star}$ is a significantly tighter objective than $\mathrm{AX_L}$ and it is thus preferable in practice.

We say that an exploration algorithm solves the AX problem if its sample complexity $\mathcal{C}_{\mathrm{AX}}(\mathfrak{A}, L, \varepsilon, \delta)$ in Def. 5 is polynomial in $|\mathcal{K}|$, A, L, ε^{-1} and $\log(S)$. Notice that requiring a logarithmic dependency on the size of $\mathcal{S}$ is crucial but nontrivial, since the overall state space may be large and we do not want the agent to waste time trying to reach states that are not L-controllable. The dependency on the (algorithmic-dependent and random) set $\mathcal{K}$ can be always replaced using the upper bound $|\mathcal{K}| \leq |\mathcal{S}_{L+\varepsilon}^{\to}|$, which is implied with high probability by both $\mathrm{AX_L}$ and $\mathrm{AX^\star}$ conditions. Finally, notice that the error threshold $\varepsilon > 0$ has a two-fold impact on the performance of the algorithm. First, ε defines the largest set $\mathcal{S}_{L+\varepsilon}^{\to}$ that could be returned by the algorithm: the larger ε, the bigger the set. Second, as ε increases, the quality (in terms of controllability and navigational precision) of the output policies worsens w.r.t. the shortest-path policy restricted on $\mathcal{S}_L^{\to}$.

3 The `DisCo` Algorithm

The algorithm `DisCo` — for `Discover and Control` — is detailed in Alg. 1. It maintains a set $\mathcal{K}$ of "controllable" states and a set $\mathcal{U}$ of states that are considered "uncontrollable" *so far*. A state s is tagged as controllable when a policy to reach s in at most $L + \varepsilon$ steps (in expectation from s_0) has been found with high confidence, and we denote by π_s such policy. The states in $\mathcal{U}$ are states that have been discovered as potential members of $\mathcal{S}_L^{\to}$, but the algorithm has yet to produce a policy to control any of them in less than $L + \varepsilon$ steps. The algorithm stores an estimate of the transition model and it proceeds through rounds, which are indexed by k and incremented whenever a state in $\mathcal{U}$ gets transferred to the set $\mathcal{K}$, i.e., when the transition model reaches a level of accuracy sufficient

[5]Note that we translated in the condition in [1] of a relative error of $L\varepsilon$ to an absolute error of ε, to align it with the common formulation of sample complexity in RL.

Algorithm 1: Algorithm `DisCo`

```
Input: Actions $A$, initial state $s_0$, confidence parameter $\delta \in (0,1)$, error threshold $\varepsilon > 0$, $L \geq 1$ and
       (possibly adaptive) allocation function $\phi : \mathcal{P}(\mathcal{S}) \to \mathbb{N}$ (where $\mathcal{P}(\mathcal{S})$ denotes the power set of $\mathcal{S}$).
1  Initialize $k := 0$, $\mathcal{K}_0 := \{s_0\}$, $\mathcal{U}_0 := \{\}$ and a restricted policy $\pi_{s_0} \in \Pi(\mathcal{K}_0)$.
2  Set $\varepsilon := \min\{\varepsilon, 1\}$ and continue := True.
3  while continue do
4      Set $k$ += 1. //new round
       // ① Sample collection on K
5      For each $(s,a) \in \mathcal{K}_k \times A$, execute policy $\pi_s$ until the total number of visits $N_k(s,a)$ to $(s,a)$ satisfies
         $N_k(s,a) \geq n_k := \phi(\mathcal{K}_k)$. For each $(s,a) \in \mathcal{K}_k \times A$, add $s' \sim p(\cdot|s,a)$ to $\mathcal{U}_k$ if $s' \notin \mathcal{K}_k$.
       // ② Restriction of candidate states U
6      Compute transitions $\widehat{p}_k(s'|s,a)$ and $\mathcal{W}_k := \left\{ s' \in \mathcal{U}_k : \exists (s,a) \in \mathcal{K}_k \times A, \widehat{p}_k(s'|s,a) \geq \frac{1-\varepsilon/2}{L} \right\}$
7      if $\mathcal{W}_k$ is empty then
8          Set continue := False. //condition STOP1
9      else
           // ③ Computation of the optimistic policies on K
10         for each state $s' \in \mathcal{W}_k$ do
11             Compute $(\widetilde{u}_{s'}, \widetilde{\pi}_{s'}) :=$ OVI$_{\text{SSP}}(\mathcal{K}_k, A, s', N_k, \frac{\varepsilon}{6L})$; see Alg. 3 in App. D.1.
12         Let $s^\dagger := \arg\min_{s \in \mathcal{W}_k} \widetilde{u}_s(s_0)$ and $\widetilde{u}^\dagger := \widetilde{u}_{s^\dagger}(s_0)$.
13         if $\widetilde{u}^\dagger > L$ then
14             Set continue := False. //condition STOP2
15         else
               // ④ State transfer from U to K
16             Set $\mathcal{K}_{k+1} := \mathcal{K}_k \cup \{s^\dagger\}$, $\mathcal{U}_{k+1} := \mathcal{U}_k \setminus \{s^\dagger\}$ and $\pi_{s^\dagger} := \widetilde{\pi}_{s^\dagger}$.
   // ⑤ Policy consolidation: computation on the final set K
17 Set $K := k$.
18 for each state $s \in \mathcal{K}_K$ do
19     Compute $(\widetilde{u}_s, \widetilde{\pi}_s) :=$ OVI$_{\text{SSP}}(\mathcal{K}_K, A, s, N_K, \frac{\varepsilon}{6L})$.
20 Output: the states $s$ in $\mathcal{K}_K$ and their corresponding policy $\pi_s := \widetilde{\pi}_s$.
```

to compute a policy to control one of the states encountered before. We denote by $\mathcal{K}_k$ (resp. $\mathcal{U}_k$) the set of controllable (resp. uncontrollable) states at the beginning of round k. `DisCo` stops at a round K when it can confidently claim that all the remaining states outside of $\mathcal{K}_K$ cannot be L-controllable.

At each round, the algorithm uses all samples observed so far to build an estimate of the transition model denoted by $\widehat{p}(s'|s,a) = N(s,a,s')/N(s,a)$, where $N(s,a)$ and $N(s,a,s')$ are counters for state-action and state-action-next state visitations. Each round is divided into two phases. The first is a *sample collection* phase. At the beginning of round k, the agent collects additional samples until $n_k := \phi(\mathcal{K}_k)$ samples are available at each state-action pair in $\mathcal{K}_k \times A$ (step ①). A key challenge lies in the careful (and adaptive) choice of the allocation function ϕ, which we report in the statement of Thm. 1 (see Eq. 19 in App. D.4 for its exact definition). Importantly, the incremental construction of $\mathcal{K}_k$ entails that sampling at each state $s \in \mathcal{K}_k$ can be done efficiently. In fact, for all $s \in \mathcal{K}_k$ the agent has already confidently learned a policy π_s to reach s in at most $L + \varepsilon$ steps on average (see how such policy is computed in the second phase). The generation of transitions (s,a,s') for $(s,a) \in \mathcal{K}_k \times A$ achieves two objectives at once. First, it serves as a discovery step, since all observed next states s' not in $\mathcal{U}_k$ are added to it — in particular this guarantees sufficient exploration at the fringe (or border) of the set $\mathcal{K}_k$. Second, it improves the accuracy of the model p in the states in $\mathcal{K}_k$, which is essential in computing near-optimal policies and thus fulfilling the AX* condition.

The second phase does not require interacting with the environment and it focuses on the *computation of optimistic policies*. The agent begins by significantly restricting the set of candidate states in each round to alleviate the computational complexity of the algorithm. Namely, among all the states in $\mathcal{U}_k$, it discards those that do not have a high probability of belonging to $\mathcal{S}_L^{\rightarrow}$ by considering a restricted set $\mathcal{W}_k \subseteq \mathcal{U}_k$ (step ②). In fact, if the estimated probability $\widehat{p}_k$ of reaching a state $s \in \mathcal{U}_k$ from any of the controllable states in $\mathcal{K}_k$ is lower than $(1 - \varepsilon/2)/L$, then no shortest-path policy restricted on $\mathcal{K}_k$ could get to s from s_0 in less than $L + \varepsilon$ steps on average. Then for each state s' in $\mathcal{W}_k$, `DisCo` computes an optimistic policy restricted on $\mathcal{K}_k$ to reach s'. Formally, for any candidate state $s' \in \mathcal{W}_k$, we define the induced stochastic shortest path (SSP) MDP M_k' with goal state s' as follows.

Definition 6. *We define the SSP-MDP $M'_k := \langle \mathcal{S}, \mathcal{A}'_k(\cdot), c'_k, p'_k \rangle$ with goal state s', where the action space is such that $\mathcal{A}'_k(s) = \mathcal{A}$ for all $s \in \mathcal{K}_k$ and $\mathcal{A}'_k(s) = \{\text{RESET}\}$ otherwise (i.e., we focus on policies restricted on $\mathcal{K}_k$). The cost function is such that for all $a \in \mathcal{A}$, $c'_k(s', a) = 0$, and for any $s \neq s'$, $c'_k(s, a) = 1$. The transition model is $p'_k(s'|s', a) = 1$ and $p'_k(\cdot|s, a) = p(\cdot|s, a)$ otherwise.*[6]

The solution of M'_k is the shortest-path policy from s_0 to s' restricted on $\mathcal{K}_k$. Since p'_k is unknown, `DisCo` cannot compute the exact solution of M'_k, but instead, it executes optimistic value iteration (OVI_{SSP}) for SSP [27, 28] to obtain a value function $\widetilde{u}_{s'}$ and its associated greedy policy $\widetilde{\pi}_{s'}$ restricted on $\mathcal{K}_k$ (see App. D.1 for more details).

The agent then chooses a candidate goal state $s^\dagger$ for which the value $\widetilde{u}^\dagger := \widetilde{u}_{s^\dagger}(s_0)$ is the smallest. This step can be interpreted as selecting the optimistically most promising new state to control. Two cases are possible. If $\widetilde{u}^\dagger \leq L$, then $s^\dagger$ is added to $\mathcal{K}_k$ (step ¯), since the accuracy of the model estimate on the state-action space $\mathcal{K}_k \times \mathcal{A}$ guarantees that the policy $\widetilde{\pi}_{s^\dagger}$ is able to reach the state $s^\dagger$ in less than $L + \varepsilon$ steps in expectation with high probability (i.e., $s^\dagger$ is incrementally $(L + \varepsilon)$-controllable). Otherwise, we can guarantee that $\mathcal{S}^{\rightarrow}_L \subseteq \mathcal{K}_k$ with high probability. In the latter case, the algorithm terminates and, using the current estimates of the model, it recomputes an optimistic shortest-path policy π_s restricted on the final set $\mathcal{K}_K$ for each state $s \in \mathcal{K}_K$ (step °). This policy consolidation step is essential to identify near-optimal policies restricted on the final set $\mathcal{K}_K$ (and thus on $\mathcal{S}^{\rightarrow}_L$): indeed the expansion of the set of the so far controllable states may alter and refine the optimal goal-reaching policies restricted on it (see App. A).

Computational Complexity. Note that algorithmically, we do not need to define M'_k (Def. 6) over the whole state space $\mathcal{S}$ as we can limit it to $\mathcal{K}_k \cup \{s'\}$, i.e., the candidate state s' and the set $\mathcal{K}_k$ of so far controllable states. As shown in Thm. 1, this set can be significantly smaller than $\mathcal{S}$. In particular this implies that the computational complexity of the value iteration algorithm used to compute the optimistic policies is independent from S (see App. D.9 for more details).

4 Sample Complexity Analysis of `DisCo`

We now present our main result: a sample complexity guarantee for `DisCo` for the $\text{AX}^\star$ objective, which directly implies that AX_L is also satisfied.

Theorem 1. *There exists an absolute constant $\alpha > 0$ such that for any $L \geq 1$, $\varepsilon \in (0, 1]$, and $\delta \in (0, 1)$, if we set the allocation function ϕ as*

$$\phi : \mathcal{X} \to \alpha \cdot \left(\frac{L^4 \widehat{\Theta}(\mathcal{X})}{\varepsilon^2} \log^2\left(\frac{LSA}{\varepsilon\delta}\right) + \frac{L^2 |\mathcal{X}|}{\varepsilon} \log\left(\frac{LSA}{\varepsilon\delta}\right) \right), \tag{2}$$

with $\widehat{\Theta}(\mathcal{X}) := \max_{(s,a)\in\mathcal{X}\times\mathcal{A}} \left(\sum_{s'\in\mathcal{X}} \sqrt{\widehat{p}(s'|s,a)(1 - \widehat{p}(s'|s,a))}\right)^2$, *then the algorithm* `DisCo` *(Alg. 1) satisfies the following sample complexity bound for* $\text{AX}^\star$

$$\mathcal{C}_{\text{AX}^\star}(\texttt{DisCo}, L, \varepsilon, \delta) = \widetilde{O}\left(\frac{L^5 \Gamma_{L+\varepsilon} S_{L+\varepsilon} A}{\varepsilon^2} + \frac{L^3 S^2_{L+\varepsilon} A}{\varepsilon} \right), \tag{3}$$

where $S_{L+\varepsilon} := |\mathcal{S}^{\rightarrow}_{L+\varepsilon}|$ *and*

$$\Gamma_{L+\varepsilon} := \max_{(s,a)\in\mathcal{S}^{\rightarrow}_{L+\varepsilon}\times\mathcal{A}} \|\{p(s'|s,a)\}_{s'\in\mathcal{S}^{\rightarrow}_{L+\varepsilon}}\|_0 \leq S_{L+\varepsilon}$$

is the maximal support of the transition probabilities $p(\cdot|s, a)$ restricted to the set $\mathcal{S}^{\rightarrow}_{L+\varepsilon}$.

Given the definition of $\text{AX}^\star$, Thm. 1 implies that `DisCo` **1)** terminates after $\mathcal{C}_{\text{AX}^\star}(\texttt{DisCo}, L, \varepsilon, \delta)$ time steps, **2)** discovers a set of states $\mathcal{K} \supseteq \mathcal{S}^{\rightarrow}_L$ with $|\mathcal{K}| \leq S_{L+\varepsilon}$, **3)** and for each $s \in \mathcal{K}$ outputs a policy π_s which is ε-optimal w.r.t. policies restricted on $\mathcal{S}^{\rightarrow}_L$, i.e., $v_{\pi_s}(s_0 \to s) \leq V^\star_{\mathcal{S}^{\rightarrow}_L}(s_0 \to s) + \varepsilon$. Note that Eq. 3 displays only a *logarithmic* dependency on S, the total number of states. This property on the sample complexity of `DisCo`, along with its S-independent computational complexity, is significant when the state space $\mathcal{S}$ grows large w.r.t. the unknown set of interest $\mathcal{S}^{\rightarrow}_L$.

[6]In words, all actions at states in $\mathcal{K}_k$ behave exactly as in M and suffer a unit cost, in all states outside $\mathcal{K}_k$ only the reset action to s_0 is available with a unit cost, and all actions at the goal s' induce a zero-cost self-loop.

4.1 Proof Sketch of Theorem 1

While the complete proof is reported in App. D, we now provide the main intuition behind the result.

State Transfer from U to K (step ¯). Let us focus on a round k and a state $s^\dagger$ 2 U_k that gets added to K_k. For clarity we remove in the notation the round k, goal state $s^\dagger$ and starting state s_0. We denote by v and $\widetilde{v}$ the value functions of the candidate policy $\widetilde{\pi}$ in the true and optimistic model respectively, and by $\widetilde{u}$ the quantity w.r.t. which $\widetilde{\pi}$ is optimistically greedy. We aim to prove that $s^\dagger$ 2 $S^{\rightarrow}_{L+\varepsilon}$ (with high probability). The main chain of inequalities underpinning the argument is

$$v \quad jv \quad \widetilde{v}j + \widetilde{v} \overset{\text{(a)}}{\quad} \frac{\varepsilon}{2} + \widetilde{v} \overset{\text{(b)}}{\quad} \frac{\varepsilon}{2} + \widetilde{u} + \frac{\varepsilon}{2} \overset{\text{(c)}}{\quad} L + \varepsilon, \tag{4}$$

where (c) is guaranteed by algorithmic construction and (b) stems from the chosen level of value iteration accuracy. Inequality (a) has the flavor of a simulation lemma for SSP, by relating the shortest-path value function of a same policy between two models (the true one and the optimistic one). Importantly, when restricted to K these two models are close in virtue of the algorithmic design which enforces the collection of a minimum amount of samples at each state-action pair of $K \quad A$, denoted by n. Specifically, we obtain that

$$jv \quad \widetilde{v}j = \widetilde{O}\Big(\sqrt{\frac{L^4\Gamma_{\mathcal{K}}}{n}} + \frac{L^2 jKj}{n}\Big), \qquad \text{with} \quad \Gamma_{\mathcal{K}} := \max_{(s,a)\in\mathcal{K}\times\mathcal{A}} kfp(s'js,a)g_{s'\in\mathcal{K}}k_0 \quad jKj.$$

Note that $\Gamma_{\mathcal{K}}$ is the branching factor restricted to the set K. Our choice of n (given in Eq. 2) is then dictated to upper bound the above quantity by $\varepsilon/2$ in order to satisfy inequality (a). Let us point out that, interestingly yet unfortunately, the structure of the problem does not appear to allow for technical variance-aware improvements seeking to lower the value of n prescribed above (indeed the AX framework requires to analytically encompass the uncontrollable states U into a single meta state with higher transitional uncertainty, see App. D for details).

Termination of the Algorithm. Since $S^{\rightarrow}_L$ is *unknown*, we have to ensure that none of the states in $S^{\rightarrow}_L$ are "missed". As such, we prove that with overwhelming probability, we have $S^{\rightarrow}_L \quad K_K$ when the algorithm terminates at a round denoted by K. There remains to justify the final near-optimal guarantee w.r.t. the set of policies $\Pi(S^{\rightarrow}_L)$. Leveraging that step ° recomputes the policies $(\pi_s)_{s\in\mathcal{K}_K}$ on the final set K_K, we establish the following chain of inequalities

$$v \quad jv \quad \widetilde{v}j + \widetilde{v} \overset{\text{(a)}}{\quad} \frac{\varepsilon}{2} + \widetilde{v} \overset{\text{(b)}}{\quad} \frac{\varepsilon}{2} + \widetilde{u} + \frac{\varepsilon}{2} \overset{\text{(c)}}{\quad} V^\star_{\mathcal{K}_K} + \varepsilon \overset{\text{(d)}}{\quad} V^\star_{\mathcal{S}^!_L} + \varepsilon, \tag{5}$$

where (a) and (b) are as in Eq. 4, (c) leverages optimism and (d) stems from the inclusion $S^{\rightarrow}_L \quad K_K$.

Sample Complexity Bound. The choice of allocation function ϕ in Eq. 2 bounds n_K which is the total number of samples required at each state-action pair in $K_K \quad A$. We then compute a high-probability bound ψ on the time steps needed to collect a given sample, and show that it scales as $\widetilde{O}(L)$. Since the sample complexity is solely induced by the sample collection phase (step ¬), it can be bounded by the quantity $\psi\, n_K jK_K jA$. Putting everything together yields the bound of Thm. 1.

4.2 Comparison with `UcbExplore` [1]

We start recalling the critical distinction that `DisCo` succeeds in tackling problem AX$^?$, while `UcbExplore` [1] fails to do so (see App. A for details on the AX objectives). Nonetheless, in the following we show that even if we restrict our attention to AX$_\text{L}$, for which `UcbExplore` is designed, `DisCo` yields a better sample complexity in most of the cases. From [1], `UcbExplore` verifies[7]

$$\mathsf{C}_{\text{AX}_\text{L}}(\texttt{UcbExplore}, L, \varepsilon, \delta) = \widetilde{O}\left(\frac{L^6 S_{L+\varepsilon} A}{\varepsilon^3}\right) \tag{6}$$

Eq. 6 shows that the sample complexity of `UcbExplore` is linear in $S_{L+\varepsilon}$, while for `DisCo` the dependency is somewhat worse. In the main-order term $\widetilde{O}(1/\varepsilon^2)$ of Eq. 3, the bound depends linearly on $S_{L+\varepsilon}$ but also grows with the branching factor $\Gamma_{L+\varepsilon}$, which is not the "global" branching factor

[7]Note that if we replace the error of " for AX$_\text{L}$ with an error of L " as in [1], we recover the sample complexity of Ө $L^3 S_{L+"} A="^3$ stated in [1, Thm. 8].

but denotes the number of possible next states in $\mathcal{S}_{L+\varepsilon}^{\rightarrow}$ starting from $\mathcal{S}_{L+\varepsilon}^{\rightarrow}$. While in general we only have $\Gamma_{L+\varepsilon} \leq S_{L+\varepsilon}$, in many practical domains (e.g., robotics, user modeling), each state can only transition to a small number of states, i.e., we often have $\Gamma_{L+\varepsilon} = O(1)$ as long as the dynamics is not too "chaotic". While DisCo does suffer from a quadratic dependency on $S_{L+\varepsilon}$ in the second term of order $\widetilde{O}(1/\varepsilon)$, we notice that for any $S_{L+\varepsilon} \leq L^3\varepsilon^{-2}$ the bound of DisCo is still preferable. Furthermore, since for $\varepsilon \to 0$, $S_{L+\varepsilon}$ tends to S_L, the condition is always verified for small enough ε.

Compared to DisCo, the sample complexity of UcbExplore is worse in both ε and L. As stressed in Sect. 2.2, the better dependency on ε both improves the quality of the output goal-reaching policies as well as reduces the number of incrementally $(L+\varepsilon)$-controllable states returned by the algorithm. It is interesting to investigate why the bound of [1] (Eq. 6) inherits a $\widetilde{O}(\varepsilon^{-3})$ dependency. As reviewed in App. E, UcbExplore alternates between two phases of state discovery and policy evaluation. The optimistic policies computed by UcbExplore solve a *finite-horizon problem* (with horizon set to H_{UCB}). However, minimizing the expected time to reach a target state is intrinsically an SSP problem, which is exactly what DisCo leverages. By computing policies that solve a finite-horizon problem (note that UcbExplore resets every H_{UCB} time steps), [1] sets the horizon to $H_{\text{UCB}} := \lceil L + L^2\varepsilon^{-1}\rceil$, which leads to a policy-evaluation phase with sample complexity scaling as $\widetilde{O}(H_{\text{UCB}}\varepsilon^{-2}) = \widetilde{O}(\varepsilon^{-3})$. Since the rollout budget of $\widetilde{O}(\varepsilon^{-3})$ is hard-coded into the algorithm, the dependency on ε of UcbExplore's sample complexity cannot be improved by a more refined analysis; instead a different algorithmic approach is required such as the one employed by DisCo.

4.3 Goal-Free Cost-Free Exploration on $\mathcal{S}_L^{\rightarrow}$ with DisCo

A compelling advantage of DisCo is that it achieves an accurate estimation of the environment's dynamics restricted to the unknown subset of interest $\mathcal{S}_L^{\rightarrow}$. In contrast to UcbExplore which needs to restart its sample collection from scratch whenever L, ε or some transition costs change, DisCo can thus be *robust* to changes in such problem parameters. At the end of its exploration phase in Alg. 1, DisCo is able to perform zero-shot planning to solve other tasks restricted on $\mathcal{S}_L^{\rightarrow}$, such as cost-sensitive ones. Indeed in the following we show how the DisCo agent is able to compute an $\varepsilon/c_{\min}$-optimal policy for *any* stochastic shortest-path problem on $\mathcal{S}_L^{\rightarrow}$ with goal state $s \in \mathcal{S}_L^{\rightarrow}$ (i.e., s is absorbing and zero-cost) and cost function lower bounded by $c_{\min} > 0$.

Corollary 1. *There exists an absolute constant $\beta > 0$ such that for any $L \geq 1$, $\varepsilon \in (0,1]$ and $c_{\min} \in (0,1]$ verifying $\varepsilon \leq \beta \cdot (L\, c_{\min})$, with probability at least $1-\delta$, for* whatever *goal state $s \in \mathcal{S}_L^{\rightarrow}$ and* whatever *cost function c in $[c_{\min}, 1]$, DisCo can compute (after its exploration phase, without additional environment interaction) a policy $\widehat{\pi}_{s,c}$ whose SSP value function $V_{\widehat{\pi}_{s,c}}$ verifies*

$$V_{\widehat{\pi}_{s,c}}(s_0 \to s) \leq V^{\star}_{\mathcal{S}_L^{\rightarrow}}(s_0 \to s) + \frac{\varepsilon}{c_{\min}},$$

where $V_\pi(s_0 \to s) := \mathbb{E}\Big[\sum_{t=1}^{\tau_\pi(s_0\to s)} c(s_t, \pi(s_t)) \,\Big|\, s_1 = s_0\Big]$ is the SSP value function of a policy π and $V^{\star}_{\mathcal{S}_L^{\rightarrow}}(s_0 \to s) := \min_{\pi\in\Pi(\mathcal{S}_L^{\rightarrow})} V_\pi(s_0 \to s)$ is the optimal SSP value function restricted on $\mathcal{S}_L^{\rightarrow}$.

It is interesting to compare Cor. 1 with the reward-free exploration framework recently introduced by Jin et al. [24] in finite-horizon. At a high level, the result in Cor. 1 can be seen as a counterpart of [24] beyond finite-horizon problems, specifically in the goal-conditioned setting. While the parameter L defines the horizon of interest for DisCo, resetting after every L steps (as in finite-horizon) would prevent the agent to identify L-controllable states and lead to poor performance. This explains the distinct technical tools used: while [24] executes finite-horizon no-regret algorithms, DisCo deploys SSP policies restricted on the set of states that it "controls" so far. Algorithmically, both approaches seek to build accurate estimates of the transitions on a specific (unknown) state space of interest: the so-called "significant" states within H steps for [24], and the incrementally L-controllable states $\mathcal{S}_L^{\rightarrow}$ for DisCo. Bound-wise, the cost-sensitive AX* problem inherits the critical role of the minimum cost $c_{\min}$ in SSP problems (see App. C and e.g., [27, 28, 29]), which is reflected in the accuracy of Cor. 1 scaling inversely with $c_{\min}$. Another interesting element of comparison is the dependency on the size of the state space. While the algorithm introduced in [24] is robust w.r.t. states that can be reached with very low probability, it still displays a *polynomial* dependency on the total number of states S. On the other hand, DisCo has only a *logarithmic* dependency on S, while it directly depends on the number of $(L+\varepsilon)$-controllable states, which shows that DisCo effectively adapts to the state space of interest and it ignores all other states. This result is significant since not only $S_{L+\varepsilon}$ can be arbitrarily smaller than S, but also because the set $\mathcal{S}_{L+\varepsilon}^{\rightarrow}$ itself is initially unknown to the algorithm.

Figure 2: Proportion of the incrementally L-controllable states identified by DisCo and UcbExplore in a confusing chain domain for $L = 4.5$ and $\epsilon \in \{0.1, 0.4, 0.8\}$. Values are averaged over 50 runs.

5 Numerical Simulation

In this section, we provide the first evaluation of algorithms in the incremental autonomous exploration setting. In the implementation of both DisCo and UcbExplore, we remove the logarithmic and constant terms for simplicity. We also boost the empirical performance of UcbExplore in various ways, for example by considering confidence intervals derived from the empirical Bernstein inequality (see [30]) as opposed to Hoeffding as done in [1]. We refer the reader to App. F for details on the algorithmic configurations and on the environments considered.

We compare the sample complexity empirically achieved by DisCo and UcbExplore. Fig. 2 depicts the time needed to identify all the incrementally L-controllable states when $L = 4.5$ for different values of ϵ, on a confusing chain domain. Note that the sample complexity is achieved soon after, when the algorithm can confidently discard all the remaining states as non-controllable (it is reported in Tab. 2 of App. F). We observe that DisCo outperforms UcbExplore for any value of ϵ. In particular, the gap in performance increases as ϵ decreases, which matches the theoretical improvement in sample complexity from $\tilde{O}(\epsilon^{-3})$ for UcbExplore to $\tilde{O}(\epsilon^{-2})$ for DisCo. On a second environment — the combination lock problem introduced in [31] — we notice that DisCo again outperforms UcbExplore, as shown in App. F.

Another important feature of DisCo is that it targets the tighter objective $AX^{\star}$, whereas UcbExplore is only able to fulfill objective AX_L and may therefore elect suboptimal policies. In App. F we show empirically that, as expected theoretically, this directly translates into higher-quality goal-reaching policies recovered by DisCo.

6 Conclusion and Extensions

Connections to existing deep-RL methods. While we primarily focus the analysis of DisCo in the tabular case, we believe that the formal definition of AX problems and the general structure of DisCo may also serve as a theoretical grounding of many recent approaches to unsupervised exploration. For instance, it is interesting to draw a parallel between DisCo and the ideas behind Go-Explore [32]. Go-Explore similarly exploits the following principles: (1) remember states that have previously been visited, (2) first return to a promising state (without exploration), (3) then explore from it. Go-Explore assumes that the world is deterministic and resettable, meaning that one can reset the state of the simulator to a previous visit to that cell. Very recently [15], the same authors proposed a way to relax this requirement by training goal-conditioned policies to reliably return to cells in the archive during the exploration phase. In this paper, we investigated the theoretical dimension of this direction, by provably learning such goal-conditioned policies for the set of incrementally controllable states.

Future work. Interesting directions for future investigation include: 1) Deriving a lower bound for the AX problems; 2) Integrating DisCo into the meta-algorithm MNM [33] which deals with incremental exploration for AX_L in non-stationary environments; 3) Extending the problem to continuous state space and function approximation; 4) Relaxing the definition of incrementally controllable states and relaxing the performance definition towards allowing the agent to have a non-zero but limited sample complexity of learning a shortest-path policy for any state at test time.

Broader Impact

This paper makes contributions to the fundamentals of online learning (RL) and due to its theoretical nature, we see no ethical or immediate societal consequence of our work.

References

[1] Shiau Hong Lim and Peter Auer. Autonomous exploration for navigating in MDPs. In *Conference on Learning Theory*, pages 40–1, 2012.

[2] Jürgen Schmidhuber. 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*, pages 222–227, 1991.

[3] Nuttapong Chentanez, Andrew G Barto, and Satinder P Singh. Intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, pages 1281–1288, 2005.

[4] Pierre-Yves Oudeyer and Frederic Kaplan. What is intrinsic motivation? a typology of computational approaches. *Frontiers in neurorobotics*, 1:6, 2009.

[5] Satinder Singh, Richard L Lewis, Andrew G Barto, and Jonathan Sorg. Intrinsically motivated reinforcement learning: An evolutionary perspective. *IEEE Transactions on Autonomous Mental Development*, 2(2):70–82, 2010.

[6] Adrien Baranes and Pierre-Yves Oudeyer. Intrinsically motivated goal exploration for active motor learning in robots: A case study. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1766–1773. IEEE, 2010.

[7] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in neural information processing systems*, pages 1471–1479, 2016.

[8] Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, Xi Chen, Yan Duan, John Schulman, Filip DeTurck, and Pieter Abbeel. # exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in neural information processing systems*, pages 2753–2762, 2017.

[9] Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Variational information maximizing exploration. *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[10] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 16–17, 2017.

[11] Mohammad Gheshlaghi Azar, Bilal Piot, Bernardo Avila Pires, Jean-Bastian Grill, Florent Altché, and Rémi Munos. World discovery models. *arXiv preprint arXiv:1902.07685*, 2019.

[12] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*, 2019.

[13] Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. In *International Conference on Learning Representations*, 2020.

[14] Víctor Campos Camúñez, Alex Trott, Caiming Xiong, Richard Socher, Xavier Giró Nieto, and Jordi Torres Viñals. Explore, discover and learn: unsupervised discovery of state-covering skills. In *International Conference on Machine Learning*, pages 1317–1327. PMLR, 2020.

[15] Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O Stanley, and Jeff Clune. First return then explore. *arXiv preprint arXiv:2004.12919*, 2020.

[16] Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. Automatic goal generation for reinforcement learning agents. In *International Conference on Machine Learning*, pages 1515–1528, 2018.

[17] Cédric Colas, Pierre Fournier, Mohamed Chetouani, Olivier Sigaud, and Pierre-Yves Oudeyer. Curious: intrinsically motivated modular multi-goal reinforcement learning. In *International conference on machine learning*, pages 1331–1340. PMLR, 2019.

[18] David Warde-Farley, Tom Van de Wiele, Tejas Kulkarni, Catalin Ionescu, Steven Hansen, and Volodymyr Mnih. Unsupervised control through non-parametric discriminative rewards. In *International Conference on Learning Representations*, 2019.

[19] Vitchyr H Pong, Murtaza Dalal, Steven Lin, Ashvin Nair, Shikhar Bahl, and Sergey Levine. Skew- t: State-covering self-supervised reinforcement learning. In *International Conference on Machine Learning*, pages 7783–7792. PMLR, 2020.

[20] Elad Hazan, Sham Kakade, Karan Singh, and Abby Van Soest. Provably ef cient maximum entropy exploration. In *International Conference on Machine Learning*, pages 2681–2691, 2019.

[21] Jean Tarbouriech and Alessandro Lazaric. Active exploration in markov decision processes. In *The 22nd International Conference on Arti cial Intelligence and Statistics*, pages 974–982, 2019.

[22] Wang Chi Cheung. Exploration-exploitation trade-off in reinforcement learning on online markov decision processes with global concave rewards. *arXiv preprint arXiv:1905.06466*, 2019.

[23] Jean Tarbouriech, Shubhanshu Shekhar, Matteo Pirotta, Mohammad Ghavamzadeh, and Alessandro Lazaric. Active model estimation in markov decision processes. In *Conference on Uncertainty in Arti cial Intelligence*, 2020.

[24] Chi Jin, Akshay Krishnamurthy, Max Simchowitz, and Tiancheng Yu. Reward-free exploration for reinforcement learning. In *International Conference on Machine Learning*, pages 4870–4879. PMLR, 2020.

[25] Martin L Puterman. *Markov Decision Processes.: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

[26] Dimitri Bertsekas. *Dynamic programming and optimal control*, volume 2. 2012.

[27] Jean Tarbouriech, Evrard Garcelon, Michal Valko, Matteo Pirotta, and Alessandro Lazaric. No-regret exploration in goal-oriented reinforcement learning. In *International Conference on Machine Learning*, pages 9428–9437. PMLR, 2020.

[28] Aviv Rosenberg, Alon Cohen, Yishay Mansour, and Haim Kaplan. Near-optimal regret bounds for stochastic shortest path. In *International Conference on Machine Learning*, pages 8210–8219. PMLR, 2020.

[29] Dimitri P Bertsekas and Huizhen Yu. Stochastic shortest path problems under weak conditions. *Lab. for Information and Decision Systems Report LIDS-P-2909, MIT*, 2013.

[30] Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 263–272. JMLR. org, 2017.

[31] Mohammad Gheshlaghi Azar, Vicenç Gómez, and Hilbert J Kappen. Dynamic policy programming. *Journal of Machine Learning Research*, 13(Nov):3207–3245, 2012.

[32] Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O Stanley, and Jeff Clune. Go-explore: a new approach for hard-exploration problems. *arXiv preprint arXiv:1901.10995*, 2019.

[33] Pratik Gajane, Ronald Ortner, Peter Auer, and Csaba Szepesvari. Autonomous exploration for navigating in non-stationary CMPs. *arXiv preprint arXiv:1910.08446*, 2019.

[34] Blai Bonet. On the speed of convergence of value iteration on stochastic shortest-path problems. *Mathematics of Operations Research*, 32(2):365–373, 2007.

[35] Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. Tuning bandit algorithms in stochastic environments. In *International conference on algorithmic learning theory*, pages 150–165. Springer, 2007.

[36] Andreas Maurer and Massimiliano Pontil. Empirical bernstein bounds and sample variance penalization. *arXiv preprint arXiv:0907.3740*, 2009.

[37] Dimitri P Bertsekas and John N Tsitsiklis. An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, 16(3):580–595, 1991.

[38] Ronan Fruit, Matteo Pirotta, and Alessandro Lazaric. Improved analysis of ucrl2 with empirical bernstein inequality. *arXiv preprint arXiv:2007.05456*, 2020.

[39] Abbas Kazerouni, Mohammad Ghavamzadeh, Yasin Abbasi, and Benjamin Van Roy. Conservative contextual linear bandits. In *Advances in Neural Information Processing Systems*, pages 3910–3919, 2017.

Appendix

A Autonomous Exploration Objectives

We recall the two AX objectives stated in Def. 5: for any length $L \geq 1$, error threshold $\varepsilon > 0$ and confidence level $\delta \in (0, 1)$, the sample complexities $C_{AX_L}(\mathfrak{A}, L, \varepsilon, \delta)$ and $C_{AX^\star}(\mathfrak{A}, L, \varepsilon, \delta)$ are defined as the number of time steps required by a learning algorithm $\mathfrak{A}$ to identify a set $\mathcal{K} \supseteq \mathcal{S}^{\rightarrow}_L$ such that with probability at least $1 - \delta$, it has learned a set of policies $\{\pi_s\}_{s \in \mathcal{K}}$ that respectively verifies the following AX requirement

$$(AX_L) \quad \forall s \in \mathcal{K}, v_{\pi_s}(s_0 \rightarrow s) \leq L + \varepsilon,$$
$$(AX^\star) \quad \forall s \in \mathcal{K}, v_{\pi_s}(s_0 \rightarrow s) \leq V^\star_{\mathcal{S}^{\rightarrow}_L}(s_0 \rightarrow s) + \varepsilon.$$

As we explain in Sect. 4, DisCo (Alg. 1) succeeds in tackling condition $AX^\star$, whereas UcbExplore [1], which is designed to tackle condition AX_L, is unable to tackle $AX^\star$. Note that the algorithmic design of UcbExplore entails that it computes policies whose value function implicitly targets $V^\star_{\mathcal{K}_t}$, with $\mathcal{K}_t$ the current set of controllable states. While $V^\star_{\mathcal{K}_t}$ is always smaller than L, UcbExplore cannot provide any tightness guarantees w.r.t. $V^\star_{\mathcal{K}_t}$ since it has no guarantee that the transition dynamics are estimated well enough on $\mathcal{K}_t$. An additional challenge with which UcbExplore fails to cope is the fact that the set $\mathcal{K}_t$ increases over time and thus unlocks new states and paths, which may be useful to improve its shortest-path policies for previously discovered states.

To better understand this phenomenon, let us introduce an alternative condition AX' — tighter than AX_L, but looser than $AX^\star$ — which stems from the challenge of not knowing $\mathcal{S}^{\rightarrow}_L$ in advance. We define AX' as follows: for any state s in $\mathcal{S}^{\rightarrow}_L$, the objective is to find a policy that can reach s from s_0 in at most $L' + \varepsilon$ steps on average, where $L' := \min\{l \leq L : s \in \mathcal{S}^{\rightarrow}_l\}$, i.e.,

$$(AX') \quad \forall s \in \mathcal{K}, v_{\pi_s}(s_0 \rightarrow s) \leq L' + \varepsilon, \text{ where } L' := \min\{l \leq L : s \in \mathcal{S}^{\rightarrow}_l\}.$$

As mentioned in [1, Corollary 9], it is possible to run separate instances of UcbExplore with increasing $L_n = 1 + n\varepsilon$ from $n = 0$ to $\lceil \frac{L-1}{\varepsilon} \rceil$ (i.e., until n satisfies $L_{n-1} \leq L \leq L_n$). This verifies the condition AX' at the cost of a worsened dependency on both ε and L as follows

$$C_{AX'}(\text{UcbExplore}, L, \varepsilon, \delta) = \widetilde{O}\left(\frac{L^7 S_{L+\varepsilon} A}{\varepsilon^4} \right).$$

While AX' is tighter than AX_L, it may be arbitrarily loose compared to $AX^\star$, which illustrates the intrinsic limitations in UcbExplore design. UcbExplore incrementally expands a set of “controllable” states $\mathcal{K}$: starting with $\mathcal{K}_0 = \{s_0\}$, at time t a state s is added to $\mathcal{K}_t$ whenever UcbExplore can confidently assess that it managed to learn a policy reaching s in less than L steps. Since at time t UcbExplore can only consider policies restricted to the controllable states $\mathcal{K}_t$, even the shortest-path policy computed to reach s at time t may not be ε-optimal w.r.t. to the *whole* set $\mathcal{S}^{\rightarrow}_L$. Indeed, every time a state is added to $\mathcal{K}_t$, this state may unlock new paths which may, for previously controllable states, allow for better shortest-path policies restricted on the updated $\mathcal{K}_t$. Fig. 3 illustrates this behavior, where the state y unlocks a fast path from y to x which should be taken in y instead of resetting to s_0. Consequently, if the agent seeks to tackle condition $AX^\star$, it must have the faculty to *backtrack*, i.e., continuously update both its belief of the vicinity ($\mathcal{K}$) and its notion of optimality on the vicinity ($V^\star_{\mathcal{K}}$). Unfortunately, UcbExplore can only compute policies targeting $V^\star_{\mathcal{K}}$ with $\mathcal{K}$ the current set of controllable states, but it fails to be accurate enough to devise such policies as the set of

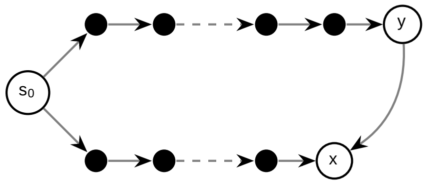


Figure 3: Let $\mathcal{X} := \{s_0\} \cup \{x\}$ and $\mathcal{Y} := \mathcal{X} \cup \{y\}$. For any $l \geq 1$, suppose that from s_0, the agent reaches x in l steps with probability $1/2$, or reaches y in $l + 1$ steps with probability $1/2$. If the goal state is x, constraining an agent to use policies restricted to $\mathcal{X}$ (i.e., that reset to s_0 outside of $\mathcal{X}$) is detrimental since x can actually be reached in 1 step from y. Formally, we can easily prove that $V^\star_{\mathcal{X}}(s_0 \rightarrow x) - V^\star_{\mathcal{Y}}(s_0 \rightarrow x) = l + 1$, which grows arbitrarily as l increases.

AX	UcbExplore [1]	DisCo (Alg. 1)
AX_L	$\tilde{O}\left(\frac{L^6 S_{L+\epsilon} A}{\epsilon^3}\right)$	$\tilde{O}\left(\frac{L^5 \Gamma_{L+\epsilon} S_{L+\epsilon} A}{\epsilon^2} + \frac{L^3 S_{L+\epsilon}^2 A}{\epsilon}\right)$
AX'	$\tilde{O}\left(\frac{L^7 S_{L+\epsilon} A}{\epsilon^4}\right)$	
$AX^\star$	Unable	

Table 1: Comparison between the sample complexity of UcbExplore and DisCo, depending on the condition AX_L, AX' or $AX^\star$.

controllable states $\mathcal{K}$ is expanded over time. In contrast, in virtue of its allocation function (Eq. 2) which enables to track the number of collected samples as $\mathcal{K}$ increases, DisCo is able to improve its candidate shortest-path policies during the consolidation step when the final set $\mathcal{K}$ is considered.

The following general and simple statement captures how the expansion of the state space of interest may alter and refine the optimal policy restricted on it.

Lemma 1. For any two sets $\mathcal{X} \subseteq \mathcal{Y}$ and any state $x \in \mathcal{X}$, we have $V^\star_{\mathcal{X}}(s_0 \to x) \geq V^\star_{\mathcal{Y}}(s_0 \to x)$. Moreover, the gap between the two quantities may be arbitrarily large.

Proof. The inequality is immediate from Asm. 1. Fig. 3 shows the gap may be arbitrarily large. □

Finally, we summarize all the sample complexity results in Tab. 1.

B Efficient Computation of Optimistic SSP Policy

In this section we recall from [27, 28] how to efficiently compute an optimistic stochastic shortest-path (SSP) policy.

B.1 Computation of Optimal Policy in Known SSP

This section details the procedure to efficiently compute an (arbitrarily near-) optimal policy in a known SSP instance with positive costs and which admits at least one proper policy. Recall that a proper policy is a policy whose execution starting from any non-goal state eventually reaches the goal state with probability one [26].

Definition 7 (SSP-MDP). An SSP-MDP is an MDP $M = \langle \mathcal{S}^\dagger, \mathcal{A}, s^\dagger, p, c \rangle$ where $\mathcal{S}^\dagger$ is the set of non-goal states with $|\mathcal{S}^\dagger| = S^\dagger$, $\mathcal{A}$ is the set of actions, p is the transition function and c is the cost function. The goal state $s^\dagger \notin \mathcal{S}^\dagger$ is zero-cost and absorbing, i.e., $p(s^\dagger|s^\dagger, a) = 1$ and $c(s^\dagger, a) = 0$ for any $a \in \mathcal{A}$.

The (possibly unbounded) value function (also called expected cost-to-go) of any policy $\pi \in \Pi$ starting from state s_0 is defined as

$$V^\pi(s_0) := \mathbb{E}\left[\sum_{t=1}^{+\infty} c(s_t, \pi(s_t)) \,\Big|\, s_0\right] = \mathbb{E}\left[\sum_{t=1}^{\tau_\pi(s_0 \to s^\dagger)} c(s_t, \pi(s_t)) \,\Big|\, s_0\right].$$

Assumption 2. We restrict the attention to SSP-MDP M (see Def. 7) such that, for any $(s, a) \in \mathcal{S}^\dagger \times \mathcal{A}$, $c(s, a) \in [c_{\min}, 1]$ with $c_{\min} > 0$. (Note that having positive costs ensures that for any non-proper policy π there exists a state s with $V^\pi(s) = +\infty$.) Moreover, we assume that there exists at least one proper policy (i.e., that reaches the goal state $s^\dagger$ with probability one starting from any state in $\mathcal{S}^\dagger$).

The procedure VI_{SSP} considers the following inputs: a goal $s^\dagger$, non-goal states $\mathcal{S}^\dagger$, a known model p and a known cost function c, with (non-goal) costs lower bounded by $c_{\min} > 0$. VI_{SSP} outputs a vector u (of size $|\mathcal{S}^\dagger|$) and a policy π which is greedy w.r.t. the vector u.

The optimal Bellman operator is defined as follows for any vector u and any non-goal state $s \in \mathcal{S}^\dagger$

$$\mathcal{L}u(s) := \min_{a \in \mathcal{A}} \left\{ c(s, a) + \sum_{s' \in \mathcal{S}^\dagger} p(s'|s, a) u(s') \right\}.$$

Algorithm 2: VI_{SSP}

```
Input: Non-goal states $S^\dagger$, action set $A$, transitions $p$, costs $c$ and accuracy $\epsilon$
Output: Value vector $u$ and greedy policy $\pi$
1 Define $Lu(s) := \min_{a\in A}\left\{ c(s,a) + \sum_{s'\in S^\dagger} p(s'|s,a)u(s') \right\}$
2 Set $u_0 = 0_{S^\dagger}$ and $j = 0$
3 $u_1 = Lu_0$
4 while $\|u_{j+1} - u_j\|_\infty > \epsilon$ do
5     $u_{j+1} = Lu_j$
6 Set $u := u_j$ and $\pi(s) \in \arg\min_{a\in A}\left\{ c(s,a) + \sum_{s'\in S^\dagger} p(s'|s,a)u(s') \right\}$ for any $s \in S^\dagger \cup \{s^\dagger\}$
```

Note that by definition, $V^\pi(s^\dagger) = 0$ for any π. We perform a value iteration (VI) scheme over this operator as explained in [e.g., 29, 34, 27]. Namely, we consider initial vector $u_0 := 0$ and set iteratively $u_{i+1} := Lu_i$ (see Alg. 2). For a predefined VI precision $\epsilon > 0$, the stopping condition is reached for the first iteration j such that $\|u_{j+1} - u_j\|_\infty \leq \epsilon$. The policy is then selected to be the greedy policy w.r.t. the vector $u := u_j$, i.e.,

$$\forall s \in S^\dagger \cup \{s^\dagger\}, \quad \pi(s) \in \arg\min_{a\in A}\left\{ c(s,a) + \sum_{s'\in S^\dagger} p(s'|s,a)u(s') \right\}. \tag{7}$$

Importantly, while u is not the value function of π, both quantities can be related according to the following lemma.

Lemma 2. *Consider an SSP-MDP $M = \langle S^\dagger, A, s^\dagger, p, c\rangle$ defined as in Def. 7 and satisfying Asm. 2. Let $(u, \pi) = \text{VI}_{\text{SSP}}(S^\dagger, A, p, c, \epsilon)$ be the solution computed by VI_{SSP}. Denote by V^π the true value function of π and by $V^\star = V^{\pi^\star} = LV^\star$ the optimal value function. The following component-wise inequalities hold*

- $u \leq V^\star \leq V^\pi$.
- *If the VI precision level verifies $\epsilon \leq \frac{c_{\min}}{2}$, then $V^\pi \leq \left(1 + \frac{2\epsilon}{c_{\min}}\right) u$.*

Proof. The result can be obtained by adapting [27, Lem. 4 & App. E]. For the first inequality, given that we consider the initial vector $u_0 = 0$, we know that $0 \leq V^\star$ with $V^\star = LV^\star$ by definition. By monotonicity of the operator L [25, 26], we obtain $u_j \leq V^\star \leq V^\pi$. As for the second inequality, we introduce the following Bellman operators of a deterministic policy π for any vector u and state s,

$$L^\pi u(s) := c(s,\pi(s)) + \sum_{s'\in S} p(s'|s,\pi(s))u(s'),$$

$$T^\pi u(s) := \underbrace{c(s,\pi(s)) - \epsilon}_{>0} + \sum_{s'\in S} p(s'|s,\pi(s))u(s').$$

Note that the SSP problem defined by the operator T^π satisfies Asm. 2 since i) it has positive costs due to the condition $\epsilon \leq \frac{c_{\min}}{2}$ and ii) the fact that M satisfies Asm. 2 guarantees the existence of at least one proper policy in the model p. We can write component-wise

$$T^\pi u_j = L^\pi u_j - \epsilon \overset{(a)}{=} Lu_j - \epsilon \overset{(b)}{\leq} u_j,$$

where (a) uses that π is the greedy policy w.r.t. u_j and (b) stems from the chosen stopping condition which yields $Lu_j \leq u_j + \epsilon$. By monotonicity of the operator T^π, we have for all $m > 0$, $(T^\pi)^m u_j \leq u_j$. The asymptotic convergence of the operator in an SSP problem satisfying Asm. 2 (see e.g., [26, Prop. 2.2.1]) guarantees that taking the limit $m \to +\infty$ yields $W^\pi_\epsilon \leq u_j$, where W^π_ϵ is defined as the value function of policy π in the model p with ϵ subtracted to all the costs, i.e.,

$$W^\pi_\epsilon(s) := \mathbb{E}\left[\sum_{t=1}^{\tau_\pi(s)} (c(s_t, \pi(s_t)) - \epsilon) \,\Big|\, s_1 = s\right] = V^\pi(s) - \epsilon\,\mathbb{E}[\tau_\pi(s)],$$

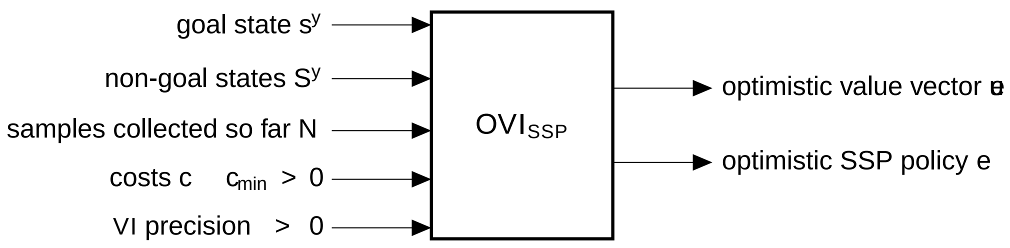

goal state $s^\dagger$
non-goal states $S^\dagger$
samples collected so far N
costs $c \ge c_{\min} > 0$
VI precision $\epsilon_{VI} > 0$
OVI_{SSP}
optimistic value vector $\tilde u$
optimistic SSP policy $\tilde\pi$

Figure 4: Optimistic Value Iteration for SSP (OVI_{SSP}).

where $\tau_\pi(s)$ denotes the (random) hitting time of policy π to reach the goal starting from state s. Moreover, we have $c_{\min}\mathbb{E}[\tau_\pi(s)] \le V^\pi(s) \le c_{\max}\mathbb{E}[\tau_\pi(s)]$. Putting everything together, we thus get $\left(1 - \frac{\epsilon}{c_{\min}}\right) V^\pi \le u_j$. Since $\epsilon \le \frac{c_{\min}}{2}$, we ultimately obtain

$$V^\pi \le \frac{1}{1 - \frac{\epsilon}{c_{\min}}} u_j \le \left(1 + \frac{2\epsilon}{c_{\min}}\right) u_j ,$$

where the last inequality uses the fact that $\frac{1}{1-x} \le 1 + 2x$ holds for any $0 \le x \le \frac{1}{2}$. □

B.2 Computation of Optimistic Model in Unknown SSP

Consider an SSP problem M defined as in Asm. 2. Consider that, at any given stage of the learning process, the agent is equipped with $N(s,a)$ samples at each state-action pair. A method to compute an optimistic model $\tilde p$ is provided in [28], which we recall below.

Denote by $\hat p$ the current empirical average of transitions: $\hat p(s'|s,a) = N(s,a,s')/N(s,a)$, and set $\hat\sigma^2(s'|s,a) := \hat p(s'|s,a)(1 - \hat p(s'|s,a))$ as well as $N^+(s,a) := \max\{1, N(s,a)\}$. For any $(s,a,s') \in S^\dagger \times A \times S^\dagger$, the empirical Bernstein inequality [35, 36] is leveraged to select the following confidence intervals (with probability at least $1-\delta$) on the transition probabilities

$$\beta(s,a,s') := 2\sqrt{\frac{\hat\sigma^2(s'|s,a)}{N^+(s,a)} \log\left(\frac{2SAN^+(s,a)}{\delta}\right)} + \frac{6\log\left(\frac{2SAN^+(s,a)}{\delta}\right)}{N^+(s,a)} ,$$

and $\beta(s,a,s^\dagger) := \sum_{s' \in S^\dagger} \beta(s,a,s')$. The selection of the optimistic model $\tilde p$ is as follows: the probability of reaching the goal $s^\dagger$ is maximized at every state-action pair, which implies minimizing the probability of reaching all other states and setting them at the lowest value of their confidence range. Formally, we set for all $(s,a,s') \in S^\dagger \times A \times S^\dagger$,

$$\tilde p(s'|s,a) := \max\left\{\hat p(s'|s,a) - \beta(s,a,s'),\ 0\right\} ,$$

and $\tilde p(s^\dagger|s,a) := 1 - \sum_{s' \in S^\dagger} \tilde p(s'|s,a)$.

B.3 Combining the two: Optimistic Value Iteration for SSP (OVI_{SSP})

OVI_{SSP} first computes an optimistic model $\tilde p$ leveraging App. B.2, and it then runs the VI_{SSP} procedure of App. B.1 in the model $\tilde p$, i.e., $(\tilde u, \tilde\pi) = \text{VI}_{\text{SSP}}(S^\dagger, A, s^\dagger, \tilde p, c, \epsilon_{VI})$. This outputs an optimistic pair $(\tilde u, \tilde\pi)$ composed of the VI vector $\tilde u$ and the policy $\tilde\pi$ that is greedy w.r.t. $\tilde u$ in the model $\tilde p$. The OVI_{SSP} scheme is recapped in Fig. 4.

C Useful Result: Simulation Lemma for SSP

Consider a stochastic shortest-path (SSP) instance (see Def. 7) that satisfies Asm. 2. We denote by $A = |\mathcal{A}|$ the number of actions, $S = |\mathcal{S}|$ the number of non-goal states, $g \notin \mathcal{S}$ the (zero-cost and absorbing) goal state, p the unknown transitions and c the known cost function. We assume that $0 < c(s,a) \le 1$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$, and set $c_{\min} := \min_{s,a} c(s,a) > 0$. We also set $\mathcal{S}' := \mathcal{S} \cup \{g\}$.

Recall that the goal state is zero-cost (i.e., $c(g, a) = 0$) and absorbing (i.e., $p(g|g, a) = 1$), and that the value function of a policy amounts to the expected cumulative costs following this policy until reaching the goal.

Definition 8. For any model p and $\eta > 0$, we introduce the set of models close to p w.r.t. the ℓ_1-norm on the non-goal states as follows

$$\mathcal{P}^{(p)}_{\eta} := \Big\{ p' \in \mathbb{R}^{S' \times A \times S'} : \forall (s,a) \in \mathcal{S} \times \mathcal{A},\ p'(\cdot|s,a) \in \Delta(\mathcal{S}'),\ p(g|g,a) = 1,\ \sum_{y \in \mathcal{S}} |p(y|s,a) - p'(y|s,a)| \leq \eta \Big\}.$$

Lemma 3 (Simulation Lemma for SSP)**.** Consider any model p and $p' \in \mathcal{P}^{(p)}_{\eta}$ such that, for each model, there exists at least one proper policy w.r.t. the goal state g. Consider any policy π that is proper in p', with value function denoted by V'^{π}, such that the following condition is verified

$$\eta \|V'^{\pi}\|_{\infty} \leq \frac{c_{\min}}{2}. \tag{8}$$

Then π is proper in p (i.e., its value function verifies $V^{\pi} < +\infty$ component-wise), and we have

$$\forall s \neq g,\ V^{\pi}(s) \leq \Big(1 + \frac{2\eta \|V'^{\pi}\|_{\infty}}{c_{\min}}\Big) V'^{\pi}(s),$$

and conversely,

$$\forall s \neq g,\ V'^{\pi}(s) \leq \Big(1 + \frac{\eta \|V'^{\pi}\|_{\infty}}{c_{\min}}\Big) V^{\pi}(s).$$

Combining the two inequalities above yields

$$\|V^{\pi} - V'^{\pi}\|_{\infty} \leq \frac{7\eta \|V'^{\pi}\|^2_{\infty}}{c_{\min}}.$$

Proof. The proof of Lem. 3 requires a result of [37] recalled in Lem. 4 and can be seen as a generalization of [28, Lem. B.4]. First, let us assume that π is proper in the model p'. This implies that its value function, denoted by V'^{π}, is bounded component-wise. Moreover, for any non-goal state $s \in \mathcal{S}$, the Bellman equation holds as follows

$$\begin{aligned} V'^{\pi}(s) &= c(s, \pi(s)) + \sum_{y \in \mathcal{S}} p'(y|s, \pi(s)) V'^{\pi}(y) \\ &= c(s, \pi(s)) + \sum_{y \in \mathcal{S}} p(y|s, \pi(s)) V'^{\pi}(y) + \sum_{y \in \mathcal{S}} \big(p'(y|s, \pi(s)) - p(y|s, \pi(s))\big) V'^{\pi}(y). \end{aligned} \tag{9}$$

By successively using Hölder's inequality and the facts that $p' \in \mathcal{P}^{(p)}_{\eta}$ and $c(s, \pi(s)) \geq c_{\min}$, we get

$$V'^{\pi}(s) \geq c(s, \pi(s)) - \eta \|V'^{\pi}\|_{\infty} + p(\cdot|s, \pi(s))^{\top} V'^{\pi} \geq c(s, \pi(s)) \Big(1 - \frac{\eta \|V'^{\pi}\|_{\infty}}{c_{\min}}\Big) + p(\cdot|s, \pi(s))^{\top} V'^{\pi}.$$

Let us now introduce the vector $V''^{\pi} := \Big(1 - \frac{\eta \|V'^{\pi}\|_{\infty}}{c_{\min}}\Big)^{-1} V'^{\pi}$. Then for all $s \in \mathcal{S}$,

$$V''^{\pi}(s) \geq c(s, \pi(s)) + p(\cdot|s, \pi(s))^{\top} V''^{\pi}.$$

Hence, from Lem. 4, π is proper in p (i.e., $V^{\pi} < +\infty$), and we have

$$V^{\pi} \leq V''^{\pi} \leq \Big(1 + 2\eta \frac{\|V'^{\pi}\|_{\infty}}{c_{\min}}\Big) V'^{\pi}, \tag{10}$$

where the last inequality stems from condition (8) and the fact that $\frac{1}{1-x} \leq 1 + 2x$ holds for any $0 \leq x \leq \frac{1}{2}$. Conversely, analyzing Eq. 9 from the other side, we get

$$V'^{\pi}(s) \leq c(s, \pi(s)) \Big(1 + \frac{\eta \|V'^{\pi}\|_{\infty}}{c_{\min}}\Big) + p(\cdot|s, \pi(s))^{\top} V'^{\pi}.$$

Let us now introduce the vector $V'' := \left(1 + \frac{\|V'\|_\infty}{c_{\min}}\right)^{-1} V'$. Then

$$V''(s) \leq c(s, \pi(s)) + p(\cdot|s, \pi(s))^\top V''.$$

We then obtain in the same vein as Lem. 4 (by leveraging the monotonicity of the Bellman operator $\mathcal{L}_\pi U(s) := c(s, \pi(s)) + p(\cdot|s, \pi(s))^\top U$) that $V'' \leq V_\pi$, and therefore

$$V' \leq \left(1 + \frac{\|V'\|_\infty}{c_{\min}}\right) V. \tag{11}$$

Combining Eq. 10 and 11 yields component-wise

$$\|V - V'\|_\infty \leq 2\frac{\|V'\|_\infty}{c_{\min}}\|V'\|_\infty + \frac{\|V'\|_\infty}{c_{\min}}\|V\|_\infty \leq 7\frac{\|V'\|_\infty^2}{c_{\min}},$$

where the last inequality uses that $\|V\|_\infty \leq 5\|V'\|_\infty$ which stems from plugging condition (8) into Eq. 10.

Note that here p and p' play symmetric roles; we can perform the same reasoning in the case where π is proper in the model p' and it would yield an equivalent result by switching the dependencies on V and V'. $\square$

Lemma 4 ([37], Lem. 1)**.** In an SSP-MDP satisfying Asm. 2, let π be any policy, then

- If there exists a vector $U : S \to \mathbb{R}$ such that $U(s) \geq c(s, \pi(s)) + \sum_{s' \in S} p(s'|s, \pi(s)) U(s')$ for all $s \in S$, then π is proper, and V^π the value function of π is upper bounded by U component-wise, i.e., $V^\pi(s) \leq U(s)$ for all $s \in S$.
- If π is proper, then its value function V^π is the unique solution to the Bellman equations $V^\pi(s) = c(s, \pi(s)) + \sum_{s' \in S} p(s'|s, \pi(s)) V^\pi(s')$ for all $s \in S$.

D Proof of Theorem 1 (Sample Complexity Analysis of DisCo)

D.1 Computation of the Optimistic Policies

At each round k, for each goal state $s^\dagger \in \mathcal{W}_k$, DisCo computes an optimistic goal-oriented policy associated to the MDP $M_k'(s^\dagger)$ constructed as in Def. 6. This MDP is defined over the entire state space S and restricts the action to the only action RESET outside $\mathcal{K}_k$. We can build an equivalent MDP by restricting the focus on $\mathcal{K}_k$. To this end, we define the following SSP-MDP.

Definition 9. Define $M_k^\dagger(s^\dagger) := \langle \mathcal{S}_k^\dagger, \mathcal{A}_k^\dagger(\cdot), c_k^\dagger, p_k^\dagger \rangle$ where $\mathcal{S}_k^\dagger := \mathcal{K}_k \cup \{s^\dagger, x\}$ and $S_k^\dagger = |\mathcal{S}_k^\dagger| = |\mathcal{K}_k| + 2$. State x is a meta-state that encapsulates all the states that have been observed so far and are not in $\mathcal{K}_k$. The action space $\mathcal{A}_k^\dagger(\cdot)$ is such that $\mathcal{A}_k^\dagger(s) = \mathcal{A}$ for all states $s \in \mathcal{K}_k$ and $\mathcal{A}_k^\dagger(s) = \{\text{RESET}\}$ for $s \in \{s^\dagger, x\}$. The cost function is $c_k^\dagger(x, a) = 0$ for any $a \in \mathcal{A}_k^\dagger(x)$ and $c_k^\dagger(s, a) = 1$ everywhere else. The transition function is defined as $p_k^\dagger(s^\dagger|s^\dagger, a) = p_k^\dagger(s_0|x, a) = 1$ for any a, $p_k^\dagger(y|s, a) = p(y|s, a)$ for any $(s, a, y) \in \mathcal{K}_k \times \mathcal{A} \times (\mathcal{K}_k \cup \{s^\dagger\})$ and $p_k^\dagger(x|s, a) = 1 - \sum_{y \in \mathcal{K}_k \cup \{s^\dagger\}} p_k^\dagger(y|s, a)$.

Note that solving $M_k^\dagger$ yields a policy effectively restricted to the set $\mathcal{K}_k$ insofar as we can interpret the meta-state x as $S \setminus \{\mathcal{K}_k \cup \{s^\dagger\}\}$. Since p is unknown, we cannot construct $M_k^\dagger(s^\dagger)$. Let N_k be the state-action counts accumulated up until now. We denote by $\hat{p}_k$ the "global" empirical estimates, i.e., $\hat{p}_k(y|s, a) = N_k(s, a, y)/N_k(s, a)$. Given them, we define the "restricted" empirical estimates $\hat{p}_k^\dagger$ as follows: $\hat{p}_k^\dagger(y|s, a) := \hat{p}_k(y|s, a)$ for any $(s, a, y) \in \mathcal{K}_k \times \mathcal{A} \times (\mathcal{K}_k \cup \{s^\dagger\})$ and $\hat{p}_k^\dagger(x|s, a) := 1 - \sum_{y \in \mathcal{K}_k \cup \{s^\dagger\}} \hat{p}_k^\dagger(y|s, a)$. Denoting $N_k^+(s, a) := \max\{1, N_k(s, a)\}$, we then define the following bonuses for any $(s, a, y) \in \mathcal{K}_k \times \mathcal{A} \times (\mathcal{K}_k \cup \{s^\dagger\})$,

$$\beta_k(s, a, y) := 2\sqrt{\frac{\hat{p}_k(y|s, a)(1 - \hat{p}_k(y|s, a))}{N_k^+(s, a)} \log\left(\frac{2SAN_k^+(s, a)}{\delta}\right)} + \frac{6\log\left(\frac{2SAN_k^+(s, a)}{\delta}\right)}{N_k^+(s, a)}, \tag{12}$$

$$\beta_k(s, a, x) := \sum_{y \in \mathcal{K}_k \cup \{s^\dagger\}} \beta_k(s, a, y). \tag{13}$$

```
Algorithm 3: OVI_SSP
Input: $\mathcal{K}_k, \mathcal{A}, s^\dagger, N_k, \varepsilon > 0$
Output: Value vector $\widetilde{u}^\dagger$ and policy $\widetilde{\pi}^\dagger$
1 Estimate transitions probabilities $\widehat{p}_k$ using $N_k$
2 Compute the optimistic SSP-MDP $\widetilde{M}_k^\dagger$ as detailed in Def. 10
3 Compute $(\widetilde{u}_k^\dagger, \widetilde{\pi}_k^\dagger) = \text{VI}_{\text{SSP}}(\mathcal{S}_k^\dagger, \mathcal{A}_k^\dagger, c_k^\dagger, \widetilde{p}_k^\dagger, \varepsilon)$ (see Alg. 2)
```

Moreover, we set the uncertainty about the MDP at the meta-state x and at the goal state $s^\dagger$ to 0 by construction (since their outgoing transitions are deterministic, respectively to s_0 and $s^\dagger$).

We now leverage the optimistic construction mentioned in App. B.1.

Definition 10. We denote by $\widetilde{M}_k^\dagger(s^\dagger) = \langle \mathcal{S}_k^\dagger, \mathcal{A}_k^\dagger(\cdot), c_k^\dagger, \widetilde{p}_k^\dagger \rangle$ the optimistic MDP associated to $M_k^\dagger(s^\dagger)$ defined in Def. 9. Then $\forall (s,a) \in \mathcal{K}_k \times \mathcal{A}$,

$$\widetilde{p}_k^\dagger(y|s,a) := \max\{\widehat{p}_k(y|s,a) - \beta_k(s,a,y), 0\}, \quad \forall y \in \mathcal{K}_k \cup \{x\}; \tag{14}$$

$$\widetilde{p}_k^\dagger(s^\dagger|s,a) := 1 - \sum_{y \in \mathcal{K}_k \cup \{x\}} \widetilde{p}_k^\dagger(y|s,a); \tag{15}$$

$$\widetilde{p}_k^\dagger(s^\dagger|s^\dagger,a) = \widetilde{p}_k^\dagger(s_0|x,a) = 1. \tag{16}$$

Given this MDP, we can compute the optimistic value vector $\widetilde{u}_k^\dagger$ and policy $\widetilde{\pi}_k^\dagger$ using value iteration for SSP: $(\widetilde{u}_k^\dagger, \widetilde{\pi}_k^\dagger) = \text{VI}_{\text{SSP}}(\mathcal{S}_k^\dagger, \mathcal{A}_k^\dagger, c_k^\dagger, \widetilde{p}_k^\dagger, \frac{\varepsilon}{4L})$. We summarize the construction of the optimistic model and the computation of value function and policy in Alg. 3 (OVI_{SSP}).

Remark. Note that the structure of the problem does not appear to allow for variance-aware improvements in the analysis of Thm. 1 (specifically, when the analysis will apply an SSP simulation lemma argument). Indeed, given the possibly large number of states in the total environment $\mathcal{S}$, the computation of the optimistic policies requires the construction of the meta-state x that encapsulates all the states in $\mathcal{S} \setminus \{\mathcal{K}_k \cup \{s^\dagger\}\}$, where $s^\dagger$ is the candidate goal state considered at round k. As a result, the uncertainty on the transitions reaching x needs to be summed over multiple states, as shown in Eq. 13. This extra uncertainty at a single state in the induced MDP has the effect of canceling out Bernstein techniques seeking to lower the prescribed requirement of the state-action samples that the algorithm should collect. In turn this implies that such variance-aware techniques would not lead to any improvement in the final sample complexity bound.

D.2 High-Probability Event

Lemma 5. It holds with probability at least $1-\delta$ that for any time step $t \geq 1$ and for any state-action pair (s,a) and next state s',

$$|\widehat{p}_t(s'|s,a) - p(s'|s,a)| \leq 2\sqrt{\frac{\widehat{\sigma}_t^2(s'|s,a)}{N_t^+(s,a)} \log\left(\frac{2SAN_t^+(s,a)}{\delta}\right)} + \frac{6\log\left(\frac{2SAN_t^+(s,a)}{\delta}\right)}{N_t^+(s,a)}, \tag{17}$$

where $N_t^+(s,a) := \max\{1, N_t(s,a)\}$ and where $\widehat{\sigma}_t^2$ are the population variance of transitions, i.e., $\widehat{\sigma}_t^2(s'|s,a) := \widehat{p}_t(s'|s,a)(1 - \widehat{p}_t(s'|s,a))$.

Proof. The confidence intervals in Eq. 17 are constructed using the empirical Bernstein inequality, which guarantees that the considered event holds with probability at least $1-\delta$, see e.g., [38]. □

Define the set of plausible transition probabilities as

$$C_k^\dagger := \bigcap_{(s,a) \in \mathcal{S}_k^\dagger \times \mathcal{A}} C_k^\dagger(s,a),$$

where

$C_k^\dagger(s,a) := \{\tilde{p} \in \mathcal{C} \mid \tilde{p}(\cdot \mid s^\dagger, a) = 1_{s^\dagger}, \tilde{p}(\cdot \mid x, a) = 1_{s_0}, |\tilde{p}(s' \mid s,a) - \hat{p}_k(s' \mid s,a)| \leq \beta_k(s,a,s')\},$

with $\mathcal{C}$ the $S_k^\dagger$-dimensional simplex and $\hat{p}_k$ the empirical average of transitions.

Lemma 6. Introduce the event $\mathcal{E} := \bigcap_{k=1}^{K+1} \bigcap_{s^\dagger \in \mathcal{W}_k} \{p_k^\dagger \in C_k^\dagger\}$. Then $\mathbb{P}(\mathcal{E}) \geq 1 - \frac{\delta}{3}$.

Proof. We have with probability at least $1 - \frac{\delta}{3}$ that, for any $y \neq x$, $|p_k^\dagger(y|s,a) - \hat{p}_k^\dagger(y|s,a)| \leq \beta_k(s,a,y)$ from the empirical Bernstein inequality (see Eq. 17), and moreover $|\hat{p}_k^\dagger(x|s,a) - p_k^\dagger(x|s,a)| = |1 - \sum_{y \in \mathcal{K}_k \cup \{s^\dagger\}} p_k^\dagger(y|s,a) - 1 + \sum_{y \in \mathcal{K}_k \cup \{s^\dagger\}} \hat{p}_k^\dagger(y|s,a)| \leq \sum_{y \in \mathcal{K}_k \cup \{s^\dagger\}} |p_k^\dagger(y|s,a) - \hat{p}_k^\dagger(y|s,a)| \leq \beta_k(s,a,x)$. □

Lemma 7. Under the event $\mathcal{E}$, for any round k and any goal state $s^\dagger \in \mathcal{W}_k$, the optimistic model $\tilde{p}_k^\dagger$ constructed in Def. 10 verifies $\tilde{p}_k^\dagger \in \mathcal{P}_{\eta_k}^{(p_k^\dagger)}$, with $\eta_k := 4\beta_k(s,a,x)$ where β_k is defined in Eq. 13.

Proof. Combining the construction in Def. 10, the proof of Lem. 6 and the triangle inequality yields

$$\begin{aligned}\sum_{y \in \mathcal{K}_k \cup \{x\}} |\tilde{p}_k^\dagger(y|s,a) - p_k^\dagger(y|s,a)| &\leq \sum_{y \in \mathcal{K}_k \cup \{x\}} |\tilde{p}_k^\dagger(y|s,a) - \hat{p}_k^\dagger(y|s,a)| + |\hat{p}_k^\dagger(y|s,a) - p_k^\dagger(y|s,a)| \\ &\leq \sum_{y \in \mathcal{K}_k \cup \{x\}} \beta_k(s,a,y) + 2\beta_k(s,a,x) \\ &\leq 4\beta_k(s,a,x).\end{aligned}$$

□

Throughout the remainder of the proof, we assume that the event $\mathcal{E}$ holds.

D.3 Properties of the Optimistic Policies and Value Vectors

We recall notation. Let us fix any round k and any goal state $s^\dagger \in \mathcal{W}_k$. We denote by $\tilde{\pi}_k^\dagger$ the greedy policy w.r.t. $\tilde{v}_k^\dagger(\cdot \rightarrow s^\dagger)$ in the optimistic model $\tilde{p}_k^\dagger$. Let $v_k^\dagger(s \rightarrow s^\dagger)$ be the value function of policy $\tilde{\pi}_k^\dagger$ starting from state s in the model $p_k^\dagger$. We can apply Lem. 2 given that the conditions of Asm. 2 hold (indeed, we have $c_{\min} = 1 > 0$ and there exists at least one proper policy to reach the goal state $s^\dagger$ since it belongs to $\mathcal{W}_k$). Moreover, we have that $\tilde{v}_{\mathcal{K}_k}^\star(s_0 \rightarrow s^\dagger) \leq V_{\mathcal{K}_k}^\star(s_0 \rightarrow s^\dagger)$ given the way the optimistic model $\tilde{p}_k^\dagger$ is computed (i.e., by maximizing the probability of transitioning to the goal at any state-action pair), see [28, Lem. B.12]. Hence we get the two following important properties.

Lemma 8. For any round k, goal state $s^\dagger \in \mathcal{W}_k$ and state $s \in \mathcal{K}_k \cup \{x\}$, we have under the event $\mathcal{E}$,

$$\tilde{v}_k^\dagger(s \rightarrow s^\dagger) \leq V_{\mathcal{K}_k}^\star(s \rightarrow s^\dagger).$$

Lemma 9. For any round k, goal state $s^\dagger \in \mathcal{W}_k$ and state $s \in \mathcal{K}_k \cup \{x\}$, we have

$$v_k^\dagger(s \rightarrow s^\dagger) \leq (1 + 2\epsilon)\tilde{v}_k^\dagger(s \rightarrow s^\dagger).$$

D.4 State Transfer from $\mathcal{U}$ to $\mathcal{K}$ (step ¯)

We fix any round k and any goal state $s^\dagger \in \mathcal{W}_k$ that is added to the set of "controllable" states $\mathcal{K}$, i.e., for which $\tilde{v}_k^\dagger(s_0 \rightarrow s^\dagger) \leq L$.

Lemma 10. Under the event $\mathcal{E}$, we have both following inequalities

$$\begin{cases} v_k^\dagger(s_0 \rightarrow s^\dagger) \leq L + \varepsilon; \\ v_k^\dagger(s_0 \rightarrow s^\dagger) \leq V_{\mathcal{K}_k}^\star(s_0 \rightarrow s^\dagger) + \varepsilon. \end{cases}$$

In particular, the first inequality entails that $s^\dagger \in \mathcal{S}_{L+\varepsilon}^{\rightarrow}$, which justifies the validity of the state transfer from $\mathcal{U}$ to $\mathcal{K}$.

Proof. We have

$$\widetilde{v}_k^{\dagger}(s_0 \to s^{\dagger}) \overset{(a)}{\le} (1+2\gamma)\widetilde{u}_k^{\dagger}(s_0 \to s^{\dagger}) \le \left\{ \begin{array}{l} \overset{(b)}{\le} L + \frac{\varepsilon}{3} \\ \overset{(c)}{\le} V^{\star}_{\mathcal{K}_k}(s_0 \to s^{\dagger}) + \frac{\varepsilon}{3}; \end{array} \right. \tag{18}$$

where inequality (a) comes from Lem. 9, inequality (b) combines the algorithmic condition $\widetilde{u}_k^{\dagger}(s_0 \to s^{\dagger}) \le L$ and the VI precision level $\gamma := \frac{\varepsilon}{6L}$, and finally inequality (c) combines Lem. 8 and the VI precision level. Moreover, for any state $s \in \mathcal{K}_k$,

$$\widetilde{v}_k^{\dagger}(s \to s^{\dagger}) \overset{(a)}{\le} V^{\star}_{\mathcal{K}_k}(s \to s^{\dagger}) + \frac{\varepsilon}{3} \overset{(b)}{\le} V^{\star}_{\mathcal{K}_k}(s_0 \to s^{\dagger}) + 1 + \frac{\varepsilon}{3} \le \widetilde{v}_k^{\dagger}(s_0 \to s^{\dagger}) + 1 + \frac{\varepsilon}{3},$$

where (a) comes from Lem. 8 and (b) stems from the presence of the RESET action (Asm. 1).

We now provide the exact choice of allocation function ϕ in Alg. 1. We introduce

$$\psi := \frac{2\varepsilon}{12(L+1+\varepsilon)(L+\frac{\varepsilon}{3})}.$$

(Note that $\psi = O(\varepsilon/L^2)$.) We set the following requirement of samples for each state-action pair (s,a) at round k,

$$n_k = \phi(\mathcal{K}_k) = \left\lceil \frac{57 X_k^2}{\psi^2} \log\left(\frac{8eX_k}{\sqrt{\psi}}\sqrt{\frac{2SA}{\delta}}\right)^2 + \frac{24|\mathcal{S}_k^{\dagger}|}{\psi} \log\left(\frac{24|\mathcal{S}_k^{\dagger}|SA}{\psi\delta}\right) \right\rceil, \tag{19}$$

where we define

$$X_k := \max_{(s,a) \in \mathcal{S}_k^{\dagger} \times \mathcal{A}} \sum_{s' \in \mathcal{S}_k^{\dagger}} \sqrt{\widehat{\sigma}_k^2(s'|s,a)},$$

with $\widehat{\sigma}_k^2(s'|s,a) := \widehat{p}_k^{\dagger}(s'|s,a)(1-\widehat{p}_k^{\dagger}(s'|s,a))$ the estimated variance of the transition from (s,a) to s'. Leveraging the empirical Bernstein inequality (Lem. 5) and perfoming simple algebraic manipulations (see e.g., [39, Lem. 8 and 9]) yields that $\epsilon_k(s,a,x) \le \psi$. From Lem. 7, this implies that $\widetilde{p}_k^{\dagger} \in \mathcal{P}^{(\widehat{p}_k^{\dagger})}_{\eta}$ with $\eta := 4\psi$. We can then apply Lem. 3 (whose condition 8 is verified), which gives

$$\begin{aligned} v_k^{\dagger}(s_0 \to s^{\dagger}) &\le \left(1 + \eta\|\widetilde{v}_k^{\dagger}(\cdot \to s^{\dagger})\|_{\infty}\right)\widetilde{v}_k^{\dagger}(s_0 \to s^{\dagger}) \qquad (20) \\ &\le (1 + \eta(L+1+\varepsilon))\widetilde{v}_k^{\dagger}(s_0 \to s^{\dagger}) \\ &\le \widetilde{v}_k^{\dagger}(s_0 \to s^{\dagger}) + \frac{2\varepsilon}{3}, \end{aligned}$$

where the last inequality uses that $\eta(L+1+\varepsilon)(L+\frac{\varepsilon}{3}) = \frac{2\varepsilon}{3}$ by definition of η. Plugging in Eq. 18 yields the sought-after inequalities.

$\square$

D.5 Termination of the Algorithm

Lemma 11 (Variant of Lem. 17 of [1])**.** *Suppose that for every state $s \in \mathcal{S}$, each action $a \in \mathcal{A}$ is executed $b \ge \lceil L \log\left(\frac{3ALS}{\delta}\right)\rceil$ times. Let $\mathcal{S}'_{s,a}$ be the set of all next states visited during the b executions of (s,a). Denote by Ω the complementary of the event*

$$\left\{\exists (s', s, a) \in \mathcal{S}^2 \times \mathcal{A} : p(s'|s,a) \ge \frac{1}{L} \wedge s' \notin \mathcal{S}'_{s,a}\right\}.$$

Then $\mathbb{P}(\Omega) \ge 1 - \frac{\delta}{3}$.

Lemma 12. *Under the event $\Omega \cap \Theta$, for any round k, either $\mathcal{S}_L^{\to} \subseteq \mathcal{K}_k$, or there exists a state $s^{\dagger} \in \mathcal{S}_L^{\to} \setminus \mathcal{K}_k$ such that $s^{\dagger} \in \mathcal{W}_k$ and is L-controllable with a policy restricted to $\mathcal{K}_k$. Moreover, $|\mathcal{W}_k| \le 2LA|\mathcal{K}_k|$.*

Proof of Lem. 12. Consider a round k such that $S_L^{\to} \setminus K_k$ is non-empty. Due to the incremental construction of the set $S_L^{\to}$ (Def. 4), there exists a state $s^{y} \in S_L^{\to}$ and a policy restricted to K_k that can reach s^{y} in at most L steps (in expectation). Hence there exists a state-action pair $(s,a) \in K_k \quad A$ such that $p(s^{y}|s,a) \quad \frac{1}{L}$. Since $(K_k) \quad \lceil L \log \frac{3ALS}{} \rceil$ samples are available at each state-action pair, according to Lem. 11, we get that, under the event , s^{y} is found during the sample collection procedure for the state-action pair (s,a) (step ¬), which implies that $s^{y} \in U_k$.

Moreover, the choice of allocation function guarantees in particular that there are more than $(\frac{4L^2}{"^2}\log(\frac{2LSA}{"}))$ samples available at each state-action pair $(s,a) \in K_k \quad A$. From the empirical Bernstein inequality of Eq. 17, we thus have that $|p(s^{y}|s,a) \quad \hat{p}_k(s^{y}|s,a)| \quad \frac{"}{2L}$ under the event . Consequently we have

$$\hat{p}_k(s^{y}|s,a) \quad \frac{1}{L} \quad |\ p(s^{y}|s,a) \quad \hat{p}_k(s^{y}|s,a)| \quad \frac{1 \quad \frac{"}{2}}{L};$$

which implies that $s^{y} \in W_k$. Furthermore, we can decompose W_k the following way

$$W_k = \bigcup_{(s,a) \in K_k \quad A} Y_k(s,a);$$

where we introduce the subset

$$Y_k(s,a) := \quad s^0 \in U_k : \hat{p}_k(s^0|s,a) \quad \frac{1 \quad \frac{"}{2}}{L} \quad :$$

We then have

$$1 = \sum_{s^0 \in S} \hat{p}_k(s^0|s,a) \quad \sum_{s^0 \in Y_k(s,a)} \hat{p}_k(s^0|s,a) \quad \frac{1 \quad \frac{"}{2}}{L}|Y_k(s,a)|:$$

We conclude the proof by writing that

$$|W_k| \quad \sum_{(s,a) \in K_k \quad A} |Y_k(s,a)| \quad \frac{L}{1 \quad \frac{"}{2}} A|K_k| \quad 2LA\,|K_k|;$$

where the last inequality uses that $" \quad 1$ (from line 2 of Alg. 1). □

Lemma 13. Under the event $\cap$, when either condition STOP1 or STOP2 is triggered (at a round indexed by K), we have $S_L^{\to} \quad K_K$.

Proof. If condition STOP1 is triggered, Lem. 12 immediately guarantees that $S_L^{\to} \quad K_K$ under the event . If condition STOP2 is triggered, we have for all $s \in W_K$, $\underline{v}_s(s_0 \to s) > L$. From Lem. 8 this means that, under the event , for all $s \in W_K$, $V^{?}_{K_K}(s_0 \to s) > L$. Hence none of the states in W_K can be reached in at most L steps (in expectation) with a policy restricted to K_K. We conclude the proof using Lem. 12. □

Lemma 14. Under the event $\cap$, when DisCo terminates at round K, for any state $s \in K_K$, the policy $_s$ computed during step ° verifies

$$v_{_s}(s_0 \to s) \quad \min_{\in (S_L^{\to})} v(s_0 \to s) + ":$$

Moreover, we have that $S_L^{\to} \quad K_K \quad S^{\to}_{L+"}$.

Proof. Assume that the event $\cap$ holds. Then when the final set K_K is considered and the new policies are computed using all the samples, Lem. 10 yields for all $s \in K_K$,

$$v_{_s}(s_0 \to s) \quad \min_{\in (K_K)} v(s_0 \to s) + ":$$

Moreover Lem. 13 entails that $K_K \quad S^{\to}_L$. This implies from Lem. 1 that

$$\min_{\in (K_K)} v(s_0 \to s) \quad \min_{\in (S_L^{\to})} v(s_0 \to s);$$

which means that $K_K \quad S^{\to}_{L+"}$. □

D.6 High Probability Bound on the Sample Collection Phase (step)

Denote by K the (random) index of the last round during which the algorithm terminates. We focus on the sample collection procedure for any state $s \in K_K$. We denote by k_s the index of the round during which s was added to the set of "controllable" states K. To collect samples at state s, the learner uses the shortest-path policy $_s$. We say that an attempt to collect a specific sample is a rollout. We denote by $Z_K := |K_K|AN_K$ the total number of samples that the learner needs to collect. As such, at most Z_K rollouts must take place. Assume that the event holds. Then from Lem. 14, we have $K_K \quad S^{!}_{L+"}$. Hence, denoting $S_{L+"} := |S^{!}_{L+"}|$, we have $Z_K \quad Z_{L+"} := S_{L+"}A(\quad S^{!}_{L+"})$. The following lemma provides a high-probability upper bound on the time steps required to meet the sampling requirements.

Lemma 15. Assume that the event holds. Set

$$:= 4(L + " + 1)\log \frac{6Z_{L+"}}{} ;$$

and introduce the following event

$$T := \left\{ \exists \text{ one rollout (with goal state } s\text{) s.t. } \quad _s(s_0 \to s) > \right\} :$$

We have $P(T) \quad \frac{}{3}$.

Proof. Assume that the event holds. Leveraging a union bound argument and applying Lem. 16 to policy $_s$ which verifies $v_{_s}(s^0 \to s) \quad L + " + 1$ for any $s^0 \in K_{k_s}$, we get

$$P(T) \quad \sum_{\text{rollouts}} 2\exp\left(\frac{}{4(L + " + 1)}\right) \quad 2Z_{L+"}\exp\left(\frac{}{4(L + " + 1)}\right) \quad \frac{}{3};$$

where the last inequality comes from the choice of . □

Lemma 16 ([28], Lem. B.5). Let be a proper policy such that for some $d > 0$, $V(s) \quad d$ for every non-goal state s. Then the probability that the cumulative cost of to reach the goal state from any state s is more than m, is at most $2e^{m=(4d)}$ for all $m \quad 0$. Note that a cost of at most m implies that the number of steps is at most $m=c_{\min}$.

D.7 Putting Everything Together: Sample Complexity Bound

The sample complexity of the algorithm is solely induced by the sample collection procedure (step). Recall that we denote by K the index of the round at which the algorithm terminates. With probability at least $1 \quad \frac{2}{3}$, Lem. 13 holds, and so does the event . Hence the algorithm discovers a set of states $K_K \quad S^{!}_{L}$. Moreover, from Lem. 14, the algorithm outputs for each $s \in K_K$ a policy $_s$ with $E[\ _s(s_0 \to s)] \quad V^{?}_{S^{!}_L}(s) + "$. Hence we also have $jK_K j \quad S_{L+"} := jS^{!}_{L+"}j$.

We denote by $Z_K := jK_K jA \quad (K_K)$ the total number of samples that the learner needs to collect. From Lem. 15, with probability at least $1 \quad \frac{}{3}$, the total sample complexity of the algorithm is at most Z_K, where $:= 4(L + " + 1)\log \frac{6Z_{L+"}}{}$.

Now, from Eq. 19 there exists an absolute constant > 0 such that DisCo selects as allocation function

$$: X \to \left(\frac{L^4 b(X)}{"^2} \log^2 \frac{LSA}{"} + \frac{L^2 jXj}{"} \log \frac{LSA}{"} \right);$$

where

$$b(X) := \max_{(s;a)\in X A} \left(\sum_{s^0 \in X} \sqrt{b p(s^0 js; a)(1 \quad b p(s^0 js; a))} \right)^2 :$$

The total requirement is (K_K). Note that from Cauchy-Schwarz's inequality, we have

$$b(K_K) \quad _K := \max_{(s;a)\in K_K A} kf p(s^0 js; a)g_{s^0 \in K_K} k_0 \quad jK_K j:$$

Combining everything yields with probability at least $1-\delta$,

$$Z_K = \tilde{O}\left(\frac{L^5 \Gamma_{\mathcal{K}_K} |\mathcal{K}_K| A}{\varepsilon^2} + \frac{L^3 |\mathcal{K}_K|^2 A}{\varepsilon}\right).$$

We finally use that $\mathcal{K}_K \subseteq \mathcal{S}^{\to}_{L+\varepsilon}$ from Lem. 14, which implies that

$$\mathcal{C}_{AX^\star}(\text{DisCo}, L, \varepsilon, \delta) = \tilde{O}\left(\frac{L^5 \Gamma_{L+\varepsilon} S_{L+\varepsilon} A}{\varepsilon^2} + \frac{L^3 S_{L+\varepsilon}^2 A}{\varepsilon}\right),$$

where $\Gamma_{L+\varepsilon} := \max_{(s,a)\in\mathcal{S}^{\to}_{L+\varepsilon}\times\mathcal{A}} \|\{p(s'|s,a)\}_{s'\in\mathcal{S}^{\to}_{L+\varepsilon}}\|_0$. This concludes the proof of Thm. 1.

D.8 Proof of Corollary 1

The result given in Cor. 1 comes from retracing the analysis of Lem. 14 and therefore Lem. 10 by considering non-uniform costs between $[c_{\min}, 1]$ instead of costs all equal to 1. Specifically, Eq. 20 needs to account for the inverse dependency on $c_{\min}$ of the simulation lemma of Lem. 3. This induces the final $\varepsilon/c_{\min}$ accuracy level achieved by the policies output by DisCo. There remains to guarantee that condition 8 of Lem. 3 is verified. In particular the condition holds if $\eta(L+1+\varepsilon) \leq 2c_{\min}$, where η is the model accuracy prescribed in the proof of Lem. 10. We see that this is the case whenever we have $\varepsilon = O(Lc_{\min})$ due to the fact that $\eta = \Omega(\varepsilon/L^2)$.

D.9 Computational Complexity of DisCo

The overall computational complexity of DisCo can be expressed as $\sum_{k=1}^{K} |\mathcal{W}_k|\, C(\text{OVI}_{\text{SSP}})$, where $C(\text{OVI}_{\text{SSP}})$ denotes the complexity of an OVI_{SSP} procedure and where we recall that K denotes the (random) index of the last round during which the algorithm terminates. Note that it holds with high probability that $K \leq |\mathcal{S}^{\to}_{L+\varepsilon}|$ and $|\mathcal{W}_k| \leq 2LA|\mathcal{K}_k| \leq 2LA|\mathcal{S}^{\to}_{L+\varepsilon}|$. Moreover $C(\text{OVI}_{\text{SSP}})$ captures the complexity of the value iteration (VI) algorithm for SSP, which was proved in [34] to converge in time quadratic w.r.t. the size of the considered state space (here, $\mathcal{K}_k$) and $\|V^\star\|_\infty/c_{\min}$. Here we have $c_{\min} = 1$, and we can easily prove that in all the SSP instances considered by DisCo, the optimal value function $V^\star$ verifies $\|V^\star\|_\infty = O(L^2)$, due to the restriction of the goal state in $\mathcal{W}_k$ (indeed this restriction implies that there exists a state-action pair in $\mathcal{K}_k \times \mathcal{A}$ that transitions to the goal state with probability $\Omega(1/L)$ in the true MDP). Putting everything together gives DisCo's computational complexity. Interestingly, we notice that while it depends polynomially on $S_{L+\varepsilon}$, L and A, it is independent from S the size of the global state space.

E The UcbExplore Algorithm [1]

E.1 Outline of the Algorithm

The UcbExplore algorithm was introduced by Lim and Auer [1] to specifically tackle condition AX_L. The algorithm maintains a set $\mathcal{K}$ of "controllable" states and a set $\mathcal{U}$ of "uncontrollable" states. It alternates between two phases of *state discovery* and *policy evaluation*. In a state discovery phase, new candidate states are discovered as potential members of the set of controllable states. Any policy evaluation phase is called *round* and it relies on an optimistic principle: it attempts to reach an "optimistic" states (i.e., the easiest state to reach based on information collected so far) among all the candidate states by executing an optimistic policy π_s that minimizes the optimistic expected hitting time truncated at a horizon of $H_{\text{UCB}} := \lceil L + L^2\varepsilon^{-1}\rceil$. Within the round of evaluation of policy π_s, the algorithm proceeds through at most $\lambda_{\text{UCB}} := \lceil 6L^3\varepsilon^{-3}\log(16|\mathcal{K}|^2\delta^{-1})\rceil$ episodes, each of which begins at s_0 and ends either when π_s successfully reaches s or when H_{UCB} steps have been executed. If the empirical performance of π_s is poor (measured through a performance check done after each episode), the round is said to have *failed*. Otherwise, the round is *successful* which means that s is controllable and an acceptable policy (π_s) has been discovered. A failure round leads to selecting another candidate state-policy pair for evaluation, while a success round leads to a state discovery phase which in turn adds more candidate states for the subsequent rounds. As explained in App. A, UcbExplore is unable to tackle the more challenging objective $AX^\star$.

E.2 Minor Issue and Fix in the Analysis of UcbExplore

The key insight of UcbExplore is to bound the number of *failure rounds* of the algorithm, by lower- and upper-bounding the so-called "regret" contribution of failure rounds, where the regret of a failure round k is defined as

$$\sum_{j=1}^{e_k} \left[H_{UCB} - L - \sum_{i=0}^{H_{UCB}-1} r_i \right];$$

where e_k $_{\mathrm{UCB}}$ is the actual number of episodes executed in round k and where the reward $r_i \in \{0; 1\}$ is equal to 1 only if the state is the goal state. However, upper bounding the regret contribution of failure rounds implies applying a concentration inequality on *only* specific rounds that are chosen given their *empirical performance*. Hence Lim and Auer [1, Lem. 18] improperly use a martingale argument to bound a sum whose summands are chosen in a non-martingale way, i.e., depending on their realization.

To avoid the aforementioned issue, one must upper and lower bound the cumulative regret of the *entire* set of rounds and not *only* the failure rounds in order to obtain a bound on the number of failure rounds. However, this would yield a sample complexity that has a second term scaling as $\tilde{O}(\varepsilon^{-4})$. Following personal communication with the authors, the fix is to change the definition of regret of a round, making it equal to

$$\sum_{j=1}^{e_k} \left(\widetilde{e}_{H_{UCB}}(s_0 \to s) - \sum_{i=0}^{H_{UCB}-1} r_i \right);$$

where s is the considered goal state and $\widetilde{e}_{H_{UCB}}(s_0 \to s)$ is the optimistic H_{UCB}-step reward (where the reward is equal to 1 only at state s). With this new definition, it is possible to recover the sample complexity provided in [1] scaling as $\tilde{O}(\varepsilon^{-3})$.

E.3 Issue with a Possibly Infinite State Space

Lim and Auer [1] claim that their setting can cope with a countable, possibly infinite state space. However, this leads to a technical issue, which has been acknowledged by the authors via personal communication and as of now has not been resolved. Indeed, it occurs when a union bound over the unknown set $\mathcal{U}$ is taken to guarantee high-probability statements (e.g., the Lem. 14 or 17 of [1]). Yet for each realization of the algorithm, we do not know what the set $\mathcal{U}$, or equivalently $\mathcal{K}$, looks like, hence it is improper to perform a union bound over a set of unknown identity. Simple workarounds to circumvent this issue are to impose a finite state space, or to assume prior knowledge over a finite superset of $\mathcal{U}$. In this paper we opt for the first option. It remains an open and highly non-trivial question as to how (and whether) the framework can cope with an infinite state space.

E.4 Effective Horizon of the AX Problem and its Dependency on ε

UcbExplore [1] designs finite-horizon problems with horizon $H_{UCB} := \lceil L + L^2 \varepsilon^{-1} \rceil$ and outputs policies that reset every H_{UCB} time steps. In the following we prove that the effective horizon of the AX problem actually scales as $O(\log(L\varepsilon^{-1})L)$, i.e., only *logarithmically* w.r.t. ε^{-1}. We begin by defining the concept of "resetting" policies as follows.

Definition 11. For any $\pi \in \Pi$ and horizon $H \geq 0$, we denote by $\pi^{|H}$ the non-stationary policy that executes the actions prescribed by π and performs the RESET action every H steps, i.e.,

$$\pi_t^{|H}(a|s) := \begin{cases} \text{RESET} & \text{if } t \equiv 0 \ (\text{mod} H); \\ \pi(a|s) & \text{otherwise.} \end{cases}$$

We denote by $\Pi^{|H}$ the set of such "resetting" policies.

The following lemma captures the effective horizon H_{eff} of the problem, in the sense that restricting our attention to $\Pi^{|H}(S_L^{\to})$ for $H \geq H_{\text{eff}}$ does not compromise the possibility of finding policies that achieve the performance required by $\text{AX}^{\star}$ (and thus also by AX_L).

Lemma 17. For any " 2 (0; 1] and L 1, whenever

$$H \quad H_{\mathrm{eff}} := 4(L+1) \quad \log \frac{4(L+1)}{"} \quad ;$$

we have for any s^y 2 $S_L^!$,

$$\min_{|H\ 2\ |H\ (S_L^!\ )} v \ _{|H} (s_0 \ ! \quad s^y) \quad V^?_{S_L^!} \ (s_0 \ ! \quad s^y) + ":$$

Proof. Consider any goal state s^y 2 $S_L^!$. Set $"^0 := \frac{"}{2(L+1)} \quad \frac{1}{2}$. Denote by 2 ($S_L^!$) the minimizer of $V^?_{S_L^!}$ $(s_0 \ ! \quad s^y)$. For any horizon H 0, we introduce the truncated value function $v_{;H}$ $(s \ ! \quad s^0) :=$ E[$(s \ ! \quad s^0) \wedge H$] and the tail probability $q_{;H}$ $(s \ ! \quad s^0) :=$ P($(s \ ! \quad s^0) > H$). Due to the presence of the RESET action, the value function of can be bounded for all states s 2 $S_L^!$ n f s^yg as

$$v \ (s \ ! \quad s^y) \quad V^?_{S_L^!} \ (s_0 \ ! \quad s^y) + 1 \quad L + 1:$$

This entails that the probability of the goal-reaching time decays exponentially. More speci cally, we have

$$q_{;H} \ (s_0 \ ! \quad s^y) \quad 2 \exp \quad \frac{H}{4(L+1)} \quad "^0; \tag{21}$$

where the rst inequality stems from Lem. 16 and the second inequality comes from the choice of H 4(L + 1) log $\frac{2}{"^0}$. Furthermore, we have $(s \ ! \quad s^0) \wedge H$ $(s \ ! \quad s^0)$ and thus E[$(s \ ! \quad s^0) \wedge H$] E[$(s \ ! \quad s^0)$]. Consequently,

$$v_{;H} \ (s_0 \ ! \quad s^y) \quad v \ (s_0 \ ! \quad s^y) = V^?_{S_L^!} \ (s_0 \ ! \quad s^y): \tag{22}$$

Now, from [1, Eq. 4], the value function of can be related to its truncated value function and tail probability as follows

$$v \ _{|H} = \frac{v_{;H} \ + q_{;H}}{1 \quad q_{;H}}: \tag{23}$$

Plugging Eq. 21 and 22 into Eq. 23 yields

$$v \ _{|H} (s_0 \ ! \quad s^y) \quad \frac{V^?_{S_L^!} \ (s_0 \ ! \quad s^y) + "^0}{1 \quad "^0}:$$

Notice that the inequalities $\frac{1}{1 \ x}$ $1 + 2x$ and $\frac{x}{1 \ x}$ $2x$ hold for any $0 < x$ $\frac{1}{2}$. Applying them for $x = "^0$ yields

$$\frac{V^?_{S_L^!} \ (s_0 \ ! \quad s^y) + "^0}{1 \quad "^0} \quad (1 + 2"^0) V^?_{S_L^!} \ (s_0 \ ! \quad s^y) + 2"^0:$$

From the inequality $V^?_{S_L^!}$ $(s_0 \ ! \quad s^y)$ L and the de nition of $"^0$, we nally obtain

$$v \ _{|H} (s_0 \ ! \quad s^y) \quad V^?_{S_L^!} \ (s_0 \ ! \quad s^y) + ";$$

which completes the proof. □

Lem. 17 reveals that the effective horizon H_{eff} of the AX problem scales only logarithmically and not linearly in " ^{1}. This highlights that the design choice in UcbExplore to tackle nite-horizon problems with horizon H_{UCB} unavoidably leads to a suboptimal dependency on " in its AX_L sample complexity bound. In contrast, by designing SSP problems and thus leveraging the intrinsic goal-oriented nature of the problem, DisCo can (implicitly) capture the effective horizon of the problem. This observation is at the heart of the improvement in the " dependency from $\tilde{O}$(" ^{3}) of UcbExplore [1] to $\tilde{O}$(" ^{2}) of DisCo (Thm. 1).

F Experiments

This section complements the experimental findings partially reported in Sect. 5. We provide details about the algorithmic configurations and the environments as well as additional experiments.

F.1 Algorithmic Configurations

Experimental improvements to UcbExplore [1]. We introduce several modifications to UcbExplore in order to boost its practical performance. We remove all the constants and logarithmic terms from the requirement for state discovery and policy evaluation (refer to [1, Fig. 1]). Furthermore, we remove the constants in the definition of the accuracy $\varepsilon' = \varepsilon/L$ used by UcbExplore (while their original algorithm requires ε' to be divided by 8, we remove this constant). We also significantly improve the planning phase of UcbExplore [1, Fig. 2]. Their procedure requires to divide the samples into $H := (1 + 1/\varepsilon')L$ disjoint sets to estimate the transition probability of each stage h of the finite-horizon MDP. This substantially reduces the accuracy of the estimated transition probability since for each stage h only $N_k(s,a)/H$ are used. In our experiments, we use all the samples to estimate a stationary MDP (i.e., $\hat{p}_k(s'|s,a) = N_k(s,a,s')/N_k(s,a)$) rather than a stage-dependent model. Estimating a stationary model instead of bucketing the data is simpler and more efficient since leads to a higher accuracy of the estimated model. To avoid to move too far away from the original UcbExplore, we decided to define the confidence intervals as if bucketing was used. We thus consider $\underline{N}_k(s,a) = N_k(s,a)/H$ for the construction of the confidence intervals. For planning, we use the optimistic backward induction procedure as in [30]. We thus leverage empirical Bernstein inequalities —which are much tighter— rather than Hoeffding inequalities as suggested in [1]. In particular, we further approximate the bonus suggested in [30, Alg. 4] as

$$b_h(s,a) = \sqrt{\frac{\mathrm{Var}_{s' \sim \hat{p}_k(\cdot|s,a)}[V_{k,h+1}(s')]}{\underline{N}_k(s,a) \vee 1}} + \frac{(H-h)}{\underline{N}_k(s,a) \vee 1}.$$

For DisCo, we follow the same approach of removing constants and logarithmic terms. We thus use the definition of as in Thm. 1 with = 1 and without log-terms. For planning, we use the procedure described in App. D with $b_k(s,a,s') = \sqrt{\frac{\hat{p}_k(s'|s,a)(1-\hat{p}_k(s'|s,a))}{N_k(s,a) \vee 1}} + \frac{1}{N_k(s,a) \vee 1}$. Finally, in the experiments we use a state-action dependent value $(s,a;K_k) = \left(\sum_{s' \in K_k} \sqrt{\hat{p}_k(s'|s,a)(1-\hat{p}_k(s'|s,a))}\right)^2$ instead of taking the maximum over (s,a).

Even though we boosted the practical performance of UcbExplore w.r.t. the original algorithm proposed in [1] (e.g., the use of Bernstein), we believe it makes the comparison between DisCo and UcbExplore as fair as possible.

F.2 Confusing Chain

The *confusing chain* environment referred to in Sect. 5 is constructed as follows. It is an MDP composed of an initial state s_0, a chain of length C (states are denoted by $s_1, \ldots, s_C$) and a set of K confusing states $(s_{C+1}, \ldots, s_{C+K})$. Two actions are available in each state. In state s_0, we have a forward action a_0 that moves to the chain with probability p_c ($p(s_1|s_0,a_0) = p_c$ and $p(s_0|s_0,a_0) = 1 - p_c$) and a confusing action that has uniform probability of reaching any confusing state ($p(s_i|s_0,a_1) = 1/K$ for any $i \in \{C+1, \ldots, C+K\}$). In the confusing states, all actions move deterministically to the end of the chain ($p(s_C|s_i,a) = 1$ for any $i \in \{C+1, \ldots, C+K\}$ and a). In each state of the chain, there is a forward action a_0 that behaves as in s_0 ($p(s_{\min(C,i+1)}|s_i,a_0) = p_c$ and $p(s_i|s_i,a_0) = 1 - p_c$, for any $i \in \{1, \ldots, C-1\}$) and a skip action a_1 that moves to m states ahead with probability p_{skip} ($p(s_{\min(C,i+m)}|s_i,a_0) = p_{skip}$ and $p(s_i|s_i,a_0) = 1 - p_{skip}$, for any $i \in \{1, \ldots, C-1\}$). Finally, $p(s_0|s_C,a) = 1$ for any action a. In our experiments, we set $m = 4$, $p_{skip} = 1/3$, $p_c = 1$, $C = 5$, $K = 6$, $L = 4.5$.

ε	DisCo	UcbExplore-Bernstein
0.1	$374,263$ $(13,906)$	$5,076,688$ $(92,643)$
0.2	$105,569$ $(4,645)$	$636,580$ $(13,716)$
0.4	$29,160$ (829)	$108,894$ $(2,305)$
0.6	$15,349$ (475)	$40,538$ (805)
0.8	$9,891$ (244)	$21,270$ (441)

Table 2: Sample complexity of DisCo and UcbExplore-Bernstein, on the confusing chain domain. Values are averaged over 50 runs and the 95%-confidence interval of the mean is reported in parenthesis.

	UcbExplore-Bernstein					
ε	Expected hitting time $v^{\pi}(s_0 \to s_i)$					
	s_0	s_1	s_2	s_3	s_4	s_5
$0.1; 0.2$	0	1	2	3	4	4
0.4	0	1	2	3	4	4.94 (0.04)
0.6	0	1	2	3.36 (0.11)	4	4.53 (0.07)
0.8	0	1	2	3.38 (0.11)	4.07 (0.07)	4.53 (0.06)

Table 3: Expected hitting time of state s_i of the goal-oriented policy π_{s_i} recovered by UcbExplore-Bernstein, on the confusing chain domain. DisCo recovers the optimal goal-oriented policy in all the runs and for all ε. The advantage of DisCo lies in its final policy consolidation step. Values are averaged over 50 runs and the 95%-confidence interval of the mean is reported in parenthesis (it is omitted when equal to 0). This shows that UcbExplore recovers the optimal goal-oriented policy in every run only for ε equal to 0.1 and 0.2.

Sample complexity. We provide in Tab. 2 the sample complexity of the algorithms for varying values of ε. As mentioned in Sect. 5, DisCo outperforms UcbExplore for any value of ε, and increasingly so when ε decreases. Fig. 7 complements Fig. 2 for additional values of ε.

Quality of goal-reaching policies. We now investigate the quality of the policies recovered by DisCo and UcbExplore. In particular, we show that DisCo is able to find the incrementally near-optimal shortest-path policies to any goal state, while UcbExplore may only recover sub-optimal policies. On the confusing chain domain, the intuition is that the set of confusing states makes s_C reachable in just 2 steps but the confusing states are not in the controllable set and thus the algorithms are not able to recover the shortest-path policy to s_C. On the other hand, state s_C is controllable through two policies: 1) the policy π_1 that takes always the forward action a_0 reaches s_C in 5 steps; 2) the policy π_2 that takes the skip action a_1 in s_1 reaches s_C in 4 steps. We observed empirically that DisCo always recovers policy π_1 (i.e., the fastest policy) while UcbExplore selects policy π_2 in several cases. This is highlighted in Tab. 3 where we report the expected hitting time of the policies recovered by the algorithms. This finding is not surprising since, as we explain in Sect. 4 and App. A, UcbExplore is designed to find policies reaching states in at most L steps on average, yet it is not able to recover incrementally near-optimal shortest-path policies, as opposed to DisCo.

F.3 Combination Lock

We consider the combination lock problem introduced in [31]. The domain is a stochastic chain with $S = 6$ states and $A = 2$ actions. In each state s_k, action right (a_1) is deterministic and leads to state s_{k+1}, while action left (a_0) moves to a state s_{k-l} with probability proportional to $1/(k-l)$ (i.e., inversely proportional to the distance of the states). Formally, we have that

$$n(x_k; x_l) = \begin{cases} \frac{1}{k-l} & \text{if } l < k \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad p(x_l|x_k; a_0) = \frac{n(x_k; x_l)}{\sum_s n(x_k; s)}.$$

We set the initial state to be at $2/3$ of the chain, i.e., $\lfloor 2N/3 \rfloor$. The actions in the end states are absorbing, i.e., $p(s_0|s_0; a_0) = 1$ and $p(s_{N-1}|s_{N-1}; a_1) = 1$, while the remaining actions behave normally. See Fig. 5 for an illustration of the domain.

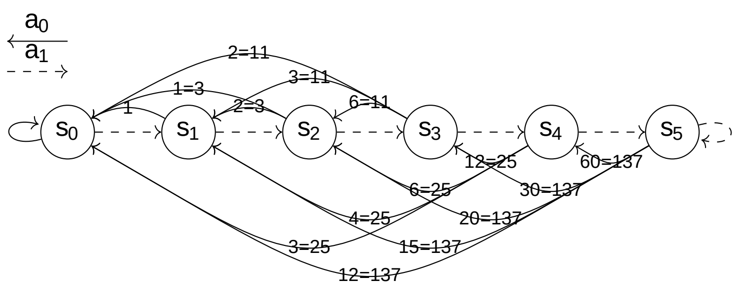
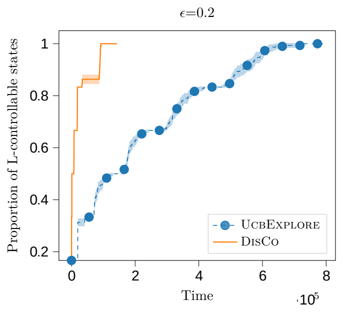


Figure 5: Combination lock domain with $S = 6$ states. Expected hitting times from the initial state s_3 are $v(s_3 \to s) = (2.18, 1.91, 1.64, 0, 1, 2)$. Consider $L = 3$, the set of incrementally L-controllable states is $\mathcal{S}_L^{\to} = \{s_2, s_3, s_4, s_5\}$. The goal-oriented policy to reach s_4 and s_5 takes always the right action a_1, while the policy for s_2 always selects the left action a_0.

Figure 6: Proportion of the incrementally L-controllable states identified by DisCo and UcbExplore in the combination lock domain for $L = 2.7$ and $\varepsilon = 0.2$. Values are averaged over 20 runs.

Sample complexity. We evaluate the two algorithms DisCo and UcbExplore on the combination lock domain, for $\varepsilon = 0.2$ and $L = 2.7$. We further boost the empirical performance of UcbExplore by using N instead of $\underline{N}$ for the construction of the confidence intervals (i.e., we do not account for the data bucketing in [1], see App. F.1). To preserve the robustness of the algorithm, we use $\log(|\mathcal{K}_k|^2)/(\varepsilon')^3$ episodes for UcbExplore's policy evaluation phase (indeed we noticed that the removal of the logarithmic term here sometimes leads UcbExplore to miss some states in $\mathcal{S}_L^{\to}$ in this domain). For the same reason, in DisCo we use the value $\widehat{\phi}(\mathcal{K}_k) = \max_{s,a} \widehat{\phi}(s, a, \mathcal{K}_k)$ prescribed by the theoretical algorithm instead of the state-action dependent values used in the previous experiment. We average the experiments over 20 runs and obtain a sample complexity of $30,117$ $(2,087)$ for DisCo and $90,232$ $(2,592)$ for UcbExplore. Fig. 6 reports the proportion of incrementally L-controllable states identified by the algorithms as a function of time. We notice that once again DisCo clearly outperforms UcbExplore.



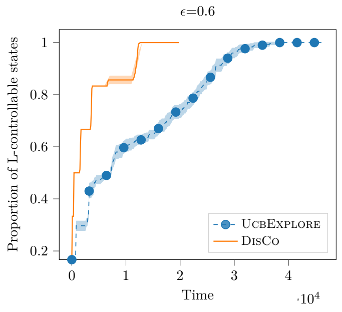


Figure 7: Proportion of the incrementally L-controllable states identified by `DisCo` and `UcbExplore` on the confusing chain domain for $L = 4.5$ and ε 2 f$0.1, 0.2, 0.4, 0.6, 0.8$g. Values are averaged over 50 runs. `UcbExplore` uses Bernstein confidence intervals for planning.