The Dissertation Committee for Mustafa O. Karabag
certifies that this is the approved version of the following dissertation:

# Decision-Making for Autonomous Agents in Adversarial or Information-Scarce Settings

**Committee**:

Ufuk Topcu, Supervisor

Aryan Mokhtari

Sanjay Shakkottai

Takashi Tanaka

Melkior Ornik

# Decision-Making for Autonomous Agents in Adversarial or Information-Scarce Settings

**by**

**Mustafa O. Karabag**

**Dissertation**

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

**Doctor of Philosophy**

**The University of Texas at Austin**

**August 2023**

# Dedication

To my family.

# Acknowledgments

First and foremost, I thank my parents, Ayse and Mikail, and my sisters, Semra and Esra. None of this work would have been possible without your unconditional support and love. Mom, you always made me happy and comfortable with your love and care. Dad, you not only supported my education journey but also taught me every skill that you know. Semra and Esra, I always enjoyed the time I spent with you and am extremely lucky to have you as my siblings and friends. I also would like to thank my nephew Marsel and my niece Ipek. Seeing and talking with you always put a smile on my face.

I am grateful to have Ufuk Topcu as my advisor. You gave me the freedom to find my research problems and guided me in the right direction when I was lost. I appreciate that you always challenged me to think about the broader picture and were always available when I needed your help.

I thank my committee members, Aryan Mokhtari, Melkior Ornik, Sanjay Shakkottai, and Takashi Tanaka. I am grateful for their valuable feedback on my research that shaped the final form of this dissertation.

During my studies, I was fortunate to work with many great collaborators whose insights and technical discussions made me a better researcher. In particular, I would like to thank Melkior Ornik for always challenging me with interesting research ideas, and Cyrus Neary for his professionalism during our joint projects. I also would like to thank my fellow PhD students at the Autonomous Systems Group for their help with research and for being great friends.

I would like to thank my friends with whom I enjoyed my years in Austin. Special thanks to Yagiz for supporting me in research and being the brother that I never had. I would also like to thank Cyrus, Michael, Suda, Steve, Nick, Jesse, Maansi, and Gen for their friendship. I had a lot of fun spending time with you, and

I am fortunate to have you in my life.

I would like to thank Bulent Onaran, who encouraged my decision to become an engineer and guided me in almost every aspect of life.

Finally, I have to thank Anna for her love and support. Thank you for all the joy you have brought me and your support that helped me finish my PhD journey.

# Abstract

## Decision-Making for Autonomous Agents in Adversarial or Information-Scarce Settings

Mustafa O. Karabag, PhD
The University of Texas at Austin, 2023

SUPERVISOR: Ufuk Topcu

Autonomous agents often operate in adversarial or information-scarce settings. These settings exist due to various factors, such as the coexistence of non-cooperative agents, computation limitations, communication losses, and imperfect sensors. To ensure high performance in the presence of such factors, decision-making algorithms for autonomous agents must limit the amount of sensitive information leaked to adversaries and rely on minimal information about their environment. We consider a variety of problems where an autonomous agent operates in an adversarial or information-scarce setting, and present novel theory and decision-making algorithms for these problems. First, we focus on an adversarial setting where a malicious agent aims to deceive its supervisor in probabilistic supervisory control setting. We formulate the deception problem as an expected cost minimization problem in a Markov decision process (MDP) where the cost function is motivated by the results from hypothesis testing. We show the existence of an optimal stationary deceptive policy and provide algorithms for the synthesis of optimal deceptive policies. From the perspective of the supervisor, we prove the NP-hardness of synthesizing optimal reference policies that prevent deception. We also show that synthesizing optimal deceptive policies under partial observations is NP-hard and provide synthesis algorithms by considering special classes of policies and MDPs. Second, as a part of decision-making in

information-scarce settings, we consider a multiagent decision-making problem where a group of agents cooperates under communication losses. We model this problem with a multiagent MDP, quantify the intrinsic dependencies between the agents induced by their joint policy, and develop a decentralized policy execution algorithm for communication losses. For a variety of communication loss models, we provide performance lower bounds that are functions of the dependencies between the agents. We develop an algorithm for the synthesis of minimally dependent policies that optimize these lower bounds and thereby remain performant under communication losses. Finally, we consider the problem of optimization under limited information since autonomous agents often perform optimization as a part of their operation. We develop optimization algorithms for smooth convex optimization using sub-zeroth-order oracles that provide less information than zeroth and first-order oracles. For the directional preference oracle that outputs the sign of the directional derivative at the query point and direction, we show a $\tilde{\mathcal{O}}(n^4)$ sample complexity upper bound where $n$ is the number of dimensions. For the comparator oracle that compares the function value at two query points and outputs a binary comparison value, we show a $\tilde{\mathcal{O}}(n^4)$ sample complexity upper bound. For the noisy value oracle, we develop an algorithm with $\tilde{\mathcal{O}}(n^{3.75}T^{0.75})$ high probability regret bound where $T$ is the number of queries.

# Table of Contents

# List of Figures

# Chapter 1: Introduction

Autonomous agents perform tasks without human control and often operate in adversarial or information-scarce settings. These settings exist due to various factors such as the coexistence of non-cooperative agents, computation limitations, communication losses, and imperfect sensors. To ensure high performance in the presence of such factors, decision-making algorithms for autonomous agents must limit the amount of sensitive information revealed to adversaries and rely on minimal information about their environments and the other agents in their environments. The goal of this dissertation is to develop theoretical performance guarantees and algorithms for decision-making in an adversarial or information-scarce settings.

**Decision-making in adversarial settings** Autonomous agents are expected to operate in the presence of their adversaries. These adversaries may be different agents with conflicting objectives in the same environment or external attackers that aim to exploit the vulnerabilities of the agent. In such cases, an agent can remain performant by limiting sensitive information revealed to its adversaries, e.g., by preserving opacity (Jacob et al., 2016; Bérard et al., 2015), secrecy (Alur et al., 2006) or privacy (Farokhi and Sandberg, 2019; Abadi et al., 2016).

We explore methods to limit the revealed information on an agent's intentions and study a deceptive decision-making problem. Deception is present in many fields that involve two agents, at least one of which is performing a task that is undesirable to the other. Deceptive strategies exploit information asymmetries between the agents or the irrationality of the others allowing the deceptive agent to gain advantage. Some example domains of deception include cyber systems (Carroll and Grosu, 2011; Almeshekah and Spafford, 2016), autonomous vehicles (McEneaney and Singh, 2005), warfare strategy (Lloyd, 2003), and robotics (Shim and Arkin, 2013). In Chapter 2, we focus on application of deceptive strategies in probabilistic supervisory control. In

detail, we formulate a problem where an agent is supposed to follow the instructions of its supervisor but instead aims to achieve a task that is potentially malicious toward the supervisor. Hence, it follows a deceptive strategy not to be detected by the supervisor. We study optimal deceptive decision-making for the agent that maintains plausible deniability and does not reveal the agent's intentions. On the flip side, we study the supervisor's problem, i.e., giving instructions that would make malicious agents reveal the most information and allow the supervisor to distinguish the malicious agents from the well-intentioned ones.

**Decision-making in information-scarce settings** Autonomous agents often have to operate under limited information. Collecting information is costly, and processing information is challenging due to energy, communication, and computational limitations (Bernstein et al., 2018; König et al., 2021; Rasmussen et al., 2018). On the other hand, the capabilities of an autonomous system are proportional to the quality and quantity of the available information about its environment and the other agents in the environment. Hence, agents should rely on minimal information about their surroundings and utilize the available information efficiently to achieve high performance in information-scarce settings. Towards this goal, we study two decision-making problems in information-scarce settings.

First, we consider a multiagent decision-making problem where a group of agents cooperate under communication losses and thus may not always have perfect information on each other's state. In cooperative multiagent systems, a team of decision-making agents aims to achieve a common objective through repeated interactions with each other and with a shared environment. Such multiagent systems are ubiquitous; many applications of autonomous systems — such as the coordination of autonomous vehicles, the control of networks of mobile sensors, or the control of traffic lights — can be modeled as collections of interacting agents (Cao et al., 2012; Parker et al., 2016).

Inter-agent communication plays an essential role in the successful deployment of such multiagent systems. In particular, the coordination between agents via communication – their agreement upon the particular actions to collectively take at any given point in time – is often necessary for the successful implementation of an optimal joint policy (Boutilier, 1996). However, many possible sources of communication disruption exist in practice, such as radio interference, hardware failure, or even adversarial attacks intended to sabotage the team. Lost or unreliable communication can result in substantial degradation of the team's performance, because it removes the agents' ability to coordinate. Despite this reliance of the team's performance on communication, multiagent planning algorithms typically do not offer robustness guarantees against possible losses in communication. In Chapter 3, we focus on finding controllers that remain performant under communication losses by removing intrinsic dependencies between the agents and making their policies require minimal information about each other.

Second, we consider the problem of optimization under limited information since autonomous systems often perform optimization as a part of their operation. Derivative-free optimization methods use limited information and are necessary when explicit access to the objective function is not available, or when the function's gradient is hard to compute (Conn et al., 2009). Utility functions, a concept from economics, provide an example of a type of objective function which may be hard to explicitly characterize. However, while a consumer may not be able to quantify their utility for a given prospect, they will likely be able to rank the available prospects. From a human's perspective, ranking the prospects may be simple, even if it is difficult to directly assign them values (Abbas and Howard, 2015). For example, consider a reinforcement learning scenario in which a robot learns to perform a task via human feedback. The human may not be able to assign explicit rewards to the demonstrations performed by the robot, but she can rank them (Akrour et al., 2012; Fürnkranz et al., 2012; Wilson et al., 2012).

While necessary in a range of applications (Conn et al., 2009; Audet and Hare,

2017), the theoretical analysis of derivative-free optimization methods is limited in comparison with that of first-order optimization methods (Conn et al., 2009). In Chapter 4, we consider various sub-zeroth-order oracles that provide less information than zeroth and first-order oracles and leverage the smoothness and convexity of the objective function to develop sample-efficient optimization algorithms for derivative-free smooth convex optimization.

## 1.1 Dissertation Overview

In Chapters 2–4, we study the aforementioned decision-making problems and conclude with extensions and future directions in Chapter 5.

**Chapter 2: Deception in Probabilistic Supervisory Control** We consider a deception problem in a probabilistic supervisory control setting where an agent is supposed to follow a reference policy provided by its supervisor to achieve some tasks but instead uses a deceptive policy to achieve a malicious task. The agent aims to achieve its task while not being detected by the supervisor. The supervisor, on the other hand, aims to distinguish well-intentioned agents from malicious ones. We model this problem using a Markov decision process (MDP) and a Kullback-Leibler divergence objective function motivated by results from hypothesis testing. In this formulation, the supervisor observes the agent's paths in the environment for detection. The optimal deceptive policy for the agent minimizes the KL divergence between the distribution of paths under the agent's policy and the distribution of paths under the reference policy subject to the agent's task constraint. We show that a stationary optimal deceptive policy exists for the agent when the supervisor's policy is stationary, and this policy can be synthesized computationally efficiently by solving a convex optimization problem. On the flip side, we establish that the supervisor's problem, i.e., synthesizing optimal reference policies that prevent deception and achieves the supervisor's tasks, is NP-hard. We also extend this problem to a partially observ-

able setting where the supervisor gets partial observations of the agent's state. We establish that the synthesis of optimal deceptive policies is NP-hard in the partially observable setting. As an approximation, we consider special classes of control policies and MDPs, and provide policy synthesis algorithms for these special cases. The material presented in this chapter was published in (Karabag et al., 2021b, 2022b).

**Chapter 3: Minimally-Dependent Multiagent Systems that are Robust to Communication Loss** We consider a multiagent control problem where a team of agents cooperates to achieve a joint task under communication losses. We model the agents' environment with a transition-independent multiagent MDP (Becker et al., 2003), i.e., an MDP that is a Cartesian product of multiple MDPs, and the joint task with a reach-avoid specification. We introduce a simulation-based policy execution mechanism to be used for the decentralized execution of the team's joint policy during communication losses. Under this mechanism, we quantify the intrinsic dependencies between the agents that are induced by the joint policy, i.e., the total correlation of the joint policy. We then consider different communication loss models, e.g., a Bernoulli process and an adversarial loss model, and give upper bounds on the performance loss under communication losses. The performance loss is upper bounded by an increasing function of total correlation. For the Bernoulli process loss model, the upper bound is also an increasing function of the communication dropout rate. Finally, we use total correlation as a regularizer for "soft decentralization" and provide a synthesis procedure for the minimally dependent policies that remain performant under communication losses. The material presented in this chapter was published in (Karabag et al., 2022a).

**Chapter 4: Smooth Convex Optimization Using Sub-Zeroth-Order Oracles** We consider the minimization of a smooth convex function on a convex domain using sub-zeroth-order oracles. We use three different sub-zeroth-order oracles: a directional preference oracle that outputs the sign of the directional derivative at the query point

and direction, a comparator oracle that compares the function value at two query points and outputs a binary comparison value, and a noisy value oracle that outputs a value that is the function value at the query point plus a subgaussian noise. We rely on estimating inexact gradient directions and develop optimization algorithms based on the ellipsoid method. For the directional preference and comparator oracles, we develop optimization algorithms and show $\tilde{\mathcal{O}}(n^4)$ sample complexity upper bounds where $n$ is the number of dimensions. To the best of our knowledge, these optimization algorithms are the first algorithms with a linear rate of convergence for smooth convex optimization using directional preference or comparator oracles. For the noisy value oracle, we develop an algorithm with $\tilde{\mathcal{O}}(n^{3.75}T^{0.75})$ (ignoring other factors) high probability regret bound where $T$ is the number of queries[1]. The material presented in this chapter was published in (Karabag et al., 2021a).

[1]The publication Karabag et al. (2021a) included the above regret bound. The bound can be improved to $\tilde{\mathcal{O}}(n^{2.25}T^{0.75})$ by changing the analysis as described in §4.

# Chapter 2: Deception in Probabilistic Supervisory Control

We consider a setting with a supervisor and an agent where the supervisor provides a reference policy to the agent and expects the agent to achieve a task by following the reference policy. However, the agent aims to achieve another task that is potentially malicious towards the supervisor and follows a different, deceptive policy. In this chapter[1], we study the synthesis of deceptive policies for such agents and the synthesis of reference policies for such supervisors that try to prevent deception besides achieving a task.

In the described supervisory control setting, the agent's deceptive policy is misleading in the sense that the agent follows his own policy, but convinces the supervisor that he follows the reference policy. Misleading acts result in plausibly deniable outcomes (Doody, 2018). Hence, the agent's misleading behavior should have plausible outcomes for the supervisor. In detail, the supervisor has an expectation of the probabilities of the possible events. The agent should manipulate these probabilities such that he achieves his task while closely adhering to the supervisor's expectations.

We measure the closeness between the reference policy and the agent's policy by Kullback–Leibler (KL) divergence. KL divergence, also called relative entropy, is a measure of dissimilarity between two probability distributions (Cover and Thomas, 2012). KL divergence quantifies the extra information needed to encode a posterior distribution using the information of a given prior distribution. We remark that this interpretation matches the definition of plausibility: The posterior distribution is plausible if the KL divergence between the distributions is low.

[1]The research presented in this chapter is published in (Karabag et al., 2021b, 2022b). Mustafa O. Karabag formulated the problem, derived the technical results, performed the experiments, and wrote the paper.

We use a Markov decision process (MDP) to represent the stochastic environment and reachability specifications to represent the supervisor's and the agent's tasks. We formulate the synthesis of optimal deceptive policies as an optimization problem that minimizes the KL divergence between the distributions of paths under the agent's policy and reference policy subject to the agent's task specification. In order to preempt the agent's deceptive policies, the supervisor may aim to design its reference policy such that any deviations from the reference policy that achieves some malicious task do not have a plausible explanation. We formulate the synthesis of optimal reference policies as a maximin optimization problem where the supervisor's optimal policy is the one that maximizes the KL divergence between itself and the agent's deceptive policy subject to the supervisor's task constraints.

The agent's problem, the synthesis of optimal deceptive policies, and the supervisor's problem, the synthesis of optimal reference policies, lead to the following questions: Is it computationally tractable to synthesize an optimal deceptive policy? Is it computationally tractable to synthesize an optimal reference policy? We show that given the supervisor's stationary policy, there exists an optimal stationary deceptive policy for the agent, and the agent's problem reduces to a convex optimization problem, which can be solved efficiently. The existence of a stationary optimal policy is not trivial since the formulated problem corresponds to a total expected cost minimization problem for constrained MDPs in the infinite undiscounted horizon where there is not an optimal stationary policy in general. On the other hand, the supervisor's problem results in a nonconvex optimization problem even when the agent uses a predetermined policy. We show that the supervisor's problem is NP-hard. We propose the gradient descent-ascent (GDA) algorithm (Nedić and Ozdaglar, 2009; Lin et al., 2020) and the alternating direction method of multipliers (ADMM) (He and Yang, 1998; Boyd et al., 2011) to solve the supervisor's optimization problem. We also give a relaxation of the problem that is a linear program.

As an extension, we consider the deception problem under partial observations. In detail, the supervisor receives partial observations of the agent's state via an ob-

servation function. The agent, on the other hand, has full observability of its own state and knows the observation function of the supervisor. Given the MDP and the observation function, the agent's policy induces a hidden Markov model (HMM). The supervisor receives observation sequences from this HMM and uses them to decide whether the agent followed the reference policy.

We use the KL divergence between the distribution of observation sequences under the agent's policy and the distribution of observation sequences under the reference policy. The value of the KL divergence is the expectation of the log-likelihood ratio between the HMM generated by the agent's policy and the HMM generated by the reference policy for a random observation sequence. The agent's problem is to find a policy that would minimize the KL divergence, making, in that sense, the two HMMs indistinguishable.

The minimization of KL divergence between the distributions of observation sequences for two HMMs is a computationally challenging task. The agent's partial observability provides greater opportunities for deception because the optimal value for the KL divergence objective function for the partially observable setting is lower than the fully observable case. However, exploiting the partial observability is computationally challenging. We show that the 3-SAT problem (Karp, 1972) can be reduced to an instance of the deception problem in the partially observable setting. Consequently, the agent's problem is NP-hard. Furthermore, we show that there is no polynomial time approximation scheme for it unless P = NP.

The computational hardness of the agent's problem in the partially observable setting is due to the large size of the policy space, the large number of observation sequences, and the stochasticity of the MDP or observation function. One can synthesize an optimal deceptive policy by solving a convex optimization problem that considers the class of history-dependent policies. However, this optimization problem would have exponentially many variables in the length of the time horizon.

We consider a smaller policy space as an approximation to the agent's problem.

A mixture policy (Collins and McNamara, 1998) is a weighted set of basis policies. We use mixture policies as the search space for the agent's problem. Since the KL objective function is a convex function of the weight vector, one can find the best mixture of any given set of policies by solving a convex optimization problem. On the other hand, the construction of the optimization problem still requires a parameter for each observation sequence. Since the number of observation sequences is potentially large, the full construction of the optimization problem is impractical. Instead, we propose to use stochastic optimization to solve this problem. We give an iterative algorithm based on stochastic gradient descent that asymptotically converges to the optimal value and outputs an optimal mixture of a given set of policies. The advantage of the algorithm is that the full construction is not required and every iteration takes polynomial time in the size of the problem.

When the transition and observation functions are deterministic, one can synthesize the optimal policy by directly optimizing the probabilities of the observation sequences. However, synthesizing an explicit policy is generally infeasible since the number of observation sequences is large. Instead of synthesizing an explicit policy, we propose a randomized algorithm that generates a single path. The algorithm boosts the probabilities of the observation sequences for which there is a path that satisfies the agent's task. The algorithm induces the optimal distribution of observation sequences and generates a path for the agent in polynomial time.

**Summary of Contributions**

- We model the deception problem in the probabilistic supervisory control setting using an MDP and the KL divergence objective function.
- In the fully observable setting,
  - we show that there exists an optimal stationary deceptive policy, and this policy can be synthesized by solving a convex optimization problem,

  - we show that the synthesis of optimal reference policies is NP-hard,
  - we propose two algorithms for the synthesis of reference policies and provide a linear programming relaxation for this problem, and
  - we demonstrate the optimal deceptive and reference policies on different numerical examples.

- In the partially observable setting,
  - we show that the synthesis of optimal deceptive policies is NP-hard and there is no polynomial time approximation scheme,
  - we consider the class of mixture policies and provide a policy synthesis algorithm that asymptotically converges to the optimal mixture,
  - we consider the class of deterministic MDPs, i.e., directed graphs, and provide a randomized algorithm that runs polynomial time in expectation and induces the optimal distribution of observation sequences, and
  - we demonstrate the synthesis of an optimal mixture policy on a numerical example.

**Outline** The rest of the chapter is organized as follows. We discuss the related work in §2.1. §2.2 provides necessary background. §2.3 focuses on the deception problem in the fully observable setting. In §2.3.1, the agent's and the supervisor's problems are presented. §2.3.2 explains the synthesis of optimal deceptive policies. In §2.3.3, we give the NP-hardness result on the synthesis of optimal reference policies. We derive the optimization problem to synthesize the optimal reference policy and give the ADMM algorithm to solve the optimization problem. In this section, we also give a relaxed problem that relies on a linear program for the synthesis of optimal reference policies. We present numerical examples in §2.3.4. We provide the proofs for the technical results in §2.3.5. §2.4 focuses on the deception problem in the partially observable setting. In §2.4.1, the agent's problems are presented. In §2.4.2, we focus

on the complexity of optimal deception under partial observability. §2.4.3 provides two algorithms for the sythesis of deceptive policies using the mixture policies and for the deterministic MDPs. We present a numerical example in §2.4.4. We provide the proofs for the technical results in §2.4.5.

## 2.1 Related Work

**Deception** Deception has been studied in the game theory framework, e.g., (Li and Cruz Jr, 2009; Zhang and Zhu, 2018; Almeshekah and Spafford, 2016), as a game between deceiving and deceived players where there is an information asymmetry or capability difference between the players. Different from existing works that consider static games or state-dependent utility functions, we consider a sequential setting with an observation sequence-dependent utility function motivated by hypothesis testing. Deception is also interpreted as the exploitation of an adversary's inaccurate beliefs on the agent's behavior (Ornik and Topcu, 2018; Karabag et al., 2019). The work (Ornik and Topcu, 2018) focuses on generating unexpected behavior conflicting with the beliefs of the adversary, and (Karabag et al., 2019) focuses on generating noninferable behavior leading to inaccurate belief distributions. On the other hand, the deceptive policy that we present generates behavior that is closest to the beliefs of the other party in order to hide the agent's malicious intentions.

The concept of opacity (Saboori and Hadjicostis, 2013; Keroglou and Hadjicostis, 2018; Bérard et al., 2015) is closely related to the notion of deception: Hiding properties of the system from an outside observer. Probabilistic system opacity (Keroglou and Hadjicostis, 2018) is the problem of determining the source of an observation sequence given a set of HMMs. Two HMMs are pairwise probabilistically opaque if the misclassification rate is a positive constant for any observation sequence. Under strong assumptions, e.g., HMMs can start from any initial state with nonzero probability, (Keroglou and Hadjicostis, 2018) shows that probabilistic opacity can be verified in polynomial time. In the course of this chapter, we show that, when

the initial state distribution is not strictly positive, the verification is NP-hard. We also consider the optimization problem of finding an HMM that is closest to a target HMM which is not studied in (Keroglou and Hadjicostis, 2018).

We explore the synthesis of optimal reference policies, which, to the best of our knowledge, has not been discussed before. We propose to use ADMM to synthesize the optimal reference policies. Similarly, (Fu et al., 2015) also used ADMM for the synthesis of optimal policies for MDPs. While we use the same method, the objective functions of these papers differ since (Fu et al., 2015) is concerned with the average reward case, whereas we use ADMM to optimize the KL divergence between the distributions of paths.

**KL divergence objective function** Similar to our approach, Bakshi and Prabhakaran (2018) used KL divergence as a proxy for the plausibility of messages in broadcast channels. While we use the KL divergence for the same purpose, the context of this chapter differs from (Bakshi and Prabhakaran, 2018). In the setting of transition systems, the works (Boularias et al., 2011; Levine and Abbeel, 2014) used the metric proposed in this chapter, the KL divergence between the distributions of paths under the agent's policy and the reference policy, for inverse reinforcement learning. In addition to the contextual difference, the proposed method of this work differs from (Boularias et al., 2011; Levine and Abbeel, 2014). We work in a setting with known transition dynamics and provide a convex optimization problem to synthesize the optimal policy while (Boularias et al., 2011; Levine and Abbeel, 2014) work with unknown dynamics and use sampling-based gradient descent to synthesize the optimal policy. KL control framework with the total cost criterion (Todorov, 2006, 2009) considers the sum of state-dependent costs and the KL divergences between the action distributions of the agent's control policy and reference policy as the objective function. When the reference policy is stationary, the sum of KL divergences for the action distributions is equivalent the objective function considered in this chapter, and the class of stationary policies suffice for optimality. Different from

(Todorov, 2006, 2009), we consider a constrained problem where stationary policies are not necessarily optimal and show the optimality of them. Entropy maximization for MDPs (Savas et al., 2019) is a special case of the deception problem where the reference policy follows every possible path with equal probability. One can synthesize optimal deceptive policies by maximizing the entropy of the agent's path distribution minus the cross-entropy of the supervisor's path distribution relative to the agent's. For the synthesis of optimal deceptive policies, we use a method similar to (Savas et al., 2019) as we represent the objective function using transition probabilities. However, our proofs for the existence and synthesis of the optimal deceptive policies significantly differ from the results of (Savas et al., 2019). In particular, (Savas et al., 2019) restricts attention to stationary policies without optimality guarantees whereas we prove the optimality of stationary policies for the deception problem. In the security framework, Kung et al. (2016); Bai et al. (2017) study the detectability of an attacker using KL divergence. While we consider an agent whose goal is to perform a reachability task in an MDP, (Kung et al., 2016; Bai et al., 2017) consider an attacker whose goal is to maximize the state estimation error of a controller in a linear dynamical system. In the context of distributionally robust optimization, Hu and Hong (2013) also considers a minimax problem with a KL divergence constraint. Using the terminology in this chapter, in Hu and Hong (2013), the agent chooses a probability distribution that (i) maximizes the objective function that is an expectation over its probability distribution and (ii) has a limited the KL divergence from a fixed reference distribution, and the supervisor aims to minimize the same objective function by choosing a set of parameters, for example the cost function of an MDP. The work that we present in this chapter differs from KL divergence constrained distributionally robust optimization since we consider that the reference distribution is chosen by the supervisor and is not fixed.

**Partially observable MDPs and decision problems for regular languages** Partially observable MDPs (POMDPs) are commonly used to model the environ-

ment of an agent with partial observability of its state. While there are existing results on the hardness of policy synthesis for POMDPs (Papadimitriou and Tsitsiklis, 1987; Madani et al., 1999; Bonet, 2009), these results do not apply to the partial observability problem studied in this chapter since we consider partial observability for an outside observer, i.e., the supervisor, and not for the ego agent. For example, planning in deterministic POMDPs (Bonet, 2009) is provably hard due to the initial state ambiguity whereas we provide an efficient algorithm for this case thanks to the full observability of the agent.

Decision problems for regular languages (Kozen, 2012; Stearns and Hunt III, 1985; Stockmeyer and Meyer, 1973) are closely related to the deception problem due to the objective function that we consider. In the course of this chapter, we show that the language containment problem (Stearns and Hunt III, 1985) is equivalent to deciding the finiteness of the KL divergence. We use the results from automata theory to establish the computational hardness of the deception problem in the partially observable setting. In addition to the qualitative analysis, we quantitatively optimize the closeness of two languages using KL divergence.

**Probabilistic supervisory control** The setting described in this chapter can be considered as a probabilistic discrete event system under probabilistic supervisory control (Pantelic et al., 2009; Lawford and Wonham, 1995). The probabilistic supervisor induces an explicit probability distribution over the language generated by the system by random disablement of the events. The supervisory control model considered in this chapter is similar in that the reference policy induces an explicit probability distribution over the paths of the MDP. Different from (Pantelic et al., 2009; Lawford and Wonham, 1995), we consider that the random disablement is done by the agent, and the supervisor is only responsible for providing the explicit random disablement strategy.

## 2.2 Preliminaries

The set $\{x = (x_1, \ldots, x_n) | x_i \geq 0\}$ is denoted by $\mathbb{R}^n_+$. The set $\{1, \ldots, n\}$ is denoted by $[n]$. The set $\{x = (x_1, \ldots, x_n) | \sum_{i=1}^n x_i = 1, x_i \geq b\}$ is denoted by $\Delta^n_b$. $|C|$ denotes the size of set $C$. The power set of $C$ is denoted by $2^C$. $Ber(p)$ is the distribution of a Bernoulli random variable with parameter $p$. $Uniform(C)$ is the uniform probability distribution over set $C$.

The characteristic function $\mathfrak{I}_C(x)$ of a set $C$ is defined as $\mathfrak{I}_C(x) = 0$ if $x \in C$ and $\infty$ otherwise. The indicator function $\mathbb{1}_y(x)$ of an element $y$ is defined as $\mathbb{1}_y(x) = 1$ if $x = y$ and 0 otherwise. The projection $Proj_C(x)$ of a variable $x \in \mathbb{R}^n$ to a set $C \subseteq \mathbb{R}^n$ is equal to $\arg\min_{y \in C} \|x - y\|_2^2$.

Let $Q_1$ and $Q_2$ be discrete probability distributions with a support $\mathcal{X}$. The *Kullback–Leibler (KL) divergence* between $Q_1$ and $Q_2$ is

$$KL(Q_1||Q_2) = \sum_{x \in \mathcal{X}} Q_1(x) \log \left( \frac{Q_1(x)}{Q_2(x)} \right).$$

We define $Q_1(x) \log \left( \frac{Q_1(x)}{Q_2(x)} \right)$ to be 0 if $Q_1(x) = 0$, and $\infty$ if $Q_1(x) > 0$ and $Q_2(x) = 0$. KL divergence is a jointly convex function in its arguments.

Let $p(y|x)$ be a conditional probability mass function. Let $W_1$ and $W_2$ be discrete probability distributions with a support $\mathcal{Y}$ such that for every $y \in \mathcal{Y}$, $W_1(y) = \sum_{x \in \mathcal{X}} Q_1(x) p(y|x)$ and $W_2(y) = \sum_{x \in \mathcal{X}} Q_2(x) p(y|x)$. *Data processing inequality* states that any (potentially) stochastic transformation $p(x|y)$ satisfies

$$KL(Q_1||Q_2) \geq KL(W_1||W_2). \tag{2.1}$$

**Remark 2.1.** *KL divergence is frequently defined with logarithm to base 2 in information theory. However, we use natural logarithm for the clarity of representation in the optimization problems. The base change does not change the results.*

### 2.2.1 Markov Decision Processes and Reachability Specifications

A *Markov decision process* (MDP) is a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, s_0)$ where $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ is a finite set of actions, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition probability function, and $s_0$ is the initial state. $\mathcal{A}(s)$ denotes the set of available actions at state $s$ where $\sum_{y \in \mathcal{S}} \mathcal{T}(s, a, y) = 1$ for all $a \in \mathcal{A}(s)$. The successor states of state $s$ is denoted by $Succ(s)$ where a state $y$ is in $Succ(s)$ if and only if there exists an action $a$ such that $\mathcal{T}(s, a, y) > 0$. State $s$ is *absorbing* if $\mathcal{T}(s, a, s) = 1$ for all $a \in \mathcal{A}(s)$.

The *history* $h_t$ at time $t$ is a sequence of states and actions such that $h_t = s_0 a_0 s_1 \dots s_{t-1} a_{t-1} s_t$. The set of all histories at time $t$ is $\mathcal{H}_t$. A *policy* for $\mathcal{M}$ is a sequence $\pi = \mu_0 \mu_1 \dots$ where each $\mu_t : \mathcal{H}_t \times \mathcal{A} \to [0, 1]$ is a function such that $\sum_{a \in \mathcal{A}(s_t)} \mu_t(h_t, a) = 1$ for all $h_t \in \mathcal{H}_t$. We use $\Pr^\pi$ to denote the probability measure induced by the policy $\pi$. A *Markovian policy* is a sequence $\pi = \mu_1 \mu_2 \dots$ where $\mu_t : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is a function such that $\sum_{a \in \mathcal{A}(s)} \mu_t(s, a) = 1$ for every $s \in \mathcal{S}$ and $t \geq 1$. A *stationary policy* is a sequence $\pi = \mu\mu \dots$ where $\mu : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is a function such that $\sum_{a \in \mathcal{A}(s)} \mu(s, a) = 1$ for every $s \in \mathcal{S}$. A *deterministic* policy is a sequence $\pi = \mu_0 \mu_1 \dots$ such that $\mu_t(\cdot, a) = 0$ or $1$ where $\cdot$ is a state or a history. The set of all policies for $\mathcal{M}$ is denoted by $\Pi(\mathcal{M})$, the set of all stationary policies for $\mathcal{M}$ is denoted by $\Pi^{St}(\mathcal{M})$, and the set of all deterministic, history-dependent policies for $\mathcal{M}$ is denoted by $\Pi^{D,H}(\mathcal{M})$. For notational simplicity, we use $\pi(s, a)$ for $\mu(s, a)$ if $\pi = \mu\mu \dots$, i.e., $\pi$ is stationary.

A stationary policy $\pi$ for $\mathcal{M}$ induces a Markov chain $\mathcal{M}^\pi = (\mathcal{S}, \mathcal{T}^\pi, s_0)$ where $\mathcal{S}$ is the finite set of states, $\mathcal{T}^\pi : \mathcal{S} \times \mathcal{S} \to [0, 1]$ is the transition probability function such that $\mathcal{T}^\pi(s, y) = \sum_{a \in \mathcal{A}(s)} \mathcal{T}(s, a, y)\pi(s, a)$ for all $s, y \in \mathcal{S}$, and $s_0$ is the initial state. A state $y$ is *accessible* from a state $s$ if there exists an $n \geq 0$ such that the probability of reaching $y$ from $s$ in $n$ steps is greater than 0 . A set $C$ of states is a *communicating class* if $y$ is accessible from $s$, and $s$ is accessible from $y$ for all $s, y \in C$. A communicating class $C$ is *closed* if $y$ is not accessible from $s$ for all $s \in C$

and $y \in \mathcal{S} \setminus C$.

A *path* $\xi = s_0 s_1 s_2 \ldots$ for an MDP $\mathcal{M}$ is an infinite sequence of states under policy $\pi = \mu_0 \mu_1 \ldots$ such that $\sum_{a \in \mathcal{A}(s_t)} \mathcal{T}(s_t, a, s_{t+1}) \mu_t(h_t, a) > 0$ for all $t \geq 0$. The set of all paths is $Paths(\mathcal{M})$. The distribution of paths for $\mathcal{M}$ under policy $\pi$ is denoted by $\Gamma^\pi$. A *k-length path fragment* $\xi = s_0 s_1 \ldots s_k$ for an MDP $\mathcal{M}$ is a sequence of states under policy $\pi = \mu_0 \mu_1 \ldots$ such that $\sum_{a \in \mathcal{A}(s_t)} \mathcal{T}(s_t, a, s_{t+1}) \mu_t(s_t, a) > 0$ for all $k > t \geq 0$. The distribution of k-length path fragments for $\mathcal{M}$ under policy $\pi$ is denoted by $\Gamma_k^\pi$. For an arbitrary policy $\pi$, $\Gamma^\pi$ may have not have a finite support. For policies $\pi^1$ and $\pi^2$, we define

$$KL(\Gamma^{\pi^1} || \Gamma^{\pi^2}) = \lim_{k \to \infty} KL(\Gamma_k^{\pi^1} || \Gamma_k^{\pi^2}).$$

We note that the limit exists since $KL(\Gamma_k^{\pi^1} || \Gamma_k^{\pi^2})$ is a monotone function of $k$. The monotonicity of $KL(\Gamma_k^{\pi^1} || \Gamma_k^{\pi^2})$ can be shown by the chain rule and non-negativity of KL divergence.

For an MDP $\mathcal{M}$ and a policy $\pi$, the *state-action occupation measure* at state $s$ and action $a$ is defined by $x_{s,a}^\pi := \sum_{t=0}^{\infty} \sum_{\substack{h_t \in \mathcal{H}_t \\ s_t = s}} \Pr^\pi(h_t | s_0) \mu_t(h_t, a)$. If $\pi$ is stationary, the state-action occupation measures satisfy $x_{s,a}^\pi = \pi(s, a) \sum_{b \in \mathcal{A}(s)} x_{s,b}^\pi$ for all $s$ with finite occupation measures. The state-action occupation measure of a state-action pair is the expected number of times that the action is taken at the state over a path. We use $x_s^\pi$ for the vector of the state-action occupation measures at state $s$ under policy $\pi$ and $x^\pi$ for the vector of all state-action occupation measures.

The event of reaching set $R$ is denoted with $\Diamond R$. We also use $\Diamond R$ to denote the reachability specification to set $R$. A path $\xi = s_0 s_1 s_2 \ldots$ satisfies $\Diamond R$ if and only if there exists $i$ such that $s_i \in R$. On an MDP $\mathcal{M}$, the probability that a specification $\Diamond R$ is satisfied under a policy $\pi$, is denoted by $\Pr^\pi(s_0 \models \Diamond R)$. The event of reaching set $R$ in $T$ steps is denoted with $\Diamond_{\leq T} R$. A path $\xi = s_0 s_1 \ldots$ satisfies $\Diamond_{\leq T} R$, i.e., $\xi \models \Diamond_{\leq T} R$, if and only if $s_i \in R$ for some $0 \leq i \leq T$.

A *nondeterministic finite automaton* (NFA) is a tuple $N = (Q, \Sigma, \Delta, q_0, F)$ where $Q$ is a finite set of states, $\Sigma$ is a finite set of input symbols, $\Delta : Q \times \Sigma \to 2^Q$

is a transition function, $q_0$ is an initial state, and $F$ is a set of accepting states such that $F \subseteq Q$.

## 2.3 Deception Under Full Observability

We consider a setting in which an agent operates in a discrete stochastic environment modeled with an MDP $\mathcal{M}$, and a supervisor provides a reference policy $\pi^{\mathsf{s}}$ to the agent. The supervisor expects the agent to follow $\pi^{\mathsf{s}}$ on $\mathcal{M}$, thereby performing $K^{\mathsf{s}}$ tasks that are specified by reachability specifications $\Diamond R_i^{\mathsf{s}}$ for all $i \in [K^{\mathsf{s}}]$. The agent aims to perform another task that is specified by the reachability specification $\Diamond R^{\mathsf{a}}$ and may deviate from the reference policy to follow a different policy $\pi^{\mathsf{a}}$. In this setting, both the agent and the supervisor know the environment, i.e., the components of $\mathcal{M}$.

While the agent operates in $\mathcal{M}$, the supervisor observes the transitions, but not the actions of the agent, to detect any deviations from the reference policy. An agent that does not want to be detected must use a deceptive policy $\pi^{\mathsf{a}}$ that limits the amount of deviations from reference policy $\pi^{\mathsf{s}}$ and achieves $\Diamond R^{\mathsf{a}}$ with high probability.

### 2.3.1 Problem Statement

We use Kullback-Leibler (KL) divergence to measure the deviation from the supervisor's policy. Recall that $\Gamma^{\pi^{\mathsf{s}}}$ and $\Gamma^{\pi^{\mathsf{a}}}$ are the distributions of paths under $\pi^{\mathsf{s}}$ and $\pi^{\mathsf{a}}$, respectively. We consider $KL(\Gamma^{\pi^{\mathsf{a}}}||\Gamma^{\pi^{\mathsf{s}}})$ as a proxy for the agent's deviations from the reference policy.

The perspective of information theory provides multiple motivations for the choice of KL divergence. The obvious motivation is that this value corresponds to the amount of information bits that the reference policy lacks while encoding the agent's path distribution. By limiting the deviations from the reference policy, we aim to make the agent's behavior easily explainable by the reference policy. Sanov's theorem (Cover and Thomas, 2012) provides the second motivation. We note that

satisfying the agent's objective with high probability is a rare event under the supervisor's policy. By minimizing the KL divergence between the policies, we make the agent's policy mimic the rare event that satisfies the agent's objective and is most probable under the supervisor's policy. Formally, let $\pi^*$ be a solution to

$$\begin{aligned} \inf_{\pi^{\mathsf{a}} \in \Pi(\mathcal{M})} \quad & KL\left(\Gamma^{\pi^{\mathsf{a}}} || \Gamma^{\pi^{\mathsf{s}}}\right) \\ \text{subject to} \quad & \Pr^{\pi^{\mathsf{a}}}(s_0 \models \Diamond R^{\mathsf{a}}) \geq \nu^{\mathsf{a}}. \end{aligned}$$

Assume that we simulate $n$ paths under the supervisor's policy. The probability that the observed paths satisfy $\Diamond R^{\mathsf{a}}$ with probability higher than $\nu^{\mathsf{a}}$ is approximately equal to $\exp(-nKL(\Gamma^{\pi^*} || \Gamma^{\pi^{\mathsf{s}}}))$. Furthermore, given that the observed path distribution satisfies $\Diamond R^{\mathsf{a}}$ with a probability higher than $\nu^{\mathsf{a}}$, the most likely distribution is $\Gamma^{\pi^*}$ (Cover and Thomas, 2012).

The choice of KL divergence is also justified from the perspective of statistical hypothesis testing. Likelihood-ratio test, the most powerful test for a given significance level (Neyman and Pearson, 1933), is a hypothesis test method to compare two models according to their goodness in terms of fitting the data. Assume that the supervisor uses likelihood-ratio test to decide whether the agent followed the reference policy or a different policy. Let $\xi_1, \ldots, \xi_n$ be the agent's paths that are observed by the supervisor. We denote the probability of $\xi_1, \ldots, \xi_n$ under $\pi^{\mathsf{s}}$ and $\pi^{\mathsf{a}}$ by $\Pr^{\pi^{\mathsf{s}}}(\xi_1, \ldots, \xi_n)$ and $\Pr^{\pi^{\mathsf{a}}}(\xi_1, \ldots, \xi_n)$, respectively. By the likelihood-ratio test, the supervisor decides that the agent uses $\pi^{\mathsf{a}}$ if $\log(\Pr^{\pi^{\mathsf{a}}}(\xi_1, \ldots, \xi_n)) - \log(\Pr^{\pi^{\mathsf{s}}}(\xi_1, \ldots, \xi_n)) \geq c$, and the agent uses $\pi^{\mathsf{s}}$ otherwise where $c$ is a constant. We note that $\pi^*$ minimizes

$$nKL(\Gamma^{\pi^{\mathsf{a}}} || \Gamma^{\pi^{\mathsf{s}}}) = \mathbb{E}\left[\log(\Pr^{\pi^{\mathsf{a}}}(\xi_1, \ldots, \xi_n)) - \log(\Pr^{\pi^{\mathsf{s}}}(\xi_1, \ldots, \xi_n))\right]$$

subject to $\Pr^{\pi^{\mathsf{a}}}(s_0 \models \Diamond R^{\mathsf{a}}) \geq \nu^{\mathsf{a}}$. Therefore, in expectation $\pi^*$ is most likely policy to be not detected by the supervisor among the policies that satisfy $\Pr^{\pi^{\mathsf{a}}}(s_0 \models \Diamond R^{\mathsf{a}}) \geq \nu^{\mathsf{a}}$.

The other motivations for the choice of KL divergence is due the Chernoff-Stein lemma and Bratagnolle-Huber inequality (Bretagnolle and Huber, 1979). Minimizing

the KL divergence is in fact equivalent to minimizing the detection rate of an attacker for an ergodic process as proved in (Bai et al., 2017) using the Chernoff-Stein lemma. For a single sample path, Bretagnolle–Huber inequality shows that the sum of the supervisor's false positive and negative rates is lower bounded by a decreasing function of the KL divergence between the distributions $\Gamma^{\pi^{\mathsf{a}}}$ and $\Gamma^{\pi^{\mathsf{s}}}$.

We propose the following problem for the synthesis of deceptive policies for the agents.

**Problem 2.1** (Synthesis of Optimal Deceptive Policies)**.** *Given an MDP* $\mathcal{M}$*, a reachability specification* $\Diamond R^{\mathsf{a}}$*, a probability threshold* $\nu^{\mathsf{a}}$*, and a reference policy* $\pi^{\mathsf{s}}$*, solve*

$$\inf_{\pi^{\mathsf{a}} \in \Pi(\mathcal{M})} \quad KL\left(\Gamma^{\pi^{\mathsf{a}}} || \Gamma^{\pi^{\mathsf{s}}}\right) \tag{2.2a}$$

$$\text{subject to} \quad \Pr^{\pi^{\mathsf{a}}}(s_0 \models \Diamond R^{\mathsf{a}}) \geq \nu^{\mathsf{a}}. \tag{2.2b}$$

*If the optimal value is attainable, find a policy* $\pi^{\mathsf{a}}$ *that is a solution to* (2.2)*.*

In order to preempt the possibility of that the agent uses a policy $\pi^{\mathsf{a}}$ that is the best deceptive policy against $\pi^{\mathsf{s}}$, the supervisor aims to find a reference policy $\pi^{\mathsf{s}}$ that maximizes the divergence between $\pi^{\mathsf{a}}$ and $\pi^{\mathsf{s}}$ subject to $\Pr^{\pi^{\mathsf{s}}}(s_0 \models \Diamond R_i^{\mathsf{s}}) \geq \nu_i^{\mathsf{s}}$ for all $i \in [K^{\mathsf{s}}]$. We assume that the supervisor knows the agent's task and propose the following problem for the synthesis of reference policies for the supervisor.

**Problem 2.2** (Synthesis of Optimal Reference Policies)**.** *Given an MDP* $\mathcal{M}$*, reachability specifications* $\Diamond R^{\mathsf{a}}$ *and* $\Diamond R_i^{\mathsf{s}}$ *for all* $i \in [K^{\mathsf{s}}]$*, probability thresholds* $\nu^{\mathsf{a}}$ *and* $\nu_i^{\mathsf{s}}$ *for all* $i \in [K^{\mathsf{s}}]$*, solve*

$$\sup_{\pi^{\mathsf{s}} \in \Pi(\mathcal{M})} \inf_{\pi^{\mathsf{a}} \in \Pi(\mathcal{M})} \quad KL\left(\Gamma^{\pi^{\mathsf{a}}} || \Gamma^{\pi^{\mathsf{s}}}\right) \tag{2.3a}$$

$$\text{subject to} \quad \Pr^{\pi^{\mathsf{a}}}(s_0 \models \Diamond R^{\mathsf{a}}) \geq \nu^{\mathsf{a}}, \tag{2.3b}$$

$$\Pr^{\pi^{\mathsf{s}}}(s_0 \models \Diamond R_i^{\mathsf{s}}) \geq \nu_i^{\mathsf{s}}, \qquad \forall i \in [K^{\mathsf{s}}]. \tag{2.3c}$$

*If the supremum is attainable, find a policy* $\pi^{\mathsf{s}}$ *that is a solution to* (2.3)*.*

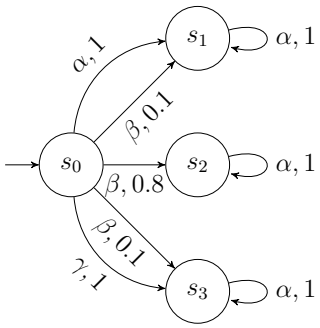



Figure 2.1: An MDP with 4 states. A label $a, p$ of a transition refers to the transition that happens with probability $p$ when action $a$ is taken.

**Example.** *We explain the synthesis of optimal deceptive policies and reference policies through the MDP* $\mathcal{M}$ *given in Figure 2.1. Note that the policies for* $\mathcal{M}$ *may vary only at* $s_0$ *since it is the only state with more than one action.*

*We first consider the synthesis of optimal deceptive policies where the reference policy satisfies* $\pi^{\mathsf{s}}(s_0, \beta) = 1$*. Consider* $\Diamond R^{\mathsf{a}} = \Diamond\{s_3\}$ *and* $\nu^{\mathsf{a}} = 0.2$*. Assume that the agent's policy has* $\pi^{\mathsf{a}}(s, \gamma) = 1$*. The value of the KL divergence is* 2.30*. However, note that as* $\pi^{\mathsf{a}}(s, \beta)$ *increases, the KL divergence decreases. In this case, the optimal policy satisfies* $\pi^{\mathsf{a}}(s, \beta) = 0.89$ *and* $\pi^{\mathsf{a}}(s, \gamma) = 0.11$ *and the optimal value for the KL divergence is* 0.04*.*

*We now consider the synthesis of optimal reference policies where the supervisor has a single specification* $\Diamond R^{\mathsf{s}} = \Diamond\{s_1, s_2\}$ *and* $\nu^{\mathsf{s}} = 0.9$*. Consider* $\Diamond R^{\mathsf{a}} = \Diamond\{s_3\}$ *and* $\nu^{\mathsf{a}} = 0.1$*. Assume that we have* $\pi^{\mathsf{s}}(s_0, \beta) = 1$*. In this case, the agent can directly follow the reference policy and make the KL divergence zero. This reference policy is not optimal; the supervisor, knowing the malicious objective of the agent, can choose the reference policy with* $\pi^{\mathsf{s}}(s_0, \alpha) = 1$*, which does not allow any deviations and makes the KL divergence infinite.*

### 2.3.2 Synthesis of Optimal Deceptive Policies

In this section, we explain the synthesis of optimal deceptive policies. Before proceeding to the synthesis step, we make assumptions to simplify the problem. Then, we show the existence of an optimal deceptive policy and give an optimization problem to synthesize one.

Without loss of generality, we make the following assumption on the target states of the agent and the supervisor for the clarity of representation. This assumption ensures that the probability of completing a task is constant, either 0 or 1, upon reaching a target state.

**Assumption 2.1.** *Every* $s \in R^{\mathsf{a}} \cup R^{\mathsf{s}}_1 \cup \ldots \cup R^{\mathsf{s}}_{K^{\mathsf{s}}}$ *is absorbing.*

We remark that in the absence of Assumption 2.1, one can still find the optimal deceptive policy by constructing a product MDP that encodes both the state of the original MDP and the statuses of the tasks. In detail, we need to construct a joint deterministic finite automaton whose states encode the statuses of the specifications for the agent and the supervisor. After creating the joint deterministic finite automaton (DFA), we construct a product MDP by combining the original MDP and the joint DFA and synthesize a policy on the product state space. Since there is a one-to-one mapping between the paths of the original MDP and the product MDP, the synthesized policy for the product MDP can be translated into a policy for the original MDP (Baier and Katoen, 2008).

If the reference policy is not stationary, we may need to compute the optimal deceptive policy by considering the parameters of the reference policy at different time steps. Such computation leads to a state explosion, which we avoid by adopting the following assumption.

**Assumption 2.2.** *The reference policy* $\pi^{\mathsf{s}}$ *is stationary on* $\mathcal{M}$.

In many applications the supervisor aims to achieve the specifications with the maximum possible probabilities. Under Assumption 2.1, stationary policies suffice to achieve the Pareto optimal curve for maximizing the probabilities of multiple reachability specifications (Etessami et al., 2007).

Without loss of generality, we assume that the optimal value of Problem 2.1 is finite. One can easily check whether the optimal value is finite in the following way. Assume that the transition probability between a pair of states is zero under the reference policy. One can create a modified MDP from $\mathcal{M}$ by removing the actions that assign a positive value to such state-state pairs. If there exists a policy that satisfies the constraint (2.2b) then the value is finite.

Given that the optimal value of Problem 2.1 is finite, we first identify the three sets of states where the agent should follow the reference policy. Firstly, the agent's policy should not be different from the supervisor's policy on the states that belong to $R^{\mathsf{a}}$, since the specification of the agent is already satisfied. Secondly, the agent should follow the reference policy at states that are recurrent under the reference policy. Formally, the reference policy $\pi^{\mathsf{s}}$ induces a Markov chain $\mathcal{M}^{\mathsf{s}}$. A state is recurrent in $\mathcal{M}^{\mathsf{s}}$ if it belongs to some closed communicating class. The agent should follow the reference policy if a state is recurrent in $\mathcal{M}^{\mathsf{s}}$.

For the second claim, we first remark that every closed communicating class $C \subset \mathcal{S}$ of $\mathcal{M}^{\mathsf{s}}$ satisfy either 1) $C \cap (\mathcal{S} \setminus R^{\mathsf{a}}) \neq \emptyset$ and $C \cap R^{\mathsf{a}} = \emptyset$, or 2) $C \cap (\mathcal{S} \setminus R^{\mathsf{a}}) = \emptyset$ and $C \cap R^{\mathsf{a}} \neq \emptyset$. This is due to the fact that $R^{\mathsf{a}}$ is a closed set, i.e., a state in $R^{\mathsf{a}}$ is reached and the states in $\mathcal{S} \setminus R^{\mathsf{a}}$ are not accessible. Hence, there cannot be a closed communicating class of $\mathcal{M}^{\mathsf{s}}$ that has states in both $R^{\mathsf{a}}$ and $\mathcal{S} \setminus R^{\mathsf{a}}$ . Let $C^{cl}$ be the union of all closed communicating classes of $\mathcal{M}^{\mathsf{s}}$, i.e., the recurrent states of $\mathcal{M}^{\mathsf{s}}$. Note that $C^{cl} \setminus R^{\mathsf{a}}$ is a closed set in $\mathcal{M}^{\mathsf{s}}$ and the states in $R^{\mathsf{a}}$ are not accessible from $C^{cl} \setminus R^{\mathsf{a}}$ in $\mathcal{M}^{\mathsf{s}}$ due to the above discussion.

Assume that under the agent's policy $\pi^{\mathsf{a}}$, there exists a path that visits a state in $C^{cl} \setminus R^{\mathsf{a}}$ and leaves $C^{cl} \setminus R^{\mathsf{a}}$ with positive probability. In this case, the KL divergence

is infinite since an event that happens with probability zero under the supervisor's policy happens with a positive probability under the agent's policy. Hence, $C^{cl} \setminus R^{\mathsf{a}}$ must also be a closed set under $\pi^{\mathsf{a}}$. Furthermore, since the agent cannot leave $C^{cl} \setminus R^{\mathsf{a}}$, and the probability of satisfying $\Diamond R^{\mathsf{a}}$ is zero upon entering $C^{cl} \setminus R^{\mathsf{a}}$, the agent should choose the same policy as the supervisor to minimize the KL divergence between the distributions of paths. Note that for the recurrent states in $R^{\mathsf{a}}$, i.e., $C^{cl} \cap R^{\mathsf{a}}$, the second claim is trivially satisfied by the first claim.

For all $s \in \mathcal{S} \setminus (C^{cl} \cup R^{\mathsf{a}})$, $s$ is transient in $\mathcal{M}^{\mathsf{s}}$, and the agent's policy must eventually stop visiting $s$, since otherwise we have infinite divergence. Furthermore, we have the following proposition.

**Proposition 2.1.** *If the optimal value of Problem 2.1 is finite and the optimal policy is $\pi^{\mathsf{a}}$, the state-action occupation measure $x^{\pi^{\mathsf{a}}}_{s,a}$ is finite for all $s \in S \setminus (C^{cl} \cup R^{\mathsf{a}})$ and $a \in \mathcal{A}(s)$.*

The occupation measures are bounded for the states that the agent's policy may differ from the supervisor's policy. Since the occupation measures are bounded, the stationary policies suffice for the synthesis of optimal deceptive policies (Altman, 1999).

**Proposition 2.2.** *For any policy $\pi^{\mathsf{a}} \in \Pi(\mathcal{M})$ that satisfies $\Pr^{\pi^{\mathsf{a}}}(s_0 \models \Diamond R^{\mathsf{a}}) \geq \nu^{\mathsf{a}}$, there exists a stationary policy $\pi^{\mathsf{a},St} \in \Pi(\mathcal{M})$ that satisfies $\Pr^{\pi^{\mathsf{a},St}}(s_0 \models \Diamond R^{\mathsf{a}}) \geq \nu^{\mathsf{a}}$ and*

$$KL\left(\Gamma^{\pi^{\mathsf{a},St}} || \Gamma^{\pi^{\mathsf{s}}}\right) \leq KL\left(\Gamma^{\pi^{\mathsf{a}}} || \Gamma^{\pi^{\mathsf{s}}}\right).$$

We remark that the existence of a stationary optimal policy is not trivial since the formulated problem corresponds to a total expected cost minimization problem for constrained MDPs in the infinite undiscounted horizon where there is not an optimal stationary policy in general.

We denote the set of states for which the agent's policy can differ from the supervisor's policy by $\mathcal{S}_d = \mathcal{S} \setminus (C^{cl} \cup R^{\mathsf{a}})$. We solve the following optimization problem

to compute the occupation measures of an optimal deceptive policy:

$$\inf \quad \sum_{s\in\mathcal{S}_d}\sum_{a\in\mathcal{A}(s)}\sum_{y\in Succ(s)} x^{\mathsf{a}}_{s,a}\mathcal{T}(s,a,y)\log\left(\frac{\sum_{b\in\mathcal{A}(s)} x^{\mathsf{a}}_{s,b}\mathcal{T}(s,b,y)}{\mathcal{T}^{\pi^{\mathsf{s}}}(s,y)\sum_{b\in\mathcal{A}(s)} x^{\mathsf{a}}_{s,b}}\right) \tag{2.4a}$$

$$\text{subject to} \quad x^{\mathsf{a}}_{s,a} \geq 0, \qquad \forall s\in\mathcal{S}_d,\ \forall a\in\mathcal{A}(s), \tag{2.4b}$$

$$\sum_{a\in\mathcal{A}(s)} x^{\mathsf{a}}_{s,a} - \sum_{y\in\mathcal{S}_d}\sum_{a\in\mathcal{A}(y)} x^{\mathsf{a}}_{y,a}\mathcal{T}(y,a,s) = \mathbb{1}_{s_0}(s), \quad \forall s\in\mathcal{S}_d, \tag{2.4c}$$

$$\sum_{y\in R^{\mathsf{a}}}\sum_{s\in\mathcal{S}_d}\sum_{a\in\mathcal{A}(s)} x^{\mathsf{a}}_{s,a}\mathcal{T}(s,a,y) + \mathbb{1}_{s_0}(y) \geq \nu^{\mathsf{a}} \tag{2.4d}$$

where $\mathcal{T}^{\pi^{\mathsf{s}}}(s,y)$ is the transition probability from $s$ to $y$ under $\pi^{\mathsf{s}}$ and the decision variables are $x^{\mathsf{a}}_{s,a}$ for all $s\in\mathcal{S}_d$ and $a\in\mathcal{A}(s)$. The objective function (2.4a) is obtained by reformulating the KL divergence between the path distributions as the sum of the KL divergences between the successor state distributions for every time step (See Lemma 2.3 in §2.3.5). The constraint (2.4c) encodes the feasible policies and the constraint (2.4d) represents the task constraint.

**Proposition 2.3.** *The optimization problem given in* (2.4) *is a convex optimization problem that shares the same optimal value with* (2.1). *Furthermore, there exists a policy* $\pi\in\Pi^{St}(\mathcal{M})$ *that attains the optimal value of* (2.4).

The optimization problem given in (2.4) gives the optimal state-action occupation measures for the agent. One can synthesize the optimal deceptive policy $\pi^{\mathsf{a}}$ using the relationship $x^{\mathsf{a}}_{s,a} = \pi^{\mathsf{a}}(s,a)\sum_{b\in\mathcal{A}(s)} x^{\mathsf{a}}_{s,b}$ for all $s\in\mathcal{S}_d$ and $\pi^{\mathsf{a}}(s,a) = \pi^{\mathsf{s}}(s,a)$ for the other states.

The optimization problem given in (2.4) can be considered as a constrained MDP problem with an infinite action space (Altman, 1999) and a nonlinear cost function. This equivalence follows from that there exists a deterministic policy that incurs the same cost on the infinite action MDP for every randomized policy for $\mathcal{M}$. Since there exists a deterministic optimal policy for the infinite MDP, we can represent the objective function and constraints of Problem 2.1 with the occupancy measure. However, we remark that (2.4) is a convex nonlinear optimization problem

whereas the constrained MDPs are often modeled with a linear cost function and solved using linear optimization methods.

**Remark 2.2.** *The methods provided in this section can be generalized to task constraints that are co-safe LTL specifications. In detail, every co-safe LTL can be translated into a DFA (Kupferman and Lampert, 2006). By combining the MDP and the DFA, we get the product MDP. Since the co-safe LTL specifications translates into reachability specifications on the product MDP and there is a one-to-one mapping between the paths of the original MDP and the product MDP, we can apply the methods described in this section to compute an optimal deceptive policy.*

### 2.3.3 Synthesis of Optimal Reference Policies

In this section, we prove the hardness of Problem 2.2. We propose the alternating direction method of multipliers and the gradient descent-ascent algorithm for synthesis of reference policies. We also derive a lower bound on the objective function and give a linear programming relaxation of Problem 2.2.

The optimization problem given in (2.4) has the supervisor's policy parameters as constants. We want to solve the optimization problem given in (2.4) to formulate the synthesis of optimal reference policies by adding the supervisor's policy parameters as additional decision variables. The set $C^{cl}$ is the set of states that belong to a closed communicating class of $\mathcal{M}^{\mathsf{s}}$. In (2.4), $C^{cl}$ is a constant set for a given reference policy, but it may vary under different reference policies. We make the following assumption to prevent set $C^{cl}$ from varying under different reference policies.

**Assumption 2.3.** *The set $C^{cl}$ is the same for all reference policies considered in Problem 2.2.*

**Remark 2.3.** *Assumption 2.3 is made for the clarity of representation. In the absence of Assumption 2.3, one can to compute the optimal reference policy for different values of $C^{cl}$. However, we remark that since, in general, $C^{cl}$ can have $\mathcal{O}(2^{|\mathcal{S}|})$ values,*

*computing the optimal reference policy for different values of $C^{cl}$ may have exponential complexity in $|\mathcal{S}|$.*

Under Assumptions 2.2 and 2.3, the optimal value of Problem 2.2 is equal to the optimal value of the following optimization problem:

$$\sup_{x^{\mathsf{s}}_{s,a}} \inf_{x^{\mathsf{a}}_{s,a}} \quad \sum_{s\in\mathcal{S}_d}\sum_{a\in\mathcal{A}(s)}\sum_{y\in Succ(s)} x^{\mathsf{a}}_{s,a}\mathcal{T}(s,a,y)\log\left(\frac{\sum_{b\in\mathcal{A}(s)} x^{\mathsf{a}}_{s,b}\mathcal{T}(s,b,y)}{\mathcal{T}^{\pi^{\mathsf{s}}}(s,y)\sum_{b\in\mathcal{A}(s)} x^{\mathsf{a}}_{s,b}}\right) \tag{2.5a}$$

subject to (2.4b) – (2.4d)

$$\mathcal{T}^{\pi^{\mathsf{s}}}(s,y) = \sum_{a\in\mathcal{A}(s)} \mathcal{T}(s,a,y)\frac{x^{\mathsf{s}}_{s,a}}{\sum_{b\in\mathcal{A}(s)} x^{\mathsf{s}}_{s,b}},\ \forall s\in\mathcal{S}_d,\ \forall y\in\mathcal{S}, \tag{2.5b}$$

$$x^{\mathsf{s}}_{s,a}\geq 0, \qquad \forall s\in\mathcal{S}_d,\ \forall a\in\mathcal{A}(s), \tag{2.5c}$$

$$\sum_{a\in\mathcal{A}(s)} x^{\mathsf{s}}_{s,a} - \sum_{y\in\mathcal{S}_d}\sum_{a\in\mathcal{A}(y)} x^{\mathsf{s}}_{y,a}\mathcal{T}(y,a,s) = \mathbb{1}_{s_0}(s), \forall s\in\mathcal{S}_d, \tag{2.5d}$$

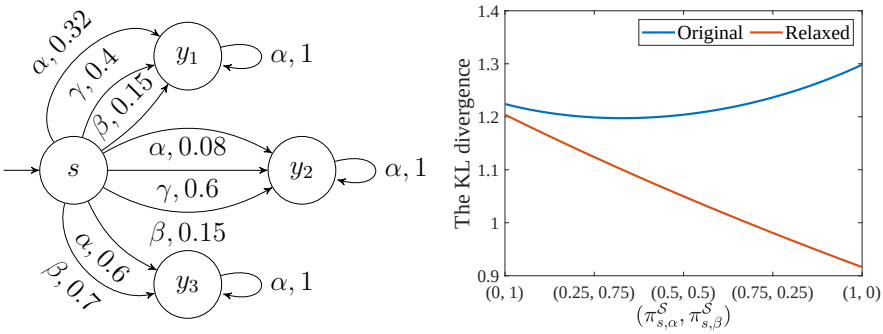
$$\sum_{y\in R^{\mathsf{s}}_i}\sum_{s\in\mathcal{S}_d\setminus C_{\mathsf{s}}}\sum_{a\in\mathcal{A}(s)} x^{\mathsf{s}}_{s,a}\mathcal{T}(s,a,y) + \mathbb{1}_{s_0}(y) \geq \nu^{\mathsf{s}}_i,\ \forall i\in[K^{\mathsf{s}}] \tag{2.5e}$$

where $x^{\mathsf{s}}_{s,a}$ variables are the decision variables for the supervisor and $x^{\mathsf{a}}_{s,a}$ variables are the decision variables for the agent.

**Remark 2.4.** *The optimization problem given in* (2.5) *has undefined points due to the denominators in* (2.5a) *and* (2.5b)*, that are ignored in the above optimization problem for the clarity of representation. If $\sum_{a\in\mathcal{A}(s)} x^{\mathsf{s}}_{s,a} = 0$, then the state $s$ is unreachable and if the KL divergence between the policies is finite, the state must be unreachable also under $\pi^{\mathsf{a}}$. Hence there is no divergence at state $s$. If $\mathcal{T}^{\pi^{\mathsf{s}}}(s,y) = 0$ and if the KL divergence between the policies is finite, $x^{\mathsf{a}}_{s,y}$ must be 0. Hence there is no divergence for state $s$ and successor state $y$.*

We can show the existence of an optimal reference policy if the condition given in Proposition 2.4 is satisfied. This condition ensures that the objective function of the problem in (2.5) is finite for all pairs of the supervisor's and the agent's policies.

**Proposition 2.4.** *If $\mathcal{T}(s,a,y) > 0$ for all $s\in\mathcal{S}_d$, $a\in\mathcal{A}(s)$, and $y\in Succ(s)$, then there exists a policy $\pi^{\mathsf{s}}$ that attains the optimal value of the optimization problem given in* (2.5).




(a) (b)

Figure 2.2: (a) An MDP with 4 states. A label $a, p$ of a transition refers to the transition that happens with probability $p$ when action $a$ is taken. (b) The KL divergence between the path distributions of the agent and the supervisor for different reference policies. Note that there are two local optima that maximizes the KL divergence.

We note that the optimization problem given in (2.5) is nonconvex. One might wonder whether there exists a problem formulation that yields a convex optimization problem. We first observe that it is possible that there are multiple locally optimal reference policies. For example, consider the MDP given in Figure 2.2a where the specification of the agent is $\Pr^{\pi^{\mathsf{a}}}(s \models \Diamond\{y_1, y_2\}) = 1$. Regardless of the reference policy, the agent's policy must have $\pi^{\mathsf{a}}(s, \gamma) = 1$ due to his specification. For simplicity, there is no specification for the supervisor, i.e., $\nu^{\mathsf{s}}$ is 0. The optimal reference policy maximizes $0.4 \log(0.4/(0.32x^{\mathsf{s}}_{s_0,\alpha} + 0.15x^{\mathsf{s}}_{s_0,\beta} + 0.4x^{\mathsf{s}}_{s_0,\gamma})) + 0.6 \log(0.6/(0.08x^{\mathsf{s}}_{s_0,\alpha} + 0.15x^{\mathsf{s}}_{s_0,\beta} + 0.6x^{\mathsf{s}}_{s_0,\gamma}))$, which is a convex function of $x^{\mathsf{s}}_{s_0,\alpha}$, $x^{\mathsf{s}}_{s_0,\beta}$, and $x^{\mathsf{s}}_{s_0,\gamma}$. There are two locally optimal reference policies for Problem 2.2: the policy that satisfies $\pi^{\mathsf{s}}(s, \alpha) = 1$ and the policy that satisfies $\pi^{\mathsf{s}}(s, \beta) = 1$. Hence, the problem is not only nonconvex but also possibly multimodal.

We consider a new parametrization to reformulate the optimization problem given in (2.5). Consider a continuous and bijective transformation from the occupation measures to the new parameters, that makes new parameters to span all stationary policies. After this transformation, an optimal solution to (2.5) yields an optimal

solution in the new parameter space. If the optimization problem given in (2.5) has multiple local optima, then any reformulation spanning all stationary policies for the supervisor has multiple local optima. Furthermore, in §2.3.3.1, we show that Problem 2.2 is a provably hard problem. In §2.3.3.2 and §2.3.3.3, we describe two approaches based on alternating direction method of multipliers and gradient descent-ascent to solve (2.5). Finally, we present a relaxation of the problem in §2.3.3.4 that relies on solving a linear program.

#### 2.3.3.1 The Complexity of the Synthesis of Optimal Reference Policies

In this section, we show that the synthesis of an optimal reference policy is NP-hard whereas the feasibility problem for reference policies can be solved in polynomial time.

Finding a feasible policy under multiple reachability constraints has polynomial complexity in the number states and actions for a given MDP. When the target states are absorbing, the complexity of the problem is also polynomial in the number of constraints (Etessami et al., 2007). This result follows from that a feasible policy can be synthesized with a linear program where the numbers of variables and constraints are polynomial in the number of states, actions, and task constraints.

Matsui (1996) transformed the set partition problem to the decision version of an instance of linear multiplicative programming and proved the NP-hardness of linear multiplicative programming. In the proof of Proposition 2.5, we give an instance of Problem 2.2 whose decision problem is equivalent into the decision problem of the instance of linear multiplicative programming that Matsui provided.

While a feasible reference policy can be synthesized in polynomial time by solving a linear program, the complexity of finding an optimal reference policy is NP-hard even when the target states are absorbing. Formally we have the following result.

**Proposition 2.5.** *Problem 2.2 is NP-hard even under Assumption 2.1.*

#### 2.3.3.2 Alternating Direction Method of Multipliers (ADMM)-based Approach for the Synthesis of Optimal Reference Policies

The alternating direction method of multipliers (ADMM) (He and Yang, 1998; Boyd et al., 2011) is an algorithm to solve decomposable optimization problems by solving smaller pieces of the problem. We use the ADMM to solve the optimization problem given in (2.5). The objective function of (2.5) is decomposable since it is a sum across $\mathcal{S}_d$ where each summand consists of different variables. We exploit this feature to solve smaller problems size via the ADMM.

For every state $s \in \mathcal{S}_d$, we introduce $z_s^{\mathsf{a}}$ and $z_s^{\mathsf{s}}$ such that $z_s^{\mathsf{a}} = x_s^{\mathsf{a}}$ and $z_s^{\mathsf{s}} = x_s^{\mathsf{s}}$. With these extra variables, the augmented Lagrangian of (2.5) is

$$
\begin{aligned}
&L(x^{\mathsf{s}}, x^{\mathsf{a}}, z^{\mathsf{s}}, z^{\mathsf{a}}, \lambda^{\mathsf{s}}, \lambda^{\mathsf{a}}) \\
&= \left( \sum_{s \in \mathcal{S}_d} \left( \sum_{a \in \mathcal{A}(s)} \sum_{y \in \mathcal{S}} x_{s,a}^{\mathsf{a}} \mathcal{T}(s,a,y) \log \left( \frac{\sum_{b \in \mathcal{A}(s)} x_{s,b}^{\mathsf{a}} \mathcal{T}(s,b,y)}{\mathcal{T}^{\pi^{\mathsf{s}}}(s,y) \sum_{b \in \mathcal{A}(s)} x_{s,b}^{\mathsf{a}}} \right) \right) + \mathcal{I}_{\mathbb{R}_{\geq 0}^{|\mathcal{A}(s)|}}(x_s^{\mathsf{s}}) + \mathcal{I}_{\mathbb{R}_{\geq 0}^{|\mathcal{A}(s)|}}(x_s^{\mathsf{a}}) \right. \\
&\quad \left. - -\rho^{\mathsf{s}}(x_s^{\mathsf{s}} - z_s^{\mathsf{s}})^T \lambda_s^{\mathsf{s}} + \rho^{\mathsf{a}}(x_s^{\mathsf{a}} - z_s^{\mathsf{a}})^T \lambda_s^{\mathsf{a}} - \frac{\rho^{\mathsf{s}}}{2} \|x_s^{\mathsf{s}} - z_s^{\mathsf{s}}\|_2^2 + \frac{\rho^{\mathsf{a}}}{2} \|x_s^{\mathsf{a}} - z_s^{\mathsf{a}}\|_2^2 \right) \\
&\quad - \mathcal{I}_{X^{\mathsf{s}}}(z^{\mathsf{s}}) + \mathcal{I}_{X^{\mathsf{a}}}(z^{\mathsf{a}}),
\end{aligned}
$$

where $\rho^{\mathsf{s}}$ and $\rho^{\mathsf{a}}$ are positive constants, $\lambda^{\mathsf{s}}$ and $\lambda^{\mathsf{a}}$ are the dual parameters, $X^{\mathsf{a}}$ is the set of occupation measures of the agent that satisfy (2.4c) and (2.4d), $X^{\mathsf{s}}$ is the set of occupation measures of the supervisor that satisfy (2.5d) and (2.5e), and $\mathcal{T}^{\pi^{\mathsf{s}}}(s,y) = \sum_{a \in \mathcal{A}(s)} \mathcal{T}(s,a,y) x_{s,a}^{\mathsf{s}} / (\sum_{b \in \mathcal{A}(s)} x_{s,b}^{\mathsf{s}})$ for all $s \in \mathcal{S}_d$ and $a \in \mathcal{A}(s)$. In Algorithm 1, we give the ADMM for the synthesis of reference policies. Note that we optimize $x^{\mathsf{s}}$ and $x^{\mathsf{a}}$ together to capture the characteristics of the maximin problem.

We remark that Algorithm 1 still requires solving a maximin optimization problem (see line 7). However, the maximin optimization problem in Algorithm 1 can be solved as a local maximin problem separately for each state since $x_s^{\mathsf{s}}$ and $x_s^{\mathsf{a}}$ are decoupled from $x_y^{\mathsf{s}}$ and $x_y^{\mathsf{a}}$ for all $s \neq y \in \mathcal{S}_d$. While the number of variables for the original maximin problem is $\mathcal{O}(|\mathcal{S}||\mathcal{A}|)$, it is $\mathcal{O}(|\mathcal{A}|)$ for the local problems in

**Algorithm 1:** The ADMM for the synthesis of reference policies

**Input:** An MDP $\mathcal{M}$, reachability specifications $\Diamond R_i^{\mathsf{s}}$ for all $i \in K^{\mathsf{s}}$ and $\Diamond R^{\mathsf{a}}$, probability thresholds $\nu_i^{\mathsf{s}}$ for all $i \in [K^{\mathsf{s}}]$ and $\nu^{\mathsf{a}}$.
**Output:** A reference policy $\pi^{\mathsf{s}}$.
Set $x^{\mathsf{s},0}$ and $z^{\mathsf{s},0}$ arbitrarily from $X^{\mathsf{s}}$.
Set $x^{\mathsf{a},0}$ and $z^{\mathsf{a},0}$ arbitrarily from $X^{\mathsf{a}}$.
Set $\lambda^{\mathsf{s},0}$ and $\lambda^{\mathsf{a},0}$ to 0. $k = 0$.
**while** *stopping criteria are not satisfied* **do**
  Set $x^{\mathsf{s},k+1}$ and $x^{\mathsf{a},k+1}$ as the solution of $\max_{x^{\mathsf{s}}} \min_{x^{\mathsf{a}}} L(x^{\mathsf{s}}, x^{\mathsf{a}}, z^{\mathsf{s},k}, z^{\mathsf{a},k}, \lambda^{\mathsf{s},k}, \lambda^{\mathsf{s},k})$.
  $z^{\mathsf{s},k+1} := Proj_{X^{\mathsf{s}}}(x^{\mathsf{s},k+1} + \lambda^{\mathsf{s},k})$.
  $z^{\mathsf{a},k+1} := Proj_{X^{\mathsf{a}}}(x^{\mathsf{a},k+1} + \lambda^{\mathsf{a},k})$.
  $\lambda^{\mathsf{s},k+1} := \lambda^{\mathsf{s},k} + x^{\mathsf{s},k+1} - z^{\mathsf{s},k+1}$.
  $\lambda^{\mathsf{a},k+1} := \lambda^{\mathsf{a},k} + x^{\mathsf{a},k+1} - z^{\mathsf{a},k+1}$.
  $k := k + 1$.
Compute $\pi^{\mathsf{s}}$ using $z^{\mathsf{s},k}$ as the occupation measure.
**return** $\pi^{\mathsf{s}}$

the ADMM algorithm. To solve the local maximin problems, one can use the dual problem for the agent's decision variables and solve nonconvex maximization problems (see §2.3.4 for numerical results). The details of the dualization-based approach is given in (Karabag et al., 2021b).

**Remark 2.5.** *Convergence of ADMM for monotone variational inequalities and convex-concave saddle-point problems has been studied (He and Yang, 1998). To the best of our knowledge, the ADMM for the nonconvex-concave optimization problem given in* (2.5) *has no convergence guarantees and does not match with the any of the existing convergence results.*

#### 2.3.3.3 Gradient Descent-Ascent for the Synthesis of Optimal Reference Policies

In this section, we describe the gradient descent-ascent (GDA) method (Nedić and Ozdaglar, 2009; Lin et al., 2020) for the synthesis of reference policies. The

objective function using the occupancy measures of the agent and supervisor is

$$f(x^{\mathsf{a}}, x^{\mathsf{s}}) = \sum_{s \in \mathcal{S}_d} \sum_{a \in \mathcal{A}(s)} \sum_{y \in Succ(s)} x^{\mathsf{a}}_{s,a} \mathcal{T}(s,a,y) \log\left(\left(\frac{\sum_{b \in \mathcal{A}(s)} x^{\mathsf{a}}_{s,b} \mathcal{T}(s,b,y)}{\sum_{b \in \mathcal{A}(s)} x^{\mathsf{a}}_{s,b}}\right)\left(\frac{\sum_{b \in \mathcal{A}(s)} x^{\mathsf{s}}_{s,b}}{\sum_{b \in \mathcal{A}(s)} x^{\mathsf{s}}_{s,b} \mathcal{T}(s,b,y)}\right)\right).$$

Algorithm 2 takes simultaneous gradient steps for the supervisor and the agent, and performs projections onto the respective occupancy measure spaces.

**Algorithm 2:** GDA for the synthesis of reference policies

**Input:** An MDP $\mathcal{M}$, reachability specifications $\Diamond R^{\mathsf{s}}_i$ for all $i \in K^{\mathsf{s}}$ and $\Diamond R^{\mathsf{a}}$, probability thresholds $\nu^{\mathsf{s}}_i$ for all $i \in [K^{\mathsf{s}}]$ and $\nu^{\mathsf{a}}$, step sizes $\alpha^{\mathsf{s}}$ and $\alpha^{\mathsf{a}}$.
**Output:** A reference policy $\pi^{\mathsf{s}}$.
Set $z^{\mathsf{s},0}$ arbitrarily from $X^{\mathsf{s}}$.
Set $z^{\mathsf{a},0}$ arbitrarily from $X^{\mathsf{a}}$.
$k = 0$.
**while** *stopping criteria are not satisfied* **do**
  $z^{\mathsf{s},k+1} := Proj_{X^{\mathsf{s}}}(z^{\mathsf{s},k} + \alpha^{\mathsf{s}} \nabla_{z^{\mathsf{s}}} f(z^{\mathsf{a},k}, z^{\mathsf{s},k}))$.
  $z^{\mathsf{a},k+1} := Proj_{X^{\mathsf{a}}}(z^{\mathsf{a},k} - \alpha^{\mathsf{a}} \nabla_{z^{\mathsf{a}}} f(z^{\mathsf{a},k}, z^{\mathsf{s},k}))$.
  $k := k + 1$.
Compute $\pi^{\mathsf{s}}$ using $z^{\mathsf{s},k}$ as the occupation measure.
**return** $\pi^{\mathsf{s}}$

We remark that the computation of the gradients has polynomial complexity in the number of decision variables and is decomposable over $\mathcal{S}_d$ for both the supervisor and the agent.

#### 2.3.3.4 A Linear Programming Relaxation for the Synthesis of Reference Policies

We give a convex relaxation of (2.5). Synthesizing a policy that minimizes the probability of satisfying the agent's specification is an intuitive way to increase the KL divergence between the distributions of paths. Formally, consider a transformation of the path distributions that groups paths of $\mathcal{M}$ into two subsets: the paths that satisfy $\Diamond R^{\mathsf{a}}$ and the paths that do not satisfy $\Diamond R^{\mathsf{a}}$. After this transformation,

the probability assigned to the first subset is $\Pr^{\pi^{\mathsf{s}}}(s_0 \models \Diamond R^{\mathsf{a}})$ under policy $\pi^{\mathsf{s}}$ and $\Pr^{\pi^{\mathsf{a}}}(s_0 \models \Diamond R^{\mathsf{a}})$ under policy $\Diamond R^{\mathsf{a}}$. By the data processing inequality given in (2.1), this transformation yields a lower bound on the KL divergence between the path distributions: $KL\left(\Gamma^{\pi^{\mathsf{a}}} || \Gamma^{\pi^{\mathsf{s}}}\right)$ is greater than or equal to

$$KL\left(Ber\left(\Pr^{\pi^{\mathsf{a}}}\left(s_0 \models \Diamond R^{\mathsf{a}}\right)\right) || Ber\left(\Pr^{\pi^{\mathsf{s}}}\left(s_0 \models \Diamond R^{\mathsf{a}}\right)\right)\right). \tag{2.6}$$

We use this lower bound to construct the relaxed problem

$$\begin{aligned} \sup_{\pi^{\mathsf{s}} \in \Pi(\mathcal{M})} \inf_{\pi^{\mathsf{a}} \in \Pi(\mathcal{M})} \quad & (2.6) && \text{(2.7a)} \\ \text{subject to} \quad & \Pr^{\pi^{\mathsf{a}}}(s_0 \models \Diamond R^{\mathsf{a}}) \geq \nu^{\mathsf{a}}, && \text{(2.7b)} \\ & \Pr^{\pi^{\mathsf{s}}}(s_0 \models \Diamond R_i^{\mathsf{s}}) \geq \nu_i^{\mathsf{s}},\ i \in [K^{\mathsf{s}}]. && \text{(2.7c)} \end{aligned}$$

If $\Pr^{\pi^{\mathsf{s}}}(s_0 \models \Diamond R^{\mathsf{a}}) \geq \nu^{\mathsf{a}}$, the agent may directly use the reference policy. Without loss of generality, assuming that $\Pr^{\pi^{\mathsf{s}}}(s_0 \models \Diamond R^{\mathsf{a}}) < \nu^{\mathsf{a}}$, the objective function of above optimization problem is decreasing in $\Pr^{\pi^{\mathsf{s}}}(s_0 \models \Diamond R^{\mathsf{a}})$ and increasing in $\Pr^{\pi^{\mathsf{a}}}(s_0 \models \Diamond R^{\mathsf{a}})$. Hence, the problem

$$\begin{aligned} \sup_{\pi^{\mathsf{s}} \in \Pi(\mathcal{M})} \inf_{\pi^{\mathsf{a}} \in \Pi(\mathcal{M})} \quad & \Pr^{\pi^{\mathsf{a}}}(s_0 \models \Diamond R^{\mathsf{a}}) - \Pr^{\pi^{\mathsf{s}}}(s_0 \models \Diamond R^{\mathsf{a}}) && \text{(2.8a)} \\ \text{subject to} \quad & \Pr^{\pi^{\mathsf{a}}}(s_0 \models \Diamond R^{\mathsf{a}}) \geq \nu^{\mathsf{a}}, && \text{(2.8b)} \\ & \Pr^{\pi^{\mathsf{s}}}(s_0 \models \Diamond R_i^{\mathsf{s}}) \geq \nu_i^{\mathsf{s}},\ i \in [K^{\mathsf{s}}]. && \text{(2.8c)} \end{aligned}$$

shares the same optimal policies with the problem given in (2.7). We note that the optimization problem given in (2.8) can be solved separately for the supervisor's and the agent's parameters where both of the problems are linear optimization problems. The optimal reference policy for the relaxed problem is the policy that minimizes $\Pr^{\pi^{\mathsf{s}}}(s_0 \models \Diamond R^{\mathsf{a}})$ subject to $\Pr^{\pi^{\mathsf{s}}}(s_0 \models \Diamond R_i^{\mathsf{s}}) \geq \nu_i^{\mathsf{s}}$ for all $i \in [K^{\mathsf{s}}]$.

The lower bound given in (2.6) provides a sufficient condition on the optimality of a reference policy for Problem 2.2. A policy $\pi^{\mathsf{s}}$ satisfying $\Pr^{\pi^{\mathsf{s}}}(s_0 \models \Diamond R^{\mathsf{a}}) = 0$ and $\Pr^{\pi^{\mathsf{s}}}(s_0 \models \Diamond R_i^{\mathsf{s}}) \geq \nu_i^{\mathsf{s}}$ for all $i \in [K^{\mathsf{s}}]$ is an optimal reference policy since the

optimization problem given in (2.7) has the optimal value of $\infty$. However, in general the gap due to the relaxation may get arbitrarily large, and the reference policy synthesized via (2.7) is not necessarily optimal for Problem 2.2. For example, consider the MDP given in Figure 2.2a where the agent's policy again has $\pi^{\mathsf{a}}(s, \gamma) = 1$. For simplicity, there is no specification for the supervisor, i.e., $\nu^{\mathsf{s}}$ is 0. The policy $\pi^{\mathsf{s}}$ that minimizes $\Pr^{\pi^{\mathsf{s}}}(s \models \Diamond\{y_1, y_2\})$ chooses action $\beta$ at state $s$. This policy has a KL divergence value of 1.22. On the other hand, a policy that chooses action $\alpha$ is optimal and it has a KL divergence value of 1.30 even though it does not minimize the probability of satisfying $\Diamond\{y_1, y_2\}$. The gap of the lower bound may get arbitrarily large as $\mathcal{T}(s, \alpha, y_2)$ decreases. Furthermore, the policy synthesized via the relaxed problem may not even be locally optimal as $\mathcal{T}(s, \alpha, y_2)$ decreases.

The relaxed problem focuses on only one event, achieving the malicious objective, and fails to capture all transitions of the agent. On the other hand, the objective function of Problem 2.2, the KL divergence between the path distributions, captures all transitions of the agent rather than a single event. In particular, to detect the deviations the optimal deceptive policy assigns a low probability to the transition from $s$ to $y_2$ which inevitably happens with high probability for the agent. However, the policy synthesized via the relaxed problem fails to capture that the agent have to assign high probability to the transition from $s$ to $y_2$.

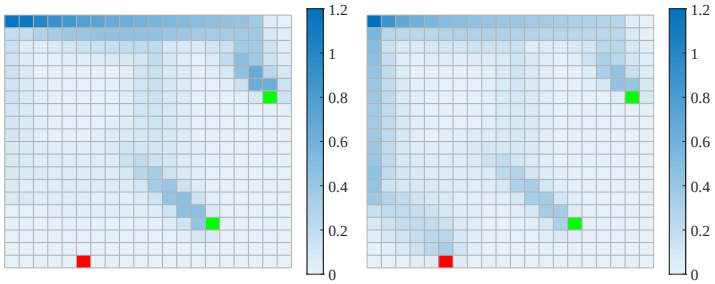
### 2.3.4 Numerical Examples

In this section we give numerical examples on the synthesis of optimal deceptive policies and optimal reference policies. In Section 2.3.4.1 we explain some characteristics of the optimal deceptive policies through different scenarios. In the second example given in Section 2.3.4.2, we compare the proposed metric, the KL divergence between the distributions of paths, to some other metrics. We demonstrate the synthesis of reference policies in Section 2.3.4.3. For the ADMM-based approach, we utilize the dual problem for the agent's decision variables for the local maximin problems and solve nonconvex maximization problems.

We solved the convex optimization problems with CVX (Grant and Boyd, 2014) toolbox using MOSEK (Aps, 2020) and the nonconvex optimization problems using IPOPT (Wächter and Biegler, 2006).

#### 2.3.4.1 Some Characteristics of Deceptive Policies

The first example demonstrates some of the characteristics of the optimal deceptive policies. The environment is a $20 \times 20$ grid world given in Figure 2.3. The green and red states are denoted with sets $g$ and $r$, respectively. At every state, there are 4 available actions, namely, up, down, left, and right. When the agent takes an action the transition happens into the target direction with probability 0.7 and in the other directions uniformly randomly with probability 0.3. If a direction is out of the grid, the transition probability of that direction is proportionally distributed to the other directions. The green and red states are absorbing. The initial state is the top-left state.



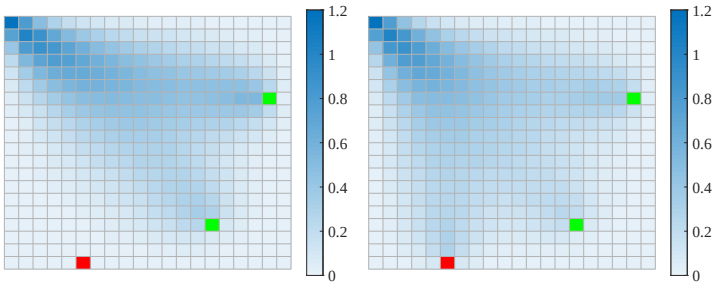

(a) Reference (b) Deceptive

Figure 2.3: Heatmaps of the occupation measures. The value of a state is the expected number of visits to the state. The deceptive policy makes the agent move towards the red state to achieve the malicious objective.

The specification of the supervisor is to reach each of the green states with probability at least 0.4. The reference policy is constructed so that it reaches the green states with probability at least $1 - 10^{-4}$ in the minimum expected time. The specification of the agent is to reach the red state. The specification is encoded with $\Diamond r$. The probability threshold $\nu^{\mathsf{a}}$ for the agent's specification is 0.3. We synthesize the

policy of the agent according to Problem 2.1, which leads to the KL divergence value of 2.662. While the reference policy satisfies $\Diamond r$ with probability $10^{-4}$, the agent's policy satisfies $\Diamond r$ with probability 0.3.

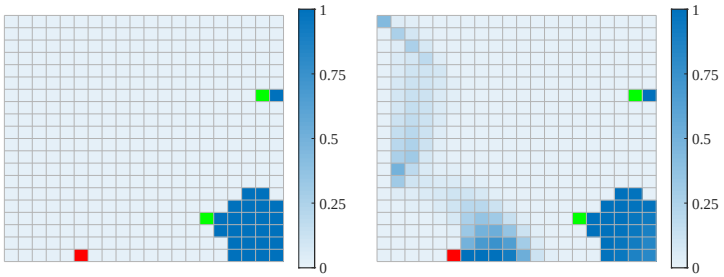
In Figure 2.3b, we observe that if the deceptive agent is close to the green states, it does not try to reach the red state since deviations from the reference policy in these regions incur high divergence. Instead, as we see in Figure 2.4, the deceptive policy makes the agent move towards left in the first steps and reach the red state by going down. The misleading occurs during this period: while the agent goes left on purpose, it may hold the stochasticity of the environment accountable for this behavior. We also observe a significant detail in the agent's deceptive policy. The deceptive policy aims to reach the left border since the reference policy takes action down in this region. The agent wants to drive himself to this region to directly follow the reference policy without any divergence. Thus the agent deviates from the reference policy at a particular state to be close to the reference policy as much as possible in the rest of the path. Once the agent is close to the red state, it again deviates from the reference policy and takes action down with a high probability to reach the red state.



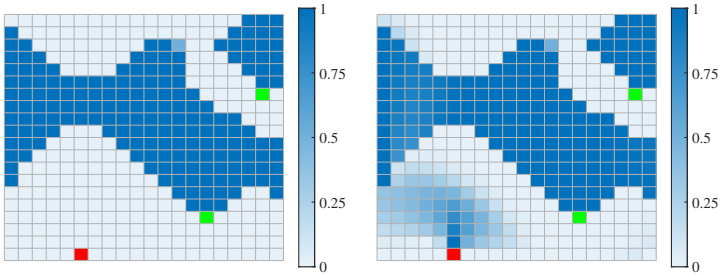

(a) Supervisor (b) Agent

Figure 2.5: Heatmaps of the occupation measures under the alternative reference policy. The deceptive policy is hard to detect under a reference policy that is not restrictive.

We note that the reference policy is restrictive in this case; as can be seen in Figure 2.3a, it follows almost a deterministic path. Under such a reference policy,




(a) Supervisor - Left (b) Agent - Left




(c) Supervisor - Down (d) Agent - Down

Figure 2.4: The assigned probabilities to the actions when the yellow state was visited, but the red state was not visited.

even the policy that is synthesized via Problem 2.1 is easy to detect. To observe the effect of the reference policy on the deceptive policy, we consider a different reference policy as shown in Figure 2.5a, which satisfies $\Diamond r$ with probability $10^{-3}$. When the reference policy is not as restrictive, the deceptive policy becomes hard to detect. Formally, the value of the KL divergence reduces to 1.462.

#### 2.3.4.2 Detection of a Deceptive Agent

In this example, by comparing KL divergence with some common metrics to synthesize the deceptive policies, we show how the choice of KL divergence helps with preventing detection. We compare the metrics using a randomly generated MDP and an MDP modeling a region from San Francisco.

The randomly generated MDP consists of 21 states. In particular, there are 20 transient states with 4 actions and an absorbing state with 1 action. For the transient states, each action has a successor state that is chosen uniformly randomly

among the transient states. In addition to these actions, every transient state has an action that has the absorbing state as the successor state. At every transient state, the reference policy goes to the absorbing state with probability 0.15 and the other successor states with probability 0.85. The agent's specification $\phi^{\mathsf{a}}$ is to reach to a specific transient state.

We randomly generate a reference policy for the randomly generated MDP. The reference policy satisfies the agent's specification with probability 0.30. For the reference policy, we synthesize three candidate policies for deception: by minimizing the KL divergence between the path distributions of the agent's policy and the reference policies, by minimizing the $L_1$-norm between the occupation measures of the state-action pairs for the agent's policy and the reference policies, and by minimizing the $L_2$-norm between the occupation measures of the state-action pairs for the agent's policy and the reference policies. The candidate policies are constructed so that they satisfy the agent's specification $\phi^{\mathsf{a}}$ with probability 0.9. For each candidate policy, we run 100 simulations each of which consists of 100 independently sampled paths.

We also simulate the agent's trajectories under the reference policies. In particular, we aim to observe the case where the empirical probability of satisfying $\phi^{\mathsf{a}}$ is approximately 0.9. Note that this is a rare event under the reference policy. We simulate this rare event in the following way. Let $\Gamma^{\pi^{\mathsf{s}}}$ be the probability distribution of paths under the reference policy. We create two conditional probability distributions $\Gamma_{+}^{\pi^{\mathsf{s}}}$ and $\Gamma_{-}^{\pi^{\mathsf{s}}}$ which are the distribution of paths under the reference policy given that the paths satisfy $\phi^{\mathsf{a}}$ and do not satisfy $\phi^{\mathsf{a}}$, respectively. We sample from $\Gamma_{-}^{\pi^{\mathsf{s}}}$ with probability 0.9 and $\Gamma_{-}^{\pi^{\mathsf{s}}}$ with probability 0.1.

In addition to the randomly generated MDP, we use a different MDP to show that the deceptive policy can help patrolling without being detected. The MDP models a region in the north east of San Francisco. The map of the region is given in Figure 2.6 where each intersection is represented with a state and each road is represented with an action. We design the reference policy to represent the average

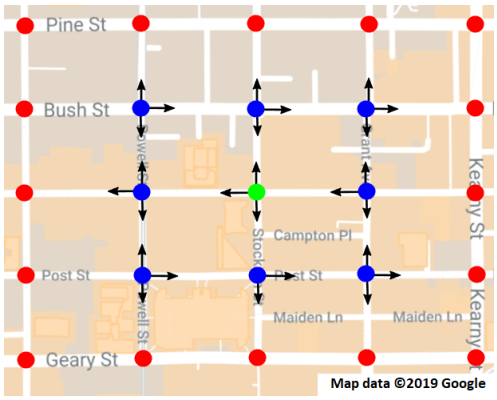
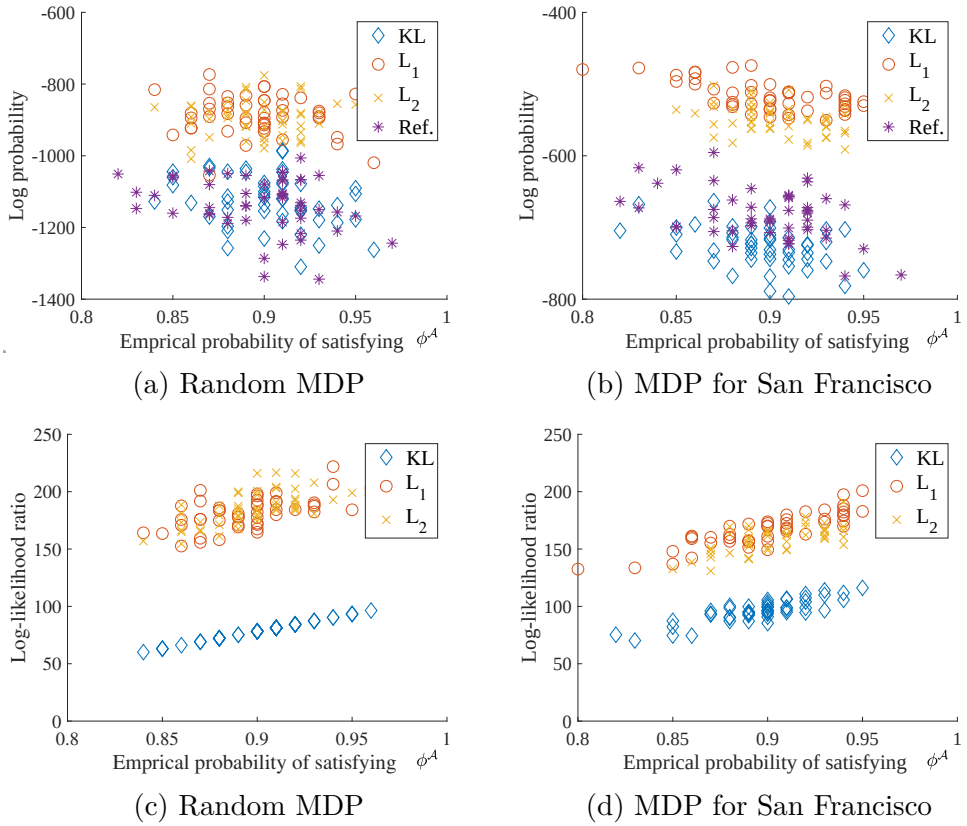



Figure 2.6: The map of a region from northeast of San Francisco. The green dot indicates the intersection at which the highest number of crimes happened. The data is from (Alamdari et al., 2014). The dots on the map represent the states of the MDP and the arrows represent the available actions. The initial state is chosen uniformly randomly among the blue states and the red states are absorbing. The agent aims to patrol the green state.

driver behavior. We obtain the traffic density data from Google Maps (Google) and synthesize the reference policy by fitting a stationary policy to the data. The aim of the agent is to patrol the intersection at which the highest number of crimes happens. Formally, the agent's policy reaches the intersection with probability at least 0.9 whereas the reference policy reaches the intersection with probability 0.28. For the simulation, we use the steps as in the randomly generated MDP.

For each simulation, we plot the log-probability under the reference policy and the log-likelihood ratio compared to the reference policy in Figure 2.7. In Figures 2.7a and 2.7b, we observe that the metrics based on $L_1$ and $L_2$ norms generate paths that are more probable under the reference policy compared to KL divergence metric. However, we also observe that the probabilities for the KL divergence metric looks closer to the probabilities of the simulations of the reference policy that are satisfying $\phi^{\mathsf{a}}$ with high probability. The paths generated via the KL divergence coincides with the typical set of this rare event. In other words, the other metrics provide paths that have higher probabilities under the reference policy. However, given that the empirical distribution of the paths satisfy $\phi^{\mathsf{a}}$ with high probability, the likelihood

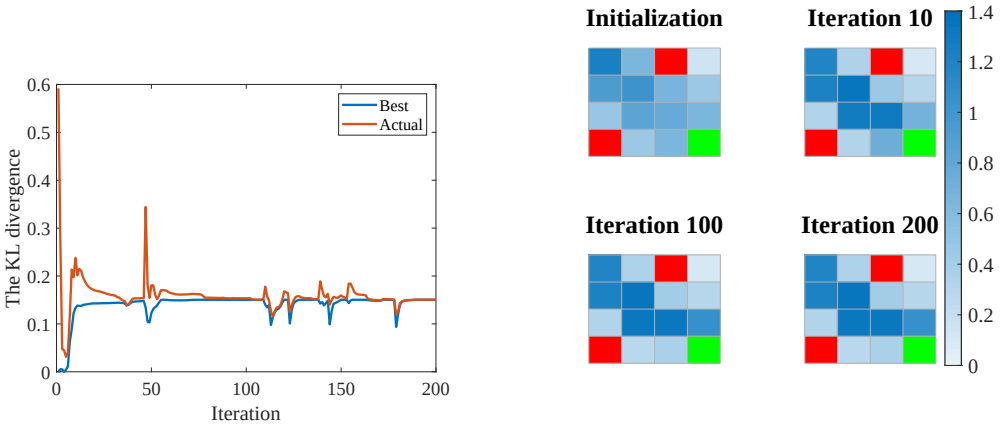



(a) Random MDP (b) MDP for San Francisco

(c) Random MDP (d) MDP for San Francisco

Figure 2.7: (a)-(b) The log-probabilities of the generated paths under the reference policy. 'Ref.' refers to the rare events of the reference policy that satisfies $\phi^{\mathsf{a}}$ with high probability. 'KL', '$L_1$', and '$L_2$' refer to the candidate deceptive policies. (c)-(d) The log-likelihood ratios between candidate deceptive policies and the reference policy for different simulations.

of the path generated by the KL divergence is higher and the supervisor expects to see paths that are close to the paths generated by the KL divergence metric. This phenomenon confirms the motivation provided by Sanov's theorem which is explained in Section 2.3.1. In Figures 2.7c and 2.7d, we observe that the paths generated under the KL divergence metric has a lower log-likelihood ratio compared to the other metrics as explained in §2.3.1. This result shows that compared to the other candidate deceptive policies, the deceptive policy generated with the KL divergence is the least likely to be detected under the likelihood-ratio test.



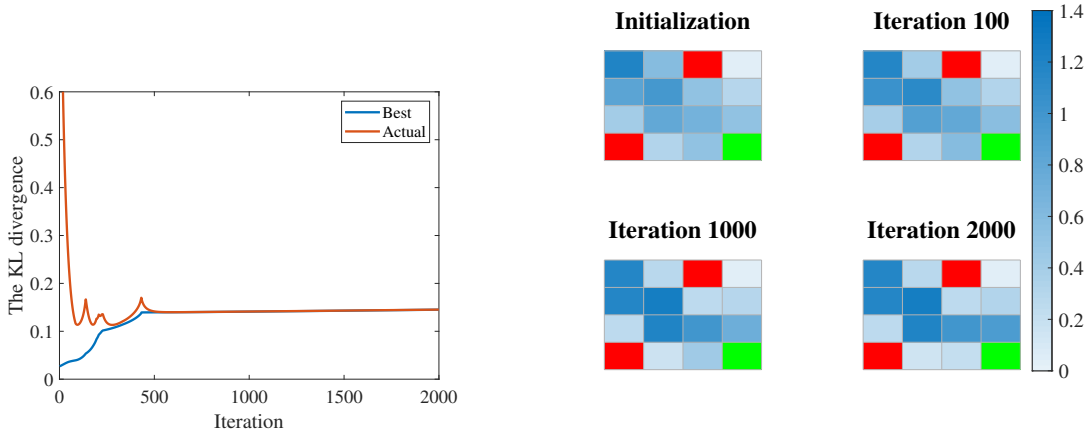

(a) (b)

Figure 2.8: (a) The KL divergence between the agent's policy and the reference policy. The curve "Best" refers to the case that the agent's policy is the best deceptive policy against the reference policy synthesized during the ADMM algorithm. The curve "Actual" refers to the case that the agent's policy is the policy synthesized during the ADMM algorithm. (b) Heatmaps of the occupation measures for the reference policy, i.e., $z^{s,k}$ parameters of the Algorithm 1. The value of a state is the expected number of visits to the state.

#### 2.3.4.3 Optimal Reference Policies

We present an example of synthesis of optimal reference policies. The environment is a $4 \times 4$ grid world given in Figure 2.8b and is similar to the environment described in the example for the characteristics of deceptive policies. The green and red states are denoted with sets $g$ and $r$, respectively. At every state, there are 4 available actions, namely, up, down, left, and right, at every state. When the agent takes an action the transition happens into the target direction with probability 0.7 and in the other directions uniformly randomly with probability 0.3. If a direction is out of the grid the transition probability to that direction is proportionally distributed to the other directions. The green state is absorbing and the initial state is the top-left state.

The specification of the supervisor is to reach the green state, i.e., $\Diamond g$. Note that the specification of the supervisor is satisfied with probability 1 under any policy.




(a) (b)

Figure 2.9: (a) The KL divergence between the agent's policy and the reference policy. The curve "Best" refers to the case that the agent's policy is the best deceptive policy against the reference policy synthesized during the GDA algorithm. The curve "Actual" refers to the case that the agent's policy is the policy synthesized during the GDA algorithm. (b) Heatmaps of the occupation measures for the reference policy, i.e., $z^{\mathsf{s},k}$ parameters of the Algorithm 2. The value of a state is the expected number of visits to the state.

The specification of the agent is to reach one of the red states, i.e., $\Diamond r$. The probability threshold for the agent's task is 0.3.

We synthesize reference policies via Algorithms 1 and 2. For both algorithms, $z^{\mathsf{s},k}$ represents the reference policy synthesized at iteration $k$. Similarly, $z^{\mathsf{a},k}$ represents the deceptive policy synthesized at iteration $k$. For the ADMM algorithm, we plot the values of the KL divergences between these policies in Figure 2.8a and give the heatmaps for the occupation measures in Figure 2.8b. For the GDA algorithm, we plot the values of the KL divergences between these policies in Figure 2.9a and give the heatmaps for the occupation measures in Figure 2.9b. After few tens of iterations of the ADMM algorithm, the KL divergence value is near to the limit value which is 0.150. GDA algorithm generates a similar result in nearly 500 iterations. However, we remark that the per iteration complexity of the GDA algorithm is significantly lower than the ADMM algorithm.

In Figure 2.8a, we also note that if the actual KL divergence value increases suddenly, the best response KL divergence value decreases. The reference policy tries to exploit suboptimal deceptive policies. While this exploitation increases the actual value, it causes suboptimality for the reference policy against the best deceptive policy.

The reference policy gradually gets away from the red states as shown in Figures 2.8b and 2.9b. Based on this observation, we expect that the relaxed problem given in §2.3.3.4 provides useful reference policies for the original problem. This expectation is indeed verified numerically: The reference policy synthesized via the relaxed problem, has a KL divergence of 0.150, which within 2% of the objective values generated by the ADMM and GDA algorithms.

### 2.3.5 Proofs for the Technical Results

We use the following definition and lemmas in the proof of Proposition 2.1. We use $\Pr^{\pi}(s \models \bigcirc\Diamond s)$ to denote the probability that $s$ is visited again from initial state $s$ under the stationary policy $\pi$ .

**Definition 2.1.** *Let $Q$ be a probability distribution with a countable support $\mathcal{X}$. The entropy of $Q$ is $H(Q) = -\sum_{x\in\mathcal{X}} Q(x)\log(Q(x))$.*

**Lemma 2.1** (Theorem 5.7 of (Conrad, 2004))**.** *Let $\mathcal{D}$ be the set of a distributions with support $\{1, 2, \ldots\}$ and the expected value of $c$. A geometric random variable $X^* \sim Geo(1/c)$ maximizes $H(X)$ subject to $X \in \mathcal{D}$ where*

$$H(X^*) = c\left(-\frac{1}{c}\log\left(\frac{1}{c}\right) - \left(1-\frac{1}{c}\right)\log\left(1-\frac{1}{c}\right)\right) = cH\left(Ber\left(\frac{1}{c}\right)\right).$$

**Lemma 2.2.** *Consider an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, s_0)$. Let $N_s^{\pi}$ denote the number of visits to the state $s$ under a stationary policy $\pi$ such that $\mathbb{E}[N_s^{\pi}] < \infty$. $N_s^{\pi}$ satisfies*

$$\Pr(N_s^{\pi} = 0) = \Pr^{\pi}(s_0 \not\models \Diamond s)$$

*and*

$$\Pr(N_s^{\pi} = i) = \Pr^{\pi}(s_0 \models \Diamond s)\Pr^{\pi}(s \models \bigcirc\Diamond s)^{i-1}\Pr^{\pi}(s \not\models \bigcirc\Diamond s).$$

*Proof of Proposition 2.1.* We prove this proposition by contradiction. We first provide a lower bound for the objective function of Problem 2.1. Then, we show that as the state-action occupation measures approach to infinity, the lower bound approaches to infinity. Hence, the state-action occupation measures must be bounded in order to have a finite value for the objective function of Problem 2.1.

Let $d^*$ be the optimal value of Problem 2.1. For a state $s \in S \setminus C^{cl}$, first consider the case $\Pr^{\pi^{\mathsf{s}}}(s_0 \models \Diamond s) = 0$, i.e., $s$ is unreachable under $\pi^{\mathsf{s}}$. In this case, the agent's policy $\pi^{\mathsf{a}}$ must satisfy $\Pr^{\pi^{\mathsf{a}}}(s_0 \models \Diamond s) = 0$, i.e., $s$ must be unreachable under $\pi^{\mathsf{a}}$, otherwise the KL divergence is infinite. Hence the occupation measure is zero in this case.

Consider $\Pr^{\pi^{\mathsf{s}}}(s_0 \models \Diamond s) > 0$. For this case, we will show that if the occupation measure is greater than some finite value, then the KL divergence between the path distributions is greater than $d^*$. Denote the number visits to $s$ with $N_s^{\pi^{\mathsf{a}}}$ and $N_s^{\pi^{\mathsf{s}}}$ under $\pi^{\mathsf{a}}$ and $\pi^{\mathsf{s}}$, respectively. We have the following claim: Given $\Pr^{\pi^{\mathsf{s}}}(s_0 \models \Diamond s) > 0$, $\Pr^{\pi^{\mathsf{s}}}(s \models \bigcirc\Diamond s) \in [0, 1)$, and $d^* > 0$, there exists an $M_s$ such that for all $\pi^{\mathsf{a}}$ that satisfies $\mathbb{E}[N_s^{\pi^{\mathsf{a}}}] > M_s$, we have $KL(\Gamma^{\pi^{\mathsf{a}}} || \Gamma^{\pi^{\mathsf{s}}}) > d^*$.

We consider a partitioning of paths according to the number of times $s$ appears in a path. By the data processing inequality given in (2.1), we have that $KL(\Gamma^{\pi^{\mathsf{a}}} || \Gamma^{\pi^{\mathsf{s}}}) \geq KL(N_s^{\pi^{\mathsf{a}}} || N_s^{\pi^{\mathsf{s}}})$, i.e., the KL divergence between the path distributions is lower bounded by the KL divergence between the distributions of number visits to $s$. Therefore it suffices to prove the following claim: Given $\Pr^{\pi^{\mathsf{s}}}(s_0 \models \Diamond s) > 0$, $\Pr^{\pi^{\mathsf{s}}}(s \models \bigcirc\Diamond s) \in [0, 1)$, and $d^* > 0$, there exists an $M_s$ such that for all $\pi^{\mathsf{a}}$ that satisfies $\mathbb{E}[N_s^{\pi^{\mathsf{a}}}] > M_s$, we have $KL(N_s^{\pi^{\mathsf{a}}} || N_s^{\pi^{\mathsf{s}}}) > d^*$.

Define a random variable $\hat{N}_s^{\pi^{\mathsf{a}}}$ such that $\Pr(\hat{N}_s^{\pi^{\mathsf{a}}} = i) = \Pr(N_s^{\pi^{\mathsf{a}}} = i | N_s^{\pi^{\mathsf{a}}} > 0)$. For notational convenience denote $r^{\mathsf{s}} = 1 - \Pr^{\pi^{\mathsf{s}}}(s_0 \models \Diamond s)$, $l^{\mathsf{s}} = \Pr^{\pi^{\mathsf{s}}}(s \models \bigcirc\Diamond s)$, $p_i = \Pr(N_s^{\pi^{\mathsf{a}}} = i)$ and $\hat{p}_i = \Pr(\hat{N}_s^{\pi^{\mathsf{a}}} = i)$. Also define $M_s^{\mathsf{a}} := \mathbb{E}[N_s^{\pi^{\mathsf{a}}}]$, $\hat{M}_s^{\mathsf{a}} := \mathbb{E}[\hat{N}_s^{\pi^{\mathsf{a}}}] = \frac{M_s^{\mathsf{a}}}{1-p_0}$, and $M_s^{\mathsf{s}} := \mathbb{E}[N_s^{\pi^{\mathsf{s}}}]$.

We want to show that $M_s^{\mathsf{a}}$ is bounded for a finite $d^*$. Assume that $M_s^{\mathsf{a}} \leq M_s^{\mathsf{s}}$.

In this case the $M_s^{\mathsf{a}}$ is finite since $M_s^{\mathsf{s}}$ is finite. If $M_s^{\mathsf{a}} > M_s^{\mathsf{s}}$, we have

$$
\begin{aligned}
&KL(N_s^{\pi^{\mathsf{a}}} || N_s^{\pi^{\mathsf{s}}}) \\
&= p_0 \log\left(\frac{p_0}{r^{\mathsf{s}}}\right) + \sum_{i=1}^{\infty} p_i \log\left(\frac{p_i}{(1-r^{\mathsf{s}})(l^{\mathsf{s}})^{i-1}(1-l^{\mathsf{s}})}\right) && \text{(2.9a)} \\
&= p_0 \log\left(\frac{p_0}{r^{\mathsf{s}}}\right) + \sum_{i=1}^{\infty} (1-p_0)\hat{p}_i \log\left(\frac{1-p_0}{1-r^{\mathsf{s}}}\right) + \sum_{i=1}^{\infty} (1-p_0)\hat{p}_i \log\left(\frac{\hat{p}_i}{(l^{\mathsf{s}})^{i-1}(1-l^{\mathsf{s}})}\right) && \text{(2.9b)} \\
&= p_0 \log\left(\frac{p_0}{r^{\mathsf{s}}}\right) + (1-p_0) \log\left(\frac{1-p_0}{1-r^{\mathsf{s}}}\right) + \sum_{i=1}^{\infty} (1-p_0)\hat{p}_i \log\left(\frac{\hat{p}_i}{(l^{\mathsf{s}})^{i-1}(1-l^{\mathsf{s}})}\right) && \text{(2.9c)} \\
&\geq (1-p_0) \sum_{i=1}^{\infty} \hat{p}_i \log\left(\frac{\hat{p}_i}{(l^{\mathsf{s}})^{i-1}(1-l^{\mathsf{s}})}\right) && \text{(2.9d)} \\
&= (1-p_0) \sum_{i=1}^{\infty} \hat{p}_i \log\left(\hat{p}_i\right) - (1-p_0) \sum_{i=1}^{\infty} \hat{p}_i \log\left((l^{\mathsf{s}})^{i-1}(1-l^{\mathsf{s}})\right) && \text{(2.9e)} \\
&= -(1-p_0) H(\hat{N}_s^{\pi^{\mathsf{a}}}) - (1-p_0) \sum_{i=1}^{\infty} \hat{p}_i \log\left((l^{\mathsf{s}})^{i-1}(1-l^{\mathsf{s}})\right) && \text{(2.9f)}
\end{aligned}
$$

where the equality (2.9a) follows from Lemma 2.2. The inequality in (2.9d) holds since the removed terms correspond to $KL(Ber(p_0)||Ber(r^{\mathsf{s}}))$ which is nonnegative.

By using Lemma 2.1 to upper bound $H(\hat{N}_s^{\pi^{\mathsf{a}}})$ and the definitions we have the following inequality.

$$KL(N_s^{\pi^{\mathsf{a}}}||N_s^{\pi^{\mathsf{s}}}) \geq (p_0 - 1)\left(H(\hat{N}_s^{\pi^{\mathsf{a}}}) + \sum_{i=1}^{\infty} \hat{p}_i \log\left((l^{\mathsf{s}})^{i-1}(1-l^{\mathsf{s}})\right)\right) \tag{2.10a}$$

$$\geq (p_0 - 1)\left(\hat{M}_s^{\mathsf{a}} H\left(Ber\left(\frac{1}{\hat{M}_s^{\mathsf{a}}}\right)\right) + \sum_{i=1}^{\infty} \hat{p}_i \log\left((l^{\mathsf{s}})^{i-1}(1-l^{\mathsf{s}})\right)\right) \tag{2.10b}$$

$$= (p_0 - 1)\left(\hat{M}_s^{\mathsf{a}} H\left(Ber\left(\frac{1}{\hat{M}_s^{\mathsf{a}}}\right)\right) + \sum_{i=1}^{\infty} \hat{p}_i \left((i-1)\log(l^{\mathsf{s}}) + \log(1-l^{\mathsf{s}})\right)\right) \tag{2.10c}$$

$$= (p_0 - 1)\left(\hat{M}_s^{\mathsf{a}} H\left(Ber\left(\frac{1}{\hat{M}_s^{\mathsf{a}}}\right)\right) + \left(\log(1-l^{\mathsf{s}}) + \left(\hat{M}_s^{\mathsf{a}} - 1\right)\log(l^{\mathsf{s}})\right)\right) \tag{2.10d}$$

$$= (1 - p_0)\left(-\hat{M}_s^{\mathsf{a}} H\left(Ber\left(\frac{1}{\hat{M}_s^{\mathsf{a}}}\right)\right)\right.$$

$$= M_s^{\mathsf{a}}\left(KL\left(Ber\left(\frac{1}{\hat{M}_s^{\mathsf{a}}}\right)||Ber(1-l^{\mathsf{s}})\right)\right) \tag{2.10e}$$

Now assume that $M_s^{\mathsf{a}} \geq \frac{c}{1-l^{\mathsf{s}}}$ where $c > 1$ is a constant. In this case, we have

$$KL(N_s^{\pi^{\mathsf{a}}}||N_s^{\pi^{\mathsf{s}}}) \geq M_s^{\mathsf{a}}\left(KL\left(Ber\left(\frac{1}{\hat{M}_s^{\mathsf{a}}}\right)||Ber(1-l^{\mathsf{s}})\right)\right) \tag{2.11a}$$

$$\geq M_s^{\mathsf{a}}\left(KL\left(Ber\left(\frac{1}{M_s^{\mathsf{a}}}\right)||Ber\left(1-l^{\mathsf{s}}\right)\right)\right) \tag{2.11b}$$

$$\geq M_s^{\mathsf{a}}\left(KL\left(Ber\left(\frac{1-l^{\mathsf{s}}}{c}\right)||Ber\left(1-l^{\mathsf{s}}\right)\right)\right) \tag{2.11c}$$

since $\hat{M}_s^{\mathsf{a}} > M_s^{\mathsf{a}}$ and for a variable $x$ such that $x \geq \frac{1}{1-l^{\mathsf{s}}}$, the value of $KL(Ber(\frac{1}{x})||Ber(1-l^{\mathsf{s}}))$ is increasing in $x$.

Note that $KL\left(Ber\left(\frac{1-l^{\mathsf{s}}}{c}\right)||Ber\left(1-l^{\mathsf{s}}\right)\right)$ is a positive constant. We can easily see that there exists an $M_s$ such that $KL(N_s^{\pi^{\mathsf{a}}}||N_s^{\pi^{\mathsf{s}}}) > d^*$ if $M_s^{\mathsf{a}} > M_s$.

We proved that for a given constant, for every transient state of the supervisor the occupancy measure under the agent's policy must be bounded by some constant otherwise the KL divergence between distributions for the number of states to this

state is greater than the constant. Since the KL divergence between the path distributions is lower bounded by the KL divergence for states, the finiteness of the KL divergence between the path distributions implies that the occupancy measure under the agent's policy for every transient state of the supervisor.

Thus, if the optimal value of Problem 2.1 is finite, the occupation measures under $\pi^{\mathsf{a}}$ must be bounded by some $M_s < \infty$ for all $s \in S \setminus C^{cl}$. ■

***Proof Sketch for Proposition 2.2***. Assume that the KL divergence between the path distributions is finite. Note that the occupation measures of $\pi^{\mathsf{a}}$ are finite for all $s \in \mathcal{S}_d = \mathcal{S} \setminus (C^{cl} \cup R^{\mathsf{a}})$.

When the reference policy is stationary, we may transform $\mathcal{M}$ into a *semi-infinite MDP*. The semi-infinite MDP shares the same states with $\mathcal{M}$, but has continuous action space such that for all states every randomized action of $\mathcal{M}$ is an action of the semi-infinite MDP. Also the states belong to $R^{\mathsf{a}}$ and $C^{cl}$ are absorbing in the semi-infinite MDP.

Let $P_s^{\mathsf{s}}$ be the successor state distribution at state $s$ under the reference policy in the semi-infinite MDP. At state $s \in \mathcal{S}_d$, an action $a$ with successor state distribution $P_{s,a}$ has cost $KL(P_{s,a}||P_s^{\mathsf{s}})$. The cost is 0 for the other states that do not belong to $\mathcal{S}_d$. Consider an optimization problem that minimizes the expected cost subject to reaching $R^{\mathsf{a}}$ with probability at least $\nu^{\mathsf{a}}$. The result of this optimization problem shares the same value with the result of Problem 2.1. This problem is a constrained cost minimization for an MDP where the only decision variables are the state-action occupation measures. An optimal policy can be characterized by the state-action occupation measures.

The occupation measures must be finite for all $s \in \mathcal{S}_d$ as we showed in Proposition 2.1. Since every finite occupation measure vector of $\mathcal{S}_d$ can also be achieved by a stationary policy, there exists a stationary policy which shares the same occupation measures with an optimal policy (Altman, 1999). Hence, this stationary policy is also optimal.

Now assume that the stationary optimal policy $\pi^*$ is randomized. Let $\pi^*_s$ be the action distribution and $P^{\pi^*}_s$ be the successor state distribution at state $s$ under $\pi^*$. Note that at state $s$ there exists an action $a^*$ that has $\mathcal{T}(s, a^*, y) = P^{\pi}_s(y)$ since the action space is convex for the semi-infinite MDP. Also due to the convexity of KL divergence we have $\int_{\Delta^{|\mathcal{A}(s)|}} KL(P_{s,a}||P^{\mathsf{s}}_s)d\pi^*_s(a) \geq KL(P^{\pi^*}_s||P^{\mathsf{s}}_s)$ where $\Delta^{|\mathcal{A}(s)|}$ is $|\mathcal{A}(s)|$-dimensional probability simplex. Hence, deterministically taking action $a^*$ is optimal for state $s$. By generalizing this argument to all $s \in \mathcal{S}_s$, we conclude that there exists an optimal stationary deterministic policy for the semi-infinite MDP. Without loss of generality we assume $\pi^*$ is stationary deterministic.

We note that the stationary deterministic policy $\pi^*$ of the semi-infinite MDP corresponds to a stationary randomized policy for the original MDP $\mathcal{M}$. Hence the proposition holds. ■

We remark that the proof of Lemma 2.3 is fairly similar with the proof of Lemma 2 from (Savas et al., 2019).

**Lemma 2.3.** *The KL divergence $KL(\Gamma^{\pi^{\mathsf{a}}}_k||\Gamma^{\pi^{\mathsf{s}}}_k)$ between the distributions of $k$-length path fragments for stationary policies $\pi^{\mathsf{a}}$ and $\pi^{\mathsf{s}}$ is equal to the expected sum of KL divergences between the successor state distributions of $\pi^{\mathsf{a}}$ and $\pi^{\mathsf{s}}$ that is*

$$\sum_{t=0}^{k-1} \sum_{s \in \mathcal{S}_d} \Pr^{\pi^{\mathsf{a}}}(s_t = s) \sum_{y \in Succ(s)} \sum_{a \in \mathcal{A}(s)} \mathcal{T}(s, a, y)\pi^{\mathsf{a}}(s, a) \log \left( \frac{\sum_{b \in \mathcal{A}(s)} \mathcal{T}(s, b, y)\pi^{\mathsf{a}}(s, b)}{\sum_{b \in \mathcal{A}(s)} \mathcal{T}(s, b, y)\pi^{\mathsf{s}}(s, b)} \right).$$

*Furthermore, if $KL(\Gamma^{\pi^{\mathsf{a}}}||\Gamma^{\pi^{\mathsf{s}}})$ is finite, it is equal to*

$$\sum_{s \in \mathcal{S}_d} \sum_{y \in \mathcal{S}_d} \sum_{a \in \mathcal{A}(s)} \mathcal{T}(s, a, y) x^{\mathsf{a}}_{s,a} \log \left( \frac{\sum_{b \in \mathcal{A}(s)} \mathcal{T}(s, b, y) x^{\mathsf{a}}_{s,b}}{\mathcal{T}^{\pi^{\mathsf{s}}}(s, y) \sum_{b \in \mathcal{A}(s)} x^{\mathsf{a}}_{s,b}} \right)$$

*Proof of Lemma 2.3.* For MDP $\mathcal{M}$, denote the set of $k$-length path fragments by $\Xi_k$ and the probability of the $k$-length path fragment $\xi_k = s_0 s_1 \dots s_k$ under the stationary policy $\pi$ by $\Pr^{\pi}(\xi_k)$. We have $\Pr^{\pi}(\xi_k) = \prod_{t=0}^{k-1} \sum_{a \in \mathcal{A}(s_t)} \mathcal{T}(s_t, a, s_{t+1})\pi(s_t, a)$. Consequently, we have

$$
\begin{aligned}
&KL(\Gamma_k^{\pi^{\mathsf{a}}}||\Gamma_k^{\pi^{\mathsf{s}}}) \\
&= \sum_{\xi_k \in \Xi_k} \Pr^{\pi^{\mathsf{a}}}(\xi_k) \log\left(\frac{\Pr^{\pi^{\mathsf{a}}}(\xi_k)}{\Pr^{\pi^{\mathsf{s}}}(\xi_k)}\right) \\
&= \sum_{t=0}^{k-1} \sum_{\xi_k \in \Xi_k} \Pr^{\pi^{\mathsf{a}}}(\xi_k) \log\left(\frac{\sum_{b\in\mathcal{A}(s_t)} \mathcal{T}(s_t,b,s_{t+1})\pi^{\mathsf{a}}(s_t,b)}{\sum_{b\in\mathcal{A}(s_t)} \mathcal{T}(s_t,b,s_{t+1})\pi^{\mathsf{s}}(s_t,b)}\right) \\
&= \sum_{t=0}^{k-1} \sum_{\xi_k \in \Xi_k} \Pr^{\pi^{\mathsf{a}}}(\xi_k) \sum_{s\in\mathcal{S}_d} \mathbb{1}_s(s_t) \sum_{y\in Succ(s)} \mathbb{1}_y(s_{t+1}|s_t=s) \log\left(\frac{\sum_{b\in\mathcal{A}(s_t)} \mathcal{T}(s,b,y)\pi^{\mathsf{a}}(s,b)}{\sum_{b\in\mathcal{A}(s_t)} \mathcal{T}(s,b,y)\pi^{\mathsf{s}}(s,b)}\right) \\
&= \sum_{t=0}^{k-1} \sum_{s\in\mathcal{S}_d} \Pr^{\pi^{\mathsf{a}}}(s_t=s) \sum_{y\in Succ(s)} \sum_{a\in\mathcal{A}(s_t)} \mathcal{T}(s,a,y)\pi^{\mathsf{a}}(s,b) \log\left(\frac{\sum_{b\in\mathcal{A}(s_t)} \mathcal{T}(s,b,y)\pi^{\mathsf{a}}(s,b)}{\sum_{b\in\mathcal{A}(s_t)} \mathcal{T}(s,b,y)\pi^{\mathsf{s}}(s,b)}\right)
\end{aligned}
$$

If $KL(\Gamma^{\pi^{\mathsf{a}}}||\Gamma^{\pi^{\mathsf{s}}})$ is finite, we have

$$
\begin{aligned}
&KL(\Gamma^{\pi^{\mathsf{a}}}||\Gamma^{\pi^{\mathsf{s}}}) \\
&= \lim_{k\to\infty} KL(\Gamma_k^{\pi^{\mathsf{a}}}||\Gamma_k^{\pi^{\mathsf{s}}}) \\
&= \lim_{k\to\infty} \sum_{s\in\mathcal{S}_d} \sum_{y\in Succ(s)} \sum_{a\in\mathcal{A}(s_t)} \sum_{t=0}^{k-1} \Pr^{\pi^{\mathsf{a}}}(s_t=s)\mathcal{T}(s,a,y)\pi^{\mathsf{a}}(s,a) \log\left(\frac{\sum_{b\in\mathcal{A}(s_t)} \mathcal{T}(s,b,y)\pi^{\mathsf{a}}(s,b)}{\sum_{b\in\mathcal{A}(s_t)} \mathcal{T}(s,b,y)\pi^{\mathsf{s}}(s,b)}\right) \\
&= \sum_{s\in\mathcal{S}_d} \sum_{y\in Succ(s)} \sum_{a\in\mathcal{A}(s)} \mathcal{T}(s,a,y)x_{s,a}^{\mathsf{a}} \log\left(\frac{\sum_{b\in\mathcal{A}(s)} \mathcal{T}(s,b,y)x_{s,b}^{\mathsf{a}}}{\mathcal{T}^{\pi^{\mathsf{s}}}(s,y)\sum_{b\in\mathcal{A}(s)} x_{s,b}^{\mathsf{a}}}\right).
\end{aligned}
$$

Finally, since $\mathcal{T}(s,a,y)$ is zero for all $y \notin Succ(s)$ and we defined $0\log 0 = 0$, we can safely replace $Succ(s)$ with $\mathcal{S}$. ■

*Proof of Proposition 2.3.* Assume that $KL(\Gamma^{\pi^{\mathsf{a}}}||\Gamma^{\pi^{\mathsf{s}}})$ is finite under the stationary policies $\pi^{\mathsf{a}}$ and $\pi^{\mathsf{s}}$. The objective function of the problem given in (2.2) is equal to

$$
\sum_{s\in\mathcal{S}_d} \sum_{y\in Succ(s)} \sum_{a\in\mathcal{A}(s)} \mathcal{T}(s,a,y)x_{s,a}^{\mathsf{a}} \log\left(\frac{\sum_{b\in\mathcal{A}(s)} \mathcal{T}(s,b,y)x_{s,b}^{\mathsf{a}}}{\mathcal{T}^{\pi^{\mathsf{s}}}(s,y)\sum_{b\in\mathcal{A}(s)} x_{s,b}^{\mathsf{a}}}\right)
$$

due to Lemma 2.3. The constraints (2.4b)-(2.4c) define the stationary policies that make the states in $\mathcal{S}_d$ have valid and finite occupation measures and the constraint (2.4d) encodes the reachability constraint.

Note that

$$\sum_{y\in\mathcal{S}}\sum_{a\in\mathcal{A}(s)}\mathcal{T}(s,a,y)x^{\mathsf{a}}_{s,a}\log\left(\frac{\sum_{b\in\mathcal{A}(s)}\mathcal{T}(s,b,y)x^{\mathsf{a}}_{s,b}}{\mathcal{T}^{\pi^{\mathsf{s}}}(s,y)\sum_{b\in\mathcal{A}(s)}x^{\mathsf{a}}_{s,b}}\right)$$

is the KL divergence between $\left[\sum_{a\in\mathcal{A}(s)}\mathcal{T}(s,a,y)x^{\mathsf{a}}_{s,a}\right]_{y\in Succ(s)}$ and $\left[\mathcal{T}^{\pi^{\mathsf{s}}}(s,y)\sum_{b\in\mathcal{A}(s)}x^{\mathsf{a}}_{s,b}\right]_{y\in Succ(s)}$, which is convex in $x^{\mathsf{a}}_{s,a}$ variables. Since the objective function of (2.4) is a sum of convex functions and the constraints are affine, (2.4) is a convex optimization problem.

We now show that there exists a stationary policy on $\mathcal{M}$ that achieves the optimal value of (2.1). By Proposition 2.1, we have that for all $s\in\mathcal{S}_d$, the occupation measures must be bounded. We may apply the constraints $x^{\mathsf{a}}_{s,a}\leq M_s$ for all $s$ in $\mathcal{S}_d$ and $a$ in $\mathcal{A}(s)$ without changing the optimal value of (2.4). After this modification, since the objective function is a continuous function of $x^{\mathsf{a}}_{s,a}$ values and the feasible space is compact, there exists a set of occupation measure values, and consequently a stationary policy that achieves the optimal value of (2.4). ■

*Proof of Proposition 2.4.* The condition $\mathcal{T}(s,a,y)>0$ for all $s\in\mathcal{S}_d$, $a\in\mathcal{A}(s)$, and $y\in Succ(s)$ implies that $\sum_{a\in\mathcal{A}(s)}x^{\mathsf{s}}_{s,a}\mathcal{T}(s,a,y)$ is strictly positive for all $y\in Succ(s)$. Note that for the states $y\notin Succ(s)$, we have $\sum_{a\in\mathcal{A}(s)}x^{\mathsf{a}}_{s,a}\mathcal{T}(s,a,y)=0$. We also note that by Assumption 2.3, the occupation measures are bounded for all $s\in\mathcal{S}_d$ under $\pi^{\mathsf{s}}$. Hence, the objective function of (2.5) is bounded and jointly continuous in $x^{\mathsf{s}}_{s,a}$ and $x^{\mathsf{a}}_{s,a}$.

Since in we showed that there exists a policy that attains the optimal value of Problem 2.1, we may represent the optimization problem given in (2.5) as

$$\sup_{x^{\mathsf{s}}}\min_{x^{\mathsf{a}}} f(x^{\mathsf{s}},x^{\mathsf{a}})$$

subject to $x^{\mathsf{s}}\in X^{\mathsf{s}}$ and $x^{\mathsf{a}}\in X^{\mathsf{a}}$. Note that $X^{\mathsf{s}}$ and $X^{\mathsf{a}}$ are compact spaces, since the occupation measures are bounded for all state-action pairs. Given that $X^{\mathsf{a}}$ is a compact space, the function $f'(x^{\mathsf{s}})=\min_{x^{\mathsf{a}}}f(x^{\mathsf{s}},x^{\mathsf{a}})$ is a continuous function of $x^{\mathsf{s}}$ (Clarke, 1975). The optimal value of $\sup_{x^{\mathsf{s}}}f'(x^{\mathsf{s}})$ is attained. Consequently, there exists a policy $\pi^{\mathsf{s}}$ that achieves the optimal value of (2.5). ■

*Proof Sketch for Proposition 2.5.* We can show the NP-hardness of Problem 2.2 by constructing an MDP that in based on a set partition problem. Set partition problem can be reduced to an instance of linear multiplicative programming. We construct the MDP and and the agent's policy such that the decision version of Problem 2.2 is equivalent to the decision problem of that instance of linear multiplicative programming. Since the set partition problem is NP-hard, Problem 2.2 is NP-hard[2].

In more detail, the set partition problem (Garey and Johnson, 1979; Karp, 1972) is NP-hard and is the following:

**Instance:** An $m \times n$ $0-1$ matrix $M$ satisfying $n > m$.

**Question:** Is there a $0-1$ vector $x$ satisfying $\sum_{\substack{j=1 \\ M_{ij}=1}}^{n} x_j = 1$ for all $i \in [n]$.

Linear multiplicative programming minimizes the product of two variables subject to linear inequality constraints and is NP-hard (Matsui, 1996) . Let $M$ be an $m \times n$ $0-1$ matrix with $n \geq m$ and $n \geq 5$, and $p = n^{n^4}$. The problem

$$
\begin{aligned}
&\text{min} && (2p^{4n} - p + 2p^{2n}x_0 + y_0)(2p^{4n} - p - 2p^{2n}x_0 + y_0) && \\
&\text{subject to} && x_0 = \sum_{i=1}^{n} p^i x_i && \text{(2.14a)} \\
& && y_0 = \sum_{i=1}^{n}\sum_{j=1}^{n} p^{i+j} y_{ij} && \text{(2.14b)} \\
&\forall i \in [n], && 0 \leq x_i \leq 1, y_{ii} = x_i, && \text{(2.14c)} \\
&\forall i, j \in [n] && 0 \leq y_{ij} \leq 1, && \text{(2.14d)} \\
&{\scriptstyle \forall i,j \in [n] \atop \scriptstyle i \neq j}, && x_i \geq y_{ij}, \quad x_j \geq y_{ij}, \quad y_{ij} \geq x_i + x_j - 1, && \text{(2.14e)} \\
&\forall i \in [m], && \sum_{\substack{j=1 \\ M_{ij}=1}}^{n} x_j = 1, && \text{(2.14f)}
\end{aligned}
$$

where the decision variables are $x_i$ for all $i \in [n]$ and $y_{ij}$ for all $i, j \in [n]$, is NP-hard. In detail, (Matsui, 1996) proved that the optimal value of (2.14) is less than or equal

[2]The complete proof is available at (Karabag et al., 2023).

to $4p^{8n}$ if and only if there exists a $0-1$ solution for $x_1, \ldots, x_n$ satisfying (2.14f). Since the decision problem of (2.14) correspond to solving the set partition problem, (2.14) is NP-hard.

We can construct an MDP with a size polynomial in $n$ and choose polynomial number of specifications in $n$ such that the optimal value of Problem 2.2 is

$$\begin{aligned}
&\max \frac{1}{2}\log \frac{1}{(2p^{4n}-p+2p^{2n}x_0+y_0)(2p^{4n}-p-2p^{2n}x_0+y_0)} \\
&\quad + \frac{1}{2}\log\left(4C^2(n^2+n+1)^2\right) \\
&\text{subject to} \quad \text{(2.14a)} - \text{(2.14f)}
\end{aligned}$$

where $C$ is a constant depending on $n$. Due to the result given in (Matsui, 1996), the optimal value of (2.15) is greater than or equal to

$$-\log(4p^{8n})/2 + \log\left(4C^2(n^2+n+1)^2\right)/2$$

if and only if there exists a $0-1$ solution for $x_1, \ldots, x_n$ satisfying (2.14f). Since the decision problem of (2.15) correspond to solving the set partition problem, (2.15) is NP-hard.

Since the number of states, actions, and the task constraints is polynomial in $n$ and (2.15) synthesizes an optimal reference policy, the synthesis of optimal reference policies is NP-hard.

■

## 2.4 Deception Under Partial Observability

In this section, we consider the problem described in §2.3 except for the type of supervisor's observations. We consider that while the agent operates in the environment, the supervisor receives a partial observation of the agent's state at every time step.

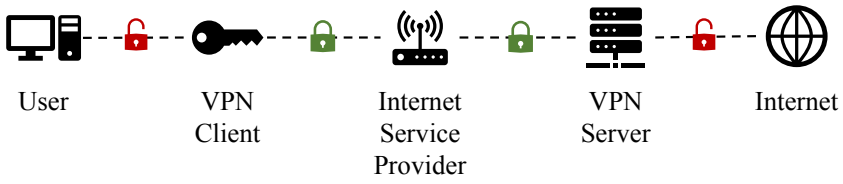



Figure 2.10: Internet access via a virtual private network (VPN). VPN client encrypts the user data, and the internet service provider (ISP) cannot observe the user's traffic.

The supervisor observes the agent's state via an *observation function* $O : \mathcal{S} \times \Omega \to [0,1]$ where $\Omega$ is a finite set of observations and $\sum_{o \in \Omega} O(s,o) = 1$ for all $s \in \mathcal{S}$. The agent has full observability of its state and knows the observation function of the supervisor. For full generality, we assume that the agent does not know the observations received by the supervisor, because in the case when the agent knows the observations received by the supervisor, we can add auxiliary states to represent the observations received by the supervisor.

**Example.** *We consider the virtual private network (VPN) example given in Figure 2.10 to demonstrate the effects of partial observability. If a user accesses the internet without a VPN, then the internet service provider (ISP) can observe the user's unencrypted traffic. In this case, the ISP can detect users with undesirable traffic. If the user accesses the internet via a VPN client, then ISP observes the user's encrypted data. The encryption makes the user's traffic partially observable; encrypted data for different types of traffic looks effectively the same for the ISP. When a VPN is used, i.e., when partial observability is exploited, ISP cannot distinguish the users with undesirable traffic.*

We remark that the setting we consider is different from partially observable MDPs (POMDPs). In POMDPs, the agent has partial observability of its state, and the goal is to find a policy that uses observations whereas in our setting the agent has full observability of its state and the goal is to shape the observation sequence.

A policy induces probability measures over paths and observation sequences.

With an abuse of notation, we denote the probability measures induced by policy $\pi$ with $\Pr^{\pi}$. We assume that $R^{\mathsf{a}}$ is a set of absorbing states, and the reference policy eventually reaches an absorbing state, i.e., $\Pr^{\pi^{\mathsf{s}}}(s_0 \models \Diamond_{\leq T} \mathcal{S}^{end}) = 1$ for $T$-step finite horizon, and $\Pr^{\pi^{\mathsf{s}}}(s_0 \models \Diamond \mathcal{S}^{end}) = 1$ for infinite horizon where $\mathcal{S}^{end}$ is the set of all absorbing states. All absorbing states share a unique observation $\varepsilon$ indicating an absorbing state has been reached. Formally, $O(s, \varepsilon) = 1$ for all $s \in \mathcal{S}^{end}$, and $O(s, \varepsilon) = 0$ for all $s \in \mathcal{S} \setminus \mathcal{S}^{end}$. When $R^{\mathsf{a}}$ is not a set of absorbing state one can use a finite automaton to represent the event of reaching $R^{\mathsf{a}}$ and synthesize the policy in the product MDP of the automaton and the original MDP.

### 2.4.1 Problem Statement

We propose the following problems for the synthesis of optimal deceptive policies under partial observability in finite and infinite horizon settings. We use Kullback-Leibler (KL) divergence as a proxy for the closeness of observations induced by the agent's policy and the reference policy. We note that KL divergences in Problems 2.3 and 2.4 are over the distributions of observation sequences.

**Problem 2.3** (Finite Horizon)**.** *Given a Markovian reference policy $\pi^{\mathsf{s}}$, solve*

$$\min_{\pi^{\mathsf{a}} \in \Pi(\mathcal{M})} \quad KL\left(\Theta^{\pi^{\mathsf{a}}}_{0:T} || \Theta^{\pi^{\mathsf{s}}}_{0:T}\right) \tag{2.16a}$$

$$\textit{subject to} \quad \Pr^{\pi^{\mathsf{a}}}\left(s_0 \models \Diamond_{\leq T} R^{\mathsf{a}}\right) \geq \nu \tag{2.16b}$$

*where $\Theta^{\pi^{\mathsf{a}}}_{0:T}$ and $\Theta^{\pi^{\mathsf{s}}}_{0:T}$ are the probability distributions of $(T+1)-$length observation sequences under $\pi^{\mathsf{a}}$ and $\pi^{\mathsf{s}}$, respectively.*

**Problem 2.4** (Infinite Horizon)**.** *Given a stationary reference policy $\pi^{\mathsf{s}}$, solve*

$$\min_{\pi^{\mathsf{a}} \in \Pi(\mathcal{M})} \quad KL\left(\Theta^{\pi^{\mathsf{a}}} || \Theta^{\pi^{\mathsf{s}}}\right) \tag{2.17a}$$

$$\textit{subject to} \quad \Pr^{\pi^{\mathsf{a}}}\left(s_0 \models \Diamond R^{\mathsf{a}}\right) \geq \nu \tag{2.17b}$$

*where $\Theta^{\pi^{\mathsf{a}}}$ and $\Theta^{\pi^{\mathsf{s}}}$ are the probability distributions of infinite length observation sequences under $\pi^{\mathsf{a}}$ and $\pi^{\mathsf{s}}$, respectively.*

**Remark 2.6.** *In our problem setting, the supervisor knows the behavioral model, i.e., reference policy, of the well-intentioned agents. If the supervisor does not know the behavioral model, it can first infer a model, e.g., as in (Wressnegger et al., 2013) using an n-Gram model, and perform detection using the learned model. We also consider that the supervisor receives an observation sequence for detection. Instead of using the sequences directly, one may consider using some features of the sequences, such as symbol frequencies or every other symbol. However, by the data processing inequality* (2.1)*, using features can only lower the KL divergence, thereby worsening the detection rate. Overall, we consider a setting that is the worst-case scenario for the deceptive agent since the supervisor knows the model of the well-intentioned agents and uses complete observation sequences.*

### 2.4.2 The Complexity of Optimal Deception Under Partial Observability

In this section, we discuss the complexity of optimal deception under partial observability. Under full observability, i.e., there is a one-to-one mapping between states and observations, the synthesis of optimal deceptive policies can be achieved in polynomial time by solving a convex optimization problem (Karabag et al., 2021b). It is easy to see that, by (2.1), the optimal values of (2.16) and (2.17) under partial observability are upper-bounded by the optimal values of (2.16) and (2.17) under full observability, respectively. This intuitively implies that the chance of being detected is lower under partial observability.

While partial observability provides a better opportunity for deception, i.e., lower objective values, and the agent still has full observability of its own state, exploiting partial observability is a hard problem. We can synthesize optimal deceptive policies under partial observability by solving a convex optimization problem with exponentially many parameters in the time horizon. The exponential complexity is due to the number of possible histories and observation sequences. Proposition 2.6 shows that Problem 2.3 is a provably hard problem, and there is no polynomial-time algorithm unless $P = NP$. The proof is due to a reduction from the 3-SAT

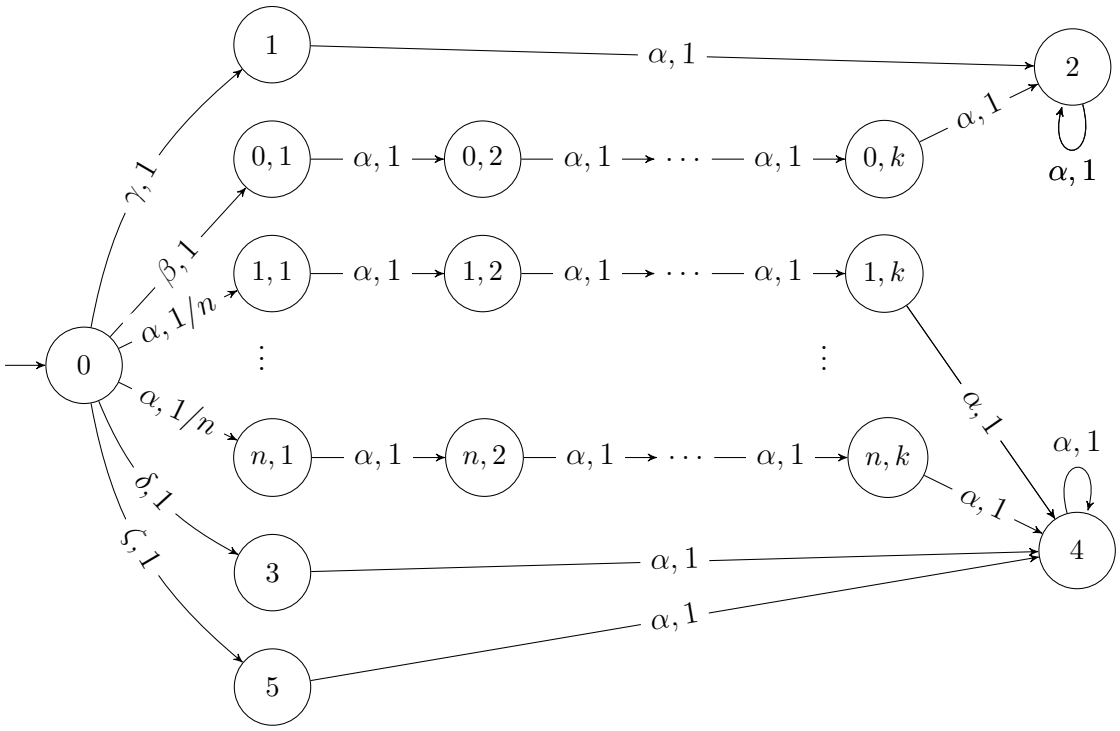



Figure 2.11: An MDP for the proof of Proposition 2.6 where nodes are the states. A label $a, p$ of an edge between nodes $s$ and $y$ refers to the transition that happens with probability $p$ under action $a$, i.e., $\mathcal{T}(s, a, y) = p$.

problem (Karp, 1972).

**Remark 2.7.** *Determining whether an observation sequence is possible for an HMM is equivalent to determining whether a word is accepted by a NFA. Formally, for a stationary policy $\pi$ and a set $C$ of end states, we can construct a NFA $N = (Q, \Sigma, \Delta, q_0, F)$ such that a word $\theta$ is accepted by $N$ if an only if there exists a path $\xi$ for MDP $\mathcal{M}$ that reach $C$ satisfying $\Pr^{\pi}(\theta|\xi)$. NFA $N$ is constructed such that $Q = \mathcal{S}$, $q_0 = s_0$, $\Sigma = \Omega$, $F = C$, and $y \in \Delta(s, o)$ if and only if $o \in O(s)$ and $\sum_{a \in \mathcal{A}(s)} \pi(s, a)\mathcal{T}(s, a, y) > 0$.*

**Proposition 2.6.** *Let $v^*$ be the optimal value of* (2.16)*. Deciding whether $v^* = \infty$ is NP-hard. If $P \neq NP$, there is no polynomial-time approximation scheme for* (2.16) *that guarantees a value lower than or equal to $(v^* + \epsilon)$ or $(1 + \epsilon)v^*$.*

Proposition 2.6 also applies to the infinite horizon optimization problem given in (2.17) as the reference policy in the proof is stationary.

We remark that the paper (Keroglou and Hadjicostis, 2018) showed that for two hidden Markov models (HMMs) deciding whether the likelihood-ratio of any observation sequence converges to a positive number is possible in polynomial time assuming that both HMMs start from any initial state with a nonzero probability, i.e., the initial state distribution is strictly positive. The proof of Proposition 2.6 shows that when the probability distribution of the initial state is not strictly positive, there is no polynomial-time algorithm for this problem unless P = NP.

We also remark that optimal deception under partial observability is a hard problem even for the simplest observation functions. For example, consider an observation function such that all transient states emit the same observation with probability 1 and all absorbing states emit another observation with probability 1. The deciding whether the optimal value of (2.17) is $\infty$ correspond to the language containment problem of unary NFA, which is shown to be coNP-complete (Stockmeyer and Meyer, 1973).

### 2.4.3 Synthesis of Deceptive Policies

In this section we discuss the synthesis of deceptive policies under partial observability. In detail, we consider the synthesis for finite horizon using mixture policies as an alternative to optimal policies. We also consider a special class of MDPs where optimal distribution of paths can be induced in polynomial time for infinite horizon.

#### 2.4.3.1 Mixture Policies for Finite Horizon

We consider a special class of policies for the finite horizon case since the synthesis of optimal deceptive policies is computationally challenging due to the size of history-dependent policies. A *mixture policy* (Collins and McNamara, 1998) is

a convex combination of a finite set of policies. In detail, a mixture policy is a tuple $([\pi^1, \ldots, \pi^N], [\alpha_1, \ldots, \alpha_N])$ where $[\pi^1, \ldots, \pi^N]$ is a vector of basis policies and $[\alpha_1, \ldots, \alpha_N] \in \Delta_0^N$ is mixing probabilities. At time 0, the agent chooses policy $\pi^i$ with probability $\alpha_i$ and follows the selected policy for the whole path.

The class of mixture policies has the following useful property: The probability distribution over paths (and over observation sequences) induced by the mixture policy is a linear combination of the probability distribution induced by each basis policy. This property provides a convex representation of the deception problem. The KL divergence between the distributions of observation sequences is a nonconvex funtion of the parameters of policies $\pi^1, \ldots, \pi^N$. On the other hand, the KL objective function is a convex function of the mixing probabilities $\alpha_1, \ldots, \alpha_N$. Hence, for a given set of basis policies $\pi^1, \ldots, \pi^N$, our goal is to optimize a convex function of the mixing probabilities $\alpha_1, \ldots, \alpha_N$ and find the best mixture policy.

The straightforward approach to find the optimal mixing probabilities is the following. First, enumerate the possible observation sequences under the reference policy for the given time horizon. Second, find the probabilities of the observation sequences under the basis policies. Finally, optimize the KL objective function. However, the enumeration of possible observation sequences is a challenging problem. Counting the possible observation sequences under the reference policy is $\sharp$P-complete due to a reduction from the problem of counting the number of satisfying assignments to a Boolean formula (Valiant, 1979). Hence, even the construction of the problem is computationally hard.

To avoid the complete construction, we propose to use stochastic optimization for the synthesis of optimal mixture policies. Algorithm 3 uses the projected stochastic gradient descent method Nemirovski et al. (2009). In every iteration $i$ of the inner loop, the algorithm samples an observation sequence uniformly randomly and adjusts the mixing probabilities according to the gradients. If the probability of the sampled observation sequence is positive under a basis policy and 0 under the reference policy,

**Algorithm 3:** Mixing algorithm

**Input:** An MDP $\mathcal{M}$, a reachability specification $\Diamond R^{\mathsf{a}}$, a probability threshold $\nu^{\mathsf{a}}$, and a set $C^{(0)}$ of basis policies.
**Output:** A mixing vector $\alpha^{(k)}$.
$\alpha^{(0,1)} \leftarrow [1/|C^{(0)}|, \ldots, 1/|C^{(0)}|]$. // Initial uniform mixing
$\beta^{(0)} \leftarrow 1$, $b^{(0)} \leftarrow 1/(2|C^{(0)}|)$, $N^{(0)} \leftarrow 1$. // Opt. parameters
**for** $k = 1, \ldots$ **do**
  $C^{(k)} \leftarrow C^{(k-1)}$, $\alpha^{(k,0)} \leftarrow \alpha^{(k-1,N^{(k-1)})}$.
  $\beta^{(k)} \leftarrow \beta^{(k-1)}/2$, $b^{(k)} \leftarrow b^{(k-1)}/\sqrt{2}$, $N^{(k)} \leftarrow 4N^{(k-1)}$.
  **for** $i = 1, \ldots, N^{(k)}$ **do**
    Uniformly sample an observation sequence $\theta$ from $\Omega^T$.
    **if** $\Pr^{\mathsf{s}}(\theta) = 0$ **then**
      **for** $\pi \in C^{(k)}$ *such that* $\Pr^{\pi}(\theta) \neq 0$ **do**
        $C^{(k)} \leftarrow C^{(k)} \setminus \{\pi\}$.
        Remove the mixing probability that correspond to $\pi$ from $\alpha^{(k,i-1)}$ and normalize $\alpha^{(k,i-1)}$.
      $f(\alpha^{(k,i-1)}, \theta) \leftarrow 0$ // No gradient step if $\Pr^{\pi}(\theta) = 0$
    **else**
      $f(\alpha^{(k,i-1)}, \theta) \leftarrow \Pr^{(C^{(k)}, \alpha^{(k,i-1)})}(\theta) \log\left(\frac{\Pr^{(C^{(k)}, \alpha^{(k,i-1)})}(\theta)}{\Pr^{\mathsf{s}}(\theta)}\right)$
    $TC = \{\alpha | \Pr^{(C^{(k)}, \alpha)}(s_0 \models \Diamond_{\leq T} R^{\mathsf{a}}) \geq \nu\}$ // Task constraint
    $\alpha^{(k,i)} \leftarrow \alpha^{(k,i-1)} - \beta^{(k)} \nabla f_{\alpha^{(k,i-1)}}(\alpha^{(k,i-1)}, \theta)$
    $\alpha^{(k,i)} \leftarrow Proj_{\Delta_{b^{(k)}} \cap TC}(\alpha^{(k,i)})$
  $\alpha^{(k)} \leftarrow \sum_{i=1}^{N^{(k)}} \alpha^{(k,i)}/N^{(k)}$ // Mixing vector after itr. $k$

then the basis policy is removed. In this case, a new iteration $k+1$ of the outer loop starts with the remaining basis policies. The gradient computation is performed in the following way. We have

$$KL\left(\Theta_{0:T}^{(C^{(k)}, \alpha^{(k,i-1)})} || \Theta_{0:T}^{\pi^{\mathsf{s}}}\right) = \sum_{\theta \in \Omega^T} f(\alpha^{(k,i-1)}, \theta)$$

where

$$f(\alpha^{(k,i-1)}, \theta) = \Pr^{(C^{(k)}, \alpha^{(k)})}(\theta) \log\left(\frac{\Pr^{(C^{(k)}, \alpha^{(k,i-1)})}(\theta)}{\Pr^{\mathsf{s}}(\theta)}\right).$$

Due to the definition of mixture policies we have

$$\frac{\partial f(\alpha^{(k,i-1)},\theta)}{\partial \alpha_j^{(k,i-1)}} = \Pr^{\pi^j}(\theta)\left(\log\left(\frac{\Pr^{(C^{(k)},\alpha^{(k,i-1)})}(\theta)}{\Pr^{\mathsf{s}}(\theta)}\right)+1\right).$$

Since computing the probability of an observation sequence for an HMM can be performed in polynomial time, every iteration in the inner **for** loop takes polynomial time in the size of the basis policies.

The output of Algorithm 3 almost surely asymptotically converges to a set of optimal mixture parameters.

**Proposition 2.7.** *Let $\alpha^*$ be a set of optimal mixing probabilities for the set $C^{(0)}$ of basis policies. Assume that there exists $\pi^i \in C^{(0)}$ with $KL\left(\Theta_{0:T}^{\pi^i}||\Theta_{0:T}^{\pi^{\mathsf{s}}}\right) < \infty$. In Algorithm 3, with probability* 1*,*

$$\lim_{k\to\infty}\mathbb{E}\left[KL\left(\Theta_{0:T}^{(C^{(k)},\alpha^{(k)})}||\Theta_{0:T}^{\pi^{\mathsf{s}}}\right)\right] = KL\left(\Theta_{0:T}^{(C^{(0)},\alpha^*)}||\Theta_{0:T}^{\pi^{\mathsf{s}}}\right).$$

*Let $v^*$ be the optimal value of* (2.16)*. If $C^{(0)} = \Pi^{D,H}(\mathcal{M})$ in Algorithm 3, with probability* 1*,*

$$\lim_{k\to\infty}\mathbb{E}\left[KL\left(\Theta_{0:T}^{(C^{(k)},\alpha^{(k)})}||\Theta_{0:T}^{\pi^{\mathsf{s}}}\right)\right] = v^*.$$

We note that we use an iterative scheme to adjust the parameters of projected stochastic gradient descent. Such a scheme is employed to eliminate the basis policies with infinite objective values and due to the unknown Lipschitz constant.

When the set of basis policies is the set of deterministic, history dependent policies, the output mixture policy is an optimal solution for Problem 2.3 in the limit since there exists a mixture of deterministic, history dependent policies that induces the same distribution of observation sequences with the optimal history-dependent policy.

Vanilla stochastic gradient descent method uses uniform sampling of the feasible observation sequences. In Algorithm 3, we sample observation sequences uniformly randomly from $\Omega^T$ for simplicity. However, some observation sequences may

be infeasible under the basis policies, i.e., $\max_i \Pr^{\pi^i}(\theta) = 0$. Algorithm 3 relies on rejection sampling and ignores such observation sequences. The convergence of Algorithm 3 might be slow in practice due to rejection sampling and the large size of $\Omega^T$. Direct sampling from the set $\{\theta | \max_i \Pr^{\pi^i}(\theta) > 0\}$ in polynomial time is possible using the method given in (Bernardi and Giménez, 2012) since this set defines a regular language. Another way to potential overcome this problem is to employ importance sampling, i.e., sample observation sequences using the current mixture policy $(C^{(k)}, \alpha^{(k,i-1)})$. In this way, we can get an unbiased estimate of the gradient. Formally, we have

$$\mathbb{E}_{\theta \sim Uniform(\{\theta | \max_i \Pr^{\pi^i}(\theta) > 0\})} \left[\nabla f(\alpha^{(k,i-1)}, \theta)\right] = \frac{\mathbb{E}_{\theta \sim (C^{(k)}, \alpha^{(k,i-1)})} \left[\frac{\nabla f(\alpha^{(k,i-1)}, \theta)}{\Pr^{(C^{(k)}, \alpha^{(k,i-1)})}(\theta)}\right]}{|\{\theta | \max_i \Pr^{\pi^i}(\theta) > 0\}|}.$$

#### 2.4.3.2 Optimal Path Distributions for Infinite Horizon Deterministic MDPs and Observation Functions

In this section, we give a path planing algorithm for infinite horizon deterministic MDPs and observation functions. For a deterministic MDP, the transition probability and observation functions are deterministic. In other words, the environment is a directed graph, and every path has a fixed observation sequence. We remark that while we consider deterministic MDPs, the reference policy can still be randomized.

Consider an agent that aims to follow predetermined path and thereby induce a predetermined observation sequence, as can be done in MDPs with deterministic transitions and observation functions. In this case, the agent can set the probability of any observation sequence to a desired value and achieve optimality for Problem 2.4.

As discussed in §2.4.3.1, the straightforward approach is to enumerate observation sequences and solve an optimization problem that minimizes the KL divergence to the reference policy's observation distribution. However, this requires solving an

**Algorithm 4:** Path planning algorithm for deterministic MDPs

**Input:** A deterministic MDP $\mathcal{M}$, a reachability specification $\Diamond R^{\mathsf{a}}$, and a probability threshold $\nu^{\mathsf{a}}$.
**Output:** A path $\xi'$
**while** *True* **do**
  Sample a path $\xi$ from $\Gamma^{\mathsf{s}}$.
  **if** $(O(\xi) \in \mathcal{L}^{\mathsf{a}}) = (c \leq \nu)$ **then**
    Find a path $\xi'$ such that $(\xi' \models \Diamond R^{\mathsf{a}})$ XOR $(c > \nu)$ is true and $O(\xi') = O(\xi)$.
    **break**
Output $\xi'$

optimization problem with infinitely many variables since the number of observation sequences is infinite. Instead, we give a randomized algorithm for path planning that runs in polynomial time and finds a random path for the agent such that the path satisfies the task constraint in expectation, and the objective value is optimal in expectation.

Algorithm 4 increases the probabilities of the observation sequences for which there is a path reaching $R^{\mathsf{a}}$. With an abuse of notation, let $O(\xi)$ be the corresponding observation sequence of path $\xi$. Also, let $\mathcal{L}^{\mathsf{a}}$ be the set of observation sequences such that for every $\theta \in \mathcal{L}^{\mathsf{a}}$ there exists a path $\xi$ satisfying $O(\xi) = \theta$ and $\xi \models \Diamond R^{\mathsf{a}}$. The algorithm relies on rejection sampling and works as follows. First, it samples a path $\xi$ using $\pi^{\mathsf{s}}$. With probability $\nu$, the algorithm accepts $\xi$ if and only if there is a path $\xi'$ that reaches $R^{\mathsf{a}}$ and $O(\xi) = O(\xi')$, i.e., $O(\xi) \in \mathcal{L}^{\mathsf{a}}$. With probability $1 - \nu$, the algorithm accepts $\xi$ if and only if there is no path $\xi'$ that reaches $R^{\mathsf{a}}$ with $O(\xi) = O(\xi')$, i.e., $O(\xi) \notin \mathcal{L}^{\mathsf{a}}$. At the end, the algorithm outputs path $\xi'$.

**Proposition 2.8.** *Assume that* $\nu \geq \Pr^{\pi^{\mathsf{s}}}(\xi | O(\xi) \in \mathcal{L}^{\mathsf{a}})$. *Let* $\mu$ *be the probability measure induced by Algorithm 4 over the paths of* $\mathcal{M}$ *and* $v^*$ *be the optimal value of* (2.17). *Algorithm 4 satisfies*

$$\Pr(\xi \models \Diamond R^{\mathsf{a}} | \xi \sim \mu) \geq \nu \text{ and } KL(\mu || \Theta^{\pi^{\mathsf{s}}}) = v^*,$$

*and it has an expected time complexity of*

$$\mathcal{O}\left(\frac{\nu|\mathcal{S}|^2|\mathcal{A}|\mathbb{E}_{\xi\sim\pi^{\mathsf{s}}}\left[len(\xi))\right]}{\Pr^{\pi^{\mathsf{s}}}(\xi|O(\xi)\in\mathcal{L}^{\mathsf{a}})^2}+\frac{(1-\nu)|\mathcal{S}|^2|\mathcal{A}|\mathbb{E}_{\xi\sim\pi^{\mathsf{s}}}\left[len(\xi)\right]}{\Pr^{\pi^{\mathsf{s}}}(\xi|O(\xi)\notin\mathcal{L}^{\mathsf{a}})^2}\right)$$

*where* $len(\xi = s_0 s_1 \ldots) = \min\{i | s_i \in \mathcal{S}^{end}\}$.

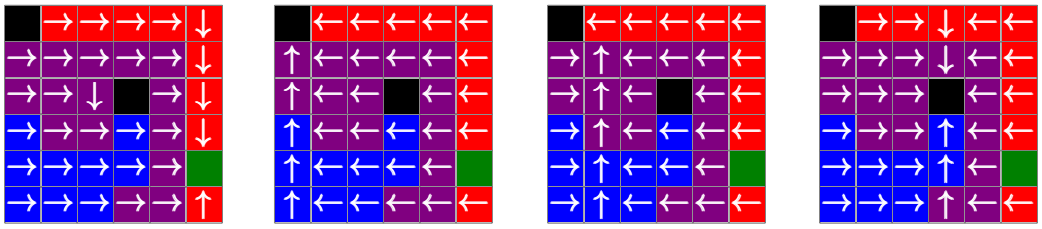
Proposition 2.8 shows that likelihood ratio for the observation sequence of the output path is optimal in expectation, as Algorithm 4 boosts the probabilities of all observation sequences for which there is a path that reaches $R^{\mathsf{a}}$ at the same ratio.

The running time of Algorithm 4 depends both on some properties of the reference policy as well as the size of the MDP. The dependencies on $\nu$ and $\Pr^{\mathsf{s}}(\xi|O(\xi)\in\mathcal{L}^{\mathsf{a}})$ are due to rejection sampling. The dependencies on $|\mathcal{S}|$, $|\mathcal{A}|$ and $\mathbb{E}_{\xi\sim\pi^{\mathsf{s}}}\left[len(\xi)\right]$ are due to checking whether the sampled observation sequence is in $\mathcal{L}^{\mathsf{a}}$.

### 2.4.4 Numerical Example

We demonstrate the synthesis of optimal mixture policies in a grid-world environment shown in Fig. 2.12. At every state, there are 4 available actions: up, down, left, right, and stay. With probability 0.9, the agent moves in the chosen direction or stays if action stay is chosen. With probability 0.1, the agent moves in the other directions or stays. If a transition is not possible because the agent is at the boundary of the grid, the transition probability is proportionally distributed among the other transitions. The observation function represents a binary temperature sensor that has two levels, *Low* and $High$. Blue cells are more likely to emit observation *Low* and red cells are more likely to emit observation $High$. Purple cells emit observations *Low* and $High$ with equal probabilities.

We consider the mixture of three basis policies shown in Fig. 2.12b–2.12d. Policy $\pi^1$ reaches the black cell on left in minimum time, and policy $\pi^3$ reaches the black cell on right in minimum time. The length of the time horizon is 8. There are 480 observation sequences such that $\max_i \Pr^{\pi^i}(\theta) > 0$. The basis policies $\pi^1$, $\pi^2$, and

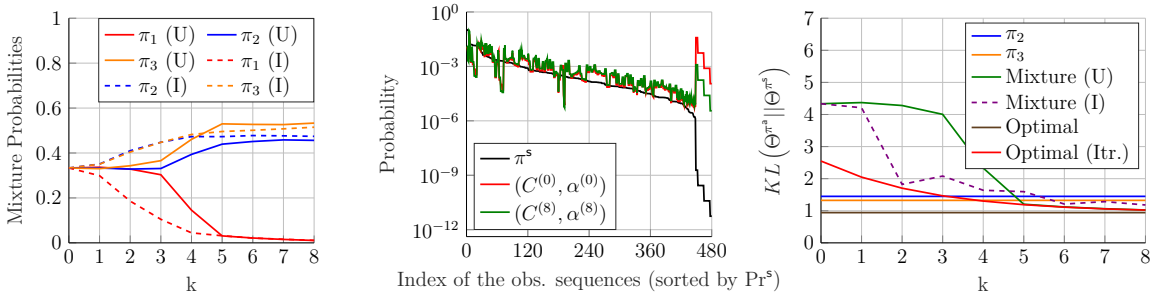


(a) $\pi^{\mathsf{s}}$ (b) $\pi^1$ (c) $\pi^2$ (d) $\pi^3$

Figure 2.12: The environment is a $6 \times 6$ grid world. The initial state is the bottom left cell. Black cells are the target set of states for the agent. The reference policy and the basis policies are shown in Fig. 2.12a–2.12d. Blue cells emit observation *Low* with probability 1/8 and *H* with probability 7/8. Purple cells emit observation *Low* with probability 1/2 and *High* with probability 1/2. Red cells emit observation *Low* with probability 1/8 and *High* with probability 7/8. Black and green are the end states, and they emit observation $\varepsilon$ with probability 1.

$\pi^3$, reach the target black cells with probabilities 0.93, 0.54, and 0.75, respectively. We set $\nu = 0.5$. Hence, every mixture of the basis policies is feasible. We initialize the mixing probabilities with a uniform distribution.

Policies $\pi^2$ and $\pi^3$ are advantageous over policy $\pi^1$. Under $\pi^{\mathsf{s}}$, the agent reaches an end state after 6 transitions with high probability (w.h.p.) Policy $\pi^3$ is advantageous since it also causes the agent to reach to an end state after 6 transitions w.h.p. The stochasticity in the observation function might lead to the same observation sequences for $\pi^{\mathsf{s}}$ and $\pi^3$. Policy $\pi^2$ is advantageous, since the observation sequences generated by $\pi^2$ resemble the observation sequences generated by $\pi^{\mathsf{s}}$. For example, $Low, Low, Low, High, High, High, \varepsilon, \varepsilon$ is among the most likely observation sequences under $\pi^{\mathsf{s}}$, and $Low, Low, Low, High, High, High, High, \varepsilon$ is among the most likely observation sequences under $\pi^2$. The stochasticity in the environment might lead to the same observation sequences for $\pi^{\mathsf{s}}$ and $\pi^2$. Policy $\pi^1$ is intuitively dissimilar to $\pi^{\mathsf{s}}$ in terms of the induced observation distributions. For example, under $\pi^1$, the agent reaches an end state after 5 transitions w.h.p. On the other hand, reaching an end state after 5 transitions is unlikely under $\pi^{\mathsf{s}}$. Overall, we expect the weights of $\pi^2$ and $\pi^3$ to be higher than the weight of $\pi^1$.




(a) (b) (c)

Figure 2.13: (a) The mixing probabilities for different values of $k$ in Algorithm 3. 'U' refers to uniform sampling, and 'I' refers to importance sampling of observation sequences using $(C^{(k)}, \alpha^{k,i-1})$. (b) The probabilities of the observation sequences under the initial and final mixture policies compared to the reference policy. (c) The objective value for different values of $k$ in Algorithm 3. 'Optimal' is the value for the optimal mixture policy. 'Optimal (Itr.)' is the value for the optimal mixture policy at iteration $k$ subject to the constraint $\alpha^{(k)} \in \Delta_{b^{(k)}}$. The objective value for $\pi^1$ is 12.23.

We run Algorithm 3 for $k = 1, \ldots, 8$. For uniform sampling, we directly sample from the observation sequences such that $\max_i \Pr^{\pi^i}(\theta) > 0$. In addition to the uniform sampling of observation sequences, we use the importance sampling method, i.e., at iteration $k$, we sample paths using policy $(C^{(k)}, \alpha^{(k,i-1)})$.

The mixture probabilities are shown in Fig. 2.13a, and the expected log-likelihood ratios are shown in Fig. 2.13c. As explained above, both sampling methods assign high weights to $\pi^2$ and $\pi^3$ and a low weight to $\pi^1$. Fig. 2.13b shows that the final mixture downweights the observation sequences that are unlikely under the reference policy. In Fig. 2.13c, we observe that as the convexity of the objective function suggests, mixture policies outperform the basis policies and converge to the optimal mixture: When the optimal mixture policy is used, the supervisor needs $\approx 40\%$ more observation sequences to achieve the same detection rate for likelihood-ratio test since

$$\min_i KL\left(\Theta^{\pi^i}_{0:T}||\Theta^{\pi^s}_{0:T}\right) / \min_{\alpha\in\Delta_0} KL\left(\Theta^{(C^{(0)},\alpha)}_{0:T}||\Theta^{\pi^s}_{0:T}\right) = 1.40.$$

Importance sampling quickly improves value of the objective function. After a single iteration, i.e., 4 sample observation sequences, importance sampling down-weights $\pi^1$. Uniform sampling outperforms importance sampling after 5 iterations, i.e, 1364 sample observation sequences. The performance of importance sampling is better than uniform sampling for the iterations where the number of samples is lower than the number of possible observation sequences. This property holds because importance sampling creates a bias towards observation sequences that have high $\Pr^{(C^{(k)},\alpha^{(k)})}(\theta)$. In detail, for any $\theta$, the value of the objective function is proportional to $\Pr^{(C^{(k)},\alpha^{(k)})}(\theta)$. Hence, using importance sampling creates a bias towards the observation sequences that highly affect the objective function. These observation sequences are sampled w.h.p. even with a small number of samples. On the other hand, when uniform sampling is used, it is more likely to sample an observation sequence with a low $\Pr^{(C^{(k)},\alpha^{(k)})}(\theta)$. Such observation sequences has a low impact on the objective function. When the number of samples is lower than the number of possible observation sequences, uniform sampling fails to sample observation sequences that have high $\Pr^{(C^{(k)},\alpha^{(k)})}(\theta)$, and performs worse than importance sampling. When the number of samples is high, uniform sampling outperforms importance sampling since importance sampling suffers from high variance.

### 2.4.5 Proofs for the Technical Results

*Proof of Proposition 2.6.* We use the MDP given in Fig. 2.11 to prove the hardness of (2.16). This MDP shares a similar structure with the NFA used to prove the hardness of language universality and containment problems for NFAs (Burghardt, 2016; Krötzsch et al., 2017). The task of the agent is to reach state 2 with probability 1, i.e., $R^{\mathsf{a}} = \{2\}$ and $\nu = 1$. The observation function is defined using a 3-SAT formula. A 3-SAT formula (Karp, 1972) is a conjunctive normal formula with $n$ clauses where each clause has three literals from a set $l_1, \ldots, l_k, \neg l_1, \ldots, \neg l_k$ of $2k$ literals. Let $\phi$ be an arbitrary instance of 3-SAT and $T = k + 3$. The observation function $O : \mathcal{S} \times \Omega \to [0, 1]$ is defined such that

- $O(0, x) = 1$, $O(1, y) = 1$, $O(2, \varepsilon) = 1$, $O(3, y) = 1$, $O(4, \varepsilon) = 1$, $O(5, z) = 1$,
- $O((0, j), \top) = 0.5$ and $O((0, j), \bot) = 0.5$,
- $O((i, j), \top) = 0.5$ and $O((i, j), \bot) = 0.5$ if $i$-th clause of $\phi$ does not contain $l_j$ and $\neg l_j$,
- $O((i, j), \top) = 1$ if $i$-th clause of $\phi$ contains $\neg l_j$, and
- $O((i, j), \bot) = 1$ if $i$-th clause of $\phi$ contains $l_j$.

To show that deciding whether the optimal value of (2.16) is $\infty$ is NP-hard, consider a reference policy such that $\pi^{\mathsf{s}}(0, \alpha) = 1$. Note that the decision at state 0 is sufficient to describe the policy since there is only one action for the other states. If $\pi^{\mathsf{a}}(0, \beta) \neq 1$ the agent violates the task constraint, i.e., $\Pr^{\pi^{\mathsf{a}}}(s_0 \models \Diamond_{\leq T} R^{\mathsf{a}}) \geq \nu$, or $KL\left(\Theta^{\pi^{\mathsf{a}}}_{0:T} || \Theta^{\pi^{\mathsf{s}}}_{0:T}\right) = \infty$ since there is a positive probability that the agent generates an observation sequence $\theta$ such that $\Pr^{\pi^{\mathsf{s}}}(\theta) = 0$.

If $\pi^{\mathsf{a}}(0, \beta) = 1$, we have $KL\left(\Theta^{\pi^{\mathsf{a}}}_{0:T} || \Theta^{\pi^{\mathsf{s}}}_{0:T}\right) < \infty$ if and only if $\Pr^{\pi^{\mathsf{s}}}(\theta) > 0$ for all $\theta \in \{x\{\top, \bot\}^k\}$ since $\Pr^{\pi^{\mathsf{a}}}(\theta) > 0$ for all $\theta \in \{x\{\top, \bot\}^k\}$. Note that by construction of the observation function, $\Pr^{\pi^{\mathsf{s}}}(\theta = o_1 \dots o_{k+1}) > 0$ if and only if $o_2 \dots o_{k+1}$ is an assignment for $l_1 \dots l_k$ that satisfies $\neg\phi$. Consequently, $\Pr^{\pi^{\mathsf{s}}}(\theta) > 0$ for all $\theta \in \{x\{\top, \bot\}^k\}$ if and only if $\neg\phi$ is true for all $\theta \in \{x\{\top, \bot\}^k\}$. Hence, $\Pr^{\pi^{\mathsf{s}}}(\theta) > 0$ for all $\theta \in \{x\{\top, \bot\}^k\}$ if and only if $\phi$ is not satisfiable. Since 3-SAT problem is NP-hard (Karp, 1972), and the size of the MDP is polynomial in the size of the 3-SAT instance, deciding whether the optimal value of (2.16) is $\infty$ is NP-hard.

To show that there is no polynomial-time $\epsilon$-approximation scheme for (2.16) unless P $\neq$ NP, we consider a reference policy such that $\pi^{\mathsf{s}}(0, \alpha) = 0.5$, $\pi^{\mathsf{s}}(0, \delta) = b$, and $\pi^{\mathsf{s}}(0, \zeta) = 0.5 - b$. If $\phi$ is not satisfiable, the optimal value of (2.16) is $\log(1/b)$, which is achieved when $\pi^{\mathsf{a}}(0, \gamma) = 1$. If $\phi$ is satisfiable, every $\theta \in \{x\{\top, \bot\}^k\}$ has $\Pr^{\pi^{\mathsf{s}}}(\theta) \geq 2^{-k+2}/n$ by construction of the observation function. If $\phi$ is satisfiable and

$\pi^{\mathsf{a}}(0,\beta)=1$, we have

$$KL\left(\Theta_{0:T}^{\pi^{\mathsf{a}}}||\Theta_{0:T}^{\pi^{\mathsf{s}}}\right)=\sum_{\theta\in\{x\{\top,\bot\}^k\}}\Pr^{\pi^{\mathsf{a}}}(\theta)\log\left(\frac{\Pr^{\pi^{\mathsf{a}}}(\theta)}{\Pr^{\mathsf{s}}(\theta)}\right)$$
$$=\sum_{\theta\in\{x\{\top,\bot\}^k\}}2^{-k}\log\left(\frac{2^{-k}}{\Pr^{\pi^{\mathsf{s}}}(\theta)}\right)\leq\log(n/4).$$

Hence, the optimal value of (2.16) is lower than or equal to $\log(n/4)$ if $\phi$ is satisfiable. Let $1/b<n/4$. If an approximation scheme assigns $\pi^{\mathsf{a}}(0,\beta)>0$ and $\phi$ is not satisfiable, then $KL\left(\Theta_{0:T}^{\pi^{\mathsf{a}}}||\Theta_{0:T}^{\pi^{\mathsf{s}}}\right)-\log(1/b)=\infty$. If an approximation scheme assigns $\pi^{\mathsf{a}}(0,\beta)=0$ and $\phi$ is satisfiable, then $KL\left(\Theta_{0:T}^{\pi^{\mathsf{a}}}||\Theta_{0:T}^{\pi^{\mathsf{s}}}\right)-\log(n/4)$ is a constant not depending on the input parameter $\epsilon$ of the approximation algorithm. Therefore, any approximation algorithm has to solve the 3-SAT problem to achieve $\epsilon$-optimality, and there is no polynomial-time $\epsilon$-approximation scheme for (2.16) unless P $\neq$ NP. ■

*Proof of Proposition 2.7.* We first show that with probability 1,

$$\lim_{k\to\infty}KL\left(\Theta_{0:T}^{(C^{(k)},\alpha^{(k)})}||\Theta_{0:T}^{\pi^{\mathsf{s}}}\right)<\infty.$$

Note that $KL\left(\Theta_{0:T}^{\pi^i}||\Theta_{0:T}^{\pi^{\mathsf{s}}}\right)<\infty$ if and only if $\Pr^{\pi^i}(\theta)=0$ for all $\theta\in\Omega^T$ such that $\Pr^{\mathsf{s}}(\theta)=0$. Let $\pi^i\in C^{(k)}$ be a policy such that $\Pr^{\mathsf{s}}(\theta)=0$ and $\Pr^{\pi^i}(\theta)\neq 0$ for some $\theta\in\Omega^T$. We have $\pi^i\in C^{(k+1)}$ with probability at most $(1-1/|\Omega^T|)^{N^{(k)}}\leq\exp(-N^{(k)}/|\Omega^T|)$. After $K$ rounds, we have $\pi^i\in C^{(K+1)}$ with probability at most $\exp(-4^KN^{(0)}/|\Omega^T|)$. Hence, $\pi^i\notin C^{(k)}$ with probability 1 as $k\to\infty$. By the union bound and the convexity of the KL divergence, $\lim_{k\to\infty}KL\left(\Theta_{0:T}^{(C^{(k)},\alpha^{(k)})}||\Theta_{0:T}^{\pi^{\mathsf{s}}}\right)\leq\sum_{i=1}^{N^{(k)}}\lim_{k\to\infty}KL\left(\Theta_{0:T}^{(C^{(k)},\alpha^{(k,i)})}||\Theta_{0:T}^{\pi^{\mathsf{s}}}\right)/N^{(k)}<\infty$ with probability 1.

We now show that with probability 1,

$$\lim_{k\to\infty}\mathbb{E}\left[KL\left(\Theta_{0:T}^{(C^{(k)},\alpha^{(k)})}||\Theta_{0:T}^{\pi^{\mathsf{s}}}\right)\right]=KL\left(\Theta_{0:T}^{(C,\alpha^*)}||\Theta_{0:T}^{\pi^{\mathsf{s}}}\right).$$

Let $v^{(k),*}=\min_{\alpha\in\Delta_{b^{(k)}}}KL\left(\Theta_{0:T}^{(C^{(k)},\alpha)}||\Theta_{0:T}^{\pi^{\mathsf{s}}}\right)$. Assume that $KL\left(\Theta_{0:T}^{\pi}||\Theta_{0:T}^{\pi^{\mathsf{s}}}\right)<\infty$ for all $\pi\in C^k$. Let $M^{(k)}=\sup_{\alpha^{(k)}\in\Delta_{b^{(k)}}}\max_{\theta\in\Omega^T}\left\|\nabla f(\alpha^{(k)},\theta)\right\|^2$. By Equation 2.19 of

(Nemirovski et al., 2009), we have

$$\mathbb{E}\left[KL\left(\Theta_{0:T}^{(C^{(k)},\alpha^{(k)})}||\Theta_{0:T}^{\pi^{\mathsf{s}}}\right)-v^{(k),*}\right]\leq\frac{4+(M^{(k)})^2N^{(k)}(\beta^{(k)})^2}{2|\Omega^T|N^{(k)}\beta^{(k)}}.$$

Let $\theta$ be an arbitrary observation sequence, and $\theta^*=\arg\min_{\theta'\in\Omega^T}\Pr^{\mathsf{s}}(\theta')$ such that $\Pr^{\mathsf{s}}(\theta')>0$. For large enough $k$, we have $\log(b^{(k)}/\Pr^{\mathsf{s}}(\theta))\leq\partial f(\alpha^{(k)},\theta)/\partial\alpha_j^{(k)}\leq 2/\Pr^{\mathsf{s}}(\theta)$. Similarly, for large $k$, we have $|\partial f(\alpha^{(k,i-1)},\theta)\ /\partial\alpha_j^{(k)}|\leq-\log(b^{(k)}/\Pr^{\mathsf{s}}(\theta))$ since $b^{(k)}\to 0$. Hence,

$$\left\|\nabla f(\alpha^{(k)},\theta)\right\|^2\leq|C^{(k)}|\log\left(\frac{b^{(k)}}{\Pr^{\mathsf{s}}(\theta^*)}\right)^2$$

for all $\alpha^{(k)}\in\Delta_{b^{(k)}}$ and $\theta\in\Omega^T$ since $\alpha^{(k,i-1)}$ has $|C^{(k)}|$ elements. Define $L^{(k)}=|C^{(k)}|\log\left(\frac{b^{(k)}}{\Pr^{\mathsf{s}}(\theta^*)}\right)^2$. There exists $k'\geq 0$ such that

$$\mathbb{E}\left[KL\left(\Theta_{0:T}^{(C^{(k)},\alpha^{(k)})}||\Theta_{0:T}^{\pi^{\mathsf{s}}}\right)-v^{(k),*}\right]\leq\frac{4+(L^{(k)})^2N^{(k)}(\beta^{(k)})^2}{2|\Omega^T|N^{(k)}\beta^{(k)}}.$$

for all $k>k'$ .

Since $\lim_{k\to\infty}N^{(k)}\beta^{(k)}=\infty$, we only need to show $\lim_{k\to\infty}(L^{(k)})^2\beta^{(k)}=0$ in order to show that the term on the right hand side goes to zero as $k\to\infty$.

Since $b^{(k+1)}=\sqrt{2}b^{(k)}$, we have

$$\lim_{k\to\infty}\log\left(\frac{b^{(k+1)}}{\Pr^{\mathsf{s}}(\theta^*)}\right)/\log\left(\frac{b^{(k)}}{\Pr^{\mathsf{s}}(\theta^*)}\right)\leq\sqrt{3}.$$

Consequently, $\lim_{k\to\infty}L^{(k+1)}/L^{(k)}\leq\sqrt{3}$. Since $\beta^{(k+1)}/\beta^{(k)}=1/2$, we have

$$\lim_{k\to\infty}(L^{(k)})^2\beta^{(k)}=0,$$

which implies $\lim_{k\to\infty}\mathbb{E}\left[KL\left(\Theta_{0:T}^{(C^{(k)},\alpha^{(k)})}||\Theta_{0:T}^{\pi^{\mathsf{s}}}\right)\right]-v^{(k),*}=0.$

Since $KL\left(\Theta_{0:T}^{(C^{(k)},\alpha^{(k)})}||\Theta_{0:T}^{\pi^{\mathsf{s}}}\right)$ is a bounded, continuous function, we have

$$\lim_{k\to\infty}v^{(k),*}=KL\left(\Theta_{0:T}^{(C,\alpha^*)}||\Theta_{0:T}^{\pi^{\mathsf{s}}}\right).$$

This property trivially holds when $\alpha^*$ is an interior point and holds due to the continuity and boundedness when $\alpha^*$ is a boundary point. Hence, with probability 1,

$$\lim_{k\to\infty} \mathbb{E}\left[KL\left(\Theta_{0:T}^{(C^{(k)},\alpha^{(k)})}||\Theta_{0:T}^{\pi^{\mathsf{s}}}\right)\right] = KL\left(\Theta_{0:T}^{(C,\alpha^*)}||\Theta_{0:T}^{\pi^{\mathsf{s}}}\right).$$

We now show that if $C^{(0)} = \Pi^{D,H}(\mathcal{M})$ in Algorithm 1, then

$$\lim_{k\to\infty} \mathbb{E}\left[KL\left(\Theta_{0:T}^{(C^{(k)},\alpha^{(k)})}||\Theta_{0:T}^{\pi^{\mathsf{s}}}\right)\right] = v^*$$

with probability 1. Theorem 3.1 of (Collins and McNamara, 1998) shows that every final state distribution of a finite horizon MDP achieved by a Markovian, randomized policy can be achieved with a mixture of Markovian, deterministic policies. Consider an MDP $\mathcal{M}'$ whose states are possible histories from $t = 0$ to $T$ of the MDP $\mathcal{M}$. The possible transitions between the states of $\mathcal{M}'$ are defined via the state-action histories on $\mathcal{M}$. A Markovian policy on $\mathcal{M}'$ is a history dependent policy on $\mathcal{M}$, and the final state distribution of $\mathcal{M}'$ is the distribution of histories of $\mathcal{M}$. By Theorem 3.1 of (Collins and McNamara, 1998), the mixture of Markovian, deterministic policies achieve every final state distribution of $\mathcal{M}'$, which implies that the mixture of history dependent, deterministic policies achieve every history distribution of $\mathcal{M}$. Consequently, if $C^{(0)} = \Pi^{D,H}(\mathcal{M})$,

$$\lim_{k\to\infty} \mathbb{E}\left[KL\left(\Theta_{0:T}^{(C^{(k)},\alpha^{(k)})}||\Theta_{0:T}^{\pi^{\mathsf{s}}}\right)\right] = KL\left(\Theta_{0:T}^{(C^{(0)},\alpha^*)}||\Theta_{0:T}^{\pi^{\mathsf{s}}}\right) = v^*$$

with probability 1. ■

*Proof of Proposition 2.8.* We first show that $KL(\mu||\Theta^{\pi^{\mathsf{s}}}) = v^*$.

Let $\theta$ be an arbitrary observation sequence. If $\theta \in \mathcal{L}^{\mathsf{a}}$, we have

$$\Pr^{\pi^{\mathsf{a}}}(\theta) = \nu \sum_{\substack{\xi\in Paths(\mathcal{M}) \\ O(\xi)=\theta}} \frac{\Pr^{\pi^{\mathsf{s}}}(\xi)}{\Pr^{\pi^{\mathsf{s}}}(\xi|O(\xi)\in\mathcal{L}^{\mathsf{a}})} = \frac{\nu\Pr^{\pi^{\mathsf{s}}}(\theta)}{\Pr^{\pi^{\mathsf{s}}}(\xi|O(\xi)\in\mathcal{L}^{\mathsf{a}})}.$$

Similarly, if $\theta \notin \mathcal{L}^{\mathsf{a}}$, $\Pr^{\pi^{\mathsf{a}}}(\theta) = (1-\nu)\Pr^{\pi^{\mathsf{s}}}(\theta)/\Pr^{\pi^{\mathsf{s}}}(\xi|O(\xi) \notin \mathcal{L}^{\mathsf{a}})$.

The KL divergence is equal to

$$KL(\mu||\Theta^{\pi^{\mathsf{s}}}) = KL\left(Ber(\nu)||Ber(\Pr^{\pi^{\mathsf{s}}}(\xi|O(\xi) \in \mathcal{L}^{\mathsf{a}}))\right).$$

We now show that the optimal value of (2.17) is lower bounded by $KL(\mu||\Theta^{\pi^{\mathsf{s}}})$. Consider a binary clustering $C$ of the observation sequences such that an observation sequence $\theta$ is in $C$ if and only if $\theta \in \mathcal{L}^{\mathsf{a}}$. By definition $\Pr^{\pi^{\mathsf{s}}}(C) = \Pr^{\pi^{\mathsf{s}}}(\xi|O(\xi) \in \mathcal{L}^{\mathsf{a}})$.

Let $\mu^*$ be an optimal distribution of observation sequences for (2.17). If $\Pr(C|\mu^*) < \nu$ then $\Pr^{\pi^{\mathsf{a}}}(s_0 \models \Diamond R^{\mathsf{a}}) < \nu$. Hence, $\Pr(C|\mu^*) \geq \nu$. Using (2.1) and the binary clustering, we have

$$KL(\mu^*||\Theta^{\pi^{\mathsf{s}}}) = v^* \geq KL\left(Ber(\nu)||Ber(\Pr^{\pi^{\mathsf{s}}}(\xi|O(\xi) \in \mathcal{L}^{\mathsf{a}}))\right).$$

Since $v^*$ is the optimal value of (2.17), we have $KL(\mu||\Theta^{\pi^{\mathsf{s}}}) = v^*$.

Note that $\Pr(\xi \models \Diamond R^{\mathsf{a}}|\xi \sim \mu) = \nu$ due to the acceptance condition in the **if** statement.

We now derive the time complexity of Algorithm 4. Sampling a path under a stationary reference policy $\pi^{\mathsf{s}}$ takes $\mathcal{O}(|\mathcal{S}||\mathcal{A}|\mathbb{E}_{\xi\sim\pi^{\mathsf{s}}}[len(\xi))])$ time in expectation. For a given random observation sequence $O(\xi)$, determining whether $O(\xi) \in \mathcal{L}^{\mathsf{a}}$ is equivalent to the string acceptance problem for NFAs, which has a time complexity of $\mathcal{O}(|\mathcal{S}|^2\mathbb{E}_{\xi\sim\pi^{\mathsf{s}}}[len(\xi)])$ in expectation. If $c \leq \nu$, sampling a path $\xi$ such that $O(\xi) \in \mathcal{L}^{\mathsf{a}}$ takes $\Pr^{\pi^{\mathsf{s}}}(\xi|O(\xi) \in \mathcal{L}^{\mathsf{a}})^{-1}$ time in expectation. Otherwise, sampling a path $\xi$ such that $O(\xi) \notin \mathcal{L}^{\mathsf{a}}$ takes $\Pr^{\pi^{\mathsf{s}}}(\xi|O(\xi) \notin \mathcal{L}^{\mathsf{a}})^{-1}$ time in expectation. If $c \leq \nu$, finding a path $\xi'$ such that $\xi' \models \Diamond R^{\mathsf{a}}$ and $O(\xi') = O(\xi)$ is equivalent to finding an accepting trace for a given string in NFAs, which has a time complexity of $\mathcal{O}(|\mathcal{S}|^2\mathbb{E}_{\xi\sim\pi^{\mathsf{s}}}[len(\xi)|O(\xi) \in \mathcal{L}^{\mathsf{a}}])$ in expectation. Similarly, if $c > \nu$, finding a path $\xi'$ such that $\xi' \not\models \Diamond R^{\mathsf{a}}$ and $O(\xi') = O(\xi)$ has a time complexity of $\mathcal{O}(|\mathcal{S}|^2\mathbb{E}_{\xi\sim\pi^{\mathsf{s}}}[len(\xi)|O(\xi) \notin \mathcal{L}^{\mathsf{a}}])$ in expectation. Overall, the expected time complexity of Algorithm 4 is at the order of

$$\frac{\nu|\mathcal{S}|^2|\mathcal{A}|\mathbb{E}_{\xi\sim\pi^{\mathsf{s}}}\left[len(\xi)|O(\xi) \in \mathcal{L}^{\mathsf{a}})\right]}{\Pr^{\pi^{\mathsf{s}}}(\xi|O(\xi) \in \mathcal{L}^{\mathsf{a}})} + \frac{(1-\nu)|\mathcal{S}|^2|\mathcal{A}|\mathbb{E}_{\xi\sim\pi^{\mathsf{s}}}\left[len(\xi)|O(\xi) \notin \mathcal{L}^{\mathsf{a}}\right]}{\Pr^{\pi^{\mathsf{s}}}(\xi|O(\xi) \notin \mathcal{L}^{\mathsf{a}})}.$$

Since $len(\xi) \geq 0$, the expected time complexity is bounded by

$$\mathcal{O}\left(\frac{\nu|\mathcal{S}|^2|\mathcal{A}|\mathbb{E}_{\xi\sim\pi^{\mathsf{s}}}\left[len(\xi))\right]}{\Pr^{\pi^{\mathsf{s}}}(\xi|O(\xi)\in\mathcal{L}^{\mathsf{a}})^2}+\frac{(1-\nu)|\mathcal{S}|^2|\mathcal{A}|\mathbb{E}_{\xi\sim\pi^{\mathsf{s}}}\left[len(\xi)\right]}{\Pr^{\pi^{\mathsf{s}}}(\xi|O(\xi)\notin\mathcal{L}^{\mathsf{a}})^2}\right).$$

■

# Chapter 3: Minimally-Dependent Multiagent Systems that are Robust to Communication Loss

In this chapter[1], we study multiagent systems that are robust to communication losses. In detail, we study sequential multiagent decision problems formulated as transition-independent multiagent MDPs[2] (Becker et al., 2003) and reach-avoid objectives (Baier and Katoen, 2008). In this setting, a group of agents cooperate by following a *joint policy* that is a mapping from the agents' states to their actions. However, due to the communication losses, they may not always have perfect information on each other's state.

We develop a simulation-based decentralized policy execution mechanism, *imaginary play*, for the execution of the joint policy in a transition-independent multiagent MDP during communication losses. Under this mechanism, each agent maintains imaginary versions of their teammates' states and actions using the pre-agreed-upon joint policy and a model of the environment's stochastic dynamics during periods of lost communication. By maintaining such imaginary copies of their teammates, each agent may act according to a model of how their teammates are likely to behave, without receiving any communicated information from them. Once communication is re-established the agents share updates, correct their imaginary models, and proceed with policy execution as normal until communication is lost again, or until the team's task is complete.

[1]The research presented in this chapter is published in (Karabag et al., 2022a). Mustafa O. Karabag formulated the problem, derived the technical results, and wrote the paper.

[2]For transition-independent multiagent MDPs, we use the definition given in (Becker et al., 2003). In a transition-independent multiagent MDP, each agent has its own state and action spaces, the joint state and action spaces are Cartesian products of the individual state and action spaces, respectively, and the next state distribution of an agent is independent of the other agents' states and actions given the agent's state and action. While Becker et al. (2003) considers decentralized control policies, i.e., the action of an agent is a function of only its own state, we consider centralized joint control policies, i.e., the action of an agent is a function of all agents' states.

We use the total correlation (Watanabe, 1960) – a generalization of the mutual information – of the stochastic state-action process induced by the joint policy as a measure of how reliant that particular policy is on communication. To relate this measure to the performance of the policy, we provide lower bounds on the value function achieved during intermittent communication, in terms of the total correlation of the policy and the value function it achieves when communication is available. In addition to the policy synthesis algorithm described below, this lower bound provides a means to select communication resources that are sufficient to achieve a particular performance while using noisy communication channels.

To synthesize *minimum-dependency policies* that remain performant under intermittent communication, we present an algorithm that maximizes a proxy to the lower bound described above. This optimization problem is formulated as a difference of convex terms. We solve for local optima using the convex-concave procedure (Yuille and Rangarajan, 2001).

Numerical results empirically demonstrate the effectiveness of the proposed algorithms for communication-free policy execution and for the synthesis of minimum-dependency joint policies. When communication is not restricted, the synthesized minimum-dependency policies enjoy task performance that is similar to a baseline policy that does not take potential communication losses into account. However, the minimum-dependency policies require minimal coordination between agents; the total correlation value of their joint state-action processes is significantly lower than the total correlation value of the process induced by the baseline policy.

As a result, the performance of the minimum-dependency policies remain constant, even when communication between agents is restricted to be entirely unavailable. By contrast, we observe a significant degradation in the performance of the baseline policy when communication is lost.

### Summary of Contributions

- We develop a simulation-based decentralized policy execution algorithm, *imaginary play*, for the execution of the joint policy in a transition-independent multiagent MDP during communication losses.
- We propose an information theoretical measure, *total correlation*, to quantify the dependencies between the agents.
- We consider different communication loss models and prove lower bounds on the value function achieved under intermittent communication, in terms of the total correlation of the policy and the value function it achieves when communication is available.
- We present an optimization problem that maximizes a proxy to lower bounds to find *minimum-dependency policies* that remain performant under intermittent communication.
- We demonstrate our framework on different numerical examples and empirically show that minimum-dependency policies do not suffer from significant performance degradation when communication is lost.

**Outline** In §3.1, we discuss related work. In §3.2, we introduce preliminary background material as well as the notation used throughout the chapter. We present our problem statement and an illustrative running example in §3.3. The proposed algorithms for policy execution during communication losses are presented in §3.4. The theoretical results and their implications are discussed in §3.5 and §3.6. In §3.7, we present the proposed formulation and solution to the policy synthesis problem, before presenting the experimental results in §3.8. We include the details of the proofs of the theoretical results §3.9.

## 3.1 Related Work

**Multiagent MDPs** Multiagent decision-making problems have been formulated using several models, e.g., multiagent Markov decision processes (MDPs) (Boutilier, 1996). Our problem setting, in which each agent has independent transitions and may only observe their own local state, is most similar to transition-independent decentralized MDPs (Dec-MDPs) (Becker et al., 2003). However, while this work considers the fully decentralized setting – the agents cannot communicate at all – we consider the setting in which communication is allowed but unreliable. We note that Dec-MDPs are a special case of decentralized partially observable MDPs (Dec-POMDPs) (Oliehoek and Amato, 2016), which are notoriously difficult to solve in general when the agents cannot communicate. In fact, even policy synthesis for finite-horizon transition-independent Dec-MDPs without communication is NP-complete (Goldman and Zilberstein, 2004).

**Policy execution with communication constraints** Prior work for multiagent systems considers imposing specific communication structures between the agents, either as a dependency graph (Guestrin et al., 2001), or as a subset of joint states at which the agents may communicate (Melo and Veloso, 2011). In addition to these fixed communication structures, the papers (Becker et al., 2009; Wu et al., 2011) consider communication as an explicit action that can be taken by the agents, leading to dynamic communication structures that change over time. While all of the above works consider synthesizing optimal behavior according to specific communication structures, our work studies multiagent systems that are robust to unpredictable communication losses.

To render the multiagent systems robust to communication loss, our work aims to minimize intrinsic dependencies between the agents. As a measure of such dependencies, we use the total correlation (Watanabe, 1960) – an information theoretic measure – of the state-action process induced by the joint policy. Information theo-

retic measures have been studied in single-agent MDPs (Savas et al., 2019; Leibfried and Grau-Moya, 2020; Tanaka et al., 2021; Eysenbach et al., 2021). In particular, (Tanaka et al., 2021) synthesizes single-agent policies that minimize the transfer entropy from the state process to the action process with the purpose of minimizing the reliance of the policy on the underlying state process. By contrast, our work considers a multiagent setting and introduces information theoretic measures with the specific purpose of providing guarantees on the performance of the team under communication loss. In the context of single-agent reinforcement learning, (Eysenbach et al., 2021) proposes to minimize the mutual information between the underlying state process and a latent state process that influences the agent's actions. By contrast, we study the multiagent setting and provide bounds on the performance of the entire team, when the agents have intermittent communication. Furthermore, we provide an optimization problem to synthesize joint policies that are robust to communication loss. In the multiagent reinforcement learning setting, (Wang et al., 2020) consider minimizing the mutual information between the state processes and the messages shared between the agents, but do not provide theoretical result on the performance of the team when communication is only intermittently available.

The centralized training decentralized execution paradigm in has recently drawn attention in multiagent reinforcement learning (Rashid et al., 2018; Sunehag et al., 2018; Son et al., 2019; Mahajan et al., 2019). These works enforce independence between the agents by imposing that the team's value function can be decomposed into local functions for each of the agents. In our work, we do not consider decomposition of the value function, but instead directly synthesize a joint policy that leads to intrinsic independence between agents. Another method to compute policies for decentralized execution, is to post-process a given joint policy. For example, (Dobbe et al., 2017) uses the rate-distortion framework (Cover and Thomas, 2012) for this purpose. Our work does not assume a joint policy to be given a priori; we instead directly synthesize a joint policy that minimizes dependencies.

**Partially observable MDPs** As discussed above, prior works tackle communication loss by making the policies fully decentralized (Rashid et al., 2018; Son et al., 2019), or by having the agents maintain beliefs about their teammates (Becker et al., 2009; Wu et al., 2011). While belief-based myopic approaches lead to high reward for a single step, they do not guarantee optimality over entire paths. Meanwhile, using belief-based approaches to reason over extended time horizons necessitates maintaining consistent beliefs between the agents, which is challenging. Unlike for single-agent POMDPs, in a multiagent MDP with communication limitations the policy of a particular agent is a function of its beliefs not only over its own state, but also over those of its teammates. Symmetrically, the control processes of the teammates also depend on their beliefs over the particular agent's state. This cyclic relationship induces a coupling between the belief processes of the agents. To the best of our knowledge, there has been limited work in resolving this issue to use beliefs for multiagent planning. One solution is to communicate agent histories whenever an agent receives an observation that is inconsistent with its belief (Wu et al., 2011). Another approach is to design the control policies of the agents from the perspective of a single centralized planner by considering the multiagent MDP as a single-agent POMDP problem (Nayyar et al., 2013); this approach requires shared memory between the agents that is sufficient to ensure belief consistency. Instead of maintaining such belief distributions, in our work each agent creates imaginary copies of its teammates when communication is lost; this idea is similar in spirit to the concept of digital twins (Boschert and Rosen, 2016). Combined with total correlation, the proposed imaginary play algorithm leads to performance guarantees over the entire path.

## 3.2 Preliminaries

In this section, we outline several definitions and notation used throughout the chapter. Given a finite collection of $N$ agents – which we index by $i \in [N] = \{1, 2, \ldots, N\}$ – we model the dynamics of each individual agent using a Markov

decision process (MDP) $\mathcal{M}^i = (\mathcal{S}^i, \mathcal{A}^i, \mathcal{T}^i, s_0^i)$ as introduced in §2.2. We use $\Delta(\mathcal{S}^i)$ to denote the set of all probability distributions over the state space $\mathcal{S}^i$. With an abuse of notation, we use $\mathcal{T}^i(s^i, a^i)$ to denote the probability distribution over $\mathcal{S}^i$ at state $s^i$ under action $a^i$. A *(state-action) path* $\xi^i$ in the MDP $\mathcal{M}^i$ is an infinite sequence $\xi^i = s_0^i a_0^i s_1^i a_1^i \ldots$ of state-action pairs such that for every $t = 0, 1, \ldots$, $\mathcal{T}^i(s_t^i, a_t^i, s_{t+1}^i) > 0$.

Given such a collection of agents, along with their corresponding MDPs $\mathcal{M}^i$, we formulate the team's decision problem as a *cooperative transition-independent multiagent MDP* $\boldsymbol{\mathcal{M}}$. The multiagent MDP setting is considered cooperative because all agents share a common objective. A transition-independent multiagent MDP involving $N$ agents, each of which is modeled by an MDP $\mathcal{M}^i = (\mathcal{S}^i, \mathcal{A}^i, \mathcal{T}^i, s_0^i)$, is given by the tuple $\boldsymbol{\mathcal{M}} = (\boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{T}}, \boldsymbol{s_0})$. Here, $\boldsymbol{\mathcal{S}} = \mathcal{S}^1 \times \mathcal{S}^2 \times \ldots \times \mathcal{S}^N$ is the finite set of joint states, $\boldsymbol{\mathcal{A}} = \mathcal{A}^1 \times \mathcal{A}^2 \times \ldots \times \mathcal{A}^N$ is the finite set of joint actions, $\boldsymbol{\mathcal{T}} : \boldsymbol{\mathcal{S}} \times \boldsymbol{\mathcal{A}} \times \boldsymbol{\mathcal{S}} \to [0, 1]$ is the joint transition probability function, and $\boldsymbol{s_0} = (s_0^1, \ldots, s_0^N)$ is the joint initial state. The joint transition function $\boldsymbol{\mathcal{T}}$ is defined as $\boldsymbol{\mathcal{T}}(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{y}) = \prod_{i=1}^{N} \mathcal{T}(s^i, a^i, y^i)$ for all $\boldsymbol{s} = (s^1, \ldots, s^N), \boldsymbol{y} = (y^1, \ldots, y^N) \in \boldsymbol{\mathcal{S}}$ and $\boldsymbol{a} = (a^1, \ldots, a^N) \in \boldsymbol{\mathcal{A}}$. We note that the definition of the joint transition function $\boldsymbol{\mathcal{T}}$ assumes that the dynamics of the individual agents are independent. With an abuse of notation, we use $\boldsymbol{\mathcal{T}}(\boldsymbol{s}, \boldsymbol{a})$ to denote the probability distribution over $\boldsymbol{\mathcal{S}}$ at joint state $\boldsymbol{s}$ under joint action $\boldsymbol{a}$. We use $\boldsymbol{\xi} = \boldsymbol{s}_0 \boldsymbol{a}_0 \boldsymbol{s}_1 \boldsymbol{a}_1 \ldots$ to denote the joint state-action path of the agents. Throughout this chapter, we use *path* to refer state-action sequences. The joint path $\boldsymbol{\xi}$ is the union of individual paths $\xi^1, \ldots, \xi^N$.

A *(stationary) joint policy* $\pi_{joint} : \boldsymbol{\mathcal{S}} \to \Delta(\boldsymbol{\mathcal{A}})$ is a mapping from a particular joint state to a probability distribution over joint actions. We use $\pi_{joint}(\boldsymbol{s}, \boldsymbol{a})$ to denote the probability that action $\boldsymbol{a}$ is selected by $\pi_{joint}$ given the team is in joint state $\boldsymbol{s}$.

In this chapter, we consider team reach-avoid problems. That is, the team's objective is to collectively reach some target set $\boldsymbol{\mathcal{S}}_{\mathcal{T}} \subseteq \boldsymbol{\mathcal{S}}$ of states, while avoiding a set $\boldsymbol{\mathcal{S}}_{\mathcal{A}} \subseteq \boldsymbol{\mathcal{S}}$ of states. The centralized planning problem then, is to solve for a team policy

$\pi_{joint}$ maximizing the probability of reaching $\boldsymbol{\mathcal{S}}_{\mathcal{T}}$ from the team's initial joint state $\boldsymbol{s_0}$, while avoiding $\boldsymbol{\mathcal{S}}_{\mathcal{A}}$. We call this probability value the reach-avoid probability. More formally, we say that a *path* $\boldsymbol{\xi} = \boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots$ successfully satisfies the reach-avoid specification if there exists some time $M$ such that $\boldsymbol{s}_M \in \boldsymbol{\mathcal{S}}_{\mathcal{T}}$ and for all $t < M$, $\boldsymbol{s}_t \notin \boldsymbol{\mathcal{S}}_{\mathcal{A}}$.

For notational convenience, we use $\boldsymbol{s}^{-i} \in \mathcal{S}^1 \times \ldots \times \mathcal{S}^{i-1} \times \mathcal{S}^{i+1} \times \ldots \times \mathcal{S}^N$ to denote the states of agent $i$'s teammates, excluding agent $i$ itself. By $\boldsymbol{\mathcal{S}}^{-i} = \mathcal{S}^1 \times \ldots \times \mathcal{S}^{i-1} \times \mathcal{S}^{i+1} \times \ldots \mathcal{S}^N$, we denote the set of all collections of the states of agent $i$'s teammates. We similarly use $\boldsymbol{a}^{-i}$ and $\boldsymbol{\mathcal{A}}^{-i}$ to denote the actions of agent $i$'s teammates and the set of all possible such collections of actions, respectively.

We use $x_{\boldsymbol{s},\boldsymbol{a}}$ to denote the occupancy measure of the state-action pair $(\boldsymbol{s}, \boldsymbol{a})$, i.e., the expected number of times that action $\boldsymbol{a}$ is taken at state $\boldsymbol{s}$. Similarly, $x_{s^i,a^i}$ denotes the the occupancy measure of the state-action pair $(s^i, a^i)$ for agent $i$. We note that $x_{s^i,a^i} = \sum_{\boldsymbol{s}^{-i}\in\boldsymbol{\mathcal{S}}^{-i}} \sum_{\boldsymbol{a}^{-i}\in\boldsymbol{\mathcal{A}}^{-i}} x_{s^i,\boldsymbol{s}^{-i},a^i,\boldsymbol{a}^{-i}}$.

$Partitions([N])$ to denote the set of all possible partitions of $[N] = \{1, \ldots, N\}$. The notation $[b_a]_{a\in C}$ denotes the unordered list of elements $b_{a_1}, \ldots, b_{a_{|C|}}$ where $a_1, \ldots, a_{|C|}$ are distinct elements of set $C$.

The entropy (Cover and Thomas, 2012) of a discrete random variable $Y$ with a support $\mathcal{Y}$ is $H(Y) = -\sum_{y\in\mathcal{Y}} \Pr(Y = y) \log(\Pr(Y = y))$.

## 3.3 Problem Statement

In this work we study the cooperative execution of joint policies when communication between the agents is intermittent, and in some cases entirely absent. We begin by discussing the inter-agent communication that is necessary, in general, for team policy execution before we present the problem statement.

The agents operate in the environment by collectively executing a joint policy $\pi_{joint}$. Each agent only has access to its own local state and action information. The

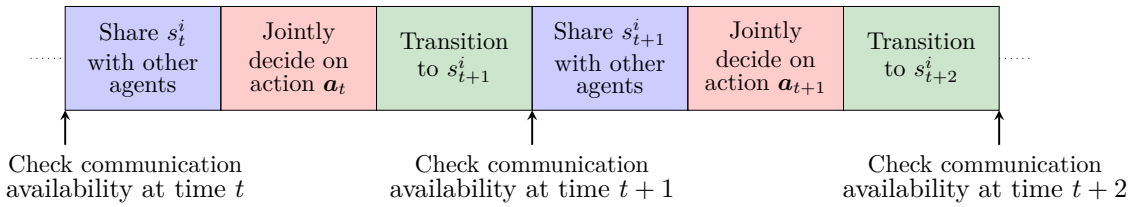



Figure 3.1: An illustration of the procedure for joint policy execution. At each decision step, all agents simultaneously check whether communication is available. If it is available, the agents share their local states in order to obtain the current joint state $\boldsymbol{s}_t$ before agreeing upon a joint action $\boldsymbol{a}_t$ sampled from the joint policy $\pi_{joint}$. Otherwise, the agents execute $\pi_{joint}$ using imaginary play, outlined in Algorithm 5.

agents must communicate their local states $s_t^i$ at each timestep $t$ and use $\pi_{joint}$ to collectively decide on a joint action $\boldsymbol{a}$, as is illustrated in Figure 3.1. Each agent executes its own local component $a^i$ of the selected joint action and resultingly transitions to its next local state $s_{t+1}^i$.

We note that the joint policy requires communication between the agents at every time step. On the other hand, if the team suffers a communication failure at any given timestep, then they will not be able to share the necessary information to execute the joint policy in the manner described above.

**Problem 3.1.** (1) *Create a planning algorithm that enables the agents to perform decentralized execution of the joint policy when communication is lost.* (2) *Quantify the performance of the team when such an algorithm is used during communication losses.* (3) *Synthesize joint policies that remain performant, even when communication is lost.*

**Remark 3.1.** *We consider a control problem where the environments, i.e., MDPs of the agents, and the reward function, i.e., reach-avoid specification, are fully known.*

**Example.** *We present the running example illustrated in Figure 3.2 to help motivate the above problems. Two robots $R_1$ and $R_2$ must simultaneously navigate to their respective targets $T_1$ and $T_2$. The robots must also maintain a pre-specified minimum*

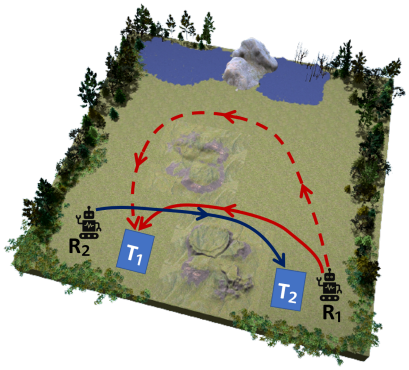



Figure 3.2: A two-agent navigation example. Two robots, $R_1$ and $R_2$, must navigate to their respective targets, $T_1$ and $T_2$, while avoiding collisions with each other. The terrain necessitates that each robot navigates through one of two valleys, while avoiding the water at the top of the map. During policy execution, each robot may only observe its own location, however, the agents communicate their locations with each other when such communication is possible. The colored curves illustrate different paths that the robots might take, depending on the availability of communication.

*distance from each other during navigation to reduce the risk of the robots colliding. Furthermore, rough terrain makes large portions of the navigation environment impassable, requiring the robots to navigate through one of two narrow valleys in order to reach their targets. Finally, a lake of water presents risk to the robots; if either of them accidentally falls into the water, then the team fails its task. The team's task is only considered complete once both robots have safely navigated to their respective targets. The objective of the agents is to complete this task with as high a probability as possible.*

*Given this team task, the robots may both choose to navigate through the bottom valley in order to reach their targets. This route is shorter than traveling through the top valley for both robots, and it avoids passing near the dangerous body of water. However, they must take turns when passing through the shared bottom valley to ensure that the robots never get too close to each other. Such behavior requires communication; both agents should share their current location and intended next action in order to avoid simultaneously entering the valley.*

*By contrast, if no communication is available, the robots may instead choose to navigate through different valleys altogether. This joint behavior increases the risk that one might fall into the water, but it removes the requirement that the robots communicate.*

## 3.4 Decentralized Policy Execution Under Communication Loss

Consider a scenario in which the team of agents lose communication during the execution of a joint policy. Under such circumstances, the agents cannot execute the policy as outlined in the previous section and as illustrated in Figure 3.1. Each agent must instead decide on its local action for itself, without knowing the local states or actions of other teammates. To achieve this decentralized execution of the joint policy, we propose to use *imaginary play*; each agent maintains imaginary copies of its teammates during periods of communication loss. That is, given the joint policy, the stochastic dynamics of the multigent MDP, and the states of their teammates at the last timestep before communication was lost, each agent in the group maintains simulated copies of their teammates' states. Each agent then uses its own imaginary version of the entire team to sample a joint action from the policy, executes its own local component of that joint action, and then simulates the next states of its imaginary teammates. In the next time step, this process repeats.

Algorithm 5 details this process of joint policy execution through imaginary play. Before the communication breaks, every Agent $i$ shares its state $s_t^i$ with its teammates at every time step, and the agents collectively decide on a joint action $\boldsymbol{a}_t$. When the communication breaks at time $t_{loss}$, every agent $i$ starts to play with imaginary teammates. That is, based on the last joint action $\hat{\boldsymbol{a}}_{t_{loss-1},i}$ prior to communication loss, every Agent $i$ uses the joint transition function $\boldsymbol{\mathcal{T}}$ to sample an imaginary state $\hat{s}^j_{t_{loss-1}+1,i}$ for each of its teammates. Here, $\hat{s}^j_{t,i}$ denotes Agent $i$'s imagined copy of Agent $j$'s state at time $t$, and $\hat{\boldsymbol{a}}_{t,i}$ denotes Agent $i$'s imagined copy of the joint action

**Algorithm 5:** Policy Execution with Imaginary Play

$t_{loss} = \infty$.
**for** $t = 0, 1, \ldots$ **do**
  **if** *Communication is possible* **then**
    **For every** $i \in [N]$ **do in parallel**
      Share $s_t^i$ with other agents.
      Set $\hat{s}_{t,i}^j = s_t^j$ for all $j \neq i$.
      Jointly decide on an action $\boldsymbol{a}_t \sim \pi_{joint}(\boldsymbol{s}_t)$.
      Set $\hat{\boldsymbol{a}}_{t,i} = \boldsymbol{a}_t$.
      Execute $a_t^i$ and transition to $s_{t+1}^i \sim \mathcal{T}(s_t^i, a_t^i)$.
  **else**
    Set $t_{loss} = t$.
    **break**
**for** $t = t_{loss}, t_{loss} + 1, \ldots$ **do**
  **For every** $i \in [N]$ **do in parallel**
    **if** *t = 0* **then**
      Set $\hat{s}_{t,i}^j = s_0^j$ for all $j \neq i$.
    **else**
      Sample $\hat{s}_{t,i}^j \sim \mathcal{T}^j(\hat{s}_{t-1,i}^j, \hat{a}_{t-1,i}^j)$ for all $j \neq i$.
    Decide on an action $\hat{\boldsymbol{a}}_{t,i} \sim \pi_{joint}(\hat{s}_{t,i}^1, \ldots, \hat{s}_{t,i}^{i-1}, s_t^i, \hat{s}_{t,i}^{i+1}, \ldots, \hat{s}_{t,i}^N)$.
    Execute $\hat{a}_{t,i}^i$ and transition to $s_{t+1}^i \sim \mathcal{T}(s_t^i, \hat{a}_{t,i}^i)$.

at time $t$. Then, at every time step $t \geq t_{loss}$, every agent $i$ samples a joint action $\hat{\boldsymbol{a}}_{t,i}$ using the joint policy and these imagined teammate states $\hat{s}_{t,i}^1, \ldots, \hat{s}_{t,i}^N$. Every agent $i$ then executes the local part $\hat{a}_{t,i}^i$ of its joint action $\hat{\boldsymbol{a}}_{t,i}$ and transitions to its next local state $s_{t+1,i}^i$. Based on its imagined joint action $\hat{\boldsymbol{a}}_{t,i}$ and the previous imagined teammate states $\hat{s}_{t,i}^1, \ldots, \hat{s}_{t,i}^N$, every Agent $i$ also samples next imaginary states $\hat{s}_{t+1,i}^1, \ldots, \hat{s}_{t+1,i}^N$ for its teammates.

We remark that while every agent operates cooperatively with its imaginary teammates under a communication loss, the objective of the team is evaluated with respect to the true joint state. We also remark that the proposed decentralized policy execution mechanism assumes that every agent has knowledge on the other agents' MDPs.

**Algorithm 6:** Policy Execution with Intermittent Communication

**for** $t = 0, 1, \ldots$ **do**
  **if** *Communication is possible* **then**
    **For every** $i \in [N]$ **do in parallel**
      Share $s_t^i$ with other agents.
      Set $\hat{s}_{t,i}^j = s_t^j$ for all $j \neq i$.
      Jointly decide on an action $\boldsymbol{a}_t \sim \pi_{joint}(\boldsymbol{s}_t)$.
      Set $\hat{\boldsymbol{a}}_{t,i} = \boldsymbol{a}_t$.
      Execute $a_t^i$ and transition to $s_{t+1}^i \sim \mathcal{T}(s_t^i, a_t^i)$.
  **else**
    **For every** $i \in [N]$ **do in parallel**
      **if** $t = 0$ **then**
        Set $\hat{s}_{t,i}^j = s_0^j$ for all $j \neq i$.
      **else**
        Sample $\hat{s}_{t,i}^j \sim \mathcal{T}^j(\hat{s}_{t-1,i}^j, \hat{a}_{t-1,i}^j)$ for all $j \neq i$.
      Decide on an action $\hat{\boldsymbol{a}}_{t,i} \sim \pi_{joint}(\hat{s}_{t,i}^1, \ldots, \hat{s}_{t,i}^{i-1}, s_t^i, \hat{s}_{t,i}^{i+1}, \ldots, \hat{s}_{t,i}^N)$.
      Execute $\hat{a}_{t,i}^i$ and transition to $s_{t+1}^i \sim \mathcal{T}(s_t^i, \hat{a}_{t,i}^i)$.

We note that while the agents do not communicate after the communication loss, for every agent the distribution of its imaginary path is the same as the distribution of the team's joint path under full communication. This is because the agents follow the same joint policy and (imaginary and real) transitions happen according to the same model. At the same time, every agent's process is conditionally independent from the other agents given $\boldsymbol{s}_{t_{loss}}$ and $\boldsymbol{a}_{t_{loss}}$. Consequently, team's joint path distribution under imaginary play after $t_{loss}$ is the product distribution of marginals of the distribution under full communication conditioned on $\boldsymbol{s}_{t_{loss}}$ and $\boldsymbol{a}_{t_{loss}}$.

**Intermittent communication loss** In some scenarios, communication failures may be intermittent as opposed to being persistent. That is, the agents may re-gain communication capabilities after periods of communication loss. For such scenarios, we propose that the agents follow imaginary play whenever communication is lost, update their imaginary representations when communication is re-established, and

coordinate directly with their real teammates for as long as communication remains available. Algorithm 6 describes this proposed approach for policy execution with intermittent communication. Different from Algorithm 6, when the communication is available every Agent $i$ updates its imaginary state $\hat{s}^j_{t,i}$ with the true state $s^j_t$ for every other Agent $j$. We note that Algorithm 6 is the same as Algorithm 5 when the communication loss is permanent.

Similar to the imaginary, while the agents do not communicate during a communication loss happening between time steps $t' + 1$ and $t''$, for every agent the distribution of its imaginary path is the same as the distribution of the team's joint path under full communication. This is because the agents follow the same joint policy and (imaginary and real) transitions happen according to the same model. At the same time, every agent's process between $t' + 1$ and $t''$ is conditionally independent from the other agents given $\boldsymbol{s}_{t'}$ and $\boldsymbol{a}_{t'}$. Consequently, team's joint path distribution between $t'+1$ and $t''$ is the product distribution of marginals of the distribution under full communication conditioned on $\boldsymbol{s}_{t'}$ and $\boldsymbol{a}_{t'}$.

**Partition communication groups** In some scenarios, a subset of agents can communicate with each other even when the whole team cannot communicate. As a more general communication availability scheme, we use a partition of $[N]$ to define the communication availability between the agents. For example, the partition $\{\{1, 4\}, \{2, 3, 6\}, \{5\}\}$ of $[N]$ at time $t$ means that Agents 1 and 4 knows each other's state at time t, and can coordinate on a joint action at time $t$. Similarly, Agents 2, 3, and 6 can communicate at time $t$. Agent 5, on the other hand, cannot coordinate with any other agent. We denote the partition at time $t$ with $\mathcal{P}_t$ and refer to the subsets of $\mathcal{P}_t$ as *communication groups*. We have the following assumption on the structure of partition groups that ensures the information symmetry between agents.

**Assumption 3.1.** *For all $t > 0$, if $\mathcal{P}_{t-1} \neq \{[N]\}$, then $\mathcal{P}_t = \mathcal{P}_{t-1}$ or $\mathcal{P}_t = \{[N]\}$.*

**Algorithm 7:** Policy Execution with Intermittent Communication with Partition Communication Groups

**for** $t = 0, 1, \ldots$ **do**
  **if** $\mathcal{P}_t = \{[N]\}$ **then**
    **For every** $i \in [N]$ **do in parallel**
      Share $s_t^i$ with other agents.
      Set $\hat{s}_{t,i}^j = s_t^j$ for all $j \neq i$.
      Jointly decide on an action $\boldsymbol{a}_t \sim \pi_{joint}(\boldsymbol{s}_t)$.
      Set $\hat{\boldsymbol{a}}_{t,i} = \boldsymbol{a}_t$.
      Execute $a_t^i$ and transition to $s_{t+1}^i \sim \mathcal{T}(s_t^i, a_t^i)$.
  **else**
    **For every** $G \in \mathcal{P}_t$ **do in parallel**
      **if** $t = 0$ **then**
        Set $\hat{s}_{t,G}^j = s_0^j$ for all $j \notin G$.
      **else**
        **if** $\mathcal{P}_{t-1} = \{[N]\}$ **then**
          For all $j \notin G$, set $\hat{s}_{t-1,G}^j = \hat{s}_{t-1,i}^j$ for some $i \in G$.
        Sample $\hat{s}_{t,G}^j \sim \mathcal{T}^j(\hat{s}_{t-1,G}^j, \hat{a}_{t-1,G}^j)$ for all $j \notin G$.
      Within $G$, jointly select action $\hat{\boldsymbol{a}}_{t,G} \sim \pi_{joint}([s_t^i]_{i \in G}, [\hat{s}_{t,G}^i]_{i \notin G})$.
      **For every** $i \in G$ **do in parallel**
        Execute $\hat{a}_{t,G}^i$ and transition to $s_{t+1}^i \sim \mathcal{T}(s_t^i, \hat{a}_{t,G}^i)$.

Assumption 3.1 means that the team can communicate as a whole for at least one time-step before switching to a different partition. We note that this assumption is already satisfied for the cases considered in Algorithms 5 and 6.

Under varying communication partitions, we use a more general version of Algorithm 7 and allow communication groups to coordinate within themselves after $t_{loss}$. Algorithm 7 details the process of joint policy execution under Assumption 3.1. Different from Algorithm 6, the agents in a communication group $G$ jointly sample an action $\hat{\boldsymbol{a}}_{t,G}$ within their communication group $G$ and keep the same imaginary states $\hat{s}_{t-1,G}^j$ for the agents $j \notin G$ that are not a part of the communication group. We note that Algorithm 7 is the same as Algorithm 6 when $\mathcal{P}_t = \{[N]\}$ or $\mathcal{P}_t = \{\{1\}, \ldots, \{N\}\}$

for all $t \geq 0$.

## 3.5 Measuring the Intrinsic Dependencies Between the Agents

Given a joint policy, the team's performance under imaginary play will differ from the performance that would have been achieved under full communication. Recall that we measure the team's performance as their probability of reaching the set $\boldsymbol{\mathcal{S}}_{\mathcal{T}} \subseteq \boldsymbol{\mathcal{S}}$ of target joint states from the initial joint state $\boldsymbol{s_0}$, while avoiding $\boldsymbol{\mathcal{S}}_{\mathcal{A}} \subseteq \boldsymbol{\mathcal{S}}$.

Intuitively, the team's performance under imaginary play will depend on how much the behavior of any particular agent changes according to the behavior of its teammates, as well as on how much the behavior of an agent's imaginary teammates differs from that of its actual teammates. In other words, if the joint policy induces high intrinsic dependencies between the agents, then policy execution using imaginary play will lead to different outcomes than policy execution with fully available communication.

Total correlation (Watanabe, 1960) measures the amount of information shared between multiple random variables. Let $X^i$ be a random variable over the paths $\xi^i = s_0^i a_0^i s_1^i a_1^i \ldots$ of Agent $i$ and $\boldsymbol{X}$ be a random variable over the joint paths $\boldsymbol{\xi} = \boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1 \ldots$ of all agents induced by the joint policy $\pi_{joint}$ under full communication. We refer to the total correlation $C_{\pi_{joint}}$ of joint policy $\pi_{joint}$ as

$$C_{\pi_{joint}} = \left[\sum_{i=1}^{N} H(X^i)\right] - H(\boldsymbol{X}).$$

There are two contributing factors to the value of the total correlation. Firstly, if the actions of a particular agent depend on the local states of its teammates, then this will increase the value of the total correlation. Secondly, if the joint policy is randomized and the agents need to coordinate on an action – the action of each agent depends on the actions simultaneously selected by its teammates – then this will also increase the value of the total correlation.

If the total correlation is 0, then there are no dependencies between the agents, i.e., the path of any given agent is independent from those of its teammates. As the dependencies between the agents increase, so too does the value of the total correlation. We additionally remark that when there are only two agents, the total correlation of the state-action processes of the agents is equivalent to the mutual information between them.

We accordingly propose to use total correlation as measure of the intrinsic dependencies between the agents induced by a particular joint policy. In the next section, we relate the value of total correlation to the team's performance under communication loss.

## 3.6 Performance Guarantees Under Communication Loss

In this section, we provide lower bounds on the team's performance under a particular joint policy during communication loss. These theoretical results are accomplished by relating the total correlation of the joint policy to the distribution over paths induced by executing that policy using imaginary play. Motivated by these lower bounds, we also give an upper bound on the best achievable performance under communication losses for some cases.

**Relating total correlation to imaginary play** Let $\mathbf{\Gamma}^{full}$ be the distribution of joint paths induced by the joint policy executed with full communication. Also, let $\mathbf{\Gamma}_0^{img}$ be the distribution of joint paths under imaginary play with no communication, i.e., $t_{loss} = 0$ in Algorithm 5. We note that the distribution $\mathbf{\Gamma}_0^{img}$ is the product distribution of marginals of $\mathbf{\Gamma}^{full}$. By the definition of total correlation, we have

$$C_{\pi_{joint}} = \left[\sum_{i=1}^{N} H(X^i)\right] - H(\boldsymbol{X}) = KL(\mathbf{\Gamma}^{full}||\mathbf{\Gamma}_0^{img}).$$

From this definition, we observe that when $C_{\pi_{joint}} = 0$, the induced distributions $\mathbf{\Gamma}^{full}$ and $\mathbf{\Gamma}_0^{img}$ must be the same since $KL(\mathbf{\Gamma}^{full}||\mathbf{\Gamma}_0^{img}) = 0$. Furthermore, as the

value of $C_{\pi_{joint}}$ increases, the Kullback-Leibler (KL) divergence between $\mathbf{\Gamma}^{full}$ and $\mathbf{\Gamma}_0^{img}$ increases as well.

**On the closeness between path distributions induced by different communication availabilities** The value of $C_{\pi_{joint}}$ measures how much the distribution over paths $\mathbf{\Gamma}_0^{img}$ differs from $\mathbf{\Gamma}^{full}$ in the setting where the agents never communicate, i.e. $t_{loss} = 0$. We now consider a scenario in which the agents communicate and operate together for some time, then lose communication and switch to imaginary play at time $t_{loss} > 0$. Let $\mathbf{\Gamma}_{t_{loss}}^{img}$ be the distribution of joint paths for an arbitrary positive value of $t_{loss}$ in Algorithm 5. Intuitively, we expect that the initial period of communication should not increase the KL divergence between $\mathbf{\Gamma}^{full}$ and $\mathbf{\Gamma}_{t_{loss}}^{img}$ in comparison with the case when $t_{loss} = 0$. Lemma 3.1 confirms this intuition.

**Lemma 3.1.** *For every $t_{loss} \in \{0, 1, \ldots\} \cup \{\infty\}$ in Algorithm 5,*

$$KL(\mathbf{\Gamma}^{full} || \mathbf{\Gamma}_0^{img}) \geq KL(\mathbf{\Gamma}^{full} || \mathbf{\Gamma}_{t_{loss}}^{img}).$$

We can similarly show that arbitrary intermittent communication does not increase the KL divergence between the induced path distributions. Let $\Lambda = \mathcal{P}_0, \mathcal{P}_1, \ldots$ be an arbitrary sequence of communication paritions. The KL divergence between $\mathbf{\Gamma}^{full}$ and $\mathbf{\Gamma}_0^{img}$ is not higher than that between $\mathbf{\Gamma}^{full}$ and $\mathbf{\Gamma}_\Lambda^{int}$, where $\mathbf{\Gamma}_\Lambda^{int}$ is the distribution of paths under intermittent communication with an arbitrary sequence $\Lambda$ of communication partitions in Algorithm 7. Furthermore, as shown in the second half of Lemma 3.2, when $\Lambda = \mathcal{P}_0, \mathcal{P}_1, \ldots$ is a random sequence of communication partitions, the communication dropout rate $q$ is related to the KL divergence between the distributions.

**Assumption 3.2.** *The sequence $\Lambda = \mathcal{P}_0, \mathcal{P}_1, \ldots$ of communication partitions is sampled from a fixed probability distribution that is independent of $\pi_{joint}$ and the team's joint history $\boldsymbol{s}_0\boldsymbol{a}_0 \ldots \boldsymbol{s}_{t-1}\boldsymbol{a}_{t-1}$.*

**Lemma 3.2.** *Let $\Lambda = \mathcal{P}_0, \mathcal{P}_1, \ldots$ be an arbitrary sequence of communication partitions in Algorithm 7 that satisfies Assumption 3.1 and is fixed a priori. Then,*

$$KL(\mathbf{\Gamma}^{full} || \mathbf{\Gamma}_0^{img}) \geq KL(\mathbf{\Gamma}^{full} || \mathbf{\Gamma}_\Lambda^{int}).$$

*Let $\Lambda = \mathcal{P}_0, \mathcal{P}_1, \ldots$ be a random sequence of communication partitions that satisfies Assumption 3.2 such that $\min_t \Pr(\mathcal{P}_t = \{[N]\}) = 1 - q$, and $\mathbf{\Gamma}^{int} = \mathbb{E}_\Lambda [\mathbf{\Gamma}_\Lambda^{int}]$. Then,*

$$KL(\mathbf{\Gamma}^{full} || \mathbf{\Gamma}_0^{img}) \geq KL(\mathbf{\Gamma}^{full} || \mathbf{\Gamma}^{int})/q.$$

Lemmas 3.1 and 3.2 bound the KL divergence between path distributions when the communication availability is independent from the histories of the agents. In practice, communication availability may depend on the state-action processes of the agents. For example, in the multiagent navigation task depicted in Figure 3.2, the agents may not be able to communicate if they do not have line-of-sight, e.g., when they are on the opposite sides of the mountains. Lemma 3.3 shows a stronger result: The distribution over joint paths under imaginary play is close to $\mathbf{\Gamma}^{full}$ even when the communication availability is an arbitrary (potentially adversarial) function of the agents' histories.

**Lemma 3.3.** *Let $f : (\boldsymbol{\mathcal{S}} \times \boldsymbol{\mathcal{A}})^* \to \{$available, not available$\}$ be an arbitrary function that determines the communication availability based on the team's joint history such that $\lambda_0 = f(\varepsilon)$ and $\lambda_t = f(\boldsymbol{s}_0 \boldsymbol{a}_0 \ldots \boldsymbol{s}_{t-1} \boldsymbol{a}_{t-1})$. Let $\mathbf{\Gamma}_f^{img}$ be the distribution over joint paths induced by imaginary play (Algorithm 5) and communication availability dictated by $f$. Then,*

$$KL(\mathbf{\Gamma}^{full} || \mathbf{\Gamma}_0^{img}) \geq KL(\mathbf{\Gamma}^{full} || \mathbf{\Gamma}_f^{img}).$$

We remark that Algorithms 5-7 are agnostic to when future communication failures happen. The lemmas do not assume a priori knowledge of the sequence of communication availability.

**On the reach-avoid probability under communication loss** We use the above results on the KL divergence between distributions of paths to derive bounds on the reach-avoid probability achieved by a particular joint policy under communication loss.

Let $\boldsymbol{v}^{full}$ be the reach-avoid probability induced by a joint policy with full communication, $\boldsymbol{v}^{img}$ be the reach-avoid probability of the same policy under imaginary play (Algorithm 5), and $\boldsymbol{v}^{int}$ be the reach-avoid probability under intermittent communication (Algorithm 7). Also, let $\boldsymbol{\mathcal{S}}_{\mathcal{D}}$ be the states from which the probability of reaching $\boldsymbol{\mathcal{S}}_{\mathcal{T}}$ is 0 under the joint policy. Define $len(\boldsymbol{\xi} = \boldsymbol{s}_0\boldsymbol{a}_0\ldots) = \min\{t+1 | \boldsymbol{s}_t \in \boldsymbol{\mathcal{S}}_{\mathcal{T}} \cup \boldsymbol{\mathcal{S}}_{\mathcal{D}}\}$ and $l^{full} = \mathbb{E}[len(\boldsymbol{\xi})|\boldsymbol{\xi} \sim \boldsymbol{\Gamma}^{full}]$.

Theorem 3.1 shows that the reach-avoid probability of a joint policy under imaginary play is lower-bounded by a function of the policy's reach-avoid probability with full communication and the value of $C_{\pi_{joint}}$, even when the communication availability depends on the agents' histories.

**Theorem 3.1.** *Let $f : (\boldsymbol{\mathcal{S}} \times \boldsymbol{\mathcal{A}})^* \to \{0, 1\}$ be an arbitrary function that determines the communication availability based on the history of the agents such that $\lambda_t = f(\boldsymbol{s}_0\boldsymbol{a}_0 \ldots \boldsymbol{s}_{t-1}\boldsymbol{a}_{t-1})$. For this system,*

$$\boldsymbol{v}^{img} \geq \boldsymbol{v}^{full} - \sqrt{1 - \exp(-C_{\pi_{joint}})}.$$

We now consider the setting in which the team's communication fails at some random time $t_{loss} \geq 0$ and does not recover thereafter. When $t_{loss}$ follows a geometric distribution, we derive a stronger bound that relates the probability of communication failure at each time step to the reach-avoid probability under imaginary play.

**Theorem 3.2.** *Consider a communication system that fails with probability $p$ at any communication step and never recovers, i.e., $\Pr(t_{loss} = t) = (1-p)^t p$ in Algorithm 5. For this system,*

$$\boldsymbol{v}^{img} \geq \max\left(\boldsymbol{v}^{full} - \sqrt{1 - \exp(-C_{\pi_{joint}})}, \boldsymbol{v}^{full}(1-p)^{\frac{l^{full}}{\boldsymbol{v}^{full}}}\right).$$

Finally, we consider intermittent communication and partition communication groups. Under Algorithm 7, coordinating with some other agents whenever possible does not degrade the performance even when the whole team cannot coordinate together, i.e., $\mathcal{P}_t \neq \{[N]\}$. Furthermore, when communication availability is intermittent, for example, in a Bernoulli process, the reach-avoid probability under intermittent communication is directly lower-bounded by a function of the communication dropout rate $q$. However, we note that the following result does not require the communication availability at different timesteps to be independent. As an example, consider a setting in which every loss in communication persists for some minimum number of consecutive timesteps. In this case, Theorem 3.3 is still applicable.

**Theorem 3.3.** *Consider a random sequence $\Lambda = \mathcal{P}_0, \mathcal{P}_1, \ldots$ of communication partitions that satisfies Assumptions 3.1 and 3.2. For this system,*

$$\boldsymbol{v}^{int} \geq \boldsymbol{v}^{full} - \sqrt{1 - \exp(-qC_{\pi_{joint}})}$$

*where $q = \max_t \Pr(\mathcal{P}_t \neq \{[N]\})$ is the maximum dropout rate per time step.*

We remark that the lower bound in Theorem 3.3 provides a means to select communication resources that are sufficient to achieve a particular performance while using noisy communication channels. In detail, consider a noisy communication channel on which the team must communicate. The code rate (Cover and Thomas, 2012) can be adjusted according to the desired value of $q$, which in turn determines the value of the lower bound on $\boldsymbol{v}^{int}$.

The lower bounds in Theorems 3.1, 3.2, and 3.3 show that the reach-avoid probability of a joint policy under communication loss depends on the total correlation of the joint policy, the reach-avoid probability achieved with full communication, the communication dropout rate, and the expected path length under the joint policy. When the total correlation is 0, the reach-avoid probability under communication loss is the same as the reach-avoid probability with full communication. As the total correlation of the joint policy increases, the values of the lower bounds decrease. During

intermittent communication, if the dropout rate is 0, then the reach-avoid probability of the joint policy executed using imaginary play (Algorithm 5) or intermittent communication (Algorithms 6 and 7) is the same as when the policy is executed with full communication. When the communication dropout rates are 1, the reach-avoid probability under communication loss depends on the value of the total correlation. We note that the bounds are tight when either the communication dropout rate or the total correlation is 0.

**An example for the worst-case highest achievable performance under communication loss** In this section, we discuss the worst-case highest achievable team performance under communication loss. In particular, we give an example where the best achievable performance under communication loss under any mechanism is bounded by a constant factor of the lower bound given in Theorem 3.3.

Proposition 3.1 shows that there exists a family of MDPs and reachability specifications where the optimal reachability probability of any possible decentralized policy execution mechanism diminishes exponentially with the increasing number of agents. Furthermore, the reachability probability of the optimal minimum-dependency policy $\pi^{MD}$ (the joint policy that maximizes the lower bound given in Theorem 3.3) under Algorithm 7 is optimal up to a constant factor for this family of MDPs and reachability specifications.

**Proposition 3.1.** *Assume that*

- *The MDP of agent $i$ is given by $\mathcal{M}^i = \mathcal{M}(i, m)$ as shown in Figure 3.3,*
- *The target set of joint states is given by $\boldsymbol{\mathcal{S}}_{\mathcal{T}} = \{\boldsymbol{s} = (s^1, \ldots, s^N) | \forall 1 \leq i < N, \exists j, s^i = z^i_{N,j,k} \wedge s^{i+1} = z^{i+1}_{N,k,l}\}$, i.e., the action index of Agent $i+1$ at time step $2i$ must match the successor state index of Agent $i$'s uniformly random transition at time step $2i-1$,*

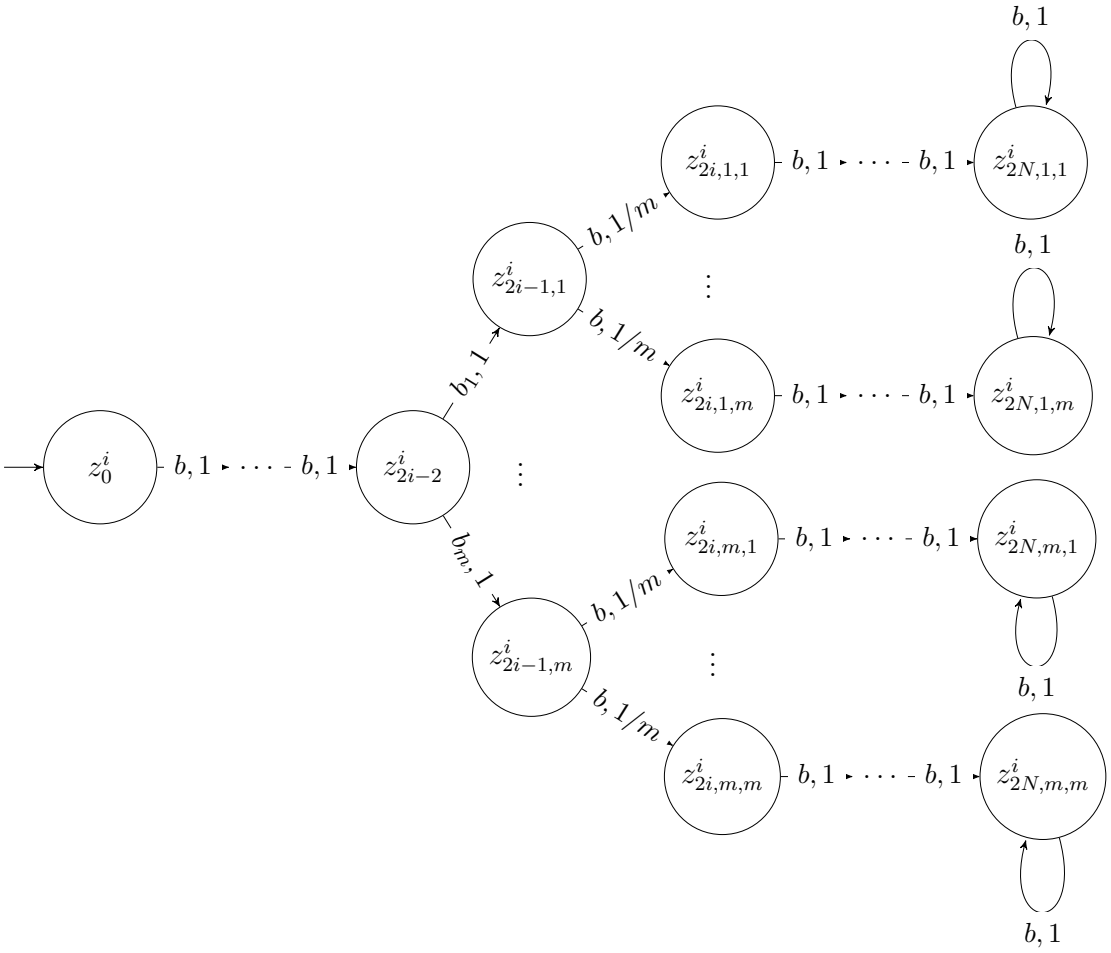



Figure 3.3: MDP $\mathcal{M}(i, m)$ of Agent $i$ for the upper bound on the worst-case reachability probability. For the first $2i-2$ time steps, the agent has a single action $b$ that transitions from $z_t^i$ to $z_{t+1}^i$ with probability 1. At time step $2i-1$, the agent has $m$ actions $b_1, \ldots, b_m$ that respectively transitions from $z_{2i-2}^i$ to $z_{2i-1,1}^i, \ldots, z_{2i-1,m}^i$ with probability 1. At time step $2i$ and state $z_{2i-1,j}^i$, the agent a single action $b$ that transitions to $z_{2i,j,1}^i, \ldots, z_{2i,j,m}^i$ with uniformly random probabilities. For $2N-2i$ time steps, the agent has a single action $b$ that transitions from $z_{t,j,k}^i$ to $z_{t+1,j,k}^i$ with probability 1. For time steps $2N, 2N+1, \ldots$, the agent has a single action $b$ that transitions from $z_{2N,j,k}^i$ to $z_{2N,j,k}^i$ with probability 1.

- *The communication availability is a Bernouilli(q) process, i.e., $\mathcal{P}_t$ are i.i.d. such that $\mathcal{P}_t = \{[N]\}$ with probability $1-q$ and $\mathcal{P}_t = \{\{1\}, \ldots, \{N\}\}$ with probability $q$.*

*Let $\pi_{MD} = \arg\max_\pi \boldsymbol{v}^{full} - \sqrt{1-\exp(-qC_\pi)}$ and $\boldsymbol{v}^{int}$ be the reachability probability of $\pi_{MD}$ under Algorithm 7 and a random communication availability. Let $\mathcal{D}$ be the optimal policy execution mechanism in terms of maximizing the reachability probability, given the MDPs and the communication availability distribution, and denote $\boldsymbol{v}^{\mathcal{D}}$ be the reachability probability of $\mathcal{D}$ under a random communication availability. We have*

$$\boldsymbol{v}^{int} \leq \boldsymbol{v}^{\mathcal{D}} \leq (1 + q/m - q)^{N-1} \leq 2\boldsymbol{v}^{int}$$

*for any $m \geq 1$, $N \geq 1$, and $q \in [0, 1]$.*

## 3.7 Joint Policy Synthesis

In this section, we discuss the synthesis of minimum-dependency joint policies $\pi_{MD}$ that are robust to communication failures.

**Entropy of paths for a single agent** Given the multiagent MDP, a stationary joint policy $\pi_{joint}$ induces a Markov chain. This Markov chain generates a Markov process $\boldsymbol{X}$, which is the joint path of the agents. The entropy $H(\boldsymbol{X})$ of a Markov process has a closed form expression in terms of the occupancy measure $x_{\boldsymbol{s},\boldsymbol{a}}$ of the joint state-action pairs $(\boldsymbol{s}, \boldsymbol{a})$ (Savas et al., 2019). The path of a single agent, on the other hand, follows a hidden Markov model where $\boldsymbol{X}$ is the underlying process and $X^i$ is the observed process. However, the entropy $H(X^i)$ of a process that follows a hidden Markov model does not admit a closed-form expression.

Let $x_{s^i,a^i}$ be the occupancy measure for the state-action pair $(s^i, a^i) \in \mathcal{S}^i \times \mathcal{A}^i$ under the joint policy $\pi_{joint}$. Consider a Markov process $\bar{X}^i$ that induces the same occupancy measure $x_{s^i,a^i}$ as the joint policy. The entropy $H(\bar{X}^i)$ of the Markov

process is greater than or equal to the entropy $H(X^i)$ of the original process (Savas et al., 2019). Since $H(X^i)$ does not admit a closed form expression, we instead upper bound $C_{\pi_{joint}}$ using $H(\bar{X}^i)$. Formally, we have

$$\bar{C}_{\pi_{joint}} = \left[\sum_{i=1}^{N} H(\bar{X}^i)\right] - H(\boldsymbol{X}) \geq C_{\pi_{joint}} = \left[\sum_{i=1}^{N} H(X^i)\right] - H(\boldsymbol{X}).$$

**The policy synthesis optimization problem** To optimize the reach-avoid probability under communication loss, we would like to maximize the lower bound given in Theorem 3.2. However, due to the complex nature of this lower bound, we propose to instead use the following optimization problem as a proxy to the original problem:

$$\sup_{\pi_{joint}} \boldsymbol{v}^{full} - \delta l^{full} - \beta \bar{C}_{\pi_{joint}} \tag{3.1}$$

where $\delta > 0$ and $\beta > 0$ are constants.

We now represent (3.1) in terms of occupancy measure variables and construct the optimization problem for synthesis. We first preprocess $\mathfrak{M}$ to ensure that $\bar{C}_{\pi_{joint}}$ is well-defined. Define $\mathfrak{S}_{\mathcal{D}} = \{\boldsymbol{s} | \max_{\pi_{joint}} \boldsymbol{v}_{joint} = 0 \text{ when the path begins at } \boldsymbol{s}\}$, the set of all states from which the reach-avoid task is violated with probability 1. We note that $\mathfrak{S}_{\mathcal{D}} \supseteq \mathfrak{S}_{\mathcal{A}}$, the absorbing states. For synthesis, we add an absorbing end state $\boldsymbol{s}_\epsilon = (s_\epsilon^1, \ldots, s_\epsilon^N)$ and a joint action $\epsilon = (\epsilon^1, \ldots, \epsilon^N)$ to $\mathfrak{M}$, which represent the end of the process in terms of the reach-avoid objective. Every $\boldsymbol{s} \in \mathfrak{S}_{\mathcal{T}} \cup \mathfrak{S}_{\mathcal{D}}$ has a single action $\epsilon$, and $\mathcal{T}(\boldsymbol{s}, \epsilon, \boldsymbol{s}_\epsilon) = 1$ for all $\boldsymbol{s} \in \mathfrak{S}_{\mathcal{T}} \cup \mathfrak{S}_{\mathcal{D}}$, i.e., the states in $\mathfrak{S}_{\mathcal{T}} \cup \mathfrak{S}_{\mathcal{D}}$ deterministically transitions to $\boldsymbol{s}_\epsilon$. For synthesis, we assume that every $\boldsymbol{s} \in \mathfrak{S} \setminus (\mathfrak{S}_{\mathcal{T}} \cup \mathfrak{S}_{\mathcal{D}})$ has a finite occupancy measure, i.e., $\sum_{\boldsymbol{a} \in \mathcal{A}} x(\boldsymbol{s}, \boldsymbol{a}) \leq K$ for some $K \geq 0$.

In the previous sections, we assumed that the joint policy is stationary. The following proposition shows that stationary policies suffice to maximize (3.1) after the preprocessing step.

**Proposition 3.2.** *There exists a stationary joint policy that is a solution to* (3.1).

Given that the stationary policies suffice, we can rewrite (3.1) as an optimization problem in terms of the occupancy measure variables $x_{\boldsymbol{s},\boldsymbol{a}}$. The constraints of this optimization problem are as follows. State $\boldsymbol{s}_\epsilon$ has an occupancy measure of zero, i.e. $x_{\boldsymbol{s}_\epsilon,\boldsymbol{a}} = 0$ for all $\boldsymbol{a} \in \boldsymbol{\mathcal{A}} \cup \{\epsilon\}$. The other states have nonnegative occupancy measure, i.e., $x_{\boldsymbol{s},\boldsymbol{a}} \geq 0$ for all $\boldsymbol{s} \in \boldsymbol{\mathcal{S}}, \boldsymbol{a} \in \boldsymbol{\mathcal{A}} \cup \{\epsilon\}$. The occupancy measure satisfy the flow equations $\sum_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}\cup\{\epsilon\}} x_{\boldsymbol{s},\boldsymbol{a}} = \sum_{\substack{\boldsymbol{y}\in\boldsymbol{\mathcal{S}} \\ \boldsymbol{b}\in\boldsymbol{\mathcal{A}}\cup\{\epsilon\}}} x_{\boldsymbol{y},\boldsymbol{b}} \boldsymbol{\mathcal{T}}(\boldsymbol{y},\boldsymbol{b},\boldsymbol{s}) + \mathbb{1}_{\{\boldsymbol{s_0}=\boldsymbol{s}\}}$ for all $\boldsymbol{s} \in \boldsymbol{\mathcal{S}}$. The objective function is

$$\max_{x} \quad \boldsymbol{v}^{full} - \delta l^{full} - \beta \left( \sum_{i=1}^{N} H(\bar{X}^i) - H(\boldsymbol{X}) \right).$$

The reach-avoid probability $\boldsymbol{v}^{full}$ can be expressed as

$$\boldsymbol{v}^{full} = \sum_{\boldsymbol{s}\in\boldsymbol{\mathcal{S}}\setminus(\boldsymbol{\mathcal{S}}_{\mathcal{D}}\cup\boldsymbol{\mathcal{S}}_{\mathcal{T}})} \sum_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}} \sum_{\boldsymbol{y}\in\boldsymbol{\mathcal{S}}_{\mathcal{T}}} x_{\boldsymbol{s},\boldsymbol{a}} \boldsymbol{\mathcal{T}}(\boldsymbol{s},\boldsymbol{a},\boldsymbol{y}).$$

The expected path length is the expected time spent in the transient states, i.e., $l^{full} = \sum_{\boldsymbol{s}\in\boldsymbol{\mathcal{S}}} \sum_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}\cup\{\epsilon\}} x_{\boldsymbol{s},\boldsymbol{a}}$. The entropy $H(\boldsymbol{X})$ (Savas et al., 2019) of the joint state-action process until reaching state $\boldsymbol{s}_\epsilon$ is

$$\sum_{\substack{\boldsymbol{s}\in\boldsymbol{\mathcal{S}} \\ \boldsymbol{a}\in\boldsymbol{\mathcal{A}}}} x_{\boldsymbol{s},\boldsymbol{a}} \log \left( \frac{\sum_{\boldsymbol{b}\in\boldsymbol{\mathcal{A}}} x_{\boldsymbol{s},\boldsymbol{b}}}{x_{\boldsymbol{s},\boldsymbol{a}}} \right) + \sum_{\substack{\boldsymbol{s}\in\boldsymbol{\mathcal{S}} \\ \boldsymbol{a}\in\boldsymbol{\mathcal{A}}}} x_{\boldsymbol{s},\boldsymbol{a}} \sum_{\boldsymbol{y}\in\boldsymbol{\mathcal{S}}} \boldsymbol{\mathcal{T}}(\boldsymbol{s},\boldsymbol{a},\boldsymbol{y}) \log \left( \frac{1}{\boldsymbol{\mathcal{T}}(\boldsymbol{s},\boldsymbol{a},\boldsymbol{y})} \right).$$

The entropy $H(\bar{X}^i)$ (Savas et al., 2019) of the state-action process $\bar{X}^i$ until reaching state $\boldsymbol{s}_\epsilon$ is

$$\sum_{\substack{s^i\in\mathcal{S}^i \\ a^i\in\mathcal{A}^i\cup\{\epsilon^i\}}} x_{s^i,a^i} \log \left( \frac{\sum_{b^i\in\mathcal{A}^i} x_{s^i,b^i}}{x_{s^i,a^i}} \right) + \sum_{\substack{s^i\in\mathcal{S}^i \\ a^i\in\mathcal{A}^i\cup\{\epsilon^i\}}} x_{s^i,a^i} \sum_{y^i\in\mathcal{S}^i\cup\{s^i_\epsilon\}} \mathcal{T}^i(s^i,a^i,y^i) \log \left( \frac{1}{\mathcal{T}^i(s^i,a^i,y^i)} \right).$$

The objective function of the optimization problem consists of convex, concave, and linear functions of the occupancy measure. $\boldsymbol{v}^{full}$ and $-\delta l^{full}$ are linear functions of the occupancy measure. $\beta H(\boldsymbol{X})$ is a concave function of occupancy measures, and $-\beta \sum_{i=1}^{N} H(\bar{X}^i)$ is a convex function of occupancy measures. Furthermore, the

problem's constraints are linear. We use the concave-convex procedure (Lanckriet and Sriperumbudur, 2009; Yuille and Rangarajan, 2001) to solve for a local optimum.

After solving for the optimal values $x^*_{\boldsymbol{s},\boldsymbol{a}}$ of the occupancy measure variables, we define the minimum-dependency joint policy as $\pi_{MD}(\boldsymbol{s},\boldsymbol{a}) = x^*(\boldsymbol{s},\boldsymbol{a}) / \sum_{\boldsymbol{b}\in\boldsymbol{\mathcal{A}}} x^*(\boldsymbol{s},\boldsymbol{b})$ for all $\boldsymbol{s} \in \boldsymbol{\mathcal{S}} \setminus (\boldsymbol{\mathcal{S}}_{\mathcal{T}} \cup \boldsymbol{\mathcal{S}}_{\mathcal{D}}), \boldsymbol{a} \in \boldsymbol{\mathcal{A}}$ such that $\sum_{\boldsymbol{b}\in\boldsymbol{\mathcal{A}}} x^*(\boldsymbol{s},\boldsymbol{b}) > 0$, and $\pi_{MD}(\boldsymbol{s},\boldsymbol{a}) = 1/|\boldsymbol{\mathcal{A}}|$ otherwise (Puterman, 2014). We note that $\pi_{MD}$ is stationary in the joint state space $\boldsymbol{\mathcal{S}}$.

## 3.8 Numerical Examples

In all of the examples, we compare the results of the minimum-dependency policy $\pi_{MD}$, synthesized by the algorithm presented in §3.7, to a baseline policy $\pi_{base}$ which does not take potential communication losses into account. The baseline policy maximizes the probability that the team will complete its task while assuming that communication will always be available. For specific implementation details surrounding the synthesis of the minimum dependency policy, we refer the reader to (Karabag et al., 2022a).

### 3.8.1 The Two-Agent Navigation Experiment

We begin by applying the proposed policy synthesis algorithm to the two-agent navigation example illustrated in Figure 3.2. The setup and objective of this task are as described in §3.3. We implement the common environment of the agents by discretizing it into a $5 \times 5$ grid of cells, each of which corresponds to a possible position of one of the agents. At any given timestep, each agent takes one of five separate actions: move left, move right, move up, move down, or remain in place. Each agent slips with probability 0.05 every time it takes an action, resulting in the agent moving instead to another one of its valid neighboring states. The resulting optimization problem has $15,625$ variables and $16,087$ constraints.

In §3.8.1.1, we consider the case in which communication is lost entirely. In

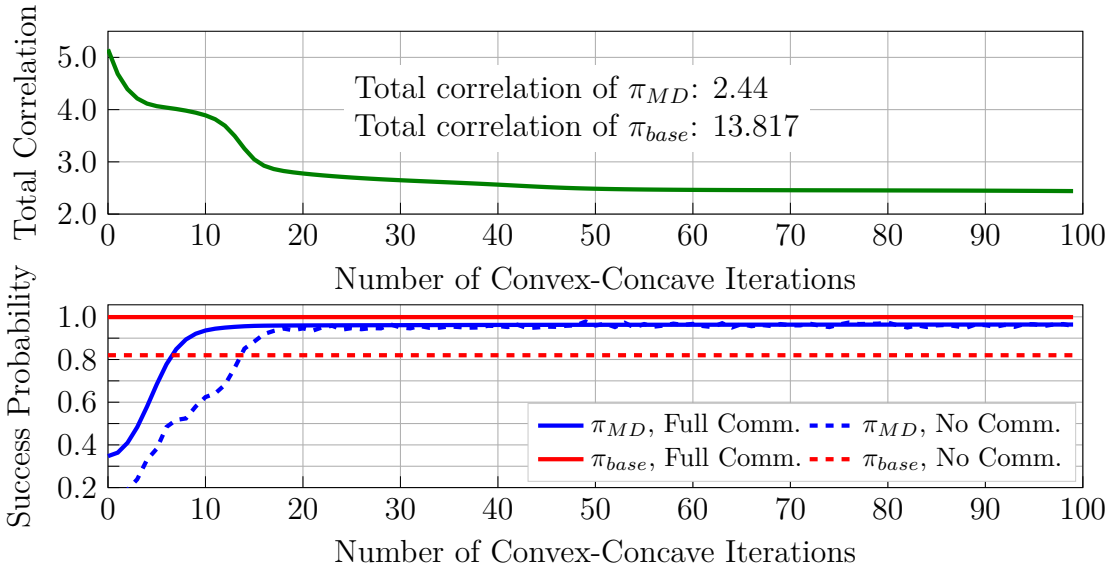



Figure 3.4: (Top) Total correlation value of the minimum-dependency policy $\pi_{MD}$ as a function of the number of elapsed iterations of the convex-concave optimization procedure. (Bottom) Probability of task success for $\pi_{MD}$ resulting from both imaginary play execution (no communication) and centralized execution (full communication). To estimate the probability of task success, we perform rollouts of the joint policy and compute the empirical rate at which the team accomplishes its objective.

§3.8.1.2 we present results for varying severities of communication loss. In both of these experimental scenarios, the values of the coefficients $\delta$ and $\beta$ in the objective of the policy synthesis problem are set to 0.01 and 0.4 respectively. These values were selected to strike a balance between the optimization objective's three competing terms.

#### 3.8.1.1 Fully Imaginary Play

Figure 3.4 compares the results of the minimum-dependency policy $\pi_{MD}$ and the baseline policy $\pi_{base}$ in two scenarios: when communication is either fully available, or when it is never available.

We observe from the top plot that our proposed policy synthesis algorithm is effective at reducing the total correlation of the induced stochastic state-action process. The total correlation value of $\pi_{MD}$ is three orders of magnitude smaller than that of $\pi_{base}$.

The bottom plot shows the strong performance of $\pi_{MD}$ when no communication is available between the agents. In particular, we observe that $\pi_{MD}$ achieves a probability of task success of 0.97, regardless of whether the agents are able to communicate. That is, by minimizing the total correlation of the policy, $\pi_{MD}$ ensures that the agents may successfully execute the policy without communicating during execution. Conversely, while $\pi_{base}$ achieves a 0.99 probability of task success when communication is available, this value falls to 0.82 if the agents lose the ability to communicate. This experiment empirically demonstrates the intuition of Theorem 3.2.

In addition to the quantitative results illustrated by Figure 3.4, we observe an interesting qualitative change in behavior between $\pi_{base}$ and $\pi_{MD}$. Figure 3.5 illustrates heatmaps of the occupancy measures of the individual agents under the synthesized joint policy $\pi_{MD}$ and the baseline policy $\pi_{base}$. Specifically, each heatmap visualizes the values of the variables $x_{s^i}$ for some agent $i$ under one of these joint

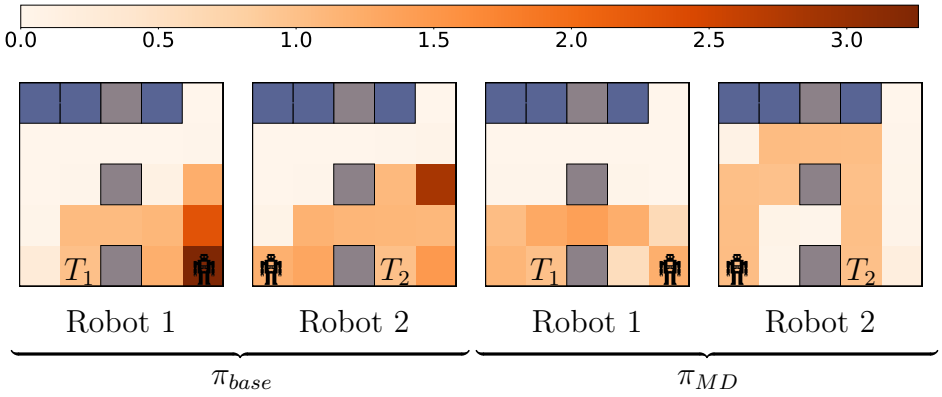



Figure 3.5: Heatmap illustrations of the occupancy measures $x_{s^i}$ of the individual robots' states under the baseline $\pi_{base}$ and minimum dependency $\pi_{MD}$ joint policies. The robot icons and the symbols $T_i$ represent the initial and goal states of the robots, respectively. Unlike the baseline policy $\pi_{base}$, the minimum dependency policy $\pi_{MD}$ assigns each robot a separate valley to navigate, reducing the probability of a crash in the event that communication is lost.

policies. These occupancy measures for the individual agents are defined as $x_{s^i} = \sum_{a^i \in \mathcal{A}} x_{s^i,a^i}$. Intuitively, we may think of the value of $x_{s^i}$ as being a measure of the frequency at which agent $i$ visits local state $s^i$ if the joint policy is repeatedly followed from the initial state.

We observe from Figure 3.5 that $\pi_{base}$ results in both of the robots navigating through the lower valley in order to arrive at their targets. This route relies heavily on teammate coordination; the robots must communicate at each timestep in order to safely take turns passing through the valley without colliding. By contrast, $\pi_{MD}$ results in robot $\mathsf{a}_2$ navigating through the top valley while $\mathsf{a}_1$ takes the bottom valley. Intuitively, by navigating through separate valleys, this team behavior is much less likely to result in collisions even if the robots don't share their locations with each other. As a result, teammate coordination is much less important for the successful execution of joint policy $\pi_{MD}$, than it is for the execution of $\pi_{base}$.

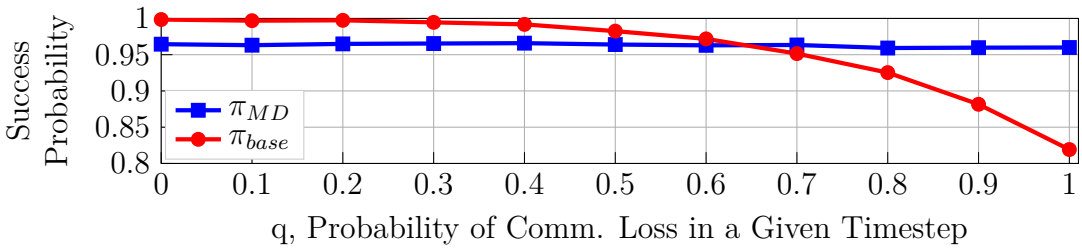



Figure 3.6: Success probability of intermittent communication for different values of $q$, which represents the probability of communication unavailability during any given timestep. When $q = 0$ communication is available at every timestep, and when $q = 1$ communication is never available.

#### 3.8.1.2 Intermittent Communication

While the previous discussion focused on the empirical performance of $\pi_{MD}$ in the setting where the agents cannot communicate at all, we now examine the setting in which random intermittent communication is available. More specifically, we assume that at each timestep communication fails with probability $q$, independently of whether or not communication is available during the other timesteps. In this setting, the agents execute the joint policy according to Algorithm 6. That is, if communication is available at a given timestep, all agents collectively share their local states and decide on a joint action. Conversely, when communication is not available, the agents execute the policy using imaginary play.

Figure 3.6 plots the team's probability of task success when they execute either $\pi_{MD}$ or $\pi_{base}$ using Algorithm 6, as a function of the probability of communication failure $q$. We observe that the probability of task success of the baseline policy $\pi_{base}$ is very high when $q = 0$, however, it begins to significantly decrease as $q$ increases beyond 0.4. Conversely, the proposed minimum-dependency policy $\pi_{MD}$ does not suffer such a drop in performance; as $q$ increases and communication becomes more sparse the task success probability of policy $\pi_{MD}$ remains constant.

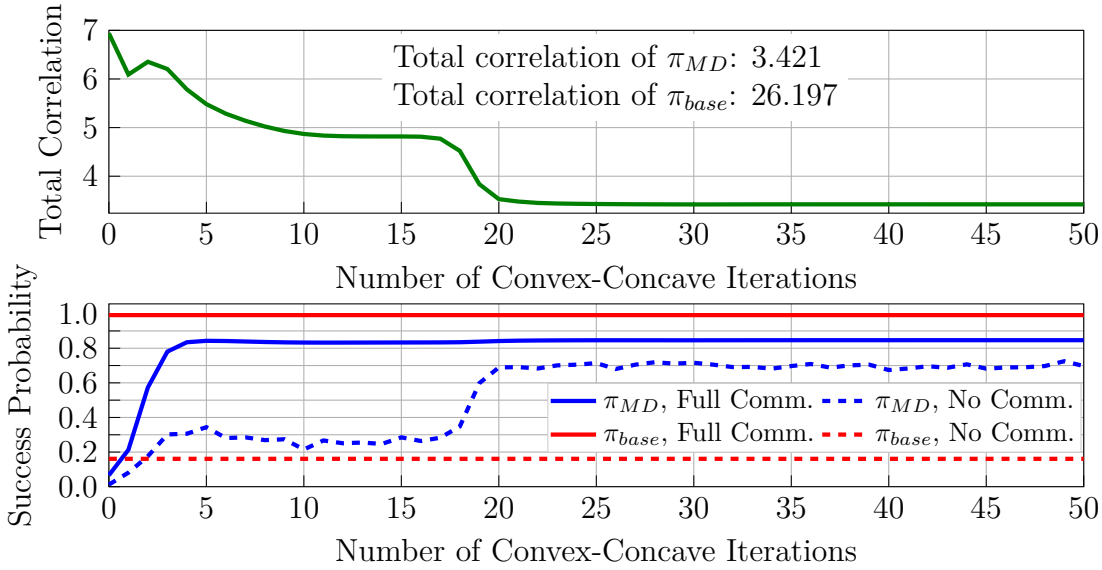



Figure 3.7: Total correlation and success probability values of the minimum-dependency policy $\pi_{MD}$ during policy synthesis on the three-agent navigation experiment. (Top) Total correlation value of the policy as a function of the number of elapsed iterations of the convex-concave optimization procedure. (Bottom) Probability of task success.

### 3.8.2 A Three-Agent Collision Avoidance Experiment

We now present a three-agent experiment, which demonstrates the ability of the proposed approach to generalize to multiagent systems including more than two agents. Robots $R_1$, $R_2$, and $R_3$ start in opposing corners of a $3 \times 3$ gridworld, as illustrated in Figure 3.8. Each robot must navigate to its respective target location $T_1$, $T_2$, or $T_3$, which are located in the corner opposite to the robot's initial position. Furthermore, while navigating to their goals, the robots must avoid collisions with each other. The actions of the agents and the slip probabilities associated with these actions are the same as for the above two-agent navigation example. In this three-agent experiment, we set the values of $\delta$ and $\beta$ in the policy synthesis problem to 0.01 and 0.1, respectively.

Figure 3.7 compares the total correlation values and the success probabilities of the synthesised minimum-dependency policy $\pi_{MD}$ and the baseline policy $\pi_{base}$. We

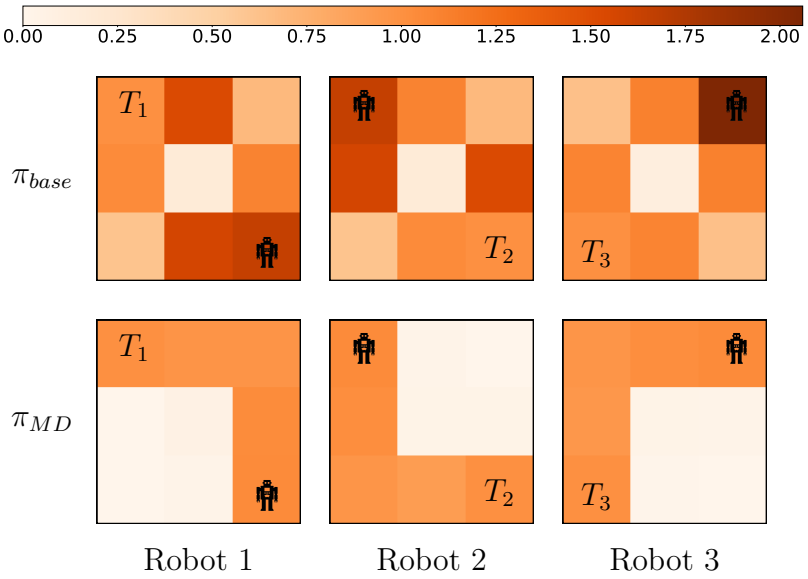



Figure 3.8: Heatmap illustrations of the occupancy measures $x_{s^i}$ of the individual robots. The minimum dependency policy $\pi_{MD}$ results in each of the robots travelling counterclockwise along the edge of the environment, regardless of the current states of their teammates.

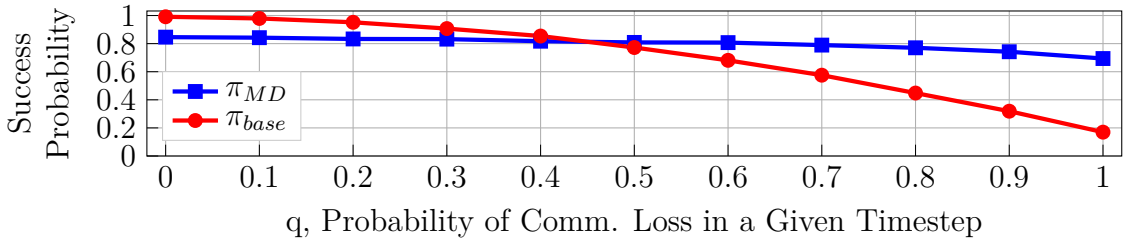



Figure 3.9: Success probability of intermittent communication for different values of $q$ on the three-agent navigation experiment.

again observe that as the total correlation of the process induced by $\pi_{MD}$ decreases during policy synthesis, the team's probability of success in the no-communication scenario increases. In particular, when communication is not available, $\pi_{MD}$ has a success probability of 70 percent, while the success probability of $\pi_{base}$ drops to 17 percent.

Figure 3.8 illustrates the occupancy measures of the individual agents under $\pi_{MD}$ and $\pi_{base}$. The minimum dependency policy $\pi_{MD}$ results in each of the robots traveling counterclockwise along the edge of the environment, regardless of the current states of their teammates. Conversely, the baseline policy $\pi_{base}$ requires that the robots react to their teammates' current states; each robot moves around the edge of the environment in a fashion that is directly dependent on the locations of the other robots. Joint policy $\pi_{base}$ thus effectively ensures that collisions are avoided when communication is available, however, its performance drops significantly as communication between the agents becomes degraded. By contrast, $\pi_{MD}$ results in consistently performant behavior. We observe this point quantitatively in Figure 3.9.

## 3.9 Proofs for Technical Results

We first define some notation to be used in the notation and provide different expressions of total correlation.

**Notation** Under the joint policy $\pi_{joint}$ with full communication, let $\boldsymbol{S}_t$ be a random variable denoting the joint state of the agents at time $t$, $\boldsymbol{A}_t$ be a random variable denoting the joint action of the agents at time $t$, $S_t^i$ be a random variable denoting the state of Agent $i$ at time $t$, and $A_t^i$ be a random variable denoting the action of Agent $i$ at time $t$.

We use $\mu^{full}$ to denote the probability measure over the (finite or infinite) state-action process under the joint policy with full communication. $\mu_{t_{loss}}^{img}$ denotes the probability measure over the (finite or infinite) state-action process under the

imaginary play (under Algorithm 5) where the first communication loss happens at time $t_{loss}$. $\mu_f^{img}$ denotes the probability measure over the (finite or infinite) state-action process under the imaginary play (under Algorithm 5) where $f : (\boldsymbol{\mathcal{S}} \times \boldsymbol{\mathcal{A}})^* \to \{0, 1\}$ determines the communication availability based on the team's joint history. $\mu_\Lambda^{int}$ denotes the probability measure over the (finite or infinite) state-action process under the intermittent communication (under Algorithm 6) with a sequence $\Lambda$ of communication availability.

The Kleene star applied to a set $V$ of symbols is the set $V^* = \bigcup_{i\geq 0} V^i$ of all finite-length words where $V^0 = \{\varepsilon\}$ and $\varepsilon$ is the empty string. The set of all infinite-length words is denoted by $V^\omega$.

**Different expressions of total correlation** The total correlation (Watanabe, 1960) of joint policy $\pi_{joint}$ is

$$C_{\pi_{joint}} = KL(\boldsymbol{\Gamma}^{full} || \boldsymbol{\Gamma}_0^{img}) = \left[\sum_{i=1}^{N} H(X^i)\right] - H(\boldsymbol{X}).$$

By the chain rule of entropy (Cover and Thomas, 2012) and the fact that $\boldsymbol{s_0}$ is a common knowledge, we have

$$\begin{aligned} C_{\pi_{joint}} &= \left[\sum_{i=1}^{N} H(S_0^i A_0^i | \boldsymbol{S}_0)\right] - H(\boldsymbol{S}_0 \boldsymbol{A}_0 | \boldsymbol{S}_0) \\ &+ \sum_{t=0}^{\infty} \left[\left[\sum_{i=1}^{N} H(S_t^i A_t^i | \boldsymbol{S}_0 A_0^i \dots S_{t-1}^i A_{t-1}^i)\right] - H(\boldsymbol{S}_t \boldsymbol{A}_t | \boldsymbol{S}_0 \boldsymbol{A}_0 \dots \boldsymbol{S}_{t-1} \boldsymbol{A}_{t-1})\right]. \end{aligned}$$

We note that for all $t = 1, 2, \dots$

$$\left[\sum_{i=1}^{N} H(S_t^i A_t^i | \boldsymbol{S}_0 A_0^i \dots S_{t-1}^i A_{t-1}^i)\right] - H(\boldsymbol{S}_t \boldsymbol{A}_t | \boldsymbol{S}_0 \boldsymbol{A}_0 \dots \boldsymbol{S}_{t-1} \boldsymbol{A}_{t-1}) \tag{3.2a}$$

$$\geq \left[\sum_{i=1}^{N} H(S_t^i A_t^i | \boldsymbol{S}_0 \boldsymbol{A}_0 \dots \boldsymbol{S}_{t-1} \boldsymbol{A}_{t-1})\right] - H(\boldsymbol{S}_t \boldsymbol{A}_t | \boldsymbol{S}_0 \boldsymbol{A}_0 \dots \boldsymbol{S}_{t-1} \boldsymbol{A}_{t-1}) \tag{3.2b}$$

$$\geq 0 \tag{3.2c}$$

where (3.2b) is because conditioning (extra information) reduces entropy and (3.2c) is due to the subadditivity of entropy. Similarly,

$$\left[\sum_{i=1}^{N} H(S_0^i A_0^i | \boldsymbol{S}_0)\right] - H(\boldsymbol{S}_0 \boldsymbol{A}_0 | \boldsymbol{S}_0) \leq 0 \tag{3.3}$$

*Proof of Lemma 3.1.* We consider three cases of $t_{loss}$ to prove the lemma: $t_{loss} = 1, 2, \ldots$, $t_{loss} = 0$, and $t_{loss} = \infty$.

If $t_{loss} = 0$, the statement trivially holds since $\boldsymbol{\Gamma}_0^{img} = \boldsymbol{\Gamma}_{t_{loss}}^{img}$. In this case, $KL(\boldsymbol{\Gamma}^{full} || \boldsymbol{\Gamma}_0^{img}) = KL(\boldsymbol{\Gamma}^{full} || \boldsymbol{\Gamma}_{t_{loss}}^{img})$.

If $t_{loss} = \infty$, the statement holds since there is always communication and $\boldsymbol{\Gamma}^{full} = \boldsymbol{\Gamma}_{\infty}^{img}$. In this case, $KL(\boldsymbol{\Gamma}^{full} || \boldsymbol{\Gamma}_{\infty}^{img}) = 0 \leq KL(\boldsymbol{\Gamma}^{full} || \boldsymbol{\Gamma}_0^{img})$.

Let $t_{loss} \geq 1$ be an arbitrary integer. We have

$$\begin{aligned}
&KL(\boldsymbol{\Gamma}^{full} || \boldsymbol{\Gamma}_{t_{loss}}^{img}) \\
&= \sum_{\boldsymbol{s}_0 \boldsymbol{a}_0 \boldsymbol{s}_1 \boldsymbol{a}_1 \ldots \in (\mathcal{S} \times \mathcal{A})^{\omega}} \mu^{full}(\boldsymbol{s}_0 \boldsymbol{a}_0 \boldsymbol{s}_1 \boldsymbol{a}_1 \ldots) \log \left( \frac{\mu^{full}(\boldsymbol{s}_0 \boldsymbol{a}_0 \boldsymbol{s}_1 \boldsymbol{a}_1 \ldots)}{\mu_{t_{loss}}^{img}(\boldsymbol{s}_0 \boldsymbol{a}_0 \boldsymbol{s}_1 \boldsymbol{a}_1 \ldots)} \right) && \text{(3.4a)} \\
&= \sum_{w \in (\mathcal{S} \times \mathcal{A})^{t_{loss}}} \mu^{full}(w) \log \left( \frac{\mu^{full}(w)}{\mu_{t_{loss}}^{img}(w)} \right) \\
&\quad + \sum_{w \in (\mathcal{S} \times \mathcal{A})^{t_{loss}}} \sum_{w' \ldots \in (\mathcal{S} \times \mathcal{A})^{\omega}} \mu^{full}(ww') \log \left( \frac{\mu^{full}(w'|w)}{\mu_{t_{loss}}^{img}(w'|w)} \right) && \text{(3.4b)} \\
&= \sum_{w \in (\mathcal{S} \times \mathcal{A})^{t_{loss}}} \sum_{w' \in (\mathcal{S} \times \mathcal{A})^{\omega}} \mu^{full}(ww') \log \left( \frac{\mu^{full}(w'|w)}{\mu_{t_{loss}}^{img}(w'|w)} \right) && \text{(3.4c)} \\
&= \sum_{w \in (\mathcal{S} \times \mathcal{A})^{t_{loss}}} \sum_{w' \in (\mathcal{S} \times \mathcal{A})^{\omega}} \mu^{full}(ww') \log \left( \frac{\mu^{full}(w'|w)}{\prod_{i=1}^{N} \mu_{t_{loss}}^{full}((w')^i|w)} \right) && \text{(3.4d)}
\end{aligned}$$

where $w = \boldsymbol{s}_0 \boldsymbol{a}_0 \ldots \boldsymbol{a}_{t_{loss}-1}$, $w' = \boldsymbol{s}_{t_{loss}} \boldsymbol{a}_{t_{loss}} \ldots$ and $(w')^i = s_{t_{loss}}^i a_{t_{loss}}^i \ldots$(3.4c) is because the imaginary play is the same with the joint policy for $t = 0, \ldots, t_{loss} - 1$ and (3.4d) is because under the imaginary play, the agents are conditionally independent for $t \geq t_{loss}$ given $\boldsymbol{s}_{t_{loss}-1} \boldsymbol{a}_{t_{loss}-1}$.

By the definition of conditional entropy,

$$
\begin{aligned}
&KL(\mathbf{\Gamma}^{full}||\mathbf{\Gamma}^{img}_{t_{loss}}) \\
&= \sum_{w\in(\mathcal{S}\times\mathcal{A})^{t_{loss}}} \sum_{w'\in(\mathcal{S}\times\mathcal{A})^{\omega}} \mu^{full}(ww') \log\left(\frac{\mu^{full}(w'|w)}{\prod_{i=1}^{N}\mu^{full}_{t_{loss}}((w')^i|w)}\right) && \text{(3.5a)}\\
&= \left[\sum_{i=1}^{N} H(S^i_{t_{loss}}A^i_{t_{loss}}S^i_{t_{loss}+1}A^i_{t_{loss}+1}\ldots|\boldsymbol{S}_0\boldsymbol{A}_0\ldots\boldsymbol{S}_{t_{loss}-1}\boldsymbol{A}_{t_{loss}-1})\right] \\
&\quad - H(\boldsymbol{S}_{t_{loss}}\boldsymbol{A}_{t_{loss}}\boldsymbol{S}_{t_{loss}+1}\boldsymbol{A}_{t_{loss}+1}\ldots|\boldsymbol{S}_0\boldsymbol{A}_0\ldots\boldsymbol{S}_{t_{loss}-1}\boldsymbol{A}_{t_{loss}-1}) && \text{(3.5b)}\\
&\leq \left[\sum_{i=1}^{N} H(S^i_{t_{loss}}A^i_{t_{loss}}S^i_{t_{loss}+1}A^i_{t_{loss}+1}\ldots|\boldsymbol{S}_0A^i_0\ldots S^i_{t_{loss}-1}A^i_{t_{loss}-1})\right] \\
&\quad - H(\boldsymbol{S}_{t_{loss}}\boldsymbol{A}_{t_{loss}}\boldsymbol{S}_{t_{loss}+1}\boldsymbol{A}_{t_{loss}+1}\ldots|\boldsymbol{S}_0\boldsymbol{A}_0\ldots\boldsymbol{S}_{t_{loss}-1}\boldsymbol{A}_{t_{loss}-1}) && \text{(3.5c)}\\
&= \sum_{t=t_{loss}}^{\infty}\left[\left[\sum_{i=1}^{N} H(S^i_tA^i_t|\boldsymbol{S}_0A^i_0\ldots S^i_{t-1}A^i_{t-1})\right] - H(\boldsymbol{S}_t\boldsymbol{A}_t|\boldsymbol{S}_0\boldsymbol{A}_0\ldots\boldsymbol{S}_{t-1}\boldsymbol{A}_{t-1})\right] && \text{(3.5d)}
\end{aligned}
$$

where (3.5c) is because conditioning reduces entropy and (3.5d) is due to the chain rule of entropy. Finally, combining (3.2),(3.3), and the definition of $C_{\pi_{joint}}$, we have

$$
\begin{aligned}
&KL(\mathbf{\Gamma}^{full}||\mathbf{\Gamma}^{img}_{t_{loss}}) && \text{(3.6a)}\\
&\leq \sum_{t=t_{loss}}^{\infty}\left[\left[\sum_{i=1}^{N} H(S^i_tA^i_t|\boldsymbol{S}_0A^i_0\ldots S^i_{t-1}A^i_{t-1})\right] - H(\boldsymbol{S}_t\boldsymbol{A}_t|\boldsymbol{S}_0\boldsymbol{A}_0\ldots\boldsymbol{S}_{t-1}\boldsymbol{A}_{t-1})\right] && \text{(3.6b)}\\
&\leq \left[\sum_{i=1}^{N} H(S^i_0A^i_0|\boldsymbol{S}_0)\right] - H(\boldsymbol{S}_0\boldsymbol{A}_0|\boldsymbol{S}_0) && \text{(3.6c)}\\
&+ \sum_{t=0}^{\infty}\left[\left[\sum_{i=1}^{N} H(S^i_tA^i_t|\boldsymbol{S}_0A^i_0\ldots S^i_{t-1}A^i_{t-1})\right] - H(\boldsymbol{S}_t\boldsymbol{A}_t|\boldsymbol{S}_0\boldsymbol{A}_0\ldots\boldsymbol{S}_{t-1}\boldsymbol{A}_{t-1})\right] && \text{(3.6d)}\\
&= C_{\pi_{joint}} && \text{(3.6e)}\\
&= KL(\mathbf{\Gamma}^{full}||\mathbf{\Gamma}^{img}_0). && \text{(3.6f)}
\end{aligned}
$$

Hence, for every $t_{loss} \in \{0, 1, \ldots\} \cup \{\infty\}$ in Algorithm 5,

$$KL(\mathbf{\Gamma}^{full}||\mathbf{\Gamma}^{img}_0) \geq KL(\mathbf{\Gamma}^{full}||\mathbf{\Gamma}^{img}_{t_{loss}}).$$

■

*Proof of Lemma 3.2.* We first show that $KL(\mathbf{\Gamma}^{full}||\mathbf{\Gamma}^{img}_0) \geq KL(\mathbf{\Gamma}^{full}||\mathbf{\Gamma}^{int}_{\Lambda})$ for an arbitrary sequence of $\Lambda = \mathcal{P}_0, \mathcal{P}_1, \ldots$ communication availability.

Define $\mathcal{P}_{-1} = \{[N]\}$. Let $l_j$ denote the starting time index of $j$-th period that communication is not available for the team as a whole, i.e., $\mathcal{P}_{l_j} \neq \{[N]\}$. Formally, $l_1 = \min\{i|\mathcal{P}_i \neq \{[N]\}, \mathcal{P}_{i-1} = \{[N]\}, i \geq 0\}$ and $l_j = \min\{i|\mathcal{P}_i \neq \{[N]\}, \mathcal{P}_{i-1} = \{[N]\}, i > l_{j-1}\}$ for all $j \geq 2$. Similarly, let $r_j$ denote the starting time index of $j$-th period that communication is available again. Formally, $r_1 = \min\{i|\mathcal{P}_i = \{[N]\}, \mathcal{P}_{i-1} \neq \{[N]\}, i \geq 0\}$ and $r_j = \min\{i|\mathcal{P}_i = \{[N]\}, \mathcal{P}_{i-1} \neq \{[N]\}, i > r_{j-1}\}$ for all $j \geq 2$. For example, for $\Lambda = \lambda_0, \lambda_1, \ldots = 0, 1, 1, 0, 0, 0, 1, 1, \ldots$, we have $l_1 = 0$, $r_1 = 1$, $l_2 = 3$, and $r_2 = 6$. For $N = 3$ and $\Lambda = \mathcal{P}_0, \mathcal{P}_1, \ldots = \Big\{[3]\Big\}, \Big\{[3]\Big\}, \Big\{\{1,3\}, \{2\}\Big\}, \Big\{\{1,3\}, \{2\}\Big\}, \Big\{\{1,3\}, \{2\}\Big\}, \Big\{[3]\Big\}, \ldots$, we have $l_1 = 2$ and $r_1 = 5$.

We consider two different cases of $\mathcal{P}_0$ separately. First, assume that $\mathcal{P}_0 = \{[N]\}$, i.e., the communication is available for the team as a whole at time 0. Let $w$ denote $\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots$, $w_{t,t'}$ denote $\boldsymbol{s}_t\boldsymbol{a}_t\ldots\boldsymbol{s}_{t'}\boldsymbol{a}_{t'}$, and $w^i_{t,t'}$ denote $s_ta_t\ldots s_{t'}a_{t'}$.

$$
\begin{aligned}
& KL(\mathbf{\Gamma}^{full}||\mathbf{\Gamma}^{int}_{\Lambda}) && \text{(3.7a)}\\
&= \sum_{w\in(\mathcal{S}\times\mathcal{A})^\omega} \mu^{full}(w)\log\left(\frac{\mu^{full}(w)}{\mu^{int}_{\Lambda}(w)}\right) && \text{(3.7b)}\\
&= \sum_{w\in(\mathcal{S}\times\mathcal{A})^\omega} \mu^{full}(w)\log\left(\frac{\mu^{full}(w_{0,l_1-1})}{\mu^{int}_{\Lambda}(w_{0,l_1-1}))}\right)\\
&\quad + \sum_{j=1}^{\infty} \sum_{w\in(\mathcal{S}\times\mathcal{A})^\omega} \mu^{full}(w)\log\left(\frac{\mu^{full}(w_{l_j,l_{j+1}-1}|w_{0,l_j-1})}{\mu^{int}_{\Lambda}(w_{l_j,l_{j+1}-1}|w_{0,l_j-1})}\right) && \text{(3.7c)}\\
&= \sum_{w\in(\mathcal{S}\times\mathcal{A})^\omega} \mu^{full}(w)\log\left(\frac{\mu^{full}(w_{0,l_1-1})}{\mu^{int}_{\Lambda}(w_{0,l_1-1}))}\right)\\
&\quad + \sum_{j=1}^{\infty} \sum_{w\in(\mathcal{S}\times\mathcal{A})^\omega} \mu^{full}(w)\log\left(\frac{\mu^{full}(w_{l_j,r_j-1}|w_{0,l_j-1})}{\mu^{int}_{\Lambda}(w_{l_j,r_j-1}|w_{0,l_j-1})}\right)\\
&\quad + \sum_{j=1}^{\infty} \sum_{w\in(\mathcal{S}\times\mathcal{A})^\omega} \mu^{full}(w)\log\left(\frac{\mu^{full}(w_{r_j,l_{j+1}-1}|w_{0,r_j-1})}{\mu^{int}_{\Lambda}(w_{r_j,l_{j+1}-1}|w_{0,r_j-1})}\right) && \text{(3.7d)}
\end{aligned}
$$

We note that when the communication is available for the team as a whole the state-action process under the intermittent communication and the state-action process under the joint policy with full communication follow the same Markov chain. Also

note that communication is available between $[0, l_1]$ and $[r_j, l_{j+1} - 1]$ for all $j \geq 1$. Consequently,

$$\begin{aligned} KL(\mathbf{\Gamma}^{full}||\mathbf{\Gamma}^{int}_{\Lambda}) &= \sum_{w\in(\mathcal{S}\times\mathcal{A})^{\omega}} \mu^{full}(w) \log\left(\frac{\mu^{full}(w_{0,l_1-1})}{\mu^{int}_{\Lambda}(w_{0,l_1-1}))}\right) \\ &\quad + \sum_{j=1}^{\infty} \sum_{w\in(\mathcal{S}\times\mathcal{A})^{\omega}} \mu^{full}(w) \log\left(\frac{\mu^{full}(w_{l_j,r_j-1}|w_{0,l_j-1})}{\mu^{int}_{\Lambda}(w_{l_j,r_j-1}|w_{0,l_j-1})}\right) \\ &\quad + \sum_{j=1}^{\infty} \sum_{w\in(\mathcal{S}\times\mathcal{A})^{\omega}} \mu^{full}(w) \log\left(\frac{\mu^{full}(w_{r_j,l_{j+1}-1}|w_{0,r_j-1})}{\mu^{int}_{\Lambda}(w_{r_j,l_{j+1}-1}|w_{0,r_j-1})}\right) \\ &= \sum_{j=1}^{\infty} \sum_{w\in(\mathcal{S}\times\mathcal{A})^{\omega}} \mu^{full}(w) \log\left(\frac{\mu^{full}(w_{l_j,r_j-1}|w_{0,l_j-1})}{\mu^{int}_{\Lambda}(w_{l_j,r_j-1}|w_{0,l_j-1})}\right). \end{aligned}$$

By the same arguments, when $\mathcal{P}_0 \neq \{[N]\}$, i.e., the communication is not available for the team as a whole at time 0,

$$\begin{aligned} KL(\mathbf{\Gamma}^{full}||\mathbf{\Gamma}^{int}_{\Lambda}) &= \sum_{w\in(\mathcal{S}\times\mathcal{A})^{\omega}} \mu^{full}(w) \log\left(\frac{\mu^{full}(w)}{\mu^{int}_{\Lambda}(w)}\right) \qquad (3.9a) \\ &= \sum_{j=1}^{\infty} \sum_{w\in(\mathcal{S}\times\mathcal{A})^{\omega}} \mu^{full}(w) \log\left(\frac{\mu^{full}(w_{l_j,r_j-1}|w_{0,l_j-1})}{\mu^{int}_{\Lambda}(w_{l_j,r_j-1}|w_{0,l_j-1})}\right) \\ &\quad + \sum_{j=1}^{\infty} \sum_{w\in(\mathcal{S}\times\mathcal{A})^{\omega}} \mu^{full}(w) \log\left(\frac{\mu^{full}(w_{r_j,l_{j+1}-1}|w_{0,r_j-1})}{\mu^{int}_{\Lambda}(w_{r_j,l_{j+1}-1}|w_{0,r_j-1})}\right) \\ &= \sum_{j=1}^{\infty} \sum_{w\in(\mathcal{S}\times\mathcal{A})^{\omega}} \mu^{full}(w) \log\left(\frac{\mu^{full}(w_{l_j,r_j-1}|w_{0,l_j-1})}{\mu^{int}_{\Lambda}(w_{l_j,r_j-1}|w_{0,l_j-1})}\right). \end{aligned}$$

Hence, for every value of $\mathcal{P}_0$,

$$KL(\mathbf{\Gamma}^{full}||\mathbf{\Gamma}^{int}_{\Lambda}) = \sum_{j=1}^{\infty} \sum_{w\in(\mathcal{S}\times\mathcal{A})^{\omega}} \mu^{full}(w) \log\left(\frac{\mu^{full}(w_{l_j,r_j-1}|w_{0,l_j-1})}{\mu^{int}_{\Lambda}(w_{l_j,r_j-1}|w_{0,l_j-1})}\right).$$

Since the policy is stationary and the groups agents are conditionally independent between $[l_j, r_j - 1]$ given $\boldsymbol{s}_{l_j-1}\boldsymbol{a}_{l_j-1}$, we have

$$KL(\mathbf{\Gamma}^{full}||\mathbf{\Gamma}^{int}_{\Lambda}) = \sum_{j=1}^{\infty} \sum_{w\in(\mathcal{S}\times\mathcal{A})^{\omega}} \mu^{full}(w) \log\left(\frac{\mu^{full}(w_{l_j,r_j-1}|w_{0,l_j-1})}{\mu^{int}_{\Lambda}(w_{l_j,r_j-1}|w_{0,l_j-1})}\right) \qquad (3.10a)$$

$$= \sum_{j=1}^{\infty} \sum_{w\in(\mathcal{S}\times\mathcal{A})^{\omega}} \mu^{full}(w) \log\left(\frac{\mu^{full}(w_{l_j,r_j-1}|w_{0,l_j-1}}{\prod_{G\in\mathcal{P}_{l_j}} \mu^{full}([w^i_{l_j,r_j-1}]_{i\in\mathcal{P}_G}|\boldsymbol{s}_{l_j-1}\boldsymbol{a}_{l_j-1})}\right). \qquad (3.10b)$$

Let $W$ denote $\boldsymbol{S}_0\boldsymbol{A}_0\boldsymbol{S}_1\boldsymbol{A}_1\ldots$, $W_{t,t'}$ denote $\boldsymbol{S}_t\boldsymbol{A}_t\ldots\boldsymbol{S}_{t'}\boldsymbol{A}_{t'}$, and $W^i_{t,t'}$ denote $S_tA_t\ldots S_{t'}A_{t'}$. By the definition of conditional entropy, we have

$$
\begin{aligned}
&KL(\boldsymbol{\Gamma}^{full}||\boldsymbol{\Gamma}^{int}_{\Lambda}) && \text{(3.11a)}\\
&= \sum_{j=1}^{\infty}\left[\left[\sum_{G\in\mathcal{P}_{l_j}} H([W^i_{l_j,r_j-1}]_{i\in\mathcal{P}_G}|\boldsymbol{S}_{l_j-1}\boldsymbol{A}_{l_j-1})\right] - H(W_{l_j,r_j-1}|W_{0,l_j-1})\right] && \text{(3.11b)}\\
&\leq \sum_{j=1}^{\infty}\left[\left[\sum_{i=1}^{N} H(W^i_{l_j,r_j-1}|\boldsymbol{S}_{l_j-1}\boldsymbol{A}_{l_j-1})\right] - H(W_{l_j,r_j-1}|W_{0,l_j-1})\right]. && \text{(3.11c)}\\
&= \sum_{j=1}^{\infty}\left[\left[\sum_{i=1}^{N} H(W^i_{l_j,r_j-1}|\boldsymbol{S}_0A^i_0S^i_1A^i_1\ldots S^i_{l_j-2}A^i_{l_j-2}\boldsymbol{S}_{l_j-1}\boldsymbol{A}_{l_j-1})\right]\right.\\
&\quad \left. - H(W_{l_j,r_j-1}|W_{0,l_j-1})\right]. && \text{(3.11d)}
\end{aligned}
$$

where (3.11c) is due to that the joint entropy is less than or equal to the sum of individual entropies, and (3.11d) is due to the stationarity of $\pi_{joint}$, i.e., $S^i_{l_j}A^i_{l_j}\ldots S^i_{r_j-1}A^i_{r_j-1}$ is independent of $\boldsymbol{S}_0A^i_0S^i_1A^i_1\ldots S^i_{l_j-2}A^i_{l_j-2}$ given $\boldsymbol{S}_{l_j-1}\boldsymbol{A}_{l_j-1}$.

Since conditioning reduces entropy,

$$
\begin{aligned}
KL(\boldsymbol{\Gamma}^{full}||\boldsymbol{\Gamma}^{int}_{\Lambda}) &= \sum_{j=1}^{\infty}\left[\left[\sum_{i=1}^{N} H(W^i_{l_j,r_j-1}|\boldsymbol{S}_0A^i_0S^i_1A^i_1\ldots\boldsymbol{S}_{l_j-1}\boldsymbol{A}_{l_j-1})\right]\right.\\
&\quad \left. - H(W_{l_j,r_j-1}|W_{0,l_j-1})\right] && \text{(3.12a)}\\
&\leq \sum_{j=1}^{\infty}\left[\left[\sum_{i=1}^{N} H(W^i_{l_j,r_j-1}|\boldsymbol{S}_0A^i_0S^i_1A^i_1\ldots S^i_{l_j-1}A^i_{l_j-1})\right]\right.\\
&\quad \left. - H(W_{l_j,r_j-1}|W_{0,l_j-1})\right] && \text{(3.12b)}\\
&= \sum_{j=1}^{\infty}\sum_{t=l_j}^{r_j-1}\left[\left[\sum_{i=1}^{N} H(S^i_tA^i_t|\boldsymbol{S}_0A^i_0\ldots S^i_{t-1}A^i_{t-1})\right]\right.\\
&\quad \left. - H(\boldsymbol{S}_t\boldsymbol{A}_t|\boldsymbol{S}_0\boldsymbol{A}_0\ldots\boldsymbol{S}_{t-1}\boldsymbol{A}_{t-1})\right] && \text{(3.12c)}
\end{aligned}
$$

where the last equality is due to the definition of conditional entropy.

Let $\min_t \Pr(\mathcal{P}_t = \{[N]\}) = 1-q$, and $\boldsymbol{\Gamma}^{int} = \mathbb{E}_{\Lambda}[\boldsymbol{\Gamma}^{int}_{\Lambda}]$. Also define $\mathbb{1}_{\mathcal{P}_t}(\{[N]\}))$ be an indicator function such that $\mathbb{1}_{\mathcal{P}_t}(\{[N]\})) = 1$ if $\mathcal{P}_t = (\{[N]\}))$ and 0 otherwise. We now show that,

$$KL(\boldsymbol{\Gamma}^{full}||\boldsymbol{\Gamma}^{img}_0) \geq KL(\boldsymbol{\Gamma}^{full}||\boldsymbol{\Gamma}^{int})/q.$$

By the convexity of KL divergence (Boyd and Vandenberghe, 2004) and Assumption 3.2, Jensen's inequality (Boyd and Vandenberghe, 2004) yields

$$
\begin{aligned}
& KL(\mathbf{\Gamma}^{full}||\mathbf{\Gamma}^{int}) && \text{(3.13a)}\\
& = KL(\mathbf{\Gamma}^{full}||\mathbb{E}_\Lambda\left[\mathbf{\Gamma}^{int}_\Lambda\right]) && \text{(3.13b)}\\
& \leq \mathbb{E}_\Lambda\left[KL(\mathbf{\Gamma}^{full}||\mathbf{\Gamma}^{int}_\Lambda])\right] && \text{(3.13c)}\\
& \leq \mathbb{E}_\Lambda\left[\sum_{t=0}^{\infty}(1-(\mathbb{1}_{\mathcal{P}_t}(\{[N]\}))\left[\left[\sum_{i=1}^{N} H(S_t^iA_t^i|\boldsymbol{S}_0A_0^i\ldots S_{t-1}^iA_{t-1}^i)\right]\right.\right. && \\
& \quad \left.\left. -H(\boldsymbol{S}_t\boldsymbol{A}_t|\boldsymbol{S}_0\boldsymbol{A}_0\ldots\boldsymbol{S}_{t-1}\boldsymbol{A}_{t-1})\right]\right] && \text{(3.13d)}\\
& \leq q\sum_{t=0}^{\infty}\left[\left[\sum_{i=1}^{N} H(S_t^iA_t^i|\boldsymbol{S}_0A_0^i\ldots S_{t-1}^iA_{t-1}^i)\right] - H(\boldsymbol{S}_t\boldsymbol{A}_t|\boldsymbol{S}_0\boldsymbol{A}_0\ldots\boldsymbol{S}_{t-1}\boldsymbol{A}_{t-1})\right] && \text{(3.13e)}\\
& = qKL(\mathbf{\Gamma}^{full}||\mathbf{\Gamma}^{img}_0) && \text{(3.13f)}
\end{aligned}
$$

where the last equalities are due to the linearity of expectation and the independence of $\mathcal{P}_t$ values from the state-action processes. Rearranging the terms yields

$$KL(\mathbf{\Gamma}^{full}||\mathbf{\Gamma}^{img}_0) \geq KL(\mathbf{\Gamma}^{full}||\mathbf{\Gamma}^{int})/q.$$

■

*Proof of Lemma 3.3.* The proof is similar to the proof of Lemma 3.2.

If $f(\varepsilon) = 0$, i.e., communication is not available at time 0, then the agents use the imaginary play for the whole path, i.e., $t_{loss} = 0$, and the distribution $\mathbf{\Gamma}^{img}_f$ of paths is the same as $\mathbf{\Gamma}^{img}_0$. Then,

$$KL(\mathbf{\Gamma}^{full}||\mathbf{\Gamma}^{img}_0) = KL(\mathbf{\Gamma}^{full}||\mathbf{\Gamma}^{img}_f).$$

Without loss of generality, we will assume $f(\varepsilon) \neq 0$ for the rest of the proof.

We first define some sets for ease of notation. Let $w = \boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots\boldsymbol{s}_t\boldsymbol{a}_t$ be a finite state-action sequence. Define $Pref(w)$ as the set of all strict prefixes of $w$ such that $Pref(w = \boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots\boldsymbol{s}_t\boldsymbol{a}_t) = \{\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots\boldsymbol{s}_{t'}\boldsymbol{a}_{t'}|t' = 0,\ldots,t-1\}$. Define

$Str(w)$ as the set of all finite state-action sequences that start with $w$ such that $Str(w) = \{ww' | w' \in (\boldsymbol{\mathcal{S}} \times \boldsymbol{\mathcal{A}})^*\}$.

Let $W_{loss}$ be the set of finite state-action sequences that lead to a communication loss for the first time. Formally,

$$W_{loss} = \{w \in (\boldsymbol{\mathcal{S}} \times \boldsymbol{\mathcal{A}})^* | f(w) = 0, \text{ and } (\forall w' \in Pref(W_{loss}), f(w') = 1)\}.$$

Note that there do not exist $w, w' \in W_{loss}$ and $w \neq w'$ such that $w \in Pref(w')$ or $w' \in Pref(w)$.

Let $W_{\neg loss}$ be the set of finite shortest state-action sequences that guarantees the agents will not ever experience a communication loss. Formally,

$$V_{\neg loss} = \{w \in (\boldsymbol{\mathcal{S}} \times \boldsymbol{\mathcal{A}})^* | (\forall w' \in Str(w), f(w') = 1) \tag{3.14}$$

$$\text{and } (\forall w' \in Pref(w), \exists \bar{w} \in Str(w'), f(\bar{w}) = 0)\} \tag{3.15}$$

and

$$W_{\neg loss} = \{w \in (\boldsymbol{\mathcal{S}} \times \boldsymbol{\mathcal{A}})^* | \nexists w' \in V_{\neg loss}, w' \in Pref(w)\}.$$

Note that there do not exist $w, w' \in W_{\neg loss}$ and $w \neq w'$ such that $w \in Pref(w')$ or $w' \in Pref(w)$.

Note that $W_{loss} \cap W_{\neg loss} = \emptyset$. Also, note that

$$\bigcup_{w \in W_{loss} \cup W_{\neg loss}} \{ww' | w' \in (\boldsymbol{\mathcal{S}} \times \boldsymbol{\mathcal{A}})^\omega\} = (\boldsymbol{\mathcal{S}} \times \boldsymbol{\mathcal{A}})^\omega,$$

i.e., every path starts with a finite state-action sequence from $W_{loss}$ or $W_{\neg loss}$. Let $\tau$ denote the random hitting time to set ($W_{loss} \cup W_{\neg loss}$, i.e., $\boldsymbol{s}_0 \boldsymbol{a}_0 \boldsymbol{s}_1 \boldsymbol{a}_1 \ldots \boldsymbol{s}_\tau \boldsymbol{a}_\tau \in (W_{loss} \cup W_{\neg loss})$.

We have

$$KL(\mathbf{\Gamma}^{full}||\mathbf{\Gamma}_f^{img}) \tag{3.16a}$$

$$= \sum_{w\in(\mathcal{S}\times\mathcal{A})^\omega} \mu^{full}(w)\log\left(\frac{\mu^{full}(w)}{\mu_f^{img}(w)}\right) \tag{3.16b}$$

$$= \sum_{w\in(W_{loss}\cup W_{\neg loss})} \mu^{full}(w)\log\left(\frac{\mu^{full}(w)}{\mu_f^{img}(w)}\right)$$
$$+ \sum_{w\in(W_{loss}\cup W_{\neg loss})} \mu^{full}(w) \sum_{w'\in(\mathcal{S}\times\mathcal{A})^\omega} \mu^{full}(w'|w)\log\left(\frac{\mu^{full}(w'|w)}{\mu_f^{img}(w'|w)}\right) \tag{3.16c}$$

$$= \sum_{w\in(W_{loss}\cup W_{\neg loss})} \mu^{full}(w)\log\left(\frac{\mu^{full}(w)}{\mu^{full}(w)}\right)$$
$$+ \sum_{w\in(W_{loss}\cup W_{\neg loss})} \mu^{full}(w) \sum_{w'\in(\mathcal{S}\times\mathcal{A})^\omega} \mu^{full}(w'|w)\log\left(\frac{\mu^{full}(w'|w)}{\mu_f^{img}(w'|w)}\right) \tag{3.16d}$$

$$= \sum_{w\in(W_{loss}\cup W_{\neg loss})} \mu^{full}(w) \sum_{w'\in(\mathcal{S}\times\mathcal{A})^\omega} \mu^{full}(w'|w)\log\left(\frac{\mu^{full}(w'|w)}{\mu_f^{img}(w'|w)}\right) \tag{3.16e}$$

where (3.16d) is because the imaginary play is the same with the joint policy for $t = 0, \ldots, \tau$.

We have

$$KL(\mathbf{\Gamma}^{full}||\mathbf{\Gamma}_f^{img}) \tag{3.17a}$$

$$= \sum_{w\in W_{loss}} \mu^{full}(w) \sum_{w'\in(\mathcal{S}\times\mathcal{A})^\omega} \mu^{full}(w'|w)\log\left(\frac{\mu^{full}(w'|w)}{\prod_{i=1}^N \mu^{full}((w')^i|w)}\right) \tag{3.17b}$$

$$\leq \sum_{w\in W_{loss}} \mu^{full}(w) \sum_{w'\in(\mathcal{S}\times\mathcal{A})^\omega} \mu^{full}(w'|w)\log\left(\frac{\mu^{full}(w'|w)}{\prod_{i=1}^N \mu^{full}((w')^i|w)}\right)$$
$$+ \sum_{w\in W_{\neg loss}} \mu^{full}(w) \sum_{w'\in(\mathcal{S}\times\mathcal{A})^\omega} \mu^{full}(w'|w)\log\left(\frac{\mu^{full}(w'|w)}{\prod_{i=1}^N \mu^{full}((w')^i|w)}\right)$$
$$+ \sum_{w\in(W_{loss}\cup W_{\neg loss})} \mu^{full}(w)\log\left(\frac{\mu^{full}(w)}{\prod_{i=1}^N \mu^{full}(w^i)}\right) \tag{3.17c}$$

since the additional terms in (3.17c) are KL divergences between probability distributions, which are always nonnegative.

By the definition of conditional entropy,

$$
\begin{aligned}
&KL(\mathbf{\Gamma}^{full}||\mathbf{\Gamma}_f^{img}) && \text{(3.18a)}\\
&\leq \left[\sum_{i=1}^{N} H(S^i_{\tau+1}A^i_{\tau+1}S^i_{\tau+2}A^i_{\tau+2}\ldots|\boldsymbol{S}_0\boldsymbol{A}_0\ldots\boldsymbol{S}_\tau\boldsymbol{A}_\tau) + H(S^i_0A^i_0\ldots A^i_\tau A^i_\tau|\boldsymbol{S}_0)\right] \\
&\quad - H(\boldsymbol{S}_{\tau+1}\boldsymbol{A}_{\tau+1}\boldsymbol{S}_{\tau+2}\boldsymbol{A}_{\tau+2}\ldots|\boldsymbol{S}_0\boldsymbol{A}_0\ldots\boldsymbol{S}_\tau\boldsymbol{A}_\tau) - H(\boldsymbol{S}_0\boldsymbol{A}_0\ldots\boldsymbol{S}_\tau\boldsymbol{A}_\tau|\boldsymbol{S}_0) && \text{(3.18b)}\\
&\leq \left[\sum_{i=1}^{N} H(S^i_{\tau+1}A^i_{\tau+1}S^i_{\tau+2}A^i_{\tau+2}\ldots|\boldsymbol{S}_0A^i_0\ldots S^i_\tau A^i_\tau) + H(S^i_0A^i_0\ldots A^i_\tau A^i_\tau|\boldsymbol{S}_0)\right] \\
&\quad - H(\boldsymbol{S}_{\tau+1}\boldsymbol{A}_{\tau+1}\boldsymbol{S}_{\tau+2}\boldsymbol{A}_{\tau+2}\ldots|\boldsymbol{S}_0\boldsymbol{A}_0\ldots\boldsymbol{S}_\tau\boldsymbol{A}_\tau) - H(\boldsymbol{S}_0\boldsymbol{A}_0\ldots\boldsymbol{S}_\tau\boldsymbol{A}_\tau|\boldsymbol{S}_0) && \text{(3.18c)}
\end{aligned}
$$

where (3.18c) is because conditioning reduces entropy. Finally, we have

$$
\begin{aligned}
&KL(\mathbf{\Gamma}^{full}||\mathbf{\Gamma}_f^{img}) && \text{(3.19a)}\\
&\leq \left[\sum_{i=1}^{N} H(S^i_{\tau+1}A^i_{\tau+1}S^i_{\tau+2}A^i_{\tau+2}\ldots|\boldsymbol{S}_0A^i_0\ldots S^i_\tau A^i_\tau) + H(S^i_0A^i_0\ldots A^i_\tau A^i_\tau|\boldsymbol{S}_0)\right] \\
&\quad - H(\boldsymbol{S}_{\tau+1}\boldsymbol{A}_{\tau+1}\boldsymbol{S}_{\tau+2}\boldsymbol{A}_{\tau+2}\ldots|\boldsymbol{S}_0\boldsymbol{A}_0\ldots\boldsymbol{S}_\tau\boldsymbol{A}_\tau) - H(\boldsymbol{S}_0\boldsymbol{A}_0\ldots\boldsymbol{S}_\tau\boldsymbol{A}_\tau|\boldsymbol{S}_0) && \text{(3.19b)}\\
&= \left[\sum_{i=1}^{N} H(S^i_0A^i_0|\boldsymbol{S}_0)\right] - H(\boldsymbol{S}_0\boldsymbol{A}_0|\boldsymbol{S}_0) \\
&+ \sum_{t=0}^{\infty}\left[\left[\sum_{i=1}^{N} H(S^i_tA^i_t|\boldsymbol{S}_0A^i_0\ldots S^i_{t-1}A^i_{t-1})\right] - H(\boldsymbol{S}_t\boldsymbol{A}_t|\boldsymbol{S}_0\boldsymbol{A}_0\ldots\boldsymbol{S}_{t-1}\boldsymbol{A}_{t-1})\right] && \text{(3.19c)}\\
&= C_{\pi_{joint}} && \text{(3.19d)}\\
&= KL(\mathbf{\Gamma}^{full}||\mathbf{\Gamma}_0^{img}) && \text{(3.19e)}
\end{aligned}
$$

where (3.19c) is due to the chain rule of entropy.

■

*Proof of Theorem 3.1.* Let $R$ be the set of paths that reach $\boldsymbol{\mathcal{S}}_{\mathcal{T}}$. A path $\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots\in R$ if and only if there exists $t\geq 0$ such that $\boldsymbol{s}_t\in R$. Also let $R'$ be an arbitrary set

of paths.

$$\begin{aligned}
\boldsymbol{v}^{full} - \boldsymbol{v}^{img} &= \sum_{\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots\in R} \mu^{full}(\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots) - \mu_f^{img}(\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots) && \text{(3.20a)}\\
&\leq \left| \sum_{\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots\in R} \mu^{full}(\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots) - \mu_f^{img}(\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots) \right| && \text{(3.20b)}\\
&\leq \sup_{R'} \left| \sum_{\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots\in R'} \mu^{full}(\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots) - \mu_f^{img}(\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots) \right| && \text{(3.20c)}\\
&\leq \sqrt{1 - \exp(-KL(\boldsymbol{\Gamma}^{full}||\boldsymbol{\Gamma}_f^{img}))} && \text{(3.20d)}\\
&\leq \sqrt{1 - \exp(-C_{\pi_{joint}})} && \text{(3.20e)}
\end{aligned}$$

where (3.20d) is due to Bretagnolle-Huber inequality (Bretagnolle and Huber, 1979) and (3.20e) is due to Lemma 3.3. Rearranging the terms of (3.20e) yields to the desired result.

■

*Proof of Theorem 3.2.* We first show that

$$\boldsymbol{v}^{img} \geq \boldsymbol{v}^{full}(1-p)^{\frac{l^{full}}{\boldsymbol{v}^{full}}}.$$

Remember that $len(\boldsymbol{\xi} = \boldsymbol{s}_0\boldsymbol{a}_0\ldots) = \min\{t+1 | \boldsymbol{s}_t \in \boldsymbol{\mathcal{S}}_{\mathcal{T}} \cup \boldsymbol{\mathcal{S}}_{\mathcal{D}}\}$ and $l^{full} = \mathbb{E}[len(\boldsymbol{\xi})|\boldsymbol{\xi} \sim \boldsymbol{\Gamma}^{full}]$. Let $Success$ be an event that the path satisfies the reach-avoid specification. Define $l_+^{full} = \mathbb{E}[len(\boldsymbol{\xi})|\boldsymbol{\xi} \sim \boldsymbol{\Gamma}^{full}, Success]$ and $l_-^{full} = \mathbb{E}[len(\boldsymbol{\xi})|\boldsymbol{\xi} \sim \boldsymbol{\Gamma}^{full}, \neg Success]$. Note that

$$l^{full} = l_+^{full}\boldsymbol{v}^{full} + l_-^{full}(1-\boldsymbol{v}^{full}) \geq l_+^{full}\boldsymbol{v}^{full}.$$

Also note that

$$\boldsymbol{v}^{full} = \sum_{t=0}^{\infty} \sum_{\substack{\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots\boldsymbol{s}_t\in(\boldsymbol{\mathcal{S}}\times\boldsymbol{\mathcal{A}})^t\times\boldsymbol{\mathcal{S}} \\ \boldsymbol{s}_0,\ldots,\boldsymbol{s}_{t-1}\notin\boldsymbol{\mathcal{S}}_{\mathcal{T}}\cup\boldsymbol{\mathcal{S}}_{\mathcal{D}} \\ \boldsymbol{s}_t\in\boldsymbol{\mathcal{S}}_{\mathcal{T}}}} \mu^{full}(\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots\boldsymbol{s}_t),$$

and

$$l_{+}^{full} = \frac{1}{\boldsymbol{v}^{full}} \left( \sum_{t=0}^{\infty} (t+1) \sum_{\substack{\boldsymbol{s}_0 \boldsymbol{a}_0 \boldsymbol{s}_1 \boldsymbol{a}_1 \ldots \boldsymbol{s}_t \in (\mathcal{S} \times \mathcal{A})^t \times \mathcal{S} \\ \boldsymbol{s}_0, \ldots, \boldsymbol{s}_{t-1} \notin \mathcal{S}_{\mathcal{T}} \cup \mathcal{S}_{\mathcal{D}} \\ \boldsymbol{s}_t \in \mathcal{S}_{\mathcal{T}}}} \mu^{full}(\boldsymbol{s}_0 \boldsymbol{a}_0 \boldsymbol{s}_1 \boldsymbol{a}_1 \ldots \boldsymbol{s}_t) \right).$$

Let $L$ be the event that the agents experience a communication loss before they reach a state in $\mathcal{S}_{\mathcal{T}} \cup \mathcal{S}_{\mathcal{D}}$. Also let $\mu^{img}$ denote the probability measure over the (finite or infinite) state-action process under the imaginary play (under Algorithm 5) where $\Pr(t_{loss} = t) = (1-p)^t p$. Since $L$ and $\neg L$ are disjoint events,

$$\boldsymbol{v}^{img} = \Pr^{\mu^{img}}(Success \ \& \ L) + \Pr^{\mu^{img}}(Success \ \& \ \neg L) \geq \Pr^{\mu^{img}}(Success \ \& \ \neg L).$$

We have

$$\Pr^{\mu^{img}}(Success \ \& \ \neg L) \tag{3.21a}$$

$$= \sum_{t=0}^{\infty} \sum_{\substack{\boldsymbol{s}_0 \boldsymbol{a}_0 \ldots \boldsymbol{s}_t \in (\mathcal{S} \times \mathcal{A})^t \times \mathcal{S} \\ \boldsymbol{s}_0, \ldots, \boldsymbol{s}_{t-1} \notin \mathcal{S}_{\mathcal{T}} \cup \mathcal{S}_{\mathcal{D}} \\ \boldsymbol{s}_t \in \mathcal{S}_{\mathcal{T}}}} \mu^{img}(\boldsymbol{s}_0 \boldsymbol{a}_0 \boldsymbol{s}_1 \boldsymbol{a}_1 \ldots \boldsymbol{s}_t | t_{loss} = t) \Pr(t_{loss} = t) \tag{3.21b}$$

$$= \sum_{t=0}^{\infty} \sum_{\substack{\boldsymbol{s}_0 \boldsymbol{a}_0 \ldots \boldsymbol{s}_t \in (\mathcal{S} \times \mathcal{A})^t \times \mathcal{S} \\ \boldsymbol{s}_0, \ldots, \boldsymbol{s}_{t-1} \notin \mathcal{S}_{\mathcal{T}} \cup \mathcal{S}_{\mathcal{D}} \\ \boldsymbol{s}_t \in \mathcal{S}_{\mathcal{T}}}} \mu^{full}(\boldsymbol{s}_0 \boldsymbol{a}_0 \boldsymbol{s}_1 \boldsymbol{a}_1 \ldots \boldsymbol{s}_t)(1-p)^{t+1} \tag{3.21c}$$

since $\mu^{img} = \mu^{full}$ if there is not a communication loss.

Let $g(t) = (1-p)^{t+1}$ for $t \geq 0$. We note that $g(t)$ is a convex function of $t$. Also, let $Q$ be a probability distribution over $0, 1 \ldots$ such that

$$Q(t) = \frac{1}{\boldsymbol{v}^{full}} \left( \sum_{\substack{\boldsymbol{s}_0 \boldsymbol{a}_0 \boldsymbol{s}_1 \boldsymbol{a}_1 \ldots \boldsymbol{s}_t \in (\mathcal{S} \times \mathcal{A})^t \times \mathcal{S} \\ \boldsymbol{s}_0, \ldots, \boldsymbol{s}_{t-1} \notin \mathcal{S}_{\mathcal{T}} \cup \mathcal{S}_{\mathcal{D}} \\ \boldsymbol{s}_t \in \mathcal{S}_{\mathcal{T}}}} \mu^{full}(\boldsymbol{s}_0 \boldsymbol{a}_0 \boldsymbol{s}_1 \boldsymbol{a}_1 \ldots \boldsymbol{s}_t) \right).$$

Note that $\mathbb{E}_{t \sim Q}[t] = l_{+}^{full} - 1$.

$\Pr^{\mu^{img}}(Success \;\&\; \neg L)$ is equal to

$$\sum_{t=0}^{\infty} \sum_{\substack{\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots\boldsymbol{s}_t\in(\mathcal{S}\times\mathcal{A})^t\times\mathcal{S} \\ \boldsymbol{s}_0,\ldots,\boldsymbol{s}_{t-1}\notin\mathcal{S}_{\mathcal{T}}\cup\mathcal{S}_{\mathcal{D}} \\ \boldsymbol{s}_t\in\mathcal{S}_{\mathcal{T}}}} \mu^{full}(\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots\boldsymbol{s}_t)(1-p)^{t+1} = \boldsymbol{v}^{full}\mathbb{E}_{t\sim Q}\left[g(t)\right].$$

Since $g(t)$ is a convex function of $t$, we get

$$\boldsymbol{v}^{full}\mathbb{E}_{t\sim Q}\left[g(t)\right] \geq \boldsymbol{v}^{full}g(\mathbb{E}_{t\sim Q}\left[t\right]) = \boldsymbol{v}^{full}(1-p)^{\mathbb{E}_{t\sim Q}[t]+1} = \boldsymbol{v}^{full}(1-p)^{l_+^{full}}.$$

by Jensen's inequality (Boyd and Vandenberghe, 2004).

Finally, using $\boldsymbol{v}^{img} \geq \Pr^{\mu^{img}}(Success \;\&\; \neg L)$ and $l^{full} \geq l_+^{full}\boldsymbol{v}^{full}$, we get

$$\boldsymbol{v}^{img} \geq \boldsymbol{v}^{full}(1-p)^{\frac{l^{full}}{\boldsymbol{v}^{full}}}.$$

The proof for $\boldsymbol{v}^{img} \geq \boldsymbol{v}^{full} - \sqrt{1-\exp(-C_{\pi_{joint}})}$ follows the same structure with the proof of Theorem 3.1 and have slight differences. We give the full proof for completeness. Let $R$ be the set of paths that reach $\mathcal{S}_{\mathcal{T}}$. A path $\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots \in R$ if and only if there exists $t \geq 0$ such that $\boldsymbol{s}_t \in R$. Also let $R'$ be an arbitrary set of paths. Define $\boldsymbol{\Gamma}^{img} = \mathbb{E}_{t_{loss}}[\boldsymbol{\Gamma}^{img}_{t_{loss}}]$.

$$\begin{aligned}
\boldsymbol{v}^{full} - \boldsymbol{v}^{img} &= \sum_{\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots\in R} \mu^{full}(\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots) - \mu^{img}(\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots) && \text{(3.22a)} \\
&\leq \left| \sum_{\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots\in R} \mu^{full}(\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots) - \mu^{img}(\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots) \right| && \text{(3.22b)} \\
&\leq \sup_{R'} \left| \sum_{\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots\in R'} \mu^{full}(\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots) - \mu^{img}(\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots) \right| && \text{(3.22c)} \\
&\leq \sqrt{1-\exp(-KL(\boldsymbol{\Gamma}^{full}||\boldsymbol{\Gamma}^{img}))} && \text{(3.22d)} \\
&\leq \sqrt{1-\exp(-\mathbb{E}_{t_{loss}}[KL(\boldsymbol{\Gamma}^{full}||\boldsymbol{\Gamma}^{img}_{t_{loss}}))])} && \text{(3.22e)} \\
&\leq \sqrt{1-\exp(-\mathbb{E}_{t_{loss}}[C_{\pi_{joint}}])} && \text{(3.22f)} \\
&= \sqrt{1-\exp(-C_{\pi_{joint}})} && \text{(3.22g)}
\end{aligned}$$

where (3.22d) is due to Bretagnolle-Huber inequality (Bretagnolle and Huber, 1979), (3.22e) is due to the convexity of the KL divergence, and (3.22f) is due to Lemma 3.1. Rearranging the terms of (3.22g) yields to the desired result.

■

*Proof of Theorem 3.3.* The proof of Theorem 3.3 follows the same structure with the proof of Theorem 3.2 and have slight differences. We give the full proof for completeness.

Let $R$ be the set of paths that reach $\boldsymbol{\mathcal{S}}_{\mathcal{T}}$. A path $\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots \in R$ if and only if there exists $t \geq 0$ such that $\boldsymbol{s}_t \in R$. Also let $R'$ be an arbitrary set of paths.

$$\begin{aligned}
\boldsymbol{v}^{full} - \boldsymbol{v}^{int} &= \sum_{\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots\in R} \mu^{full}(\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots) - \mu^{int}(\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots) && \text{(3.23a)}\\
&\leq \left| \sum_{\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots\in R} \mu^{full}(\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots) - \mu^{int}(\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots) \right| && \text{(3.23b)}\\
&\leq \sup_{R'} \left| \sum_{\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots\in R'} \mu^{full}(\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots) - \mu^{int}(\boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots) \right| && \text{(3.23c)}\\
&\leq \sqrt{1 - \exp(-KL(\boldsymbol{\Gamma}^{full}||\boldsymbol{\Gamma}^{int}))} && \text{(3.23d)}\\
&\leq \sqrt{1 - \exp(-qC_{\pi_{joint}})} && \text{(3.23e)}
\end{aligned}$$

where (3.23d) is due to Bretagnolle-Huber inequality (Bretagnolle and Huber, 1979), and (3.23e) is due to Lemma 3.2. Rearranging the terms of (3.23e) yields the desired result. ■

*Proof of Proposition 3.1.* We first show that $\boldsymbol{v}^{int} \leq \boldsymbol{v}^{\mathcal{D}} \leq (1 + q/m - q)^{N-1}$.

Note that the reachability specification is satisfied if and only if for every $\forall 1 \leq i < N$, there exists $j$ such that $s^i_{2i} = z^i_{2i,j,k} \wedge a^{i+1}_{2i} = a_k$. In words, the action index of Agent $i+1$ at time step $2i$ must match the successor state index of Agent $i$'s uniformly random transition at time step $2i-1$. Also, note that

$$\Pr(\forall 1 \leq i < N, \exists j, s^i_{2i} = z^i_{2i,j,k} \wedge a^{i+1}_{2i} = a_k) \leq \prod_{i=1}^{N-1} \max \Pr(s^i_{2i} = z^i_{2i,j,k} \wedge a^{i+1}_{2i} = a_k)$$

Since the specification is a conjuction formula.

If $\mathcal{P}_{2i} = \{\{1\}, \ldots, \{N\}\}$, then $\max \Pr(s^i_{2i} = z^i_{2i,j,k} \wedge a^{i+1}_{2i} = a_k) = 1/m$ for every mechanism $\mathcal{D}$ and team history, i.e., every action distribution under $\mathcal{D}$ leads to the

same matching probability. If $\mathcal{P}_{2i} = \{[N]\}$, then given the state $s^i_{2i}$ of Agent $i$ there exists action for Agent $i+1$ at time step $2i$ that matches the successor state index of Agent $i$'s uniformly random transition at time step $2i-1$ $\max \Pr(s^i_{2i} = z^i_{2i,j,k} \wedge a^{i+1}_{2i} = a_k) = 1$.

Let $\mathbb{1}_{\mathcal{P}_t}(\{[N]\})$ be an indicator variable such that $\mathbb{1}_{\mathcal{P}_t}(\{[N]\}) = 1$ if $\mathcal{P}_t = \{[N]\}$ and 0 otherwise. Given a sequence of communication partitions $\Lambda = \mathcal{P}_0, \mathcal{P}_1, \ldots$, let $g(\Lambda) = N - 1 - \sum_{i=1}^{N-1} \mathbb{1}_{\mathcal{P}_{2i}}(\{[N]\})$, i.e., $g(\Lambda)$ is the number of times that the agent cannot communicate at the matching time steps $2, 4, \ldots, 2N-2$. Then,

$$\boldsymbol{v}^{\mathcal{D}} \leq \prod_{i=1}^{N-1} \max \Pr(s^i_{2i} = z^i_{2i,j,k} \wedge a^{i+1}_{2i} = a_k) = \mathbb{E}_\Lambda[(1/m)^{g(\Lambda)}].$$

Since the communication availability is a Bernoulli($q$) process,

$$\mathbb{E}_\Lambda[(1/m)^{g(\Lambda)}] = \sum_{h=0}^{N-1} \binom{N-1}{h} q^h (1-q)^{N-1-h} (1/m)^h = (1 + q/m - q)^{N-1}.$$

Combining this with the fact that $\mathcal{D}$ is optimal, we get $\boldsymbol{v}^{int} \leq \boldsymbol{v}^{\mathcal{D}} \leq (1+q/m-q)^{N-1}$.

We now show $(1 + q/m - q)^{N-1} \leq 2\boldsymbol{v}^{int}$. Let $\pi_{max} = \arg\max_\pi \boldsymbol{v}^{full}$, and $\boldsymbol{v}^{max}$ be the reachability probability of $\pi_{max}$ under full communication. Note that $\boldsymbol{v}^{max} = 1$ since every agent can match the previous agent's successor state with probability 1.

Under $\pi_{max}$, $H(X^i) = 2\log(m)$ for every $2 \leq i \leq N$. This is because every Agent $i$ matches Agent $i-1$'s uniformly random transition and also uniformly randomly transitions to a successor state. Consequently, for every $2 \leq i \leq N$, Agent $i$ has $m^2$ equiprobable paths. Also note that $H(X^i | X^1 \ldots X^{i-1}) = \log(m)$ since every Agent $i$ deterministically matches Agent $i-1$'s uniformly random transition and also uniformly randomly transitions to one of the $m$ successor states.

We have

$$C_{\pi_{max}} = \left[\sum_{i=1}^{N} H(X^i)\right] - H(\boldsymbol{X}) \tag{3.24a}$$

$$= \left[\sum_{i=2}^{N} H(X^i)\right] - H(X^2 \dots X^N | X^1) \tag{3.24b}$$

$$= \left[\sum_{i=2}^{N} H(X^i)\right] - \left[\sum_{i=2}^{N} H(X^i | X^1 \dots X^{i-1})\right] \tag{3.24c}$$

$$= (N-1)\log(m) \tag{3.24d}$$

where the second and third equalities are due to the chain rule of entropy.

Due to $\pi_{MD} = \arg\max_\pi \boldsymbol{v}^{full} - \sqrt{1-\exp(-qC)}$ and Theorem 3.3, we have

$$\begin{aligned}(\boldsymbol{v}^{max} - \sqrt{1-\exp(-qC_{\pi_{max}})}) &= 1 - \sqrt{1-\exp(-q(N-1)\log(m))} \\ &= 1 - \sqrt{1-(1/m)^{(N-1)q}} \\ &\leq \boldsymbol{v}^{int}.\end{aligned}$$

Since $\sqrt{1-x} \leq 1 - x/2$ for $0 \leq x \leq 1$, we get

$$(1/m)^{(N-1)q}/2 \leq 1 - \sqrt{1-(1/m)^{(N-1)q}} \leq \boldsymbol{v}^{int}.$$

Since $x^r \geq 1 + rx - r$ for $x \geq 0$ and $r \in [0,1]$, we get $(1/m)^q \geq 1 + q/m - q$. Consequently,

$$(1 + q/m - q)^{N-1} \leq 2\boldsymbol{v}^{int}.$$

Combining this with $\boldsymbol{v}^{int} \leq \boldsymbol{v}^{\mathcal{D}} \leq (1 + q/m - q)^{N-1}$, we get the desired result.

■

*Proof of Proposition 3.2.* We first show that for every policy $\pi'_{joint}$ there exists a stationary policy $\pi^{st}_{joint}$ such that the value of (3.1) for $\pi^{st}_{joint}$ is lower than equal to the value for $\pi'_{joint}$.

Since every $\boldsymbol{s} \in \boldsymbol{\mathcal{S}}$ has a finite occupancy measure and $\boldsymbol{s}_\epsilon$ is absorbing, there exists a stationary policy $\pi^{st}_{joint}$ such that the occupancy measure of $\pi'_{joint}$ and $\pi^{st}_{joint}$ are equal for all $\boldsymbol{s} \in \boldsymbol{\mathcal{S}}, \boldsymbol{a} \in \boldsymbol{\mathcal{A}} \cup \{\epsilon\}$ (Altman, 1999).

We note that

$$\boldsymbol{v}^{full} = \sum_{\substack{\boldsymbol{s}\in\boldsymbol{\mathcal{S}}\setminus(\boldsymbol{\mathcal{S}}_{\mathcal{T}}\cup\boldsymbol{\mathcal{S}}_{\mathcal{D}}) \\ \boldsymbol{a}\in\boldsymbol{\mathcal{A}} \\ \boldsymbol{y}\in\boldsymbol{\mathcal{S}}_{\mathcal{T}}}} x_{\boldsymbol{s},\boldsymbol{a}}\boldsymbol{\mathcal{T}}(\boldsymbol{s},\boldsymbol{a},\boldsymbol{y})$$

and

$$l^{full} = \sum_{\substack{\boldsymbol{s}\in\boldsymbol{\mathcal{S}} \\ \boldsymbol{a}\in\boldsymbol{\mathcal{A}}}} x_{\boldsymbol{s},\boldsymbol{a}}.$$

Hence the value of $\boldsymbol{v}^{full}$ is the same for $\pi'_{joint}$ and $\pi^{st}_{joint}$. Similarly, the value of $l^{full}$ is the same for $\pi'_{joint}$ and $\pi^{st}_{joint}$.

The entropy $H(\bar{X}^i)$ of the stationary state-action process(Savas et al., 2019) $\bar{X}^i$ is

$$\sum_{\substack{s^i\in\mathcal{S}^i \\ a^i\in\mathcal{A}^i\cup\{\epsilon^i\}}} x_{s^i,a^i}\left(\log\left(\frac{\sum\limits_{b^i\in\mathcal{A}^i} x_{s^i,b^i}}{x_{s^i,a^i}}\right)+\log\left(\frac{1}{\mathcal{T}^i(s^i,a^i,y^i)}\right)\right),$$

which is the same for for both $\pi'_{joint}$ and $\pi^{st}_{joint}$.

Given a set of policies with the same occupancy measure, the stationary policy achieves the highest entropy (Savas et al., 2019). Consequently, the value of $H(\boldsymbol{X})$ for $\pi^{st}_{joint}$ is greater than or equal to the value for$\pi'_{joint}$.

Since $\pi^{st}_{joint}$ achieves a higher value $H(\boldsymbol{X})$ and the other terms have equal values for both $\pi'_{joint}$ and $\pi^{st}_{joint}$, the value of (3.1) for $\pi^{st}_{joint}$ is lower than equal to the value for $\pi'_{joint}$.

Given that the stationary policies suffice, (3.1) can be rewritten in terms of

the occupancy measure:

$$\max_{x} \quad \boldsymbol{v}^{full} - \delta l^{full} - \beta \left( \sum_{i=1}^{N} H(\bar{X}^i) - H(\boldsymbol{X}) \right) \tag{3.25a}$$

$$\text{s.t.} \quad \boldsymbol{v}^{full} = \sum_{\substack{\boldsymbol{s} \in \boldsymbol{\mathcal{S}} \setminus (\boldsymbol{\mathcal{S}}_{\mathcal{T}} \cup \boldsymbol{\mathcal{S}}_{\mathcal{D}}) \\ \boldsymbol{a} \in \boldsymbol{\mathcal{A}} \\ \boldsymbol{y} \in \boldsymbol{\mathcal{S}}_{\mathcal{T}}}} x_{\boldsymbol{s},\boldsymbol{a}} \boldsymbol{\mathcal{T}}(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{y}) \tag{3.25b}$$

$$l^{full} = \sum_{\substack{\boldsymbol{s} \in \boldsymbol{\mathcal{S}} \\ \boldsymbol{a} \in \boldsymbol{\mathcal{A}} \cup \{\epsilon\}}} x_{\boldsymbol{s},\boldsymbol{a}} \tag{3.25c}$$

$$H(\boldsymbol{X}) = \sum_{\substack{\boldsymbol{s} \in \boldsymbol{\mathcal{S}} \\ \boldsymbol{a} \in \boldsymbol{\mathcal{A}}}} x_{\boldsymbol{s},\boldsymbol{a}} \log \left( \frac{\sum_{\boldsymbol{b} \in \boldsymbol{\mathcal{A}}} x_{\boldsymbol{s},\boldsymbol{b}}}{x_{\boldsymbol{s},\boldsymbol{a}}} \right) \tag{3.25d}$$

$$+ \sum_{\substack{\boldsymbol{s} \in \boldsymbol{\mathcal{S}} \\ \boldsymbol{a} \in \boldsymbol{\mathcal{A}}}} x_{\boldsymbol{s},\boldsymbol{a}} \sum_{\boldsymbol{y} \in \boldsymbol{\mathcal{S}}} \boldsymbol{\mathcal{T}}(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{y}) \log \left( \frac{1}{\boldsymbol{\mathcal{T}}(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{y})} \right) \tag{3.25e}$$

$$H(\bar{X}^i) = \sum_{\substack{s^i \in \mathcal{S}^i \\ a^i \in \mathcal{A}^i \cup \{\epsilon^i\}}} x_{s^i,a^i} \log \left( \frac{\sum_{b^i \in \mathcal{A}^i} x_{s^i,b^i}}{x_{s^i,a^i}} \right) \tag{3.25f}$$

$$+ \sum_{\substack{s^i \in \mathcal{S}^i \\ a^i \in \mathcal{A}^i \cup \{\epsilon^i\}}} x_{s^i,a^i} \sum_{y^i \in \mathcal{S}^i \cup \{s^i_\epsilon\}} \mathcal{T}^i(s^i, a^i, y^i) \log \left( \frac{1}{\mathcal{T}^i(s^i, a^i, y^i)} \right). \tag{3.25g}$$

$$\sum_{\boldsymbol{a} \in \boldsymbol{\mathcal{A}} \cup \{\epsilon\}} x_{\boldsymbol{s},\boldsymbol{a}} = \sum_{\substack{\boldsymbol{y} \in \boldsymbol{\mathcal{S}} \\ \boldsymbol{b} \in \boldsymbol{\mathcal{A}} \cup \{\epsilon\}}} x_{\boldsymbol{y},\boldsymbol{b}} \boldsymbol{\mathcal{T}}(\boldsymbol{y}, \boldsymbol{b}, \boldsymbol{s}) + \mathbb{1}_{\{\boldsymbol{s_0} = \boldsymbol{s}\}}, \ \forall \boldsymbol{s} \in \boldsymbol{\mathcal{S}} \tag{3.25h}$$

$$x_{\boldsymbol{s},\boldsymbol{a}} \geq 0, \ \forall \boldsymbol{s} \in \boldsymbol{\mathcal{S}}, \boldsymbol{a} \in \boldsymbol{\mathcal{A}} \cup \{\epsilon\} \tag{3.25i}$$

$$x_{\boldsymbol{s}_\epsilon,\boldsymbol{a}} = 0, \ \forall \boldsymbol{a} \in \boldsymbol{\mathcal{A}}. \tag{3.25j}$$

Since the occupancy measure is bounded and closed, i.e.,$\sum_{\boldsymbol{a} \in \boldsymbol{\mathcal{A}}} x(\boldsymbol{s}, \boldsymbol{a}) \leq K$ for all $\boldsymbol{s} \in \boldsymbol{\mathcal{S}}$, the feasible space is compact. Since the feasible space is compact and the objective function is continuous, there exists a solution to (3.25). Hence there exists a stationary policy that is a solution to (3.1).

■

# Chapter 4: Smooth Convex Optimization Using Sub-Zeroth-Order Oracles

In this chapter[1], we consider the problem of minimizing a smooth, Lipschitz continuous, convex function $f$ on a convex, compact domain $C \subset \mathbb{R}^n$ using sub-zeroth-order oracles: i) the *directional-preference oracle* that outputs the sign of the directional derivative for a given point and direction, ii) the *comparator oracle* that compares the function value for two given points, and iii) the *noisy-value oracle* that outputs the function value plus a subgaussian noise.

For the directional-preference and comparator oracles, we prove an upper bound on the sample complexity that is polynomial in the relevant parameters. Our algorithms take advantage of the convexity and smoothness of the objective function, and rely on gradient estimation. We show that the direction of the gradient can be estimated with high accuracy via the sub-zeroth-order oracles. Having estimated the direction of the gradient, we use a variant of the ellipsoid method (Shor, 1972; Yudin and Nemirovskii, 1976). We show that the sample complexity is $\tilde{\mathcal{O}}(n^4)$ for the directional-preference and comparator oracles. To the best of our knowledge, the optimization algorithm that we provide for the comparator oracle is the first algorithm with a polynomial sample complexity for smooth convex functions with logarithmic dependence on the suboptimality gap.

We also develop a sublinear regret algorithm for the noisy-value oracle. The algorithm incurs $\tilde{\mathcal{O}}(n^{3.75}T^{0.75})$[2] regret (ignoring the other factors) with high probability where $T$ is the number of queries. The best known high probability regret bound for the noisy-value oracle is $\tilde{\mathcal{O}}(n^{9.5}\sqrt{T})$ (Bubeck et al., 2017). While our algorithm

[1]The research presented in this chapter is published in (Karabag et al., 2021a). Mustafa O. Karabag formulated the problem, derived the technical results, and wrote the paper.

[2]The publication Karabag et al. (2021a) included the above regret bound. The bound can be improved to $\tilde{\mathcal{O}}(n^{2.25}T^{0.75})$ by changing the analysis as described in Remark 4.1.

requires smoothness, and its regret is not optimal in terms of the dependency on the number of queries, its lower order dependency on the number of dimensions makes it appealing compared to this existing regret bound.

**Summary of Contributions** We have the following contributions for smooth convex optimization:

- For the directional preference oracle that outputs the sign of the directional derivative at the query point and direction, we develop an algorithm with $\tilde{\mathcal{O}}(n^4)$ sample complexity where $n$ is the number of dimensions.
- For the comparator oracle that compares the function value at two query points and outputs a binary comparison value, we develop an algorithm with $\tilde{\mathcal{O}}(n^4)$ sample complexity.
- For the noisy value oracle, we develop an algorithm with $\tilde{\mathcal{O}}(n^{3.75}T^{0.75})$ (ignoring the other factors) high probability regret bound where $T$ is the number of queries.

**Outline** In §4.1, we discuss related work. In §4.2, we introduce preliminary background material. We introduce the considered sub-zeroth-order oracles in §4.3.1. We present the optimization algorithms and sample complexity results in §4.3.2. Finally, in §4.4 we provide an optimization algorithm for the noisy-value oracles that incurs sublinear regret. The proofs for the theoretical results are given in 2.3.5.

## 4.1 Related Work

The bisection method (Burden and Faires, 1985) uses the directional-preference oracle to optimize a one-dimensional function. In multiple dimensions, Qian et al. (2015) used the directional-preference oracle to optimize a linear function. Their algorithm uses a predefined set of query directions, whereas we consider a setting where

the algorithm is allowed to query any direction at any point. SignSGD (Bernstein et al., 2018) requires the sign of directional derivatives only for fixed orthogonal basis vectors and converges to the optimum for smooth convex functions. SignSGD enjoys lower order dependency $\mathcal{O}(n)$ on the number of dimensions. However, it has a sublinear rate of convergence whereas our algorithm has a linear rate of convergence. Additionally, our algorithm for the directional-preference oracle also works for non-smooth functions.

Optimization using the comparator oracle was explored with directional direct search methods (Audet and Dennis Jr, 2006), the Nelson-Mead method (Nelder and Mead, 1965), and variants of gradient descent method Jamieson et al. (2012); Cheng et al. (2020). Directional direct search is guaranteed to converge to an optimal solution in the limit for smooth convex functions. However, the algorithm does not have a known rate of convergence. Meanwhile, the Nelson-Mead method may fail to converge to a stationary point for smooth convex functions (McKinnon, 1998). Convergent variants of the Nelson-Mead method use function values in addition to comparator oracle queries (Price et al., 2002). Jamieson et al. (2012) proved a $\mathcal{O}(n \log(1/\varepsilon))$ (ignoring the other factors) sample complexity lower bound for strongly convex functions and provided a coordinate descent and line search based algorithm that matches the lower bound. For smooth convex functions, (Cheng et al., 2020) provided an algorithm that estimates the gradient by randomly selecting a direction and has a sublinear rate of convergence.

For the regret using the noisy-value oracle, a lower bound of $\Omega(n\sqrt{T})$ has been shown (Shamir, 2013). Lattimore (2020) gave an existence result for an algorithm that achieves $\tilde{\mathcal{O}}(n^{2.5}\sqrt{T})$ regret in the adversarial case. The best known upper bounds with explicit algorithms are $\tilde{\mathcal{O}}(n^{9.5}\sqrt{T})$ (Bubeck et al., 2017) and $\mathcal{O}(nT^{0.75})$ (Flaxman et al., 2005) for Lipschitz, convex functions in the adversarial case[3]. The regret bound

[3]After the publication (Karabag et al., 2021a) of the results provided in this chapter, Lattimore and Gyorgy (2021) provided a $\tilde{\mathcal{O}}(n^{4.5}\sqrt{T})$ expected regret bound for Lipschitz, convex functions in the stochastic case.

$\mathcal{O}(n^{3.75}T^{0.75})$ that we provide is better than $\tilde{\mathcal{O}}(n^{9.5}\sqrt{T})$ regret bound of (Bubeck et al., 2017) if $T = o(n^{23})$. Our result differs from (Flaxman et al., 2005) in that our algorithm succeeds with high probability whereas the algorithm given in (Flaxman et al., 2005) succeeds in expectation.

## 4.2 Preliminaries

The unit vectors in $\mathbb{R}^n$ are $e_1, \ldots, e_n$. Let $S$ be a set of vectors in $\mathbb{R}^n$. $Proj_S(x)$ denotes the orthogonal projection of $x$ onto the span of $S$ and $Proj_{S^\perp}(x)$ denotes the orthogonal projection of $x$ onto the complement space of the span of $S$. The angle between $x$ and $y$ is $\angle(x, y)$. $I$ denotes the identity matrix. The maximum and minimum eigenvalues of a square matrix $A$ is denoted by $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$, respectively. The boundary of a set $D \in \mathbb{R}^n$ is denoted by $Bd(D)$. The convex hull of a set $D$ of points is denoted by $Conv(D)$. With a slight abuse of notation, we use 0 to denote the origin, i.e., $[0, \ldots, 0]^\top \in \mathbb{R}^n$.

A convex function $f : C \to \mathbb{R}$ is said to be *L-Lipschitz* if $\|f(x) - f(y)\| \leq L \|x - y\|$ for all $x, y \in C$. A differentiable convex function $f : C \to \mathbb{R}$ is said to be *$\beta$-strongly smooth* if $|f(y) - f(x) - \langle \nabla f(x), y - x \rangle| \leq \beta \|y - x\|^2 / 2$ for all $x, y \in C$.

A *right circular cone* in $\mathbb{R}^n$ with *semi-vertical angle* $\theta \in [0, \pi/2]$ and *direction* $v \in \mathbb{R}^n$ is $\mathcal{F}(v, \theta) = \{w | W \in \mathbb{R}^n, \angle(v, w) \leq \theta\}$.

A *ball* in $\mathbb{R}^n$ is $\mathcal{B}(r, x_0) = \{x | \|x - x_0\| \leq r\}$ where $x_0 \in \mathbb{R}^n$ and $r \geq 0$. The *circumscribing ball* $\underline{\mathcal{B}}_C = \mathcal{B}(r^*, x_0^*)$ of a compact convex set $C$ satisfies $r^* = \min_{r^*, x_0} r$ where $C \subseteq \mathcal{B}(r, x_0)$. The *inscribed ball* $\overline{\mathcal{B}}_C = \mathcal{B}(r^*, x_0^*)$ of a compact convex set $C$ satisfies $r^* = \max_{r^*, x_0} r$ where $\mathcal{B}(r, x_0) \subseteq C$. The *radius* $R_C$ of a compact convex set $C$ is equal to the the radius of the circumscribing ball, i.e., $R_C = \min_{y \in C} \max_{x \in C} \|x - y\|$.

An *ellipsoid* in $\mathbb{R}^n$ is $\mathcal{E}(A, x_0) = \left\{x | (x - x_0)^T A^{-1} (x - x_0) \leq 1\right\}$ where $x_0 \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n \times n}$ is a positive definite matrix. The *isotropic transformation* $T_{A, x_0}$ of an

ellipsoid $\mathcal{E}(A, x_0)$ is $T_{A,x_0}(x) = A^{-1/2}(x - x_0)\sqrt{\lambda_{\max}(A)}$. The isotropic transformation repositions the ellipsoid at the origin and stretches the ellipsoid such that it becomes a hypersphere whose radius is equal to the largest radius of the ellipsoid. The inverse of $T_{A,x_0}$ is $T^{-1}_{A,x_0}(x) = A^{1/2}x/\sqrt{\lambda_{\max}(A)} + x_0$. With an abuse of notation we use $T_{A,x_0}(D)$ to denote the set $\{T_{A,x_0}(x)|x \in D\}$. The *circumscribing ellipsoid* $\underline{\mathcal{E}}_C = \mathcal{E}(A^*, x_0^*)$ of a compact convex set $C$ satisfies $\det(A^*) = \min_{A,x_0} \det(A)$ where $C \subseteq \mathcal{E}(A, x_0)$.

A $\sigma^2$-*subgaussian* random variable $X$ with mean $\mu$ satisfies $\Pr(|X - \mu| > t) \leq 2\exp\left(-t^2/(2\sigma^2)\right)$ for all $t > 0$.

## 4.3 Optimization Using Sub-Zeroth-Order Oracles

We consider the minimization of a $\beta$-smooth, $L$-Lipschitz, convex function $f$ on a compact, convex set $C \subseteq \mathbb{R}^n$ where $x^*$ denotes a minimizer of $f$. We assume $x^*$ is an interior point of $C$ such that $\mathcal{E}(\varepsilon I/L, x^*) \subseteq C$, where $\varepsilon$ is the desired suboptimality gap. This assumption is included for simplicity, but can be removed by considering a near-optimal interior point with a sufficiently large neighborhood. Such a point is guaranteed to exist after the isotropic transformation. We also assume $n \geq 2$, but the algorithms that we present generalize to the one-dimensional setting.

### 4.3.1 Sub-Zeroth-Order Oracles

The first oracle we consider is the directional-preference oracle which outputs a binary value indicating whether the function is increasing on the queried direction at the queried point. The *directional-preference oracle* $\psi^{DP} : C \times \mathbb{R}^n \to \{-1, 1\}$ is a function such that $\psi^{DP}(x, y) = -1$ if $\langle \nabla f(x), y \rangle < 0$, and $\psi^{DP}(x, y) = 1$ otherwise.

We also consider the comparator oracle, which compares the function at a pair of query points. The *comparator oracle* $\psi^C : C \times C \to \{-1, 1\}$ is a function such that $\psi^C(x, y) = -1$ if $f(x) \geq f(y)$, and $\psi^C(x, y) = 1$ otherwise. The comparator oracle is similar to the directional-preference oracle in that $\psi^C(x, x + ky)$ approaches $\psi^{DP}(x, y)$ in the limit as $k$ approaches zero, i.e., $\lim_{k\to 0^+} \psi^C(x, x + ky) = \psi^{DP}(x, y)$

for all $x \in C$ and $y \in \mathbb{R}^n$.

The *noisy value oracle* $\psi^{NV} : C \to \mathbb{R}$ outputs the function value plus a $\sigma^2$-subgaussian noise, i.e., $\psi^{NV}(x) = f(x) + Z$ for all $x \in C$, where $Z$ is a $\sigma^2$-subgaussian random variable with zero mean.

In addition to the sub-zeroth-order oracles, we also consider the zeroth-order value oracle as preliminary step for the noisy-value oracle. The *value oracle* $\psi^V : C \to \mathbb{R}$ outputs the function value at the queried point, i.e., $\psi^V(x) = f(x)$ for all $x \in C$.

### 4.3.2 Ellipsoid Method with Approximate Gradients

In this section, we provide optimization algorithms that employ the sub-zeroth-order oracles. We use a variation of the ellipsoid method (Shor, 1972; Yudin and Nemirovskii, 1976) that uses the approximately correct gradient direction. The ellipsoid method begins each iteration with an ellipsoid containing an optimal point, it then computes the function's gradient at the ellipsoid center and removes all points from the feasible set that lie along an ascent direction. The remaining points in the set are then enclosed in the minimum volume circumscribing ellipsoid, which is used as the starting ellipsoid in the next iteration. The volume of the generated ellipsoid decreases in each iteration. For a Lipschitz, convex function, this method is guaranteed to output a near optimal solution in a finite number of iterations.

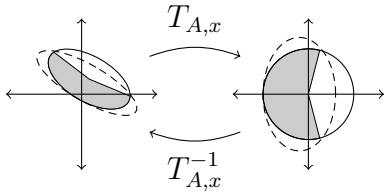



Figure 4.1: Illustrations of the ellipsoid cuts. The original coordinates are on the left and the isotropic coordinates on the right. The dashed ellipsoids enclose the shaded regions that are the possible descent directions.

While the information on the gradient direction is sufficient to apply the classical ellipsoid method, computing the exact gradient direction would require in-

finitely many queries to the sub-zeroth-order oracles. On the other hand, if the semi-vertical angle of the cone of possible gradient directions is small enough, i.e., less than $\sin^{-1}(1/n)$, in the isotropic coordinates, one can still find an ellipsoid with a smaller volume that contains all possible descent directions and the optimal solution.

**Lemma 4.1.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable, convex function. For $\theta \in [0, \sin^{-1}(1/n)]$ and $p \in \mathbb{R}^n$, if $\nabla f(0) \in \mathcal{F}(p, \theta)$, then $f(x') \geq f(0)$ for all $x' \in \mathcal{E}(I, 0) \cap \{x | \langle p/\|p\|, x\rangle > \sin\theta\}$, and there exists an ellipsoid $\mathcal{E}^*$ such that $\mathcal{E}^* \supseteq \mathcal{E}(I, 0) \cap \{x | \langle p/\|p\|, x\rangle \leq \sin\theta\}$ and*

$$\frac{Vol(\mathcal{E}^*)}{Vol(\mathcal{E}(I, 0))} = \left(\frac{n^2(1 - \sin^2(\theta))}{n^2 - 1}\right)^{(n-1)/2} \frac{n(1 + \sin(\theta))}{n + 1}$$

*If $\theta = \sin^{-1}(1/(2n))$, then*

$$Vol(\mathcal{E}^*) \leq Vol(\mathcal{E}(I, 0))e^{-\frac{1}{8(n+1)}} < Vol(\mathcal{E}(I, 0)).$$

Lemma 4.1 shows that if the semi-vertical angle is small enough, there exists an ellipsoid with a smaller volume that contains the intersection of the possible descent directions and the initial ellipsoid as shown in Figure 4.1. Since the isotropic transformation is affine, it preserves the ratio of volumes. Thus, there also exists an ellipsoid with a smaller volume in the original coordinates as shown in Figure 4.1.

We need to approximately estimate the direction of the gradient in order to employ Lemma 4.1. For the value and the noisy-value oracles, we can estimate the direction of the gradient by sampling the function on a fixed set of basis vectors. However, to estimate the gradient direction using the comparator and directional-preference oracles, we need to successively select different collections of vectors along which to sample the function. In the following two sections, we describe in detail how to estimate the direction of the gradient using the sub-zeroth-order oracles, and how to use these estimations for optimization.

#### 4.3.2.1 Optimization Using the Directional-Preference Oracle

For the directional preference oracle, we can estimate direction of the gradient by iteratively sampling the function along different sets of basis vectors. Consider

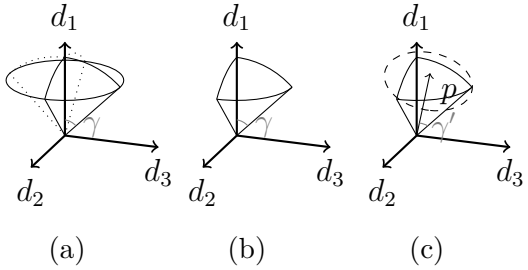



Figure 4.2: Illustrations the gradient pruning method by directional-preferences. ($a$) The cone $\mathcal{F}(d_1, \gamma)$ is the possible gradient directions. ($b$) The quarter cone is the possible gradient directions after the queries. ($c$) The dashed cone $\mathcal{F}(p, \gamma')$ overapproximates possible gradient directions.

Figure 4.2 as an example. Assume that the gradient $\nabla f(x)$ lies in $\mathcal{F}(d_1, \gamma)$ shown in Figure 4.2a. We can use $\psi^{DP}(x, d_2)$ and $\psi^{DP}(x, d_3)$ to prune the direction estimation. The query directions slice the $n$-dimensional space into $2^n$ hyperoctants that are symmetric around the direction of the cone. The query results determine the hyperoctant that the gradient lies in. For example, if $\psi^{DP}(x, d_2) = 1$ and $\psi^{DP}(x, d_3) = 1$, the gradient lies in the quarter cone given in Figure 4.2b. Before the next set of queries, we limit the possible set of gradient directions with $\mathcal{F}(p, \gamma')$ such that $\gamma' < \gamma$ as shown in Figure 4.2c.

**Lemma 4.2.** *Let* $\gamma \in (0, \pi/2]$, $d_1 = e_1$, $d_i = \cos(\gamma)e_1 + \sin(\gamma)e_i$, *for all* $i \in \{2, \ldots, n\}$, $p = \sum_{i=1}^{n} d_i$, *and* $\gamma' = \cos^{-1}(\langle p, d_2 \rangle / \|p\|))$. *Then,* $\mathcal{F}(p, \gamma') \supseteq \mathcal{F}(d_1, \gamma) \cap \{x | x_i \geq 0\}$ *and* $\sin(\gamma')/\sin(\gamma) \leq \sqrt{n-1}/\sqrt{n}$.

Lemma 4.2 shows that if we choose the direction of the new cone as the average of the extreme points of the intersection of the previous cone and the hyperoctant as in Figure 4.2c, then the semi-vertical angle of the cone of possible gradient directions is a fraction of the previous angle depending on the number of dimensions. For the directional-preference and the comparator oracles, we repeat this process until the cone of possible gradient directions is sufficiently small, i.e., less than $\sin^{-1}(1/(2n))$.

Algorithm 8 obtains a near-optimal solution for a given smooth, Lipschitz, convex function. At each iteration, we estimate the gradient direction using the direction

pruning algorithm PD-DP, which implements the procedure described above. After the gradient direction estimation, we remove the ascent directions from the feasible set and proceed to the next iteration by enclosing the feasible set using an ellipsoid.

In the classical ellipsoid method, the output is the ellipsoid center with the smallest function value. The directional-preference oracle cannot compare the function values for a given pair of points, $x_l$ and $x_r$. However, we can use the bisection method to find a point $x'$ such that $f(x') \leq \min(f(x_l), f(x_r)) + \delta$ for a given $\delta$. Since the function is Lipschitz, the search stops after a finite number of iterations. To find a point whose function value is close to the function value of the optimal ellipsoid center, we can remove $x^l$ and $x^r$ from the set of candidate points and add $x'$ to the set of the set of candidate points. Hence, the sample complexity of finding a point $x''$ such that $f(x'') \leq \min_{x \in X} f(x) + \varepsilon/2$ is linear in the size of $X$. The function COMPARE-DP implements the bisection search method on a given set $X$.

**Algorithm 8:** The optimization algorithm OPTIMIZE-DP$(X, \psi^{DP})$ for the directional preference oracle

Find $\underline{\mathcal{E}}_C = \mathcal{E}(A^{(k)}, x^{(1)})$ of $C$.
Set $X = \{x^{(1)}\}$, $C^{(1)} = C$, $K = \left\lceil 8n(n+1)\log\left(\frac{2R_C L}{\varepsilon}\right) + 1 \right\rceil$.
**for** $k = 1 \dots K$ **do**
  Set $p = \text{PD-DP}\left(\psi^{DP}, x^{(k)}, \sin^{-1}\left(1/(2n)\right), A^{(k)}\right)$.
  Set $C^{(k+1)} = C^{(k)} \cap \mathcal{E}(A^{(k)}, x^{(k)}) \cap T^{-1}_{A^{(k)}, x^{(k)}}\left(\{x |\, \langle p/\left\|p\right\|, x\rangle \leq 1/(2n)\}\right)$.
  Find $\underline{\mathcal{E}}_{C^{(k+1)}} = \mathcal{E}(A^{(k+1)}, x^{(k+1)})$ of $C^{(k+1)}$.
  Set $X = X \cup \{x^{(k+1)}\}$.
**return** COMPARE-DP$(X, \psi^{DP}, \varepsilon/2)$.

**Theorem 4.1.** *Let* $K = \left\lceil 8n(n+1)\log\left(\frac{2R_{CL}}{\varepsilon}\right)\right\rceil$. *For an L-Lipschitz,* $\beta$*-smooth convex function* $f : C \to \mathbb{R}$, *Algorithm 8 makes at most*

$$nK\left\lceil 2n\log(2n)\right\rceil + K\log_2\left(\frac{R_C L(K+1)}{\varepsilon}\right)$$

*queries to* $\psi^{DP}$ *and the output* $x'$ *of Algorithm 8 satisfies* $f(x') \leq \min_{x \in C} f(x) + \varepsilon$.

**Algorithm 8: Function** PD-DP$(x, \theta, T_{A,x})$

$p = e_1$, $r = 1$, $\gamma = \pi/2$.
**while** $\gamma > \theta$ **do**
  Find $d_i$ such that $d_1 = p$, $d_i \perp d_j$ for all $i \neq j \in [n]$, and $\|d_i\| = 1$ for all $i \in [n]$.
  Query $\psi^{DP}(x, A^{-1/2}d_1), \ldots, \psi^{DP}(x, A^{-1/2}d_n)$.
  Set $w_1 = d_1$ and for all $i \in \{2, \ldots, n\}$, set $w_i = d_1 \cos(\gamma) + d_i \psi^{DP}(x, A^{-1/2}d_i) \sin(\gamma)$.
  Set $p = (\sum_{i=1}^n w_i/n) \,/\, \|\sum_{i=1}^n w_i/n\|$,
  Set $\gamma = \cos^{-1}(\langle p, w_2 \rangle)$.
  Set $r = \sin^{-1}(\gamma)$.
**return** $p$.

The sample complexity and the correctness of Algorithm 8 follows from Lemmas 4.1 and 4.2. The sample complexity using the directional-preference oracle is $\tilde{\mathcal{O}}(n^2)$ of the classical ellipsoid algorithm. An invetable factor of $\mathcal{O}(n)$ is required to query the function in all dimensions, i.e., to slice the cone of the possible gradient directions into hyperoctants. By Lemma 4.2, a factor of $\mathcal{O}(n \log(n))$ is due to the number of iterations of the gradient pruning algorithm. While the gradient pruning method is optimal when the semi-vertical angle of the possible gradient directions is large, it is suboptimal when the semi-vertical angle is close to 0. One may improve the dependency of $\mathcal{O}(n \log(n))$ by treating this small angle regime differently. We remark that optimization using the directional-preference oracle is still possible in the absence of smoothness. One can use the same optimization method with an oracle that outputs the sign of an arbitrary directional subgradient.

#### 4.3.2.2 Optimization Using the Comparator Oracle

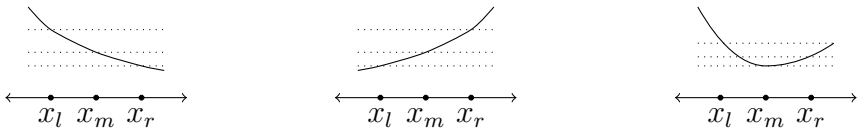
The optimization algorithm that we provide for the comparator oracle is similar to the optimization algorithm for the directional preference oracle. To solve the optimization problem, we begin by using comparisons to infer the sign of the directional derivative, i.e., we use the comparator oracle $\psi^C$ to infer the directional-

**Algorithm 8: Function** COMPARE-DP$(X, \varepsilon)$

Set $X^* = X$ and $m = |X|$.
**while** $|X^*| > 1$ **do**
  Arbitrarily pick $x^1, x^2 \in X$ such that $x^1 \neq x^2$.
  Set $X^* = X^* \setminus \{x^1, x^2\}$.
  Set $x^l = x^1$ and $x^r = x^2$.
  **while** $\|x^r - x^l\| \leq 2\varepsilon/(Lm)$ **do**
    Query $\psi^{DP}((x^r + x^l)/2, (x^r - x^l)/2)$.
    **if** $\psi^{DP}((x^r + x^l)/2, (x^r - x^l)/2) = 0$ **then**
      $x^l = (x^r + x^l)/2$.
    **else**
      $x^r = (x^r + x^l)/2$.
  $X^* = X^* \cup \{(x^r + x^l)/2\}$
**return** $x^* \in X^*$.

preference oracle $\psi^{DP}$. Then, we approximately find the direction of the gradient using the signs of the directional derivatives.

Suppose function $g$ is in isotropic coordinates and we compare the function values at three points on a line, $x_r$, $x_m$, and $x_l$. We can get the directional derivative information at the middle point if the values of the function at $x_r$, $x_m$, and $x_l$ are ordered as in Figures 4.3a and 4.3b. If the queried points are not ordered, i.e., the function value at $x_m$ is lower than or equal to the function values at both $x_r$ and $x_l$ as in Figure 4.3c, the sign of the directional derivative is unknown at $x_m$. Function FDD-C takes the isotropic transformation information and outputs directional derivative information.




(a) (b) (c)

Figure 4.3: Possible orderings for a convex function at three points on a line.

**Algorithm 8: Function** FDD-C$(A, x_0, d, t)$

Query $\psi^C(x - tA^{-1/2}d, x), \psi^C(x, x + tA^{-1/2}d)$.
**if** $f(x - tA^{-1/2}d) \leq f(x) \wedge f(x) \leq f(x + tA^{-1/2}d)$ **then return** 1.
**else if** $f(x - tA^{-1/2}d) < f(x) \wedge f(x) < f(x + tA^{-1/2}d)$ **then return** $-1$.
**else return** *unknown*.


(a) (b)

Figure 4.4: Possible cases for Algorithm 9. $(a)$ The uncertainty sets for the unknown direction, $d_1$, and the known directions, $d_2$ and $d_3$. $(b)$ Two possible cases for the gradient estimation in Algorithm 9.

In cases when the sign of the directional derivative is unknown, we can use the smoothness of the objective function to bound the magnitude of the derivative as follows. In the case shown in Figure 4.3c, there exists a point $x'$ such that $\left\langle \nabla g(x'), x^r - x^l \right\rangle = 0$ and $x' = \alpha x^r + (1 - \alpha)x^l$ for some $\alpha \in [0, 1]$. Due to the smoothness property, we have $\left\langle \nabla g(x^m), x^r - x^l \right\rangle \leq \beta \left\| x^r - x^l \right\| /2$.

The function PD-C prunes the cone of the possible gradient directions by inferring the directional derivative information on different sets of basis vectors. At each iteration, the algorithm starts with a cone of possible gradient directions. Based on the query results the algorithm identifies the unknown directions $UD$ and finds an approximate direction for the projection $Proj_{UD^\perp}(\nabla g(x))$ of the gradient onto the span of the known directions. In the next iteration, the algorithm uses the $Proj_{UD^\perp}(\nabla g(x))$ as the direction of the cone of the possible gradient directions. When the semi-vertical angle of the cone of the possible gradient directions is sufficiently small or the number of unknown directions is equal to the number of dimensions, the function returns the estimation for the direction of the gradient.

Algorithm 9, used for optimization with the comparator oracle, has two steps

**Algorithm 8: Function** PD-C$(x, \theta, A, t)$

Set $r = 1$, $\gamma = \pi/2$, $m = 0$, $UD = \emptyset$, $p = e_1$.
**while** $\gamma > \theta \wedge m < n$ **do**
  Set $\{d_1, \ldots, d_m\} = UD$.
  Find $d_i$ such that $d_{m+1} = p$, $d_i \perp d_j$ for all $i \neq j \in [n]$, and $\|d_i\| = 1$ for all $i \in [n]$.
  Set $\psi^{DP}(x, A^{-1/2}d_i) =$ FDD-C$(A, x_0, d, t)$ for all $i \in [n]$.
  **if** $\exists i \in \{m+1, \ldots, n\}$, *such that* $\psi^{DP}(x, A^{-1/2}d_i) = unknown$ **then**
    Set $UD = UD \cup d_i$, and $m = m + 1$.
  **else**
    Set $w_i = d_{m+1}\psi^{DP}(x, A^{-1/2}d_{m+1})\cos(\gamma) + d_i\psi^{DP}(x, A^{-1/2}d_i)\sin(\gamma)$ for all $i \in [n]$.
    Set $p = \left(\sum_{i=m+1}^{n} w_i/n\right) / \left\|\sum_{i=m+1}^{n} w_i/n\right\|$, $\gamma = \cos^{-1}(\langle p, w_{m+2}\rangle)$, $r = \sin^{-1}(\gamma)$.
**if** $m \neq n$ **then return** $p$, **else return** $e_1$.

in each iteration. In the first step, the algorithm identifies a candidate approximate gradient direction in the isotropic coordinates using the direction pruning function PD-C. In the second step, the algorithm performs a cut as in the classical ellipsoid method.

In order to find a near-optimal point, the algorithm exploits the fact that the direction of the projection of the gradient onto the linear subspace $Span(UD)^{\perp}$ of $\mathbb{R}^n$ is approximately correct, and the magnitude of the projection $\|Proj_{UD}(\nabla g(x))\|$ of the gradient onto the complement subspace is small. For example, in Figure 4.4a, direction $d_1$ is the unknown direction and $\|Proj_{UD}(\nabla g(x))\| \leq \delta$. Directions $d_2$ and $d_3$ are the known directions and $\angle(Proj_{UD}(\nabla g(x)), p) \leq \gamma$. There are two possible cases:

1. The angle between $Proj_{UD^{\perp}}(\nabla g(x))$ and $\nabla g(x)$ is sufficiently small.
2. The angle between $Proj_{UD^{\perp}}(\nabla g(x))$ and $\nabla g(x)$ is not sufficiently small.

Case 1 happens if $\|Proj_{UD^{\perp}}(\nabla g(x))\|$ is large enough. In this case, the estimation for the direction of the gradient $\nabla g(x)$ is approximately correct since the estimation

**Algorithm 9:** The optimization algorithm OPTIMIZE-C($\varepsilon$) for the comparator oracle

Set $C^{(1)} = C$. Find $\underline{\mathcal{E}}_{C^{(1)}} = \mathcal{E}(A^{(1)}, x^{(1)})$ of $C^{(1)}$.

Set $X = \{x^{(1)}\}$, $K = \left\lceil 8n(n+1)\log\left(\frac{R_C L}{\varepsilon}\right)\right\rceil$, $\kappa = \max\left(\frac{4}{4n-\sqrt{2}n\sqrt{\frac{4n^2-1}{4n^2}}}, 1\right)$.

**for** $k = 1 \dots K$ **do**

  Set $t^{(k)} = \frac{\min(\varepsilon, \sqrt{\lambda_{\max}(A^k)})}{\kappa n^{5/2}\max(\beta,1)\max(R_C,1)}$.

  Set $p = \text{PD-C}\left(x^{(k)}, \sin^{-1}\left(\frac{1}{2\sqrt{2}n}\right), A^{(k)}, t^{(k)}\right)$.

  Set $C^{(k+1)} = C^{(k)} \cap \mathcal{E}(A^{(k)}, x^{(k)}) \cap T^{-1}_{A^{(k)}, x^{(k)}}(\{x | \langle p/\|p\|, x\rangle \le 1/(2n)\})$.

  Find $\underline{\mathcal{E}}_{C^{(k+1)}} = \mathcal{E}(A^{(k+1)}, x^{(k+1)})$ of $C^{(k+1)}$.

  Set $X = X \cup \{x^{(k+1)}\}$.

Find $x' = \min_{x \in X} f(x)$ using $\psi^C$.

**return** $x'$.

$p$ for the direction of $Proj_{UD^\perp}(\nabla g(x))$ is approximately correct. In this case, the ellipsoid algorithm proceeds normally. For example, if $\nabla g(x) = v_2$ in Figure 4.4b, then $\angle(\nabla g(x), p)$ is small enough, say less than $\sin^{-1}(1/(2n))$. If Case 2 happens, the gradient approximation is not accurate, i.e., $\angle(\nabla g(x), p)$ might be larger than $\sin^{-1}(1/(2n))$. However, if Case 2 happens, it implies that $\|Proj_{UD^\perp}(\nabla g(x))\|$ is not large enough compared to $\|Proj_{UD}(\nabla g(x))\|$. Consequently, the magnitude $\|\nabla g(x)\|$ of the gradient is not large, say less than $\varepsilon/(nR_C)$. For example, if $\nabla g(x) = v_1$ in Figure 4.4b, then $\|\nabla g(x)\|$ is small enough. We carefully choose the sampling distance so that the current ellipsoid center $x$ is near optimal if $\|Proj_{UD^\perp}(\nabla g(x))\|$ is not large enough. Algorithm 9 is agnostic to whichever case happens: The algorithm always assumes that the direction estimation approximately correct. However, the output point is near optimal since we compare the ellipsoid centers and output the best point before the termination.

**Theorem 4.2.** *Let* $K = \left\lceil 8n(n+1)\log\left(\frac{R_C L}{\varepsilon}\right)\right\rceil$. *For an L-Lipschitz,* $\beta$*-smooth convex function* $f : C \to \mathbb{R}$*, Algorithm 9 makes at most*

$$2n\left\lceil 2n\log(2\sqrt{2}n) + n\right\rceil K + K$$

*queries to* $\psi^C$ *and the output* $x'$ *of Algorithm 9 satisfies* $f(x') \leq \min_{x \in C} f(x) + \varepsilon$.

Theorem 4.2 shows that using the comparator oracle we can find a near optimal point with $\tilde{\mathcal{O}}(n^4)$ queries, which is at the same order with the sample complexity of optimization using the directional preference oracle. We also remark that while the smoothness of the function is required to determine the sampling distance, the sample complexity is not dependent on the smoothness constant.

#### 4.3.2.3 Optimization Using the Value Oracle

The value oracle is more informative than the comparator and directional-preference oracles; we can query the function in orthogonal directions near the center point and estimate the gradient. In the limit, i.e., the sampling distance goes to 0, the gradient estimate converges to the true gradient.

Under the smoothness assumption, we can get a provably good approximation of the gradient with a finite sampling distance. Let $g$ be a $\beta$-smooth function in the isotropic coordinates. Formally, we have $g(x) - g(y) - \beta \|x - y\|^2 /2 \leq \langle \nabla g(x), x - y \rangle \leq g(x) - g(y) + \beta \|x - y\|^2 /2$.

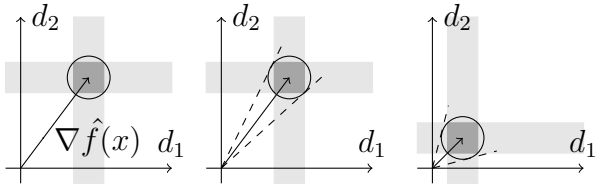



Figure 4.5: Illustrations of possible cases for the gradient $\nabla g(x)$. $(a)$ The light gray stripes are the uncertainty sets for the directional derivatives. The dark gray squares are the uncertainty sets for $\nabla g(x)$ and, the circles overapproximate the uncertainty sets. In $(b)$ and $(c)$, the angle between the empirical gradient $\nabla \hat{g}(x)$ and the dashed lines is the maximum angle between $\nabla \hat{g}(x)$ and $\nabla g(x)$.

Assume that we sample the points that have a distance of $d$ from the center point in the isotropic coordinates. After $n + 1$ queries we can bound the gradient

in a hypercube with the edge length of $\beta d$. The hypercube can be contained in a hypersphere with radius $\beta\sqrt{n}d/2$. For example, consider the case shown in Figure 4.5a. Let $\nabla\hat{g}(x)$ be the empirical gradient estimate, i.e., the center of the hypercube. We can have two cases, either the gradient is in $\mathcal{F}(0, \sin^{-1}(1/(2n))$ or the magnitude of the gradient is smaller than $(2n+1)\sqrt{n}\beta d$. The former and latter cases are illustrated in Figures 4.5b and 4.5c, respectively. In the latter case, if $d$ is sufficiently small, i.e., lower than $\varepsilon/((2n+1)\sqrt{n}\beta R_C)$, the center point is near optimal. Overall, the sample complexity of optimization using the value oracle with the ellipsoid method is $\tilde{\mathcal{O}}(n^3)$. (Nemirovsky and Yudin, 1983) provided a randomized optimization algorithm that succeeds with probability at least $1-\delta$ (where $\delta$ can be chosen to be arbitrarily small) and has a sample complexity of $\tilde{\mathcal{O}}(n^3)$ for Lipschitz continuous, convex functions. With an additional smoothness assumption, the method that we describe deterministically succeeds with the same complexity. We also remark that these bounds are inferior to the $\tilde{\mathcal{O}}(n^2)$ sample complexity result given in (Lee et al., 2018).

#### 4.3.2.4 Optimization Using the Noisy-Value Oracle

For the noisy-value oracle, we can use the same gradient direction estimation method as in the value oracle. Different from the value oracle, we also need to consider the stochasticity of the oracle outputs since the empirical estimate $(\psi^{NV}(x) - \psi^{NV}(y))/\left\|x-y\right\|$ of directional derivative is a $2\sigma^2/\left\|x-y\right\|^2$-subgaussian random variable.

We need $\tilde{\mathcal{O}}(\sigma^2/(\beta^2\left\|x-y\right\|^4))$ samples to obtain a confidence interval of $\mathcal{O}(\beta\left\|x-y\right\|)$ for the directional derivative estimate. By letting $\left\|x-y\right\| = \mathcal{O}\left(\varepsilon/(2(2n+1)\sqrt{n}\beta R_C)\right)$, we can ensure either that the ellipsoid method proceeds normally or that the current ellipsoid center is near optimal. Overall, the sample complexity of optimization using this method is $\tilde{\mathcal{O}}(n^{13}/\varepsilon^4)$. We remark that Belloni et al. (2015) provived an algorithm that has $\tilde{\mathcal{O}}(n^{7.5}/\varepsilon^2)$ sample complexity and $\varepsilon$-suboptimality in expectation.

**Algorithm 10:** The low regret algorithm REGRET-NV$(T, \delta)$ for the noisy value oracle

Set $C^{(1)} = C$. Find $\underline{\mathcal{E}}_{C^{(1)}} = \mathcal{E}(A^{(1)}, x^{(1)})$. Set $X = \{x^{(1)}\}$.
Set $K = \lceil 8n(n+1)\log(2R_C L T^{0.25})\rceil$, $\tau = \left\lceil 32\sigma^2 n^4 \log\left(\frac{2}{\delta'}\right)\right\rceil$, $\delta' = \frac{\delta}{4nK\log_{16}\left(\frac{15T}{2n}\right)}$.
**for** $k = 1, \ldots, K$ **do** // Phase 1
  Set $d = \frac{\min\left(\sqrt{\lambda_{\max}(A^{(k)})}, 1\right)}{2n}$.
  Set $\Delta = \frac{d\left(2+\beta\lambda_{\max}(A^{(k)})\right)}{2\lambda_{\max}(A^{(k)})}$.
  **for** $i = 0, 1, \ldots$ **do** // Case 2
    Set $d_i = d/2^i$, $\Delta_i = \Delta/2^i$, $\tau_i = 2^{4i}\tau$
    Query $\tau_i$ times $\psi^{NV}(x^{(k)})$ and $\psi^{NV}(T^{-1}_{A^{(k)},x^{(k)}}(de_j))$ for all $i \in [n]$.
    For every query point $x$, set $\hat{\psi}^{NV}(x)$ as the mean of queries for point $x$.
    Estimate the gradient $p$ using the mean values $\hat{\psi}^{NV}(x)$.
    **if** $(\|p\| > \sqrt{n}\Delta_i) \wedge \left(\sin^{-1}\left(\frac{\sqrt{n}\Delta_i}{\|p\|}\right) \leq \sin^{-1}\left(\frac{1}{2n}\right)\right)$ **then** // Case 1
      Set $C^{(k+1)} = C^{(k)} \cap \mathcal{E}(A^{(k)}, x^{(k)}) \cap T^{-1}_{A,x^{(k)}}\left(\{x \mid \langle p/\|p\|, x\rangle \leq 1/(2n)\}\right)$.
      Find $\underline{\mathcal{E}}_{C^{(k+1)}} = \mathcal{E}(A^{(k+1)}, x^{(k+1)})$.
      Set $X = X \cup \{x^{(k+1)}\}$.
      **break**
For all $x \in X$, query $\psi^{NV}(x)$, $\left\lceil 32\sigma^2\sqrt{T}\log\left(\frac{2(K+1)}{\delta}\right)\right\rceil$ times. Set $x'$ to the point with the highest empirical mean. // Phase 2
Repeatedly query $\psi^{NV}(x')$. // Phase 3

## 4.4 A Sublinear Regret Algorithm for the Noisy-Value Oracle

The regret of an optimization algorithm measures the performance of the algorithm during optimization. Define $x_i$ as the query point at time $i$, and $\hat{f}(x_i)$ as the output of the oracle. For a given number of queries $T$, the regret of an algorithm $\mathcal{A}$ is $\mathcal{R}^{\mathcal{A}}(T) = \sum_{t=1}^{T} f\left(\mathcal{A}\left(h_t\right)\right) - \sum_{t=1}^{T} f(x^*)$ where $h_t = (x_0, \hat{f}(x_0)) \ldots (x_{t-1}, \hat{f}(x_{t-1}))$ is the history of the algorithm. As in the previous section, we assume that $x^*$ is an interior point of $C$ such that $\mathcal{E}(T^{-0.25}I/L, x^*) \subseteq C$.

The optimization algorithm mentioned in the previous section incurs sublinear

regret when $\varepsilon = \mathcal{O}(T^{-0.2})$. However, this approach yields a regret that has high order dependencies on the other parameters since the algorithm only relies on finding a near-optimal point with a regret of $\mathcal{O}(T^{-0.2})$ if the gradient estimation fails. We give Algorithm 10 that incurs $\tilde{\mathcal{O}}(n^{3.75}R_C\sqrt{\beta\sigma}T^{0.75})$ regret with high probability when $T = \Omega(n^3L^{4/3}\sigma^2 + L^4\sigma^6)$ and $nR_C, \beta, L, \sigma \geq 1$. Different from the optimization algorithm, Algorithm 10 does not find a near-optimal point if the gradient estimation fails. Instead, Algorithm 10 finds a point with a regret that incurs the half of the regret of the previous query point. While this approach increases the number of queries for optimization purposes, it yields a low regret.

Algorithm 10 consists of three phases. In Phase 1, we start by limiting the current convex set with the circumscribing ellipsoid and apply the isotropic transformation. We query the oracle in every dimension at the center of the ellipsoid and at the points that are close to the center. Then, we estimate the gradient within a confidence interval and limit the possible gradient directions to a cone in the isotropic coordinates. There are two possible cases:

1. If the semi-vertical angle of the possible gradient directions is small enough, i.e., less than $\sin^{-1}(1/(2n))$, we cut the current ellipsoid, and start the process from the beginning using the remaining set.

2. If the semi-vertical angle of the possible gradient directions is not small enough, we halve the sampling distance and confidence interval, and start querying with the new sampling distance and confidence interval.

If Case 2 happens, it implies that the gradient at the current ellipsoid center has a small magnitude, and the regret of the next set of queries is low. If Case 1 happens sufficiently many times, then one of the ellipsoid centers is a near optimal point with low regret as in the classical ellipsoid method. After Case 1 happens sufficiently many times, the algorithm proceeds to Phase 2. In this phase, we compare the the ellipsoid

centers and find an ellipsoid center with a low regret, i.e., $\mathcal{O}(T^{-0.25})$. In Phase 3, we repeatedly query the ellipsoid center with a low regret.

**Theorem 4.3.** *Let* $K = \lceil 8n(n+1)\log(2R_C L T^{0.25})\rceil$, $\delta' = \delta/\left(4nK\log_{16}\left(\frac{15T}{2n}\right)\right)$, *and* $\tau = \left\lceil 8\sigma^2\beta^2 n^4 \log\left(\frac{2}{\delta'}\right)\right\rceil$. *For an L-Lipschitz, β-smooth convex function* $f : C \to \mathbb{R}$, *a given failure probability* $\delta > 0$, *and a time horizon* $T$, *Algorithm 10 has a regret of at most* $K\left(R_C L\tau + 5T^{0.75}n^{-0.25}\max(nR_C,1)(1+\beta)\tau^{0.25}\right) + (K+1)\left\lceil 32\sigma^2\sqrt{T}\log\left(\frac{2(K+1)}{\delta}\right)\right\rceil R_C L + T^{0.75}$ *with probability at least* $1-\delta$.

For a given $L$-Lipschitz, $\beta$-smooth function $f : C \to \mathbb{R}$, we can define $f' : C' \to \mathbb{R}$ such that $C' = \left\{x' | x' = x\sqrt{\beta}, x \in C\right\}$ and $f'(\sqrt{\beta}x) = f(x)$ for all $x \in C$. $f'$ is $L/\sqrt{\beta}$-Lipschitz, 1-smooth, and $R_{C'} = \sqrt{\beta}R_C$. If Algorithm 10 operates with the parameters of $f'$, the regret is $\tilde{\mathcal{O}}(n^{3.75}R_C\sqrt{\beta\sigma}T^{0.75})$ when $T = \Omega(n^3 L^{4/3}\sigma^2 + L^4\sigma^6)$ and $nR_C\sqrt{\beta}, L, \sigma \geq 1$.[4]

## 4.5 Proofs for the Technical Results

We use Lemmas 4.1, 4.2, 4.3, and 4.4 for the proofs.

*Proof of Lemma 4.1.* We first show that if $\nabla f(x') \in \mathcal{F}(p,\theta)$, then $f(x') \geq f(x)$ for all $x \in D = \mathcal{E}(I,x') \cap \{x | \langle p/\|p\|, x\rangle \leq \sin\theta\}$. By the convexity of $f$, we have $f(0) \geq f(x) - \langle \nabla f(0), x\rangle$ for all $x \in \mathbb{R}^n$. Since $\nabla f(0) \in \mathcal{F}(p,\theta)$ and $\langle x, y\rangle \geq 0$ for all $x \in \mathcal{F}(p, \pi/2-\theta)$ and $y \in \mathcal{F}(p,\theta)$, we have $f(0) \leq f(x) - \langle \nabla f(0), x\rangle \leq f(x)$ for all $x \in \mathcal{F}(p,\pi/2-\theta)$. Since $\mathcal{E}(I,0) \cap \{x | \langle p/\|p\|, x\rangle > \sin\theta\} \subset \mathcal{F}(p,\pi/2-\theta)$, we have $f(0) \leq f(x) - \langle \nabla f(0), x\rangle \leq f(x)$ for all $x \in \mathcal{E}(I,0) \cap \{x | \langle p/\|p\|, x\rangle > \sin\theta\}$.

By Theorem 2.1 of (Goldfarb and Todd, 1982), there exists an ellipsoid $\mathcal{E}^*$ such that $\mathcal{E}^* \supseteq \mathcal{E}(I,0) \cap \{x | \langle p/\|p\|, x\rangle \leq \sin\theta\}$ and

$$Vol(\mathcal{E}^*) = Vol(\mathcal{E}(I,0))\left(\frac{n^2(1-\sin^2(\theta))}{n^2-1}\right)^{(n-1)/2}\frac{n(1+\sin(\theta))}{n+1} < Vol(\mathcal{E}(I,0)).$$

[4]The publication Karabag et al. (2021a) included the above regret bound. The bound can be improved to $\tilde{\mathcal{O}}(n^{2.25}R_C\sqrt{\beta\sigma}T^{0.75})$ by changing the analysis as described in Remark 4.1.

Setting $\theta = \sin^{-1}(1/(2n))$, we get

$$\frac{Vol(\mathcal{E}^*)}{Vol(\mathcal{E}(I,0))} = \left(\frac{4n^2-1}{4n^2-4}\right)^{(n-1)/2} \frac{2n+1}{2n+2} = \left(1+\frac{3}{4n^2-4}\right)^{(n-1)/2} \left(1-\frac{1}{2n+2}\right).$$

By the inequality $1+x \leq e^x$, we have

$$\frac{Vol(\mathcal{E}^*)}{Vol(\mathcal{E}(I,0))} \leq e^{\frac{3(n-1)}{2(4n^2-4)}} e^{-\frac{1}{2n+2}} = e^{-\frac{1}{8(n+1)}}.$$

■

*Proof of Lemma 4.2 .* We first show that $\mathcal{F}(p,\gamma') \supseteq \mathcal{F}(d_1,\gamma) \cap \{x | x_i \geq 0\}$. It suffices to show that the semi-vertical angle $\gamma'$ of the new cone is larger than the angle between the direction $q$ of the new cone and any enclosed point. Formally, we need to show that $\gamma' \geq \max \cos^{-1}\left(\frac{\langle p,q\rangle}{\|p\|\|q\|}\right)$ where $q \in \mathcal{F}(d_1,\gamma) \cap \{x | x_i \geq 0\}$.

Without loss of generality assume that $q = ad_1 + \sum_{i=2}^{n} \sqrt{1-a^2} b_i d_i$ where $0 \leq a \leq 1$, $\sum_2^n b_i^2 = 1$, and $0 \leq b_i \leq 1$ for all $i \in \{2,\ldots,n\}$. Note that this assumption only limits the scaling of $q$ such that $\|q\| = 1$ and does not affect the maximum angle.

We have

$$\frac{\langle p,q\rangle}{\|p\|\,\|q\|} = \frac{a((n-1)\cos(\gamma)+1)}{n} + \left(\sqrt{1-a^2}\cos(\gamma)\frac{(n-1)\cos(\gamma)+1}{n}\right)\sum_{i=2}^{n} b_i + \frac{\sin^2(\gamma)}{n}\sum_{i=2}^{n} b_i.$$

For a fixed value of $a$, $\frac{\langle p,q\rangle}{\|p\|\|q\|}$ is minimized, i.e., $\cos^{-1}\left(\frac{\langle p,q\rangle}{\|p\|\|q\|}\right)$ is maximized, when $b_i = 1$ for some $i \in \{2,\ldots,n\}$ and $b_j = 0$ for others. In order to find the maximum value of $\cos^{-1}\left(\frac{\langle p,q\rangle}{\|p\|\|q\|}\right)$, without loss of generality we assume that $b_2 = 1$, $b_j = 0$ for all $j \in \{2,\ldots,n\}$. Therefore, there exists $q = ad_1 + \sqrt{1-a^2}d_2$ such that $\cos^{-1}\left(\frac{\langle p,q\rangle}{\|p\|\|q\|}\right)$ is maximized.

Define $q'$ such that $q' = bd_1 + (1-b)d_2$ where $0 \leq b \leq 1$ and $\cos^{-1}\left(\frac{\langle p,q\rangle}{\|p\|\|q\|}\right) = \cos^{-1}\left(\frac{\langle p,q'\rangle}{\|p\|\|q'\|}\right)$. Note that $q'$ is a scaled version of $q$, i.e., $q = q'/\|q'\|$.

We note that

$$\max_{q'} \cos^{-1}\left(\frac{\langle p,q'\rangle}{\|p\|\,\|q'\|}\right) \leq \max_{q'} \cos^{-1}\left(\frac{\langle p,q'\rangle}{\|p\|}\right)$$

since $\cos^{-1}(\alpha)$ is a non-increasing function of $\alpha$. We also note that $\cos^{-1}\left(\frac{\langle p,q'\rangle}{\|p\|}\right)$ is maximized when $\langle p, q'\rangle$ is minimized and $\langle p, q'\rangle$ is a linear function of $b$ on the compact, convex set $0 \leq b \leq 1$. Therefore, there exists a corner point $b \in \{0, 1\}$ such that $\cos^{-1}\left(\frac{\langle p,q'\rangle}{\|p\|}\right)$ is maximized.

For $b = 1$, we have $q' = q = d_1$ and

$$\langle p, d_1\rangle = \frac{1 + (n-1)\cos(\gamma)}{n}.$$

For $b = 0$, we have $q' = q = d_2$ and

$$\langle p, d_2\rangle = \frac{\cos(\gamma) + (n-1)\cos^2(\gamma) + \sin^2(\gamma)}{n} = \frac{1 + \cos(\gamma) + (n-2)\cos^2(\gamma)}{n}$$

for all $i \in \{2, \ldots, n\}$. Note that $\langle p, d_2\rangle \leq \langle p, d_1\rangle$ since $\cos(\gamma) \leq 1$.

We consequently have $\cos^{-1}\left(\frac{\langle p,d_1\rangle}{\|p\|}\right) \leq \cos^{-1}\left(\frac{\langle p,d_2\rangle}{\|p\|}\right)$ and $\cos^{-1}\left(\frac{\langle p,q\rangle}{\|p\|}\right)$ is maximized when $q = d_2$. Therefore,

$$\gamma' = \cos^{-1}\left(\frac{\langle p, d_2\rangle}{\|p\|}\right) = \max_{q'} \cos^{-1}\left(\frac{\langle p, q\rangle}{\|p\|}\right) \geq \max_{q'} \cos^{-1}\left(\frac{\langle p, q\rangle}{\|p\|\|q\|}\right) = \max_{q} \cos^{-1}\left(\frac{\langle p, q\rangle}{\|p\|\|q\|}\right)$$

which implies that

$$\mathcal{F}(p, \gamma') \supseteq \mathcal{F}(d_1, \gamma) \cap \{x | x_i \geq 0\}.$$

We now prove that $\frac{\sin(\gamma')}{\sin(\gamma)} \leq \sqrt{\frac{n-1}{n}}$. We have

$$\begin{aligned}
\frac{\sin(\gamma')}{\sin(\gamma)} &= \frac{\sin\left(\cos^{-1}\left(\frac{\langle p,d_2\rangle}{\|p\|\|d_2\|}\right)\right)}{\sin(\gamma)} \\
&= \frac{\sqrt{1 - \frac{(1+\cos(\gamma)+(n-2)\cos^2(\gamma))^2}{n^2\left(\left(\frac{1+(n-1)\cos(\gamma)}{n}\right)^2 + (n-1)\left(\frac{\sin(\gamma)}{n}\right)^2\right)}}}{\sin(\gamma)} \\
&= \sqrt{\frac{(n-2)^2\cos^2(\gamma) + 2(n-2)\cos(\gamma) + n - 1}{(n-1)(n-2)\cos^2(\gamma) + 2(n-1)\cos(\gamma) + n}}.
\end{aligned}$$

For $\gamma \in (0, \pi/2)$, we have

$$\begin{aligned}
\frac{\partial}{\partial\gamma}\frac{\sin(\gamma')}{\sin(\gamma)} &= \frac{\sin(\gamma)((n-2)\cos(\gamma) + 1)}{((n-1)(n-2)\cos^2(\gamma) + 2(n-1)\cos(x) + n)^2\sqrt{\frac{(n-2)^2\cos^2(\gamma)+2(n-2)\cos(\gamma)+n-1}{(n-1)(n-2)\cos^2(\gamma)+2(n-1)\cos(\gamma)+n}}} \\
&\geq 0,
\end{aligned}$$

i.e., $\frac{\sin(\gamma')}{\sin(\gamma)}$ is a non-decreasing function of $\gamma$.

Since $\frac{\sin(\gamma')}{\sin(\gamma)} = \sqrt{\frac{n-1}{n}}$ when $\gamma = \pi/2$ and $\frac{\sin(\gamma')}{\sin(\gamma)}$ is a non-decreasing function of $\gamma$, we conclude that $\frac{\sin(\gamma')}{\sin(\gamma)} \leq \sqrt{\frac{n-1}{n}}$. ■

**Lemma 4.3.** *Let $C \in \mathbb{R}^n$ be a compact convex set. The circumscribing ellipsoid $\underline{\mathcal{E}}_C = \mathcal{E}(A^*_{\mathcal{E}}, x^*_{0,\mathcal{E}})$ and the radius $R_C$ of $C$ satisfies $\sqrt{\lambda_{\max}(A^*_{\mathcal{E}})} \leq nR_C$.*

*Proof of Lemma 4.3.* Let $C_0$ be the convex set that is the isotropic transformation of $C$, i.e., $C_0 = \{x | T^{-1}_{A^*,x^*_0}(x) \in C\}$. Since $\mathcal{E}(A^*_{\mathcal{E}}, x^*_{0,\mathcal{E}})$ is the circumscribing ellipsoid of $C$, the circumscribing ellipsoid of $C_0$ is $\mathcal{B}(\sqrt{\lambda_{\max}(A^*_{\mathcal{E}})}, 0))$ and equal to the circumscribing ball $\underline{\mathcal{B}}_{C_0}$ of $C_0$. Let $\overline{\mathcal{B}}_{C_0} = \mathcal{B}(r, x_0)$ be the inscribed ball of $C_0$. Since $C_0$ is convex, we have that $\sqrt{\lambda_{\max}(A^*_{\mathcal{E}})} \leq nr$ (Henk, 2012).

We note that the transformation $T_{A^*,x^*_0}$ preserves the distances between two point if the line passing through the points is parallel to the eigenvector that is associated with the largest eigenvalue of $A^*$. Since there exist points $x, y \in \mathcal{B}(r, x_0) \subseteq C_0$ such that $\|x - y\| = r$ and $x - y$ is parallel to the eigenvector that is associated with the largest eigenvalue of $A^*$, there exist two points in $C$ such that the distance between the points is $r$. Therefore, the radius $R_C$ of $C$ satisfies $R_C \geq r$.

By combining $\sqrt{\lambda_{\max}(A^*_{\mathcal{E}})} \leq nr$ and $R_C \geq r$, we get $\sqrt{\lambda_{\max}(A^*_{\mathcal{E}})} \leq nR_C$.

■

**Lemma 4.4.** *Let $C \in \mathbb{R}^n$ be a compact convex set and $\underline{\mathcal{E}}_C = \mathcal{E}(A^*, x^*_0)$ be the circumscribing ellipsoid of $C$. If $x \in \mathcal{B}(\lambda_{\max}(A)/(2n), 0)$, then $T^{-1}_{A^*,x^*_0}(x) \in C$.*

*Proof of Lemma 4.4.* We prove the statement by contradiction: If there exists an $x \in \mathcal{B}(\lambda_{\max}(A)/(2n), 0)$, such that $T^{-1}_{A^*,x^*_0}(x) \notin C$, then $\mathcal{E}(A^*, x^*_0)$ is not the circumscribing ellipsoid of $C$.

Let $C_0$ be the convex set that is the isotropic transformation of $C$, i.e., $C_0 = \{x | T^{-1}_{A^*,x^*_0}(x) \in C\}$. Since the ratios of volumes is constant for affine transformations, the circumscribing ellipsoid of $C_0$ is $\mathcal{B}(\sqrt{\lambda_{\max}(A)}, 0)$.

Let $\mathcal{B}(r,0)$ be the ball with the maximum radius centered at the origin such that $\mathcal{B}(r,0) \in C_0$. Then, there must exists a point $x'$ such that $\|x'\| = r$ and $x' \in Bd(C_0)$.

By the supporting hyperplane theorem (Boyd and Vandenberghe, 2004) there exists a supporting hyperplane at $x$ such that the entire convex set $C_0$ is on one side of the hyperplane. Let $\mathcal{H} = \{x|\langle h,(x-x')\rangle \leq 0\}$ be the halfspace that contains $C_0$ and passes through $x'$. Assume that the hyperplane $\langle h,(x-x')\rangle = 0$ is not tangent to $\mathcal{B}(r,0)$, i.e., $h$ is not a multiple of $x'$, then we have $\mathcal{H} \cap \mathcal{B}(r,0) \neq \emptyset$ and $(\mathbb{R}^n \setminus \mathcal{H}) \cap \mathcal{B}(r,0) \neq \emptyset$. Since $\mathcal{B}(r,0) \subseteq C_0$, we also have $\mathcal{H} \cap C_0 \neq \emptyset$ and $(\mathbb{R}^n \setminus \mathcal{H}) \cap C_0 \neq \emptyset$. Therefore, the supporting hyperplane must be tangent to $\mathcal{B}(r,0)$ at $x'$, i.e., $h$ must be a multiple of $x'$. Also since $\mathcal{B}(r,0) \subset \mathcal{H} = \{x|\langle h,(x-x')\rangle \leq 0\}$, $h$ must be a positive multiple of $x'$. Without loss of generality assume that $h = x'$.

We have $C_0 \subseteq \mathcal{H} = \{x|\langle x',(x-x')\rangle \leq 0\}$ where $\|x'\| = r$. Assume that $r < \sqrt{\lambda_{\max}(A)}/(2n)$. In this case, by Lemma 4.1, there exists a an ellipsoid whose volume is smaller than $\mathcal{V}_n \lambda_{\max}(A)^{n/2}$. This leads to a contadiction as we know that the circumscribing ellipsoid of $C_0$ is $\mathcal{B}(\sqrt{\lambda_{\max}(A)},0)$. Therefore, $r \geq \sqrt{\lambda_{\max}(A)}/(2n)$.

Since $\mathcal{B}(r,0) \subseteq C_0$ and $r \geq \sqrt{\lambda_{\max}(A)}/(2n)$, we have that if $x \in \mathcal{B}(\sqrt{\lambda_{\max}(A)}/(2n),0)$, then $T^{-1}_{A^*,x_0^*}(x) \in C$. ■

To prove Theorem 4.1, we use Algorithm 8 which is similar to the optimization algorithm under the comparator oracle. Algorithm 8 estimates the gradient direction by at the current ellipsoid center by querying the directional derivatives function in different orthogonal directions. After the estimation of the gradient, Algorithm 8 proceeds to the ellipsoid cut. Before the algorithm terminates Algorithm 8 compares the ellipsoid centers and outputs a point that is near optimal. For the comparison step, we employ the function COMPARE-DP which uses bisection search to find a near optimal point from a given set of points.

**Lemma 4.5.** *For an L-Lipschitz function $f : C \to \mathbb{R}$ and a set $X$ of points with size $m$, The function COMPARE-DP makes at most $(m-1)\log_2 \frac{R_C L m}{2\varepsilon}$ queries to $\psi^{DP}$ and*

*the output $X^*$ of the above algorithm satisfies $f(x^*) \leq \min_{x \in X} f(x) + \varepsilon$, $x^* \in C$, and $x^* \in X^*$.*

*Proof of Lemma 4.5.* The proof follows from bisection search. We observe that in every iteration of the inner while loop the algorithm halves the search space $Conv(\{x^l, x^r\})$ according to the result of the directional derivative at the mid point $(x^r + x^l)/2$. We also note that since only the ascent directions are discarded, at the end of the inner while loop, there exists a point $x^* \in Conv(\{x^l, x^r\})$ such that $f(x^*) = \min_{Conv(\{x^1, x^2\})}$. Since $f$ is $L$-Lipschitz, and $\|x^r - x^l\| \leq 2\varepsilon/(Lm)$, we have $f((x^r + x^l)/2) \leq f(x^*) + \varepsilon/m$.

At the beginning of the inner while loop, the algorithm removes two points $x^1, x^2$ from $X$ and at the end of the inner while loop the algorithm adds a point $x'$ such that $f(x') \leq \min(f(x^1), f(x^2) + \varepsilon/m)$. Therefore, in each iteration of the outer while loop the size of $X^*$ decreases by 1 and the minimum function value among the points in $X^*$ increases by at most $\varepsilon/m$. Since the outer loop makes at most $m - 1$ iterations, the output point $x^*$ satisfies $f(x^*) \leq \min_{x \in X} f(x) + \varepsilon$.

Since $\|x^1 - x^2\| \leq R_C$ for all $x^1, x^2 \in X^*$, and $\|x^r - x^l\|$ is halved in each iteration, the inner while loop makes at most $\log_2 \frac{R_C L m}{2\varepsilon}$ iterations. Since the outer loop makes at most $m - 1$ iterations the number of queries is bounded by $(m - 1) \log_2 \frac{R_C L m}{2\varepsilon}$.

■

*Proof of Theorem 4.1.* We prove the theorem by showing that the output $x'$ of Algorithm 8 satisfies $f(x') \leq \min_{x \in C} f(x) + \varepsilon$ and Algorithm 8 makes at most $nK \lceil 2n \log(2n) \rceil + K \log\left(\frac{R_C L(K+1)}{\varepsilon}\right)$ queries to $\psi^{DP}$ where $K = \left\lceil 8n(n+1) \log\left(\frac{2R_{CL}}{\varepsilon}\right)\right\rceil$.

We first show that the output $x'$ of Algorithm 8 satisfies $f(x') \leq \min_{x \in C} f(x) + \varepsilon$. Note that due to Lemma 4.2, at iteration $k$, the cone $T^{-1}_{A,x^k}(\mathcal{F}(p, \sin^{-1}(1/(2n))))$ of possible gradient directions after the gradient pruning algorithm terminates, includes the gradient. Consequently, the dual cone of $T^{-1}_{A,x^k}(\mathcal{F}(p, \sin^{-1}(1/(2n))))$ includes only

the non-descent directions, i.e., $f(x) \geq f(x^k)$ for all $x \in T^{-1}_{A,x^k}(\mathcal{F}(p, \pi/2 - \sin^{-1}(1/(2n))))$. Therefore, after iteration $k$ there exists a $x^* \in C^{(k+1)}$ such that $f(x^*) = \min_{x \in C} f(x)$.

Since $f$ is $L$-Lipschitz, the volume of the set $\{x | x \in C, f(x) \leq f(x^*) + \varepsilon/2\}$ is at least $\mathcal{V}_n \left(\frac{\varepsilon}{2L}\right)^n$. Due to Lemma 4.1, we have $Vol\left(\underline{\mathcal{E}}_{C^{(K+1)}}\right) < \mathcal{V}_n \left(\frac{\varepsilon}{2L}\right)^n$. Therefore, there exists a point $x$ such that $x \notin C^{(K+1)}$ and $f(x) \leq f(x^*) + \varepsilon/2$

Since every discarded point $x$ satisfies $f(x) \geq f(x^k)$ for some $1 \leq k \leq K$, we have $f(x^k) \leq f(x^*) + \varepsilon/2$ for some $1 \leq k \leq K$. Due to Lemma 4.5, the output point $x' = \textsc{Compare-DP}(X, \psi^{DP}, \varepsilon/2)$ satisfies $f(x') \leq \min_{x \in X} f(x) + \varepsilon/2 \leq f(x^*) + \varepsilon$.

We now prove the bound on the number of queries. The gradient pruning algorithm starts with $\gamma = \pi/2$. As shown in Lemma 4.2, we have $\sin(\gamma) \leq \sqrt{\frac{n-1}{n}}^k \leq e^{\frac{-k}{2n}}$ after $k$ iterations. Since $\theta = \sin^{-1}(1/2n)$, the gradient pruning algorithm PD-DP stops after at most $\lceil 2n \log(2n) \rceil$ iterations where we make $n$ queries in each iteration. The for loop in Algorithm 8 has $K$ iterations. Therefore, the total number of queries due to the gradient pruning algorithm is $n \lceil 2n \log(2n) \rceil K$.

When Compare-DP is called in Algorithm 8, the set has $X$ has $K+1$ elements. Due to Lemma 4.5, the process $\textsc{Compare-DP}(X, \psi^{DP}, \varepsilon/2)$ makes $K \log_2 \frac{R_C L(K+1)}{\varepsilon}$ queries.

The total number of queries is bounded by $nK \lceil 2n \log(2n) \rceil + K \log\left(\frac{R_C L(K+1)}{\varepsilon}\right)$.

■

The proof of Theorem 4.2 is similar to the proof of Theorem 4.1. If the direction pruning algorithm does not encounter an unknown direction, the algorithm approximately estimates the direction of the gradient. If there is an unknown direction, then we consider two cases: the magnitude of the projection of the gradient in the known directions is large compared to the magnitude of the projection of the gradient in the known directions and otherwise. We show that, in the first case, the estimated gradient direction is still close to the direction of the gradient. In the

second case, we show that the ellipsoid center is near-optimal since the magnitude of the gradient is small.

*Proof of Theorem 4.2.* We first show that the output $x'$ of Algorithm 9 satisfies $f(x') \leq \min_{x \in C} f(x) + \varepsilon$. We then prove the bound on the number of queries.

Note that all query points are in $C$. By Lemma 4.4 we know that every $T^{-1}_{A^{(k)},x^{(k)}}(x)$ such that $\|x\| \leq \frac{\sqrt{\lambda_{\max}(A^{(k)})}}{2n}$ are in $C^{(k)}$ and consequently in $C$. All queries have distance at most $\frac{\sqrt{\lambda_{\max}(A^{(k)})}}{\kappa n^{5/2}}$ from the origin in the isotropic coordinates. Since $\kappa \geq 1$, it implies that all query points are in $C$.

At iteration $k$, due to Lemma 4.3, we have $\sqrt{\lambda_{\max}(A^{(k)})} \leq nR_{C^{(k)}} \leq nR_C$. Consequently the radius $R_{T_{A^{(k)},x^{(k)}}(C^{(k)})}$ of $C^{(k)}$ in isotropic coordinates is at most $nR_C$.

Let $E_k$ denote the event that there does not exist a point $x^* \in C^{(k)}$ such that $f(x^*) = \min_{x \in C} f(x)$. Note that $E_1$ does not happen. Assume that $E_1, \ldots, E_k$ did not happen. We show that either event $E_{k+1}$ does not happen or the algorithm finds a near optimal point at iteration $k$. If $E_1, \ldots, E_{K+1}$ do not happen, then one of the ellipsoid centers are optimal as in the classical ellipsoid method.

We consider 3 cases:

1. $\left\|\nabla\left(f \circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0)\right\| > \frac{\varepsilon}{nR_C}$ and $m = n$ when PD-C terminates,
2. $\left\|\nabla\left(f \circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0)\right\| > \frac{\varepsilon}{nR_C}$ and $m \neq n$ when PD-C terminates,
3. $\left\|\nabla\left(f \circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0)\right\| \leq \frac{\varepsilon}{nR_C}$.

**Case 1:** We note that the function $f \circ T^{-1}_{A^{(k)},x^{(k)}}$ is also $\beta$-smooth since we can only expand the coordinates via the isotropic transformation. In PD-C, if a unit vector $d$ is in $UD$, i.e., is unknown, then due to $\beta$-smoothness we have

$$\left|\left\langle \nabla\left(f \circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0), d\right\rangle\right| \leq \frac{\min(\varepsilon, \sqrt{\lambda_{\max}(A^k)})}{\kappa n^{5/2} \max(R_C, 1)}.$$

Since $m = n$, i.e., all basis directions are unknown, we have

$$\left|\left\langle \nabla \left(f \circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0), d_i \right\rangle\right| \leq \frac{\min(\varepsilon, \sqrt{\lambda_{\max}(A^k)})}{\kappa n^{5/2} \max(R_C, 1)}$$

for all $i \in [n]$. Since $d_i \perp d_j$ for all $i \neq j \in [n]$, and $\|d_i\| = 1$ for all $i \in [n]$ by construction, we have

$$\left\|\nabla \left(f \circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0)\right\| \leq \frac{\min(\varepsilon, \sqrt{\lambda_{\max}(A^k)})}{\kappa n^2 \max(R_C, 1)}.$$

This implies that

$$f(x^{(k)}) \leq f(x^*) + \frac{R_C \min(\varepsilon, \sqrt{\lambda_{\max}(A^k)})}{\kappa n \max(R_C, 1)} \leq f(x^*) + \varepsilon$$

since $\kappa \geq 1$, the function in isotropic coordinates is convex, $R_{T_{A^{(k)},x^{(k)}}(C^{(k)})} \leq nR_C$, and there exists a minimizer $x^* \in C^{(k)}$.

**Case 2:** Let $p^j$ denote the value of $p$ at iteration $j$ of PD-C. Assume that $\angle\left(p^j, Proj_{UD^\perp}\left(\nabla\left(f \circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0)\right)\right) \leq \gamma$ while PD-C runs. If a new unknown direction $d_i$ is detected at iteration $j$ of PD-C, then we have

$$\angle\left(p^{j+1}, Proj_{(UD\cup\{d_i\})^\perp}\left(\nabla\left(f \circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0)\right)\right) \leq \gamma$$

since $p^{j+1}$ and $d_i$ are orthogonal. Therefore, the angle

$$\angle\left(p, Proj_{UD^\perp}\left(\nabla\left(f \circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0)\right)\right)$$

does not increase when a new unknown direction is detected. If there is no new unknown direction, then $Proj_{UD^\perp}\left(\nabla\left(f \circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0)\right)$ is in a hyperoctant in the subspace defined by $Span(UD^\perp)$. We have

$$\angle\left(p^{j+1}, Proj_{UD^\perp}\left(\nabla\left(f \circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0)\right)\right) \leq \cos^{-1}(\left\langle p^{j+1}, w_{m+2}\right\rangle) \leq \sin^{-1}\left(\sqrt{\frac{n-1}{n}}\sin(\gamma)\right)$$

by Lemma 4.2. Since the angle $\angle\left(p, Proj_{UD^\perp}\left(\nabla\left(f \circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0)\right)\right)$ decreases by a constant factor if there is no new unknown directions and PD-C can detect an unknown direction at most $n-1$ times, we have

$$\angle\left(p, Proj_{UD^\perp}\left(\nabla\left(f \circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0)\right)\right) \leq \sin^{-1}(1/(2\sqrt{2}n))$$

when PD-C terminates. This implies

$$\frac{\left\langle p, Proj_{UD^\perp}\left(\nabla\left(f\circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0)\right)\right\rangle}{\left\|Proj_{UD^\perp}\left(\nabla\left(f\circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0)\right)\right\|}\geq\sqrt{1-\frac{1}{8n^2}}. \tag{4.3}$$

Since $p\notin Span(UD)$, (4.3) implies that

$$\frac{\left\langle p, \nabla\left(f\circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0)\right\rangle}{\left\|Proj_{UD^\perp}\left(\nabla\left(f\circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0)\right)\right\|}\geq\sqrt{1-\frac{1}{8n^2}}. \tag{4.4}$$

When PD-C terminates, we have

$$\left\|Proj_{UD}\left(\nabla\left(f\circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0)\right)\right\|\leq\frac{\sqrt{m}\min(\varepsilon,1)}{\kappa n^2\max(R_C,1)}\leq\frac{\sqrt{n}\varepsilon}{\kappa n^{5/2}R_C}=\frac{\varepsilon}{\kappa n^2 R_C}.$$

Using this we get

$$\begin{aligned}&\left\|\nabla\left(f\circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0)\right\|\\&=\left\|Proj_{UD}\left(\nabla\left(f\circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0)\right)+Proj_{UD^\perp}\left(\nabla\left(f\circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0)\right)\right\|\\&\leq\left\|Proj_{UD^\perp}\left(\nabla\left(f\circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0)\right)\right\|+\frac{\varepsilon}{\kappa n^2 R_C}.\end{aligned}$$

Since $\left\|\nabla\left(f\circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0)\right\|>\frac{\varepsilon}{nR_C}$, we have

$$\frac{\left\|Proj_{UD^\perp}\left(\nabla\left(f\circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0)\right)\right\|}{\left\|\nabla\left(f\circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0)\right\|}>1-\frac{1}{\kappa n}=\sqrt{1-\frac{1}{8n^2}}. \tag{4.6}$$

By combining (4.4) and (4.6), we finally get

$$\frac{\left\langle p, \nabla\left(f\circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0)\right\rangle}{\left\|\nabla\left(f\circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0)\right\|} \tag{4.7a}$$

$$=\frac{\left\langle p, \nabla\left(f\circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0)\right\rangle}{\left\|Proj_{UD^\perp}\left(\nabla\left(f\circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0)\right)\right\|}\frac{\left\|Proj_{UD^\perp}\left(\nabla\left(f\circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0)\right)\right\|}{\left\|\nabla\left(f\circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0)\right\|} \tag{4.7b}$$

$$\geq\sqrt{1-\frac{1}{8n^2}}\sqrt{1-\frac{1}{8n^2}} \tag{4.7c}$$

$$\geq\sqrt{1-\frac{1}{4n^2}}. \tag{4.7d}$$

We note that (4.7d) implies that when $m \neq n$ and $\left\|\nabla\left(f \circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0)\right\| > \frac{\varepsilon}{nR_C}$, we have $\nabla\left(f \circ T^{-1}_{A^{(k)},x^{(k)}}\right)(0) \in \mathcal{F}(p, \sin^{-1}(1/(2n)))$. This implies that the gradient estimate is accurate and the ellipsoid cut only removes ascent directions. If the gradient pruning algorithm succeeds in the $k$th iteration then by Lemma 4.1, $C^{(k)} \setminus C^{(k+1)}$ only includes the ascent points. After every iteration $k$, there exists a $x^* \in C^{(k)}$ such that $f(x^*) = \min_{x \in C} f(x)$. Therefore, event $E_{k+1}$ does not happen.

**Case 3:** We have $f(x^{(k)}) \leq f(x^*)+\varepsilon$ since the function in isotropic coordinates is convex, $R_{T_{A^{(k)},x^{(k)}}(C^{(k)})} \leq nR_C$, and there exists a minimizer $x^* \in C^{(k)}$.

If Case 1 or 3 happens, the output point $x'$ of Algorithm 9 satisfies $f(x') \leq f(x^{(k)}) \leq f(x^*) + \varepsilon$ since Algorithm 9 compares the ellipsoid centers before termination. If Case 1 or 3 does not happen, then event $E_1, \ldots, E_{K+1}$ does not happen, i.e., the ellipsoid method proceeds successfully. Without loss of generality we assume that Case 1 or 3 does not happen.

Since $f$ is $L$-Lipschitz the volume of the set $\{x | x \in C, f(x) \leq f(x^*) + \varepsilon\}$ is at least $\mathcal{V}_n\left(\frac{\varepsilon}{L}\right)^n$. Let $K = \left\lceil 8n(n+1)\log\left(\frac{R_C L}{\varepsilon}\right)\right\rceil$. Due to Lemma 4.1, we have $Vol\left(\underline{\mathcal{E}}_{C^{(K)}}\right) < \mathcal{V}_n\left(\frac{\varepsilon}{L}\right)^n$. Therefore, there exists a point $x$ such that $x \notin C^{(K)}$ and $f(x) \leq f(x^*) + \varepsilon$.

Since the function value of every discarded point in $C \setminus C^{(K)}$ is greater than or equal to $f(x^k)$ for some $1 \leq k \leq K$, we have $f(x^k) \leq f(x^*) + \varepsilon$ for some $1 \leq k \leq K$. Therefore, the output point $x'$ satisfies $f(x') \leq \min_{x \in X} f(x) + \varepsilon \leq f(x^*) + \varepsilon$.

We now prove the bound on the number of queries. The gradient pruning algorithm starts with $\gamma = \pi/2$. As shown in Lemma 4.2, if the there is no new unknown direction, each iteration satisfies $\frac{\sin(\gamma')}{\sin(\gamma)} \leq \sqrt{\frac{n-|UD|-1}{n-|UD|}} \leq \sqrt{\frac{n-1}{n}}$ where $\gamma'$ is the new value assigned to $\gamma$. After $k$ iterations we have $\sin(\gamma) \leq \sqrt{\frac{n-1}{n}}^k \leq e^{\frac{-k}{2n}}$. Since $\theta = \sin^{-1}(1/(2\sqrt{2}n))$, the gradient pruning algorithm stops after at most $\left\lceil 2n\log(2\sqrt{2}n)\right\rceil$ iterations where we make at most $2n$ queries in each iteration. Note that we can detect at most $n$ unknown directions while running the gradient pruning algorithm. There-

fore, the gradient pruning algorithm makes at most $2n\left\lceil 2n\log(2\sqrt{2}n)+n\right\rceil$ queries to the oracle.

The for loop in Algorithm 9 has $\left\lceil 8n(n+1)\log\left(\frac{R_C L}{\varepsilon}\right)\right\rceil$ iterations. Therefore, the total number of queries is at most

$$2n\left\lceil 2n\log(2\sqrt{2}n)+n\right\rceil\left\lceil 8n(n+1)\log\left(\frac{R_C L}{\varepsilon}\right)\right\rceil$$

before the last comparison step. The set $X$ has at most $\left\lceil 8n(n+1)\log\left(\frac{R_C L}{\varepsilon}\right)+1\right\rceil$ elements. Finding the smallest function value requires $\left\lceil 8n(n+1)\log\left(\frac{2R_C L}{\varepsilon}\right)\right\rceil$ queries to the comparator oracle. Thus, the total number of queries is at most

$$2n\left\lceil 2n\log(2\sqrt{2}n)+n\right\rceil\left\lceil 8n(n+1)\log\left(\frac{R_C L}{\varepsilon}\right)\right\rceil+\left\lceil 8n(n+1)\log\left(\frac{R_C L}{\varepsilon}\right)\right\rceil.$$

■

*Proof of Theorem 4.3.* We first show that all queries are feasible. We note that during Phase 1, all queries have distance at most $\sqrt{\lambda_{\max}(A^{(k)})}/(2n)$ from the origin in the isotropic coordinates where $\sqrt{\lambda_{\max}(A^{(k)})}$ is the radius of the current convex set in the isotropic coordinates. By Lemma 4.4, all queries in Phase 1 are feasible. The query points in Phases 2 and 3 are ellipsoid centers which are feasible due to Lemma 4.4.

We analyze the regret induced by the inner for loop. Let $D$ be the current convex set such that $\underline{\mathcal{E}}_D = \mathcal{E}(A,x)$. We first show that the gradient estimate estimation is accurate with high probability in the isotropic coordinates. Since the isotropic transformation can only stretch the coordinates, the function in the isotropic coordinates is also $\beta$-smooth and $L$-Lipschitz. We have

$$\left|\frac{\left(f\circ T_{A,x}^{-1}\right)(0)-\left(f\circ T_{A,x}^{-1}\right)(d_i e_j)}{d_i}-\frac{\left\langle\nabla\left(f\circ T_{A,x}^{-1}\right)(0),d_i e_j\right\rangle}{d_i}\right|\leq\frac{\beta d_i}{2} \tag{4.8}$$

due to $\beta$-smoothness.

At the $i$-th iteration of the inner for loop, the directional derivative estimate in direction $e_j$ is $p_j = \frac{\hat{\psi}^{NV}(T_{A,x}^{-1}(d_i e_j)) - \hat{\psi}^{NV}(x)}{d_i}$. We have

$$\Pr\left(\left|\frac{\hat{\psi}^{NV}(T_{A,x}^{-1}(d_i e_j)) - \hat{\psi}^{NV}(x)}{d_i} - \frac{\left(f \circ T_{A,x}^{-1}\right)(0) - \left(f \circ T_{A,x}^{-1}\right)(d_i e_j)}{d_i}\right| > \frac{d_i}{\min\left(\lambda_{\max}(A), 1\right)}\right) \tag{4.9a}$$

$$\leq 2\exp\left(-\frac{d_i^4 \tau_i}{2\sigma^2 \min\left(\lambda_{\max}(A), 1\right)^2}\right) \leq \delta' \tag{4.9b}$$

for each direction $e_j$. Using this in (4.8), the directional derivative estimate satisfies $\left|p_j - \left\langle \nabla\left(f \circ T_{A,x}^{-1}\right)(0), e_j \right\rangle\right| \leq \Delta_i$ at point 0 with probability at least $1 - \delta'$. Consequently, we have $\left\|\nabla\left(f \circ T_{A,x}^{-1}\right)(0) - p\right\| \leq \sqrt{n}\Delta_i$ with probability at least $1 - 2n\delta'$.

If Case 2 happens, then we have $\left\|\nabla\left(f \circ T_{A,x}^{-1}\right)(0)\right\| < (2n+1)\sqrt{n}\Delta_i$ since $\|p\| < 2n\sqrt{n}\beta\Delta_i$. Since $T_{A,x}f$ is $\beta$-smooth, the norm of the gradient is smaller than $(2n+1)\sqrt{n}\Delta_i + \beta d_i$ for every query point.

If Case 1 happens then $\nabla\left(f \circ T_{A,x}^{-1}\right)(0) \in \mathcal{F}(p, \sin^{-1}(1/(2n)))$, and the ellipsoid algorithm proceeds successfully. Note that the elliposid cuts happen only when Case 1 happens. Since every discarded point $y$ satisfies $f(y) \geq f(x)$, the set $D$ always contains a minimizer $x^*$.

We first show that Case 2 can happen at most $\log_{16}\left(\frac{15T}{2n\tau}\right)$ times. In the $i$th iteration of the inner for loop, we make $2^{4i}\tau$ queries for two points in every dimension. Let $W$ be the number of iterations of the inner for loop. We have

$$T \geq \sum_{i=0}^{W} 2^{4i} 2n\tau \tag{4.10a}$$

$$= \frac{(16^{W+1} - 1)2n\tau}{15}. \tag{4.10b}$$

By rearranging the terms, we get $W \leq \log_{16}\frac{15T}{2n\tau} - 1$. Therefore, the maximum value of $i$ is $\log_{16}\left(\frac{15T}{2n\tau}\right)$.

For each iteration of inner for loop the probability of failure is less than or equal to $2n\delta'$. Since the maximum value of $i$ is $\log_{16}\left(\frac{15T}{2n\tau}\right)$, the total probability of failure is less than or equal to $2n\delta'\log_{16}\left(\frac{15T}{2n\tau}\right)$.

We now bound the regret for each iteration of the inner loop assuming that the gradient estimation did not fail. If $i = 0$, then the regret of each query is $R_D L$ since there exists a minimizer $x^* \in D$. If $i > 0$, then $\left\|\nabla\left(f \circ T_{A,x}^{-1}\right)(0)\right\| < 2(2n+1)\sqrt{n}\Delta_i$ since Case 1 did not happen in iteration $i-1$. Due to $\beta$-smoothness, the norm of gradient at the query points is smaller than $2(2n+1)\sqrt{n}\Delta_i\Delta_i + \beta d_i$ in isotropic coordinates. The regret of each query is smaller than $\sqrt{\lambda_{\max}(A)}n(4n\sqrt{n}\Delta_i + \beta d_i)$ since $D$ contains a minimizer $x^*$ and the radius of $D$ is $\sqrt{\lambda_{\max}(A)}$ in the isotropic coordinates. The total regret induced by the inner for loop is less than or equal to

$$R_D L\tau + \sum_{i=1}^{\log_{16}\left(\frac{15T}{2n\tau}\right)} \sqrt{\lambda_{\max}(A)}(2(2n+1)\sqrt{n}\Delta_i + \beta d_i)\tau_i \tag{4.11a}$$

$$= R_D L\tau + \sum_{i=1}^{\log_{16}\left(\frac{15T}{2n\tau}\right)} 2^{3i}\sqrt{\lambda_{\max}(A)}(2(2n+1)n\sqrt{n}\Delta + \beta d)\tau \tag{4.11b}$$

$$= R_D L\tau + \frac{8^{\log_{16}\left(\frac{15T}{2n\tau}\right)} - 8}{7}\sqrt{\lambda_{\max}(A)}(2(2n+1)\sqrt{n}\Delta + \beta d)\tau \tag{4.11c}$$

$$= R_D L\tau + \frac{15^{3/4}T^{3/4}}{7(2^{3/4}n^{3/4}\tau^{3/4})}\sqrt{\lambda_{\max}(A)}2(2n+1)\sqrt{n}\Delta + \beta d)\tau \tag{4.11d}$$

$$\leq R_D L\tau + T^{3/4}\sqrt{\lambda_{\max}(A)}n^{-3/4}(2(2n+1)\sqrt{n}\Delta + \beta d)\tau^{1/4} \tag{4.11e}$$

$$= R_D L\tau + T^{3/4}\sqrt{\lambda_{\max}(A)}n^{-3/4} \tag{4.11f}$$

$$\left(2\left(2n+1\right)\sqrt{n}\frac{2+\beta\min\left(\lambda_{\max}(A),1\right)}{2n\sqrt{\min\left(\lambda_{\max}(A),1\right)}} + \frac{\beta\sqrt{\min\left(\lambda_{\max}(A),1\right)}}{n}\right)\tau^{1/4} \tag{4.11g}$$

$$\leq R_D L\tau + T^{3/4}\sqrt{\lambda_{\max}(A)}n^{-3/4}\left(2\left(2n+1\right)\sqrt{n}\frac{1+\beta\min\left(\lambda_{\max}(A),1\right)}{n\sqrt{\min\left(\lambda_{\max}(A),1\right)}}\right)\tau^{1/4} \tag{4.11h}$$

$$\leq R_D L\tau + 5T^{3/4}\sqrt{\lambda_{\max}(A)}n^{-1/4}\left(\frac{1+\beta\min\left(\lambda_{\max}(A),1\right)}{\sqrt{\min\left(\lambda_{\max}(A),1\right)}}\right)\tau^{1/4} \tag{4.11i}$$

$$\leq R_D L\tau + 5T^{3/4}n^{-1/4}\max\left(\sqrt{\lambda_{\max}(A)},1\right)(1+\beta)\tau^{1/4} \tag{4.11j}$$

$$\leq R_D L\tau + 5T^{3/4}n^{-1/4}\max\left(nR_D,1\right)(1+\beta)\tau^{1/4} \tag{4.11k}$$

$$\leq R_C L\tau + 5T^{3/4}n^{-1/4}\max\left(nR_C,1\right)(1+\beta)\tau^{1/4} \tag{4.11l}$$

where (4.11e) is due to $(15/2)^{3/4}/7 \leq 1$, (4.11h) is due to $(2n+1)\sqrt{n}+1 \leq (2n+1)\sqrt{n}$,

and (4.11i) is due to $2(2n+1) \leq 5n$. Inequality (4.11l) follows from $\lambda_{\max}(A) \leq nR_D$ for the convex set $D$ by Lemma 4.3 and $R_D \leq R_C$.

Since the outer for loop repeats at most $K$ times, the total regret incurred during Phase 1 is at most $K\left(R_C L\tau + 5T^{3/4}n^{-1/4}\max\left(nR_C, 1\right)(1+\beta)\tau^{1/4}\right)$ with probability at least $1 - 2nK\delta' \log_{16}\left(\frac{15T}{2n\tau}\right) = 1 - \delta \log_{16}\left(\frac{15T}{2n\tau}\right) / \left(2\log_{16}\left(\frac{15T}{2n}\right)\right)$. Since $\tau \geq 1$, we have the probability of failure is less than $1 - \delta/2$.

If Case 1 happens in the $k$th iteration of the outer loop, then $C^{(k)} \setminus C^{(k+1)}$ only includes the ascent points by Lemma 4.1. Since $f$ is $L$-Lipschitz the volume of the set $\{x | x \in C, f(x) \leq f(x^*) + \varepsilon\}$ is at least $\mathcal{V}_n\left(\frac{\varepsilon}{L}\right)^n$. If the iteration $K$ happens, then due to Lemma 4.1, we have $Vol\left(\underline{\mathcal{E}}_{C^{(K)}}\right) < \mathcal{V}_n\left(\frac{\varepsilon}{L}\right)^n$. Therefore, there exists a point $x$ such that $x \notin C^{(K)}$ and $f(x) \leq f(x^*) + T^{-1/4}/2$.

Since the function value of every discarded point in $C \setminus C^{(K)}$ is greater than or equal to $f(x^k)$ for some $1 \leq k \leq K$, we have $f(x^k) \leq f(x^*) + T^{-1/4}/2$ for some $1 \leq k \leq K$.

By the Hoeffding's inequality, the point $x'$ with the highest empirical mean satisfies $f(x^k) \leq f(x^*) + T^{-1/4}$ with probability at least $1 - \delta/2$.

Since set $X$ has $K+1$ elements, the regret incurred during Phase 2 is at most

$$(K+1)\left\lceil 32\sigma^2\sqrt{T}\log\left(\frac{2(K+1)}{\delta}\right)\right\rceil R_C L. \tag{4.12a}$$

Since the output point $x'$ satisfies $f(x^k) \leq f(x^*) + T^{-1/4}$. Therefore, the regret incurred during Phase 3 is at most $T^{3/4}$.

Therefore, the regret of Algorithm 10 is at most

$$\begin{aligned} &K\left(R_C L\tau + 5T^{3/4}n^{-1/4}\max\left(nR_C, 1\right)(1+\beta)\tau^{1/4}\right) \\ &+ (K+1)\left\lceil 32\sigma^2\sqrt{T}\log\left(\frac{2(K+1)}{\delta}\right)\right\rceil R_C L + T^{3/4} \end{aligned}$$

with probability at least $1 - \delta$.

■

**Remark 4.1.** *Let $\mathcal{R}^{\mathcal{A}}(T)$ be the regret of Algorithm 10. The proof of Theorem 4.3 shows that*

$$\mathcal{R}^{\mathcal{A}}(T) \leq \sum_{k=1}^{K} \left( R_D L\tau + \sum_{i=1}^{W_k} \tau_i r(i) \right)$$

*where $W_k$ is the number of iterations in the inner loop and $r(i) = \sqrt{\lambda_{\max}(A)}n(4n\sqrt{n}\Delta_i + \beta d_i)$ is an upper bound on the regret of each query in the $i$-th iteration of the inner for loop. The proof provided in (Karabag et al., 2021a) upper bounds $W_k$ with $\bar{W}$ assuming that every iteration of the inner loop takes $\sum_{i=1}^{\bar{W}} \tau_i = T$ queries and shows the regret given in Theorem 4.3. Instead, we can upper bound $\sum_{k=1}^{K} \left( R_D L\tau + \sum_{i=1}^{W_k} \tau_i r(i) \right)$ by observing that $r(i)$ is a decreasing function of $i$, and $\sum_{k=1}^{K} \sum_{i=1}^{W_k} \tau_i \leq T$, i.e., the total number of queries during Phase 1 is bounded by $T$. Via this observation, we can get a tighter bound:*

$$\mathcal{R}^{\mathcal{A}}(T) \leq \sum_{k=1}^{K} \left( R_D L\tau + \sum_{i=1}^{W^*} \tau_i r(i) \right)$$

*where $\sum_{i=1}^{W^*} \tau_i = T/K$. Carrying out the same analysis for the rest yields to*

$$\begin{aligned} &KR_C L\tau + 5K^{1/4}T^{3/4}n^{-1/4} \max\left(nR_C, 1\right)(1+\beta)\tau^{1/4} \\ &+ (K+1)\left\lceil 32\sigma^2\sqrt{T}\log\left(\frac{2(K+1)}{\delta}\right)\right\rceil R_C L + T^{3/4} \end{aligned}$$

*regret upper bound. We note that*

$$5K^{1/4}T^{3/4}n^{-1/4} \max\left(nR_C, 1\right)(1+\beta)\tau^{1/4} = \tilde{\mathcal{O}}(n^{2.25}T^{0.75})$$

*ignoring the other parameters.*

# Chapter 5: Conclusions

This dissertation focused on developing theory and algorithms for autonomous decision-making in adversarial or information-scarce settings. Towards this goal, we first considered a deception problem in the supervisory control setting as a part of decision-making in an adversarial setting. We explored the synthesis of deceptive policies for an agent that aims to deceive its supervisor and the synthesis of optimal reference policies for safeguarding against deception. For the information-scarce settings, we considered two different problems. The first problem focuses on the lack of information in a multiagent sequential decision setting due to communication interruptions. We analyzed the performance under communication loss and provided an algorithm for the synthesis of performant policies. In the second problem, we considered optimization with limited information by considering a variety of oracles and provided algorithms with polynomial sample complexities for these oracles.

The rest of this chapter provides a more detailed summary of the considered problems and results, and discusses the future work directions for these problems.

## 5.1 Summary

**Deception in probabilistic supervisory control** We considered the problem of deception under a supervisor that provides a reference policy. We modeled this problem as a hypothesis testing problem in MDPs, and used KL divergence as a proxy of deceptiveness. We showed that in the fully observable setting, there exists an optimal stationary deceptive policy, and its synthesis requires solving a convex optimization problem. We also considered the synthesis of optimal reference policies that easily prevent deception. We showed that this problem is NP-hard. We proposed two synthesis methods based and provided an approximation that can be modeled as a linear program. We then considered the partially observable setting where the

supervisor receives partial observations of the agent's state in the MDP. We showed that finding optimal deceptive policies, while possible, is computationally intractable, and there is no polynomial-time approximation algorithm. As an alternative to the synthesis of optimal policies, we considered special classes of policies where deceptive policies can be synthesized efficiently. We also considered a special class of MDPs, i.e., directed graphs, where optimal deceptive path planning can be performed efficiently.

**Minimally dependent multiagent systems** We considered the design of multiagent systems that are robust to communication loss. We provided algorithms for the decentralized execution of a joint policy when communication is lost. These decentralized policy execution algorithms rely on each agent simulating the control processes of its teammates. We considered a variety of communication loss scenarios: communication between the agents is never available during policy execution, communication availability follows a random process with arbitrary communication groups, or communication availability is chosen by an adversary. We quantified the gap between the performance of the proposed decentralized policy algorithms under these communication loss scenarios and that achieved by the joint policy under full communication. This performance gap is a function of the total correlation of the joint policy. Using these theoretical results, we proposed an optimization algorithm for the synthesis of minimally dependent joint policies that balance the team's performance with the total correlation. As a result, these minimally dependent policies remained performant under communication loss.

**Optimization using sub-zeroth order oracles** We considered the problem of minimizing a smooth, Lipschitz, convex function using sub-zeroth-order oracles that provide information on the sign of the directional derivative at the query point and direction, make a comparison between the function values at two query points, and output a noisy function value at the query point. We leveraged the smoothness property of the objective function and build variants of the ellipsoid method based

on gradient estimation. We provided optimization algorithms for these oracles that have polynomial sample complexities in the relevant parameters. We also provided an optimization algorithm for the noisy value oracle that incurs sublinear regret in the number of queries.

## 5.2 Extensions and Future Directions

**Deception in probabilistic supervisory control** A natural extension of the discrete-state setting introduced in §2.3 is the continuous-state setting. In (Patil et al., 2023), we consider the deception problem under discrete-time nonlinear continuous-state dynamics. Unlike the discrete-state settings where the optimal deceptive policies can be synthesized by solving a convex optimization problem with finite variables and constraints, in (Patil et al., 2023), we propose a path-integral-based solution (Kappen, 2005) and utilize Monte Carlo simulations of the reference policy to compute the optimal deceptive actions online. An open question is to analyze the sample complexity of this approach for the KL objective function. We also extend the deception problem to the zero-sum two-player stochastic game setting (Karabag et al., 2021c) where the agent aims to behave like an "average" agent and win the game against its opponent. The opponent, on the other hand, aims to detect whether the agent is an average agent or a "cheating" agent. We show that despite not having a discount factor, the game admits a Nash equilibrium with stationary policies.

In §2, we showed the computational hardness of synthesizing optimal reference policies under full observability and the computational hardness of synthesizing optimal deceptive policies under partial observability. A future direction is to explore whether computationally efficient approximations exist for these problems. For example, §2.3.3.4 provides a linear programming relaxation for the synthesis of optimal reference policies by considering a simple feature of the observed paths rather than considering the whole path for hypothesis testing. Similarly, for the synthesis of deceptive policies in the partially observable setting, (Fu, 2023) provides a synthesis

algorithm that does not have dependence on the time horizon and ensures qualitative deception, i.e., the observer is never sure whether the agent is deceptive, in the infinite horizon setting.

In some settings, the observer, e.g., the supervisor, may not know the behavioral models of the well-intentioned and deceptive agents. For example, in cyber settings, the users show various behavior. Learning and detection are performed together in these settings (Zhang et al., 2021; Wressnegger et al., 2013; Rosenberg et al., 2021). It would be interesting to explore the synthesis of deceptive strategies that exploit the vulnerabilities of the learning modules or worsen the detection rate by showing a different behavior to induce a wrong prior for the system manager. Relevantly, the works on goal deception (Dragan et al., 2015; Liu et al., 2021; Savas et al., 2022) assume bounded rationality models to represent the observer's prior distribution on the agent's behavior. While this assumption leads to successful deception against oblivious observers, it may not be suitable for deception-aware observers. Modeling the deception-aware observers' prior distributions would be an interesting future direction. On the flip side, the deceptive agent may not have a full knowledge of its environment and may need to learn deceptive policies. Towards this goal, the works (Liu et al., 2021; Lewis and Miller, 2023) adapt $Q$-learning for deceptive planning in goal deception, and in (Karabag et al., 2021c), we consider an offline learning problem where the agent learns a deceptive policy using the paths generated under the "average" agent's policy.

**Minimally dependent multiagent systems** In §3, we introduced a framework to synthesize minimally dependent policies that remain performant under communication losses. While these policies have provable performance guarantees and are optimal up to a constant factor for a special class of MDPs (see Proposition 3.1), they may not be optimal in general in terms of performance. Future work could investigate the gap between the best achievable performance under communication losses and under full communication.

The introduced framework maintains imaginary states instead of beliefs about the teammates. This approach allows us to avoid the belief-state explosion during the synthesis procedure but we still need to solve an optimization problem with exponentially many variables in the number of agents. This phenomenon naturally occurs in multiagent MDP planning problems due to the dependence between the transition or reward functions. To circumvent computational challenges of the multiagent setting, (Neary et al., 2021; Eappen and Jagannathan, 2023) limit the interactions between the agents to high-level events and have the agents learn its low-level behavior independently from the other agents given the high-level state. This approach is a potential way to overcome the computational issues of our framework. Another approach is to consider symmetrical agents as in mean-field games (Arabneydi and Mahajan, 2014; Lasry and Lions, 2007) and use a policy class that uses state densities as inputs.

In §3, we considered minimally dependent policies to be robust against communication losses. These policies also be used for privacy-preserving multiagent planning. In (Chen et al., 2023a), we consider a multiagent setting where a group of agents cooperate towards a common goal, but purposefully alters their intercommunication to preserve interagent privacy by employing a symbolic differential privacy mechanism Chen et al. (2023b). Since minimally dependent policies make the agents insensitive to the state information about the agents, these policies remain performant under the considered differential privacy mechanism. Future work could focus on different notions of privacy and the use of minimally dependent policies for them.

**Optimization using sub-zeroth order oracles** §4 focuses on the sample complexity of optimization using sub-zeroth-order oracles. The computational and space complexities of the provided algorithms are the same as those of the classical ellipsoid method. However, finding the exact minimum volume circumscribing ellipsoid for an arbitrary convex set is computationally intractable. In practice, we can avoid

the computation of the minimum volume ellipsoids by finding an approximate enclosing ellipsoid using a separation oracle and analytical expressions involving the ellipsoid found at the previous step (Goldfarb and Todd, 1982). We note that in the case of the comparator and noisy-value oracles, we use a property of the minimum volume circumscribing ellipsoid to give optimality and regret guarantees. We can show that a similar property holds for the approximate ellipsoids through additional feasibility cuts. For the directional preference and comparator oracles, since the sampling distance can be arbitrarily reduced through small modifications in the presented algorithms, the given sample complexities can be achieved with polynomial time complexities. For the noisy-value oracle, we expect that polynomial time complexities can be achieved while maintaining a polynomial regret in the number of dimensions.

The comparator oracle is motivated by computational constraints and learning from human preferences. While we consider a deterministic comparator oracle, a stochastic oracle (whose parameters potentially depend on the function values at the query points) would be more realistic for these motivations. The work (Jamieson et al., 2012) introduced a stochastic oracle and provided sample complexity upper and lower bounds for strongly convex functions. Future work could investigate the sample complexity for smooth convex functions with stochastic oracles.

# Bibliography


Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *ACM SIGSAC conference on computer and communications security*, pages 308–318. ACM, 2016.

Ali E Abbas and Ronald A Howard. *Foundations of decision analysis*. Pearson Higher Ed, 2015.

Riad Akrour, Marc Schoenauer, and Michèle Sebag. APRIL: Active preference learning-based reinforcement learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 116–131. Springer, 2012.

Soroush Alamdari, Elaheh Fata, and Stephen L Smith. Persistent monitoring in discrete environments: Minimizing the maximum weighted latency between observations. *International Journal of Robotics Research*, 33(1):138–154, 2014.

Mohammed H Almeshekah and Eugene H Spafford. Cyber security deception. In *Cyber Deception*, pages 23–50. Springer, 2016.

Eitan Altman. *Constrained Markov decision processes*, volume 7. CRC Press, 1999.

Rajeev Alur, Pavol Černỳ, and Steve Zdancewic. Preserving secrecy under refinement. In *International Colloquium on Automata, Languages, and Programming*, pages 107–118. Springer, 2006.

MOSEK Aps. Mosek optimizer API for Python. *Software Package, Ver*, 9, 2020.

Jalal Arabneydi and Aditya Mahajan. Team optimal control of coupled subsystems with mean-field sharing. In *IEEE Conference on Decision and Control*, pages 1669–1674. IEEE, 2014.

Charles Audet and John E Dennis Jr. Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on optimization*, 17(1):188–217, 2006.

Charles Audet and Warren Hare. *Derivative-free and blackbox optimization*. Springer, 2017.

Cheng-Zong Bai, Fabio Pasqualetti, and Vijay Gupta. Data-injection attacks in stochastic control systems: Detectability and performance tradeoffs. *Automatica*, 82:251–260, 2017.

Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT Press, 2008.

Mayank Bakshi and Vinod M Prabhakaran. Plausible deniability over broadcast channels. *IEEE Transactions on Information Theory*, 64(12):7883–7902, 2018.

Raphen Becker, Shlomo Zilberstein, Victor Lesser, and Claudia V Goldman. Transition-independent decentralized Markov decision processes. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 41–48. ACM, 2003.

Raphen Becker, Alan Carlin, Victor Lesser, and Shlomo Zilberstein. Analyzing myopic approaches for multi-agent communication. *Computational Intelligence*, 25(1):31–50, 2009.

Alexandre Belloni, Tengyuan Liang, Hariharan Narayanan, and Alexander Rakhlin. Escaping the local minima via simulated annealing: Optimization of approxi-

mately convex functions. In *Conference on Learning Theory*, pages 240–265. PMLR, 2015.

Béatrice Bérard, Krishnendu Chatterjee, and Nathalie Sznajder. Probabilistic opacity for Markov decision processes. *Information Processing Letters*, 115(1): 52–59, 2015.

Olivier Bernardi and Omer Giménez. A linear algorithm for the random sampling from regular languages. *Algorithmica*, 62(1):130–145, 2012.

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signSGD: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, pages 560–569. PMLR, 2018.

Blai Bonet. Deterministic POMDPs revisited. In *Conference on Uncertainty in Artificial Intelligence*, pages 59–66. PMLR, 2009.

Stefan Boschert and Roland Rosen. Digital twin—the simulation aspect. In *Mechatronic futures*, pages 59–74. Springer, 2016.

Abdeslam Boularias, Jens Kober, and Jan Peters. Relative entropy inverse reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pages 182–189. PMLR, 2011.

Craig Boutilier. Planning, learning and coordination in multiagent decision processes. In *Conference on Theoretical Aspects of Rationality and Knowledge*, volume 96, pages 195–210. ACM, 1996.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction

method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1): 1–122, 2011.

Jean Bretagnolle and Catherine Huber. Estimation des densités: risque minimax. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 47(2): 119–137, 1979.

Sébastien Bubeck, Yin Tat Lee, and Ronen Eldan. Kernel-based methods for bandit convex optimization. In *Annual ACM SIGACT Symposium on Theory of Computing*, pages 72–85. ACM, 2017.

Richard L Burden and J Douglas Faires. *Numerical analysis*. PWS Publishers, 1985.

Jochen Burghardt. Example to demonstrate that the subset property for regular languages is NP-hard. `https://en.wikipedia.org/wiki/File:RegSubsetNP.pdf`, 2016. Accessed Aug 5, 2021.

Yongcan Cao, Wenwu Yu, Wei Ren, and Guanrong Chen. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial Informatics*, 9(1):427–438, 2012.

Thomas E Carroll and Daniel Grosu. A game theoretic investigation of deception in network security. *Security and Communication Networks*, 4(10): 1162–1172, 2011.

Bo Chen, Calvin Hawkins, Mustafa O Karabag, Cyrus Neary, Matthew Hale, and Ufuk Topcu. Differential privacy in cooperative multiagent planning. In *Conference on Uncertainty in Artificial Intelligence*, pages 1–8. PMLR, 2023a.

Bo Chen, Kevin Leahy, Austin Jones, and Matthew Hale. Differential privacy for symbolic systems with application to markov chains. *Automatica*, 152:1–13, 2023b.

Minhao Cheng, Simranjit Singh, Patrick H. Chen, Pin-Yu Chen, Sijia Liu, and Cho-Jui Hsieh. Sign-OPT: A query-efficient hard-label adversarial attack. In *International Conference on Learning Representations*, 2020.

Frank H Clarke. Generalized gradients and applications. *Transactions of the American Mathematical Society*, 205:247–262, 1975.

Edmund J Collins and John M McNamara. Finite-horizon dynamic optimisation when the terminal reward is a concave functional of the distribution of the final state. *Advances in Applied Probability*, 30(1):122–136, 1998.

Andrew R Conn, Katya Scheinberg, and Luis N Vicente. *Introduction to derivative-free optimization*, volume 8. SIAM, 2009.

Keith Conrad. Probability distributions and maximum entropy. *Entropy*, 6 (452):1–10, 2004.

Thomas M Cover and Joy A Thomas. *Elements of Information Theory*. John Wiley & Sons, 2012.

Roel Dobbe, David Fridovich-Keil, and Claire Tomlin. Fully decentralized policies for multi-agent systems: An information theoretic approach. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Ryan Doody. Lying and denying. Preprint available at `http://rdoody.com/LyingMisleading.pdf`, 2018.

Anca Dragan, Rachel Holladay, and Siddhartha Srinivasa. Deceptive robot motion: synthesis, analysis and experiments. *Autonomous Robots*, 39:331–345, 2015.

Joe Eappen and Suresh Jagannathan. DistSPECTRL: Distributing specifications in multi-agent reinforcement learning systems. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 233–250. Springer, 2023.

Kousha Etessami, Marta Kwiatkowska, Moshe Y Vardi, and Mihalis Yannakakis. Multi-objective model checking of Markov decision processes. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 50–65. Springer, 2007.

Ben Eysenbach, Russ R Salakhutdinov, and Sergey Levine. Robust predictable control. In *Advances in Neural Information Processing Systems*, volume 34. Curran Associates, Inc., 2021.

Farhad Farokhi and Henrik Sandberg. Ensuring privacy with constrained additive noise by minimizing fisher information. *Automatica*, 99:275–288, 2019.

Abraham D Flaxman, Adam T Kalai, and H. Brendan McMahan. Online convex optimization in the bandit setting: Gradient descent without a gradient. In *ACM-SIAM Symposium on Discrete Algorithms*, page 385–394. ACM, 2005.

Jie Fu. On almost-sure intention deception planning that exploits imperfect observers. In *International Conference on Decision and Game Theory for Security*, pages 67–86. Springer, 2023.

Jie Fu, Shuo Han, and Ufuk Topcu. Optimal control in Markov decision processes via distributed optimization. In *IEEE Conference on Decision and Control*, pages 7462–7469. IEEE, 2015.

Johannes Fürnkranz, Eyke Hüllermeier, Weiwei Cheng, and Sang-Hyeun Park. Preference-based reinforcement learning: a formal framework and a policy iteration algorithm. *Machine learning*, 89(1-2):123–156, 2012.

Michael R Garey and David S Johnson. *Computers and intractability: A Guide to the Theory of NP-Completeness*, volume 174. Freeman, 1979.

Donald Goldfarb and Michael J Todd. Modifications and implementation of the ellipsoid algorithm for linear programming. *Mathematical Programming*, 23(1):1–19, 1982.

Claudia V Goldman and Shlomo Zilberstein. Decentralized control of cooperative systems: Categorization and complexity analysis. *Journal of Artificial Intelligence Research*, 22:143–174, 2004.

Google. Map of San Francisco. `https://www.google.com/maps/@37.789463,-122.4068681,16.98z`. Accessed: Jan. 25, 2019.

Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, 2014.

Carlos Guestrin, Daphne Koller, and Ronald Parr. Multiagent planning with factored MDPs. In *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001.

Bingsheng He and Hai Yang. Some convergence properties of a method of multipliers for linearly constrained monotone variational inequalities. *Operations research letters*, 23(3-5):151–161, 1998.

Martin Henk. Löwner-John ellipsoids. *Documenta Math*, pages 95–106, 2012.

Zhaolin Hu and L Jeff Hong. Kullback-leibler divergence constrained distributionally robust optimization. *Optimization Online*, 1(2):9, 2013.

Romain Jacob, Jean-Jacques Lesage, and Jean-Marc Faure. Overview of discrete event systems opacity: Models, validation, and quantification. *Annual reviews in control*, 41:135–146, 2016.

Kevin G Jamieson, Robert Nowak, and Ben Recht. Query complexity of derivative-free optimization. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

Hilbert J Kappen. Path integrals and symmetry breaking for optimal control theory. *Journal of statistical mechanics: theory and experiment*, 2005(11):1–25, 2005.

Mustafa O Karabag, Melkior Ornik, and Ufuk Topcu. Least inferable policies for Markov decision processes. In *American Control Conference*, pages 1224–1231. IEEE, 2019.

Mustafa O Karabag, Cyrus Neary, and Ufuk Topcu. Smooth convex optimization using sub-zeroth-order oracles. In *AAAI Conference on Artificial Intelligence*, pages 3815–3822. AAAI, 2021a.

Mustafa O Karabag, Melkior Ornik, and Ufuk Topcu. Deception in supervisory control. *IEEE Transactions on Automatic Control*, 67(2):738–753, 2021b.

Mustafa O Karabag, Melkior Ornik, and Ufuk Topcu. Identity concealment games: How i learned to stop revealing and love the coincidences. *arXiv preprint arXiv:2105.05377*, 2021c.

Mustafa O Karabag, Cyrus Neary, and Ufuk Topcu. Planning not to talk: Multiagent systems that are robust to communication loss. In *International Conference on Autonomous Agents and Multiagent Systems*, page 705–713. IFAAMAS, 2022a.

Mustafa O Karabag, Melkior Ornik, and Ufuk Topcu. Exploiting partial observability for optimal deception. *IEEE Transactions on Automatic Control*, pages 1–8, 2022b.

Mustafa O Karabag, Melkior Ornik, and Ufuk Topcu. Deception in supervisory control. *arXiv preprint arXiv:1902.00590*, 2023.

Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

Christoforos Keroglou and Christoforos N Hadjicostis. Probabilistic system opacity in discrete event systems. *Discrete Event Dynamic Systems*, 28(2): 289–314, 2018.

Adrian König, Lorenzo Nicoletti, Daniel Schröder, Sebastian Wolff, Adam Waclaw, and Markus Lienkamp. An overview of parameter and cost for battery electric vehicles. *World Electric Vehicle Journal*, 12(1):21, 2021.

Dexter C Kozen. *Automata and computability*. Springer Science & Business Media, 2012.

Markus Krötzsch, Tomáš Masopust, and Michaël Thomazo. Complexity of universality and related problems for partially ordered NFAs. *Information and Computation*, 255:177–192, 2017.

Enoch Kung, Subhrakanti Dey, and Ling Shi. The performance and limitations of epsilon-stealthy attacks on higher order systems. *IEEE Transactions on Automatic Control*, 62(2):941–947, 2016.

Orna Kupferman and Robby Lampert. On the construction of fine automata for safety properties. In *International Symposium on Automated Technology for Verification and Analysis*, pages 110–124. Springer, 2006.

Gert Lanckriet and Bharath K. Sriperumbudur. On the convergence of the concave-convex procedure. In *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009.

Jean-Michel Lasry and Pierre-Louis Lions. Mean field games. *Japanese journal of mathematics*, 2(1):229–260, 2007.

Tor Lattimore. Improved regret for zeroth-order adversarial bandit convex optimisation. *Mathematical Statistics and Learning*, 2(3):311–334, 2020.

Tor Lattimore and Andras Gyorgy. Improved regret for zeroth-order stochastic convex bandits. In *Conference on Learning Theory*, pages 2938–2964. PMLR, 2021.

Mark Lawford and WM Wonham. Equivalence preserving transformations for timed transition models. *IEEE Transactions on Automatic Control*, 40(7): 1167–1179, 1995.

Yin Tat Lee, Aaron Sidford, and Santosh S Vempala. Efficient convex optimization with membership oracles. In *Conference On Learning Theory*, pages 1292–1294. PMLR, 2018.

Felix Leibfried and Jordi Grau-Moya. Mutual-information regularization in Markov decision processes and actor-critic learning. In *Conference on Robot Learning*, pages 360–373. PMLR, 2020.

Sergey Levine and Pieter Abbeel. Learning neural network policies with guided policy search under unknown dynamics. In *Advances in Neural Information Processing Systems*, pages 1071–1079. Curran Associates, Inc., 2014.

Alan Lewis and Tim Miller. Deceptive reinforcement learning in model-free domains. In *International Conference on Automated Planning and Scheduling*, pages 1–8. AAAI, 2023.

Dongxu Li and Jose B Cruz Jr. Information, decision-making and deception in games. *Decision Support Systems*, 47(4):518–527, 2009.

Tianyi Lin, Chi Jin, and Michael Jordan. On gradient descent ascent for nonconvex-concave minimax problems. In *International Conference on Machine Learning*, pages 6083–6093. PMLR, 2020.

Zhengshang Liu, Yue Yang, Tim Miller, and Peta Masters. Deceptive reinforcement learning for privacy-preserving planning. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 818–826. IFAAMAS, 2021.

Mark Lloyd. *The Art of Military Deception*. Pen and Sword, 2003.

Omid Madani, Steve Hanks, and Anne Condon. On the undecidability of probabilistic planning and infinite-horizon partially observable Markov decision problems. In *AAAI Conference on Artificial Intelligence*, pages 541–548. AAAI, 1999.

Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. MAVEN: multi-agent variational exploration. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Tomomi Matsui. NP-hardness of linear multiplicative programming and related problems. *Journal of Global Optimization*, 9(2):113–119, 1996.

William McEneaney and Rajdeep Singh. Deception in autonomous vehicle decision making in an adversarial environment. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*. AIAA, 2005.

Ken IM McKinnon. Convergence of the Nelder–Mead simplex method to a nonstationary point. *SIAM Journal on optimization*, 9(1):148–158, 1998.

Francisco S Melo and Manuela Veloso. Decentralized MDPs with sparse interactions. *Artificial Intelligence*, 175(11):1757–1789, 2011.

Ashutosh Nayyar, Aditya Mahajan, and Demosthenis Teneketzis. Decentralized stochastic control with partial history sharing: A common information approach. *IEEE Transactions on Automatic Control*, 58(7):1644–1658, 2013.

Cyrus Neary, Zhe Xu, Bo Wu, and Ufuk Topcu. Reward machines for cooperative multi-agent reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 934–942. IFAAMAS, 2021.

Angelia Nedić and Asuman Ozdaglar. Subgradient methods for saddle-point problems. *Journal of optimization theory and applications*, 142:205–228, 2009.

John A Nelder and Roger Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.

Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

Arkadii S Nemirovsky and David B Yudin. *Problem complexity and method efficiency in optimization.* John Wiley & Sons, 1983.

Jerzy Neyman and Egon Sharpe Pearson. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231(694-706):289–337, 1933.

Frans A Oliehoek and Christopher Amato. *A concise introduction to decentralized POMDPs.* Springer, 2016.

Melkior Ornik and Ufuk Topcu. Deception in optimal control. In *Annual Allerton Conference on Communication, Control, and Computing*, pages 821–828. IEEE, 2018.

Vera Pantelic, Steven M Postma, and Mark Lawford. Probabilistic supervisory control of probabilistic discrete event systems. *IEEE Transactions on Automatic Control*, 54(8):2013–2018, 2009.

Christos H Papadimitriou and John N Tsitsiklis. The complexity of Markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

Lynne E Parker, Daniela Rus, and Gaurav S Sukhatme. Multiple mobile robot systems. In *Springer Handbook of Robotics*, pages 1335–1384. Springer, 2016.

Apurva Patil, Mustafa O. Karabag, Takashi Tanaka, and Ufuk Topcu. Simulator-driven deceptive control via path integral approach. *(Under review)*, 2023.

Christopher J Price, Ian D Coope, and David Byatt. A convergent variant of the Nelder–Mead algorithm. *Journal of optimization theory and applications*, 113(1):5–19, 2002.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Li Qian, Jinyang Gao, and HV Jagadish. Learning user preferences by adaptive pairwise comparison. *Proceedings of the VLDB Endowment*, 8(11):1322–1333, 2015.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 4295–4304. PMLR, 2018.

Steven Rasmussen, Derek Kingston, and Laura Humphrey. A brief introduction to unmanned systems autonomy services (uxas). In *International conference on unmanned aircraft systems*, pages 257–268. IEEE, 2018.

Ishai Rosenberg, Asaf Shabtai, Yuval Elovici, and Lior Rokach. Adversarial machine learning attacks and defense methods in the cyber security domain. *ACM Computing Surveys (CSUR)*, 54(5):1–36, 2021.

Anooshiravan Saboori and Christoforos N Hadjicostis. Current-state opacity formulations in probabilistic finite automata. *IEEE Transactions on Automatic Control*, 59(1):120–133, 2013.

Yagiz Savas, Melkior Ornik, Murat Cubuktepe, Mustafa O Karabag, and Ufuk Topcu. Entropy maximization for Markov decision processes under temporal logic constraints. *IEEE Transactions on Automatic Control*, 65(4):1552–1567, 2019.

Yagiz Savas, Christos K Verginis, and Ufuk Topcu. Deceptive decision-making under uncertainty. In *AAAI Conference on Artificial Intelligence*, pages 5332–5340, 2022.

Ohad Shamir. On the complexity of bandit and derivative-free stochastic convex optimization. In *Conference on Learning Theory*, pages 3–24. PMLR, 2013.

Jaeeun Shim and Ronald C Arkin. A taxonomy of robot deception and its benefits in HRI. In *International Conference on Systems, Man, and Cybernetics*, pages 2328–2335. IEEE, 2013.

Naum Z Shor. Utilization of the operation of space dilatation in the minimization of convex functions. *Cybernetics*, 6(1):7–15, 1972.

Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 5887–5896. PMLR, 2019.

Richard Edwin Stearns and Harry B Hunt III. On the equivalence and containment problems for unambiguous regular expressions, regular grammars and finite automata. *SIAM Journal on Computing*, 14(3):598–611, 1985.

Larry J Stockmeyer and Albert R Meyer. Word problems requiring exponential time (preliminary report). In *Annual ACM Symposium on Theory of Computing*, pages 1–9. ACM, 1973.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 2085–2087. IFAAMAS, 2018.

Takashi Tanaka, Henrik Sandberg, and Mikael Skoglund. Transfer-entropy-regularized Markov decision processes. *IEEE Transactions on Automatic Control*, 67(4):1944–1951, 2021.

Emanuel Todorov. Linearly-solvable markov decision problems. *Advances in Neural Information Processing Systems*, 19, 2006.

Emanuel Todorov. Efficient computation of optimal actions. *Proceedings of the national academy of sciences*, 106(28):11478–11483, 2009.

Leslie G Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.

Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

Rundong Wang, Xu He, Runsheng Yu, Wei Qiu, Bo An, and Zinovi Rabinovich. Learning efficient multi-agent communication: An information bottleneck approach. In *International Conference on Machine Learning*, pages 9908–9918. PMLR, 2020.

Satosi Watanabe. Information theoretical analysis of multivariate correlation. *IBM Journal of research and development*, 4(1):66–82, 1960.

Aaron Wilson, Alan Fern, and Prasad Tadepalli. A Bayesian approach for policy learning from trajectory preference queries. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

Christian Wressnegger, Guido Schwenk, Daniel Arp, and Konrad Rieck. A close look on n-grams in intrusion detection: anomaly detection vs. classification. In *ACM workshop on Artificial intelligence and security*, pages 67–76. ACM, 2013.

Feng Wu, Shlomo Zilberstein, and Xiaoping Chen. Online planning for multi-agent systems with bounded communication. *Artificial Intelligence*, 175(2): 487–511, 2011.

David B Yudin and Arkadii S Nemirovskii. Evaluation of the information complexity of mathematical programming problems. *Ekonomika i Matematicheskie Metody*, 12:128–142, 1976.

Alan L Yuille and Anand Rangarajan. The concave-convex procedure (CCCP). In *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001.

Jun Zhang, Lei Pan, Qing-Long Han, Chao Chen, Sheng Wen, and Yang Xiang. Deep learning based attack detection for cyber-physical system cybersecurity: A survey. *IEEE/CAA Journal of Automatica Sinica*, 9(3):377–391, 2021.

Tao Zhang and Quanyan Zhu. Hypothesis testing game for cyber deception. In *International Conference on Decision and Game Theory for Security*, pages 540–555. Springer, 2018.