# The Complexity of Reinforcement Learning with Linear Function Approximation

*Gellert Weisz*

A dissertation submitted in partial fulfillment

of the requirements for the degree of

**Doctor of Philosophy**

of

**University College London**.

Department of Electronic & Electrical Engineering

University College London

Monday 5th February, 2024

I, Gellert Weisz, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

# Abstract

In this thesis we present contributions to the theoretical foundations of large-scale **reinforcement learning** (**RL**) with **linear function approximation**, with a focus on establishing classes of problems that are theoretically solvable in polynomial time and ones that are not. The problem classes differ in learning paradigm and structural assumptions. We start with the problem of **planning** under $q^\star$**-realizability** (i.e., when the optimal action-value function lies in the span of a feature map), and establish exponential statistical hardness of this class. Next, we consider $v^\star$**-realizability** with a small constant number of actions, and establish an algorithm that solves the planning problem with a polynomial query complexity. Next, we present a computationally efficient algorithm that improves on the state of the art query complexity guarantees for planning under $q^\pi$**-realizability** (i.e., when $q$-values for all policies lie in the span of a feature map). Finally, we present the first algorithm that solves **online RL** under $q^\pi$-realizability with a polynomial query complexity, establishing that this problem class is statistically tractable.

# Impact Statement

All the results presented in this thesis is of a theoretical nature. As such, they do not target any specific application area. However, these results can inspire further research and algorithms targeting sequential decision-making and representation learning. As for the impact already observed, our negative results won the best student paper award at Algorithmic Learning Theory, 2021, and it are quoted in the book "Reinforcement learning: Theory and algorithms" (Agarwal et al., 2019) as a "breakthrough result". In particular, a consequence of these negative results is that reinforcement learning is fundamentally harder than supervised learning, and one should not expect the structure of $q^\star$-realizability yield tractable learning problems. In other words, one should not hope that representation learning targeting only the optimal action-value function would be the key to scaling up RL, and different (likely stronger) structural assumptions are required. This finding contrasts with the algorithms presented for $q^\pi$-realizability, which in turn implies that capturing representational structure for all $q^\pi$ functions is sufficient at scaling RL to possibly infinite state-spaces, at least from a statistical perspective. These results can thus be seen as steps towards establishing the key structural assumptions that unlock practical scaling of RL methods to real-life problems with large state spaces.

# Acknowledgements

I extend my heartfelt thanks to the individuals who have been instrumental in my PhD journey.

Foremost, I express my deepest gratitude to Professor Csaba Szepesvári, my supervisor from DeepMind. His unwavering belief in my abilities, along with his guidance and encouragement, have been invaluable. I am profoundly thankful for his mentorship.

I am also indebted to my manager, András György, whose support and insights significantly contributed to my research. His guidance provided me with crucial perspectives and helped shape my work.

I want to acknowledge the indispensable assistance of Tor Lattimore. His knowledge and willingness to assist were invaluable, and I am grateful for his contributions to my research.

I appreciate the support and guidance of my official UCL supervisor, Ilija Bugonovic. His feedback and suggestions were invaluable in refining my work.

On a personal note, I owe a debt of gratitude to my partner, Luke Egan, and my family. Their unwavering support, understanding, and encouragement sustained me through the challenges of this academic endeavor.

To everyone mentioned above, your belief in me and your support have been the cornerstone of my success. I am sincerely thankful for your contributions to my academic and personal growth.

# Declarations

## UCL Research Paper Declaration Form: referencing the doctoral candidates own published work(s)

1. **1. For a research manuscript that has already been published** (if not yet published, please skip to section 2)**:**

   (a) **What is the title of the manuscript?** Exponential Lower Bounds for Planning in MDPs With Linearly-Realizable Optimal Action-Value Functions

   (b) **Please include a link to or doi for the work:** https://proceedings.mlr.press/v132/weisz21a.html

   (c) **Where was the work published?** Conference on Algorithmic Learning Theory

   (d) **Who published the work?** Proceedings of the 32nd International Conference on Algorithmic Learning Theory, PMLR 132:1237-1264, 2021.

   (e) **When was the work published?** 2021

   (f) **List the manuscript's authors in the order they appear on the publication:** Gellert Weisz, Philip Amortila, Csaba Szepesvári

   (g) **Was the work peer reviewed?** Yes

   (h) **Have you retained the copyright?** Yes

   (i) **Was an earlier form of the manuscript uploaded to a preprint server (e.g. medRxiv)? If Yes, please give a link or doi** No

   If No, please seek permission from the relevant publisher and check the box next to the below statement:

   ☒ *I acknowledge permission of the publisher named under 1d to include in this thesis portions of the publication named as included in 1c.*

2. **For a research manuscript prepared for publication but that has not yet been published** (if already published, please skip to section 3)**:**

   (a) **What is the current title of the manuscript?**

   (b) **Has the manuscript been uploaded to a preprint server 'e.g. medRxiv'?**
   **If 'Yes', please please give a link or doi:**

   (c) **Where is the work intended to be published?**

   (d) **List the manuscript's authors in the intended authorship order:**

   (e) **Stage of publication:**

3. **For multi-authored work, please give a statement of contribution covering all authors** (if single-author, please skip to section 4)**:** Gellert Weisz: main author (responsible for research ideas, proofs, write-up); Philip Amortila: co-author (responsible for research ideas, proofs, write-up); Csaba Szepesvari: supervisory roles (responsible for strategic guidance, write-up)

4. **In which chapter(s) of your thesis can this material be found?** Chapter 1

**e-Signatures confirming that the information above is accurate** (this form should be co-signed by the supervisor/ senior author unless this is not appropriate, e.g. if the paper was a single-author work)**:**

**Candidate:** [redacted]
**Date:** Monday 5th February, 2024

**Supervisor/Senior Author signature** (where appropriate)**:** [redacted]
**Date:** Monday 5th February, 2024

## UCL Research Paper Declaration Form: referencing the doctoral candidates own published work(s)

1. **1. For a research manuscript that has already been published** (if not yet published, please skip to section 2)**:**

   (a) **What is the title of the manuscript?** On Query-efficient Planning in MDPs under Linear Realizability of the Optimal State-value Function

   (b) **Please include a link to or doi for the work:** https://proceedings.mlr.press/v134/weisz21a.html

   (c) **Where was the work published?** Conference on Learning Theory

   (d) **Who published the work?** Proceedings of Thirty Fourth Conference on Learning Theory, PMLR 134:4355-4385, 2021.

   (e) **When was the work published?** 2021

   (f) **List the manuscript's authors in the order they appear on the publication:** Gellert Weisz, Philip Amortila, Barnabás Janzer, Yasin Abbasi-Yadkori, Nan Jiang, Csaba Szepesvari

   (g) **Was the work peer reviewed?** Yes

   (h) **Have you retained the copyright?** Yes

   (i) **Was an earlier form of the manuscript uploaded to a preprint server (e.g. medRxiv)? If Yes, please give a link or doi** No

   If No, please seek permission from the relevant publisher and check the box next to the below statement:

   ☒ *I acknowledge permission of the publisher named under 1d to include in this thesis portions of the publication named as included in 1c.*

2. **For a research manuscript prepared for publication but that has not yet been published** (if already published, please skip to section 3)**:**

   (a) **What is the current title of the manuscript?**

   (b) **Has the manuscript been uploaded to a preprint server 'e.g. medRxiv'?**

   **If 'Yes', please please give a link or doi:**

   (c) **Where is the work intended to be published?**

   (d) **List the manuscript's authors in the intended authorship order:**

(e) **Stage of publication:**

3. **For multi-authored work, please give a statement of contribution covering all authors** (if single-author, please skip to section 4)**:** Gellert Weisz: main author (responsible for research ideas, proofs, write-up); Philip Amortila: co-author (responsible for research ideas, proofs, write-up); Barnabás Janzer: co-author (responsible for solving the problem presented as Problem 1.4.3 in this thesis); Yasin Abbasi-Yadkori: co-author (responsible for the lower bound presented as Theorem 3.1 in the paper, and its proof in Appendix B); Nan Jiang, Csaba Szepesvari: supervisory roles (responsible for strategic guidance, write-up)
4. **In which chapter(s) of your thesis can this material be found?** Chapters 1 and 3

**e-Signatures confirming that the information above is accurate** (this form should be co-signed by the supervisor/ senior author unless this is not appropriate, e.g. if the paper was a single-author work)**:**

**Candidate:** [redacted]

**Date:** Monday 5th February, 2024

**Supervisor/Senior Author signature** (where appropriate)**:** [redacted]

**Date:** Monday 5th February, 2024

## UCL Research Paper Declaration Form: referencing the doctoral candidates own published work(s)

1. **1. For a research manuscript that has already been published** (if not yet published, please skip to section 2)**:**

   (a) **What is the title of the manuscript?** TensorPlan and the Few Actions Lower Bound for Planning in MDPs under Linear Realizability of Optimal Value Functions

   (b) **Please include a link to or doi for the work:** https://proceedings.mlr.press/v167/weisz22a.html

   (c) **Where was the work published?** Conference on Algorithmic Learning Theory

   (d) **Who published the work?** Proceedings of The 33rd International Conference on Algorithmic Learning Theory, PMLR 167:1097-1137, 2022.

   (e) **When was the work published?** 2022

   (f) **List the manuscript's authors in the order they appear on the publication:** Gellert Weisz, Csaba Szepesvári, András György

   (g) **Was the work peer reviewed?** Yes

   (h) **Have you retained the copyright?** Yes

   (i) **Was an earlier form of the manuscript uploaded to a preprint server (e.g. medRxiv)? If Yes, please give a link or doi** No

   If No, please seek permission from the relevant publisher and check the box next to the below statement:

   ☒ *I acknowledge permission of the publisher named under 1d to include in this thesis portions of the publication named as included in 1c.*

2. **For a research manuscript prepared for publication but that has not yet been published** (if already published, please skip to section 3)**:**

   (a) **What is the current title of the manuscript?**

   (b) **Has the manuscript been uploaded to a preprint server 'e.g. medRxiv'?**
   **If 'Yes', please please give a link or doi:**

   (c) **Where is the work intended to be published?**

   (d) **List the manuscript's authors in the intended authorship order:**

   (e) **Stage of publication:**

3. **For multi-authored work, please give a statement of contribution covering all authors** (if single-author, please skip to section 4)**:** Gellert Weisz: main author (responsible for research ideas, proofs, write-up); Csaba Szepesvári and András György: supervisory roles (responsible for strategic guidance, write-up)
4. **In which chapter(s) of your thesis can this material be found?** Chapters 1 to 3

**e-Signatures confirming that the information above is accurate** (this form should be co-signed by the supervisor/ senior author unless this is not appropriate, e.g. if the paper was a single-author work)**:**

**Candidate:** [redacted]
**Date:** Monday 5th February, 2024

**Supervisor/Senior Author signature** (where appropriate)**:** [redacted]
**Date:** Monday 5th February, 2024

## UCL Research Paper Declaration Form: referencing the doctoral candidates own published work(s)

1. **1. For a research manuscript that has already been published** (if not yet published, please skip to section 2)**:**

   (a) **What is the title of the manuscript?** Confident Approximate Policy Iteration for Efficient Local Planning in $q^\pi$-realizable MDPs

   (b) **Please include a link to or doi for the work:** https://proceedings.neurips.cc/paper_files/paper/2022/hash/a3bfb116214815682a0d0d88ea95cd12-Abstr
   html

   (c) **Where was the work published?** Neural Information Processing Systems

   (d) **Who published the work?** Advances in Neural Information Processing Systems 35 (2022): 25547-25559.

   (e) **When was the work published?** 2022

   (f) **List the manuscript's authors in the order they appear on the publication:** Gellert Weisz, András György, Tadashi Kozuno, Csaba Szepesvari

   (g) **Was the work peer reviewed?** Yes

   (h) **Have you retained the copyright?** Yes

   (i) **Was an earlier form of the manuscript uploaded to a preprint server (e.g. medRxiv)? If Yes, please give a link or doi** No

   If No, please seek permission from the relevant publisher and check the box next to the below statement:

   ☒ *I acknowledge permission of the publisher named under 1d to include in this thesis portions of the publication named as included in 1c.*

2. **For a research manuscript prepared for publication but that has not yet been published** (if already published, please skip to section 3)**:**

   (a) **What is the current title of the manuscript?**

   (b) **Has the manuscript been uploaded to a preprint server 'e.g. medRxiv'?**

   **If 'Yes', please please give a link or doi:**

   (c) **Where is the work intended to be published?**

   (d) **List the manuscript's authors in the intended authorship order:**

(e) **Stage of publication:**

3. **For multi-authored work, please give a statement of contribution covering all authors** (if single-author, please skip to section 4)**:** Gellert Weisz: main author (responsible for research ideas, proofs, write-up); Tadashi Kozuno: co-author (responsible for the lower bounds in Appendix H of the paper); András György, Csaba Szepesvari: supervisory roles (responsible for strategic guidance, write-up)
4. **In which chapter(s) of your thesis can this material be found?** Chapters 1 and 4

**e-Signatures confirming that the information above is accurate** (this form should be co-signed by the supervisor/ senior author unless this is not appropriate, e.g. if the paper was a single-author work)**:**

**Candidate:** [redacted]
**Date:** Monday 5th February, 2024

**Supervisor/Senior Author signature** (where appropriate)**:** [redacted]
**Date:** Monday 5th February, 2024

## UCL Research Paper Declaration Form: referencing the doctoral candidates own published work(s)

1. **1. For a research manuscript that has already been published** (if not yet published, please skip to section 2)**:**

   (a) **What is the title of the manuscript?** Online RL in Linearly $q^\pi$-Realizable MDPs Is as Easy as in Linear MDPs If You Learn What to Ignore

   (b) **Please include a link to or doi for the work:** https://neurips.cc/virtual/2023/poster/72161

   (c) **Where was the work published?** Neural Information Processing Systems

   (d) **Who published the work?** Advances in Neural Information Processing Systems 36 (NeurIPS 2023)

   (e) **When was the work published?** 2023

   (f) **List the manuscript's authors in the order they appear on the publication:** Gellert Weisz, András György, Csaba Szepesvari

   (g) **Was the work peer reviewed?** Yes

   (h) **Have you retained the copyright?** Yes

   (i) **Was an earlier form of the manuscript uploaded to a preprint server (e.g. medRxiv)? If Yes, please give a link or doi** No

   If No, please seek permission from the relevant publisher and check the box next to the below statement:

   ☒ *I acknowledge permission of the publisher named under 1d to include in this thesis portions of the publication named as included in 1c.*

2. **For a research manuscript prepared for publication but that has not yet been published** (if already published, please skip to section 3)**:**

   (a) **What is the current title of the manuscript?**

   (b) **Has the manuscript been uploaded to a preprint server 'e.g. medRxiv'?**
   **If 'Yes', please please give a link or doi:**

   (c) **Where is the work intended to be published?**

   (d) **List the manuscript's authors in the intended authorship order:**

   (e) **Stage of publication:**

3. **For multi-authored work, please give a statement of contribution covering all authors** (if single-author, please skip to section 4)**:** Gellert Weisz: main author (responsible for research ideas, proofs, write-up); András György, Csaba Szepesvari: supervisory roles (responsible for strategic guidance, write-up)

4. **In which chapter(s) of your thesis can this material be found?** Chapters 1 and 5

**e-Signatures confirming that the information above is accurate** (this form should be co-signed by the supervisor/ senior author unless this is not appropriate, e.g. if the paper was a single-author work)**:**

**Candidate:** [redacted]

**Date:** Monday 5th February, 2024

**Supervisor/Senior Author signature** (where appropriate)**:** [redacted]

**Date:** Monday 5th February, 2024

# Contents

## Chapter 1

# Overview

## 1.1. Background

For the purposes of this overview, we assume that the reader is familiar with the reinforcement learning (RL) problem and Markov decision processes (MDPs), and related concepts, for which the precise definitions are given in Section 1.2. A great many problems of interest can be formulated as optimal sequential decision making in a stochastic environment. If the model of the environment is given, perhaps because it has been learned with a sufficient accuracy, one only has to figure out how to use the model to find good actions. This is the problem addressed by planning algorithms (Chapter 6 Mausam and Kolobov, 2012). An elegant, minimalist approach to describe such problems is to adopt the language of MDPs. Dynamic programming methods in MDPs with $S$ states, $A$ actions and a horizon of $H$ can solve the planning problem with $\text{poly}(S, A, H)$ resources (Littman et al., 1995; Kallenberg, 2002; Ye, 2011; Scherrer, 2016; Sidford et al., 2023). However, in the lack of extra information, the required resources grow at least as $\Omega(A^H)$ when the number of states is unbounded (Kearns et al., 2002). The price of simplicity (and thus generality) is therefore that efficient planning in large state-spaces is intractable, a phenomenon pointed out by Bellman (1957) and today informally referred to as Bellman's curse of dimensionality. An intriguing approach to avoid intractability when both $S$ and $H$ are large is the use of "function approximation" which promises to empower planners to extrapolate beyond the states that the planner has encountered. This approach has been proposed shortly after MDPs have been introduced when it was observed that in various problems of practical interest, value functions that the dynamic programming algorithms aim to compute can be well approximated with the linear combination of only a few basis functions, which themselves can be guessed by studying the structure of the problem to be solved (Bellman et al., 1963; Schweitzer and Seidmann, 1985). This raises the question of whether under such a favorable condition a provably efficient planner exist, i.e., whether the curse can be lifted.

While this question was arguably one of the main driving forces behind much of the research in both operations research and reinforcement learning since the 1960s, most of the early results focused on the case when the function space underlying the features have a certain completeness property when dynamic programming algorithms can be successfully adopted (e.g., Bertsekas and Tsitsiklis, 1996; Tsitsiklis and Van Roy, 1996; Munos, 2003, 2005; Szepesvári and Munos, 2005). For more recent works in this, and some other related directions, see, e.g., (Du et al., 2019a; Lattimore et al., 2020; Du et al., 2021) and the references therein.

While interesting, these works left open the question of whether efficient planners exist in the case when the function space may lack the completeness property but is still able to represent the optimal value function. Though Wen and Van Roy (2013) addresses this for deterministic systems, the focus of Chapters 2 and 3 is to investigate this question for stochastic systems. In Chapter 4 we investigate planning when the function space can represent all value functions, an assumption that is still weaker than the completeness assumption. Finally, Chapter 5 removes the need for a simulator and tackles this problem under the setting of online RL.

## 1.2. Notation and problem setup

The purpose of this section is to introduce the notation we use and the necessary definitions that will allow us to precisely formulate the problems we study. We start with the notation. This is followed by a quick review of definitions and basic concepts concerning MDPs. The section will be closed by describing the main problems considered.

### 1.2.1. Notation

Let $\mathbb{N}_+ = \{1, 2, \dots\}$ be the set of positive integers, and $\mathbb{N} = \{0\} \cup \mathbb{N}_+$. Let $\mathbb{R}$ denote the set of real numbers, $\mathcal{B}_d(r) = \{x \in \mathbb{R}^d : \|x\|_2 \le r\}$ the $d$-dimensional ball of radius $r$, and let $[i] = \{1, \dots, i\}$ be the set of integers from 1 to $i$ for an integer $i \in \mathbb{N}_+$. For $i, j \in \mathbb{N}$, we use $[i : j] = \{i, i+1, \dots, j\}$ if $i \le j$, and $[i : j] = \{\}$ otherwise. For $x \in \mathbb{R}$, $\lfloor x \rfloor$ is the largest integer that is at most $x$. For vectors $a$ and $b$ of compatible sizes, $\langle a, b \rangle = a^\top b$ denotes their inner product. For a True or False statement $X$ (possibly depending on random variables), let $\mathbb{I}\{X\}$ take 1 if $X$ is True, and 0 otherwise. Let $a \wedge b = \min(a, b)$ and $a \vee b = \max(a, b)$. For an event $E$, let $E^C$ denote its complementary event. Let () denote the empty sequence. We use $\mathcal{O}$ and $\Omega$ from the big-O notation, and we denote by $\tilde{\mathcal{O}}$ the variant of $\mathcal{O}$ that hides polylogarithmic factors.

For (column) vectors $M_1, M_2, \dots$, let us denote by $[M_1, M_2, \dots]$ their concatenation $(M_1^\top, M_2^\top, \dots)^\top$. Let $\flat(M)$ map a tensor of any rank $m$ and any shape $d_1 \times d_2 \times \dots \times d_m$ to the

vector of dimension $\prod_{i\in[m]} d_i$ by laying out its elements in a canonical order. Let $\otimes$ denote the tensor product. We will use the following result to linearize products of vectors:

**Lemma 1.2.1.** *For any positive integer $n$ and any vectors $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$ of equal dimension:*

$$\langle a_1, b_1\rangle \langle a_2, b_2\rangle \ldots \langle a_n, b_n\rangle = \langle \flat(a_1 \otimes a_2 \otimes \cdots \otimes a_n), \flat(b_1 \otimes b_2 \otimes \cdots \otimes b_n)\rangle \,.$$

### 1.2.2. Episodic Markov decision processes with bounded rewards

A **Markov decision process (MDP)** is defined by a tuple $M = (\mathcal{S}, \mathcal{A}, Q)$ of states, actions, and a transition-reward-kernel, respectively. The structure $M$ defines a discrete time sequential decision making problem where in time step $t = 0, 1, \ldots,$ an environment responds to an action $A_t \in \mathcal{A}$ of an agent by transitioning from its current state $S_t \in \mathcal{S}$ to a new random state $S_{t+1} \in \mathcal{S}$ while also generating a random reward $R_{t+1} \in \mathbb{R}$ so that the distribution of $(R_{t+1}, S_{t+1})$ given $S_0, A_0, R_1, S_1, \ldots, A_{t-1}, R_t, S_t, A_t$ is given by $Q(\cdot \,|\, S_t, A_t)$ regardless of the history before $S_t, A_t$. Formally, $Q$ is a probability kernel from state-action pairs to reward-state pairs. For simplicity, it is assumed that $R_{t+1}$ above is supported in $[0, 1]$.

To simplify the presentation, we assume that the state space is finite, noting that all the definitions and results presented in this thesis can be naturally translated to infinite state spaces by using the more technical definitions of Weisz et al. (2021a) included in Chapter 3, that involve measure theoretic considerations. Similarly, assume that the set of actions is finite and $\mathcal{A} = [A]$ for some integer $A$.

In this thesis we focus on the **fixed-horizon undiscounted total expected reward objective**. Denoting the horizon by $H$, under this objective, the goal is to find a **policy**, a way of choosing actions given the past, such that the total expected reward over $H$ steps is maximized regardless of the initial state of the process. More formally, a policy defines a probability distribution over actions ($\mathcal{A}$) given the history of $S_0, A_0, R_1, S_1, \ldots, S_t$ for $0 \le t < H$. The $H$ steps of the process is also called an **episode**. As it is well known, the optimal policy, which maximizes the stated objective, depends on the number of steps left before the episode finishes. In this thesis, we will use an equivalent formulation which avoids this dependence. In this formulation, only the first $H$ rewards can be non-zero, while the process continues indefinitely and the objective is changed to the total undiscounted expected reward. To emulate the fixed-horizon setting, one can then create $H$ disjoint copies of the state space, each corresponding to one step of the process while copying the transition structure to

transition from one copy to the next one, and add an extra state ($\bot$) such that after $H$ steps this state is reached from which point this state is never left while the reward incurred remains zero regardless of the actions taken. This is summarized below:

**Definition 1.2.2** (Fixed-horizon MDP)**.** *The state space $\mathcal{S}$ satisfies $\mathcal{S} = \cup_{h=0}^{H} \mathcal{S}_h$ with pairwise disjoint sets $\{\mathcal{S}_h\}_{h\in[0:H]}$, with $\mathcal{S}_H = \{\bot\}$, and with $Q$ is such that for any $s \in \mathcal{S}_h$, $h \in [0 : H-1]$ and $a \in \mathcal{A}$, $Q(\cdot|s,a)$ is supported on $[0,1] \times (\{\bot\} \cup \mathcal{S}_{h+1})$, while for $h = H$, this support is $\{0\} \times \mathcal{S}_H = \{0\} \times \{\bot\}$.*

Thanks to this assumption, when writing definitions, we can consider the infinite horizon total expected reward criterion. This latter criterion assigns to a policy $\pi$ used in MDP $M$ from initial state $s \in \mathcal{S}$ the value $v^\pi(s)$, which is defined as

$$v^\pi(s) = \mathbb{E}^\pi_{M,s}\left[\sum_{t=0}^{\infty} R_{t+1}\right]. \tag{1}$$

Here $\mathbb{E}^\pi_{M,s}$ is the expectation corresponding to the probability distribution $\mathbb{P}^\pi_{M,s}$ over trajectories of infinite length composed of state-action-reward triplets where this probability distribution arises from using policy $\pi$ in every step, with the first state fixed to $s$, while next states and rewards are generated according to $Q$. We will also need the action-value function of a policy. This is defined similarly as above, except that one fixes both the initial state and the initial action. Thus, for $(s,a) \in \mathcal{S} \times \mathcal{A}$,

$$q^\pi(s,a) = \mathbb{E}^\pi_{M,s,a}\left[\sum_{t=0}^{\infty} R_{t+1}\right]. \tag{2}$$

where $\mathbb{E}^\pi_{M,s,a}$ is the expectation corresponding to the probability distribution $\mathbb{P}^\pi_{M,s,a}$ over the trajectories as before, except that this time the first state-action pair is fixed to $(s,a)$ instead of just fixing the first state to $s$. Note that by Definition 1.2.2, both $v^\pi$ (mapping states to reals, the **value function of** $\pi$) and $q^\pi$ (mapping state-action pairs to reals, the **action-value function of** $\pi$) are well defined and take values in $[0,H]$ and the infinite sums can be truncated after stage $H$.

Define $v^\star : \mathcal{S} \to \mathbb{R}$ and $q^\star : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, the **optimal value** and, respectively, **optimal action-value function** as

$$v^\star(s) = \sup_\pi v^\pi(s), \qquad q^\star(s,a) = \sup_\pi q^\pi(s), \qquad s \in \mathcal{S}, a \in \mathcal{A}. \tag{3}$$

A policy $\pi$ is said to be optimal if $v^\star = v^\pi$. It is well known (Puterman, 1994) that in our setting an optimal policy always exists and in fact the policy that uses any maximizer of $q^\star(s,\cdot)$ when the state $S_t$ is $s$ is an optimal policy. This policy, as the choice of the action only depends on the last state, is called **memoryless**. Since the choice is also deterministic, the policy is also **deterministic**. A deterministic memoryless policy can be concisely given as a map from states to actions. By slightly abusing notation, in what follows, we will identify such policies with such maps and write $\pi : \mathcal{S} \to \mathcal{A}$ to denote a deterministic memoryless policy. Given such a policy $\pi : \mathcal{S} \to \mathcal{A}$, its value functions $v^\pi$ and $q^\pi$ satisfy the following equations:

$$v^\pi(s) = q^\pi(s, \pi(s)), \tag{4}$$

$$q^\pi(s,a) = r(s,a) + \sum_{s' \in \mathcal{S}} P(s'|s,a) v^\pi(s'), \qquad s \in \mathcal{S}, a \in \mathcal{A}, \tag{5}$$

where $P(s'|s,a)$, derived from the transition kernel $Q$, is the probability of arriving at state $s'$ when the process is in state $s$ and action $a$ is taken while $r(s,a)$ is the expected reward along this transition. Formally, $P(s'|s,a) = Q([0,1] \times \{s'\}|s,a)$ and $r(s,a) = \int_{[0,1]\times\mathcal{S}} r dQ(r,s'|s,a)$. The coupled equations Eq. (5) are known as the Bellman equations for $\pi$ (Puterman, 1994).

Oftentimes in MDPs the rewards and the next states are independently chosen. In this case, $Q(\cdot|s,a)$ takes the form of the "product" of a probability kernel $R$ mapping state-action pairs to $[0,1]$ and the probability kernel $P$ mapping state-action pairs to states. In some constructions below, we will thus specify an MDP with the help of two such kernels.

When the dependence of $v^\pi$, $v^\star$, or $q^\star$ on MDP $M$ is to be emphasized, we put $M$ in the index of these symbols. For example, for a policy $\pi$ for $M$, we will write $v^\pi_M$ to denote its value function in $M$.

### 1.2.3. Online RL and planning

We first introduce the setting of online RL, before introducing the more permissive setting of planning with access to a simulator. Online RL is the suitable framework when a simulator of the MDP is not available, or when the available simulator does not have the capability to reset the state of the simulation to any state apart from a dedicated start state. Note that resetting to the start state is a natural requirement when solving fixed-horizon MDPs (as such a reset happens whenever a new episode starts), so we consider online RL the most challenging model.

In online RL, given a $H$-horizon MDP $M = (\mathcal{S}, \mathcal{A}, Q)$ satisfying Definition 1.2.2, an **agent** is given direct access to the MDP by providing its current reward, state, and associated features, and

the capability for the agent to take any (possibly random) action in the current state of the MDP. The MDP is initialized and reset to a dedicated initial state $s_0 \in \mathcal{S}_0$ after every episode (every $H$ steps). The agent $P$ may stop at any time and output a memoryless policy $\pi_P$, which we recall is a mapping from the current state to a probability distribution over actions. At this point, the agent may not learn or update its output policy anymore. The agent $P$ is $\delta$-sound for a class of MDPs $\mathcal{M}$, if for any $M \in \mathcal{M}$, when deploying the agent in MDP $M$ as described above, its output policy is $\delta$-optimal in expectation from the $M$'s dedicated initial state $s_0$, that is,

$$\mathbb{E}_{\pi_P} v_M^{\pi_P}(s_0) \geq v_M^\star(s_0) - \delta\,,$$

where the expectation is over the agent's output policy. Another important metric to measure agents is their their expected number of actions they take on their underlying MDP before returning with a policy. We call this the query cost of an agent $P$ with an MDP $M$.

In contrast with online RL, our setting of **planning** involves a **simulation oracle** that can be **queried** with state action pairs $(s,a) \in \mathcal{S} \times \mathcal{A}$, to which the oracle responds with a reward-state pair $(R', S')$ generated from $Q$:

$$(R', S') \sim Q(\cdot|s,a)\,.$$

In our planning setting, a **planner** is used in a closed-loop configuration: in step $t \in [0 : H-1]$ of using the planner, the planner is given access to state $S_t$ of the process. By convention, $S_0 \in \mathcal{S}_0$ and thus we also have $S_t \in \mathcal{S}_t$, for every $t$, where $(\mathcal{S}_h)_{0 \leq h \leq H}$ is the decomposition of $\mathcal{S}$ from Definition 1.2.2. In each step $t$ of using the planner, the planner is given access to the simulation oracle, with the freedom to decide which queries and how many of them to use before it returns. Eventually, the planner needs to stop querying and return an action $A_t \in \mathcal{A}$, which is then used to generate the next state in $M$ and an associated reward:

$$(R_{t+1}, S_{t+1}) \sim Q(\cdot|S_t, A_t)\,.$$

The planner is then called with state $S_{t+1}$ for step $t+1$, and the process repeats until the episode is over (i.e., $t = H$). Using a planner $P$ in this closed-loop configuration is equivalent to running the planner-induced policy $\pi_P$ in MDP $M$, where the planner-induced policy responds to any history by calling the planner with the current state and relaying its output. Defined in this way, there is

an important distinction between planner-induced policies and the regular policies defined before: planner-induced policies, while executing the underlying planner, have access to the simulation oracle as described above, and to any memory that the planner saves (which is persisted between calls). In contrast, recall that the memoryless policies that agents are allowed to output in the online RL setting are simple mappings from states to action distributions. Observe that $\pi_P$ is a stochastic policy (possibly history-dependent, if the planner saves information between its calls), where the stochasticity comes from the randomness of the entire planner-oracle interaction (and possibly some independent randomization). However, note that $\pi_P$ itself is not a random element, in contrast to the output policy of an agent in the online RL setting.

When the process is started from the dedicated initial state $s_0$, the goal of the planner is to maximize total expected reward incurred in the episode, or equivalently, to maximize $v_M^{\pi_P}(s_0)$. Analogously to $\delta$-sound agents in the online RL setting, we introduce the concept of $\delta$-**sound** planners: a planner $P$ is $\delta$-sound for a class $\mathcal{M}$ of MDPs if for any $M \in \mathcal{M}$, when deploying the planner in MDP $M$ in the closed-loop fashion described above, its induced policy is $\delta$-optimal from the $M$'s dedicated initial state $s_0$, that is,

$$v_M^{\pi_P}(s_0) \geq v_M^{\star}(s_0) - \delta\,.$$

Analogously to agents, a planner $P$ also has a query cost with an MDP $M$, which is defined as the worst-case expected number of queries that the planner uses in a planning step (or call). Here, the worst-case is over all possible calls to the planner.

### 1.2.4. Featurized MDPs

As mentioned in the introduction, Without any additional information or structure, for a general class of MDPs, even $\frac{1}{2}$-sound planners must have a query cost of $\min\{|\mathcal{S}|, A^H\}$ (Kearns et al., 2002). Intuitively, this is because planners and agents need to learn about every stage separately, and this knowledge does not transfer to a new, unseen state. To scale our methods to large, possibly infinite state spaces, we therefore need to introduce additional structure. We do this in the form of a **feature map** that comes with an MDP. This maps states or state-action pairs of the MDP to $\mathbb{R}^d$, with the goal of describing the important aspects of the state or state-action pair, so that the planner or agent could generalize based on information it acquires in this $d$-dimensional space. In this setting, the number of states can be arbitrarily large, but $d$ is manageably small in the sense that a polynomial query cost in $d$ is considered efficient, while a polynomial query cost in $|\mathcal{S}|$ is infeasible.

The agent or algorithm interacting with an MDP has access to its corresponding feature maps according to an **access model**. In the case of **local access**, which can be used either with online RL or planning, the agent (or planner, respectively) can observe state features (or state-action features, depending on which is available) corresponding to each state encountered during the process (i.e., the state it is called with, and any state returned by the simulator or the MDP). In the case of planning under local access, another important restriction is that the planner can query the simulator for transitions and associated features **only at states previously encountered**. In the alternative access model of **global access** (often referred to as **generative model**), the planner is given the set of all states and associated features in advance (no queries required), and the option to query the simulator for transitions and rewards at any state of its choice. This model is only used with planning and not with online RL, and it is the most permissive setting considered. As we will see, the access model plays an important role, as there are problems that can be efficiently solved with global access, but not with local access. Intuitively, this is because global access relies on receiving (and potentially pre-processing) an amount of data that is polynomial in $|\mathcal{S}|$ at a query cost of zero, a somewhat unrealistic setting.

### 1.2.5. Query complexity

For some $\delta > 0$, denote the set of $\delta$-sound agents for some class of featurized MDPs $\mathcal{M}$ under the online RL, local access model by $\mathcal{P}_{\text{RL}}(\mathcal{M}, \delta)$. As online RL is only used with the local access model, in the future we omit specifying local access when the online RL setting is used. Denote the set of $\delta$-sound agents for $\mathcal{M}$ and $\delta$ under the local and global access models by $\mathcal{P}_{\text{LA}}(\mathcal{M}, \delta)$ and $\mathcal{P}_{\text{GA}}(\mathcal{M}, \delta)$, respectively.

What primarily differentiates the quality of agents and planners $P$ in these sets is the query cost of the agent or planner $P$ with an MDP $M \in \mathcal{M}$, as previously defined. Naturally, a measure of difficulty of solving a particular class of featurized MDPs to accuracy $\delta$ is the minimax query cost that $\delta$-sound agents incur on the class. For online RL, and planning with local and global access, we correspondingly define the **query complexity** of the class $\mathcal{M}$ under each setting as:

$$\mathcal{C}^{\star}_{\text{RL}}(\mathcal{M}, \delta) = \inf_{P \in \mathcal{P}_{\text{RL}}(\mathcal{M}, \delta)} \sup_{M \in \mathcal{M}} \text{query cost of } P \text{ with } M$$

$$\mathcal{C}^{\star}_{\text{LA}}(\mathcal{M}, \delta) = \inf_{P \in \mathcal{P}_{\text{LA}}(\mathcal{M}, \delta)} \sup_{M \in \mathcal{M}} \text{query cost of } P \text{ with } M$$

$$\mathcal{C}^{\star}_{\text{GA}}(\mathcal{M}, \delta) = \inf_{P \in \mathcal{P}_{\text{GA}}(\mathcal{M}, \delta)} \sup_{M \in \mathcal{M}} \text{query cost of } P \text{ with } M\,.$$

As alluded to before, there is a clear order between the permissivity of these settings. It is easy to see that any $\delta$-sound agent in the online RL setting is also a $\delta$-sound planner under planning with local access (by converting the agent's interaction with the MDP into queries to the simulator until the agent returns with a policy). Similarly, any $\delta$-sound agent in the local access model is also $\delta$-sound in the global access model. It follows that

$$\mathcal{C}^{\star}_{\mathrm{GA}}(\mathcal{M},\delta) \le \mathcal{C}^{\star}_{\mathrm{LA}}(\mathcal{M},\delta) \le \mathcal{C}^{\star}_{\mathrm{RL}}(\mathcal{M},\delta) \tag{6}$$

no matter the choice of $\mathcal{M}$ and $\delta$.

### 1.2.6. Classes of featurized MDPs

So far, we have made no demands that the feature maps corresponding to an MDP are in any way grounded to the MDP, or useful. The problems we are interested in involve classes of featurized MDPs where the features satisfy specific requirements, so that we can hope to construct a $\delta$-sound agent or planner for such classes, with a query cost that is polynomial in $d$ (the dimensionality of the feature map) instead of $|\mathcal{S}|$. In this thesis we are primarily concerned with establishing the query complexities ($\mathcal{C}^{\star}_{\mathrm{RL}}$, $\mathcal{C}^{\star}_{\mathrm{LA}}$, and $\mathcal{C}^{\star}_{\mathrm{GA}}$) for four classes of featurized MDPs. For $B \ge 0$ and positive integers $d, H, A$, these classes are defined as follows:

- $v^{\star}$-realizable class: $\mathcal{M}^{v^{\star}}_{B,d,H,A}$ is the class of $H$-horizon (Definition 1.2.2) finite-state-space featurized MDPs with $A$ actions, where the feature-vectors are $d$-dimensional. For any $(M,\varphi)$ in this class, $M$ is an MDP with some state space $\mathcal{S}$ and random rewards confined to (say) $[0,1]$, the associated feature-map $\varphi : \mathcal{S} \to \mathbb{R}^d$ with $\sup_{s\in\mathcal{S}} \|\varphi(s)\|_2 \le 1$ is such that for some $\theta^{\star} \in \mathbb{R}^d$ with $\|\theta^{\star}\|_2 \le B$, $v^{\star}_M(s) = \varphi(s)^{\top}\theta^{\star}$ holds for all $s \in \mathcal{S}$ where $v^{\star}_M$ is the optimal value function in $M$.[1]

- $q^{\star}$-realizable class: $\mathcal{M}^{q^{\star}}_{B,d,H,A}$ is the class of featurized MDPs as above except that here for any $(M,\varphi)$ in the class, for $[A] := \{1,\dots,A\}$, $\varphi : \mathcal{S}\times[A] \to \mathbb{R}^d$ with $\sup_{s\in\mathcal{S},a\in[A]} \|\varphi(s,a)\|_2 \le 1$ and $\theta^{\star} \in \mathbb{R}^d$ with $\|\theta^{\star}\|_2 \le B$, we now require that $q^{\star}_M(s,a) = \varphi(s,a)^{\top}\theta^{\star}$ holds for all states $s \in \mathcal{S}$ and actions $a \in [A]$, where $q^{\star}_M(s,a)$ is the optimal action-value at $(s,a)$.

1. Recall that according to Definition 1.2.2, the states of the MDP encode the stage index that the process can be at within an episode. This allows us to use a notation where the dependence on the stage index of values can be suppressed (as otherwise, e.g. the optimal policy would depend explicitly on this stage index), and also means that we can talk about the initial states in an MDP.

- Reachable-$v^\star/q^\star$-realizable class: $\mathcal{M}^{v^\star/q^\star\text{reach}}_{B,d,H,A}$ is the class of featurized MDPs as above except that here the MDPs $M$ are associated with two feature-maps, $\varphi_v : \mathcal{S} \to \mathbb{R}^d$ and $\varphi_q : \mathcal{S}\times[A] \to \mathbb{R}^d$ (their 2-norms bounded by 1 as before), and it is assumed that there exists some $\theta^\star \in \mathbb{R}^d$ with $\|\theta^\star\|_2 \le B$ such that $v^\star_M(s) = \varphi_v(s)^\top\theta^\star$ and $q^\star_M(s,a) = \varphi_q(s,a)^\top\theta^\star$ hold for any action $a$ and any state $s$ of the MDP that is reachable from the initial states (with positive probability with some policy).[2]

- $q^\pi$-realizable class: $\mathcal{M}^{q^\pi}_{B,d,H,A}$ is a variation of $\mathcal{M}^{q^\star}_{B,d,H,A}$ where linear realizability holds for the action-values of *all* memoryless policies (stochastic or deterministic). More precisely, for any $(M,\varphi)$ in the class, for $[A] := \{1,\dots,A\}$, $\varphi : \mathcal{S}\times[A] \to \mathbb{R}^d$ with $\sup_{s\in\mathcal{S},a\in[A]} \|\varphi(s,a)\|_2 \le 1$, we now require that for *any* memoryless policy $\pi$, there exist some $\theta_\pi \in \mathbb{R}^d$ with $\|\theta_\pi\|_2 \le B$, such that $q^\pi_M(s,a) = \varphi(s,a)^\top\theta_\pi$ holds for all states $s \in \mathcal{S}$ and actions $a \in [A]$, where $q^\pi_M(s,a)$ is the action-value at $(s,a)$ corresponding to policy $\pi$.

## 1.3. Related work

**Planning with simulators:** Minor variations of the online planning problem defined in Section 1.2.3 have been investigated by various groups in the literature. It is known as Model Predictive Control (MPC) in the process control literature (Meyn, 2022), while in Artificial Intelligence (AI) this problem is called planning (Chapter 6 Mausam and Kolobov, 2012). Without explicitly realizing its importance at the time, Rust (1997) used online planning in the closed-loop fashion that we adopt for this thesis. This is the key reason why the corresponding positive result does not contradict the negative result of Chow and Tsitsiklis (1989), who set up the planner to run offline, and not part of a closed-loop process. The local planning problem was introduced by Kearns et al. (2002), who noticed that a planner which is given a simulator and an input state and asked to return a good action can do so with computation/query time independent of the size of the state space. However, this runtime is exponential in $H$. Munos (2014) gives algorithms that use optimism to improve on this exponential runtime in benign cases. With linear features, a negative result of Du et al. (2019a) (see also Van Roy and Dong (2019); Lattimore et al. (2020)) states that an exponential in $\min\{H,d\}$ runtime remains for any planner with constant suboptimality, even if the feature map nearly realizes the action-value functions of *all* policies but the approximation error is $\varepsilon = \Omega(\sqrt{H/d})$. For target suboptimality $\mathcal{O}(\sqrt{d}\varepsilon)$, assuming access to the solution of a feature-map-dependent optimal design problem, Lattimore et al. (2020) gives a planner with polynomial computational (and query) com-

2. It is without loss of generality that we use that same $\theta^\star$ in the inner products that yield $v^\star_M(s)$ and $q^\star_M(s,a)$: if these parameters are not shared, we can concatenate them with only a factor 2 increase in $d$ and $B$.

plexity. These results are complemented by the lower bound of Weisz et al. (2021b) (presented in this thesis), showing that an exponential lower bound still holds when only $q^\star$ is realizable even if there are no approximation errors. When only the optimal value function is well-represented, Shariff and Szepesvári (2020) give an algorithm for the case where the features are contained in the convex hull of a "core set" of feature vectors. Their planning algorithm, which builds on top of Lakshminarayanan et al. (2017), has computational and query cost that scales polynomially in the size of the core set and the other relevant quantities. A similar approach appears in Zanette et al. (2019). By contrast, we only provide a bound only on the query complexity of our algorithm, but our query complexity is independent of the size of the core set, whose size, in general, is uncontrolled by the other quantities.

**Online learning:** Any online learning algorithm that controls regret can also be used for local planning by recommending the most frequently used action at the start state. Of the sizable literature on online learning with linear function approximation (Jiang et al., 2017; Du et al., 2019b; Jin et al., 2020a; Wang et al., 2019; Yang and Wang, 2019; Ayoub et al., 2020; Modi et al., 2020; Wang et al., 2020b; Zanette et al., 2020a), the most relevant are the works of Wen and Van Roy (2013); Jiang et al. (2017). Both works give algorithms for the online setting with realizable function approximation, and are based on the principle of optimism. The algorithm of Wen and Van Roy (2013) is restricted to MDPs with deterministic rewards and deterministic transitions, and guarantees that at most $d$ trajectories will be suboptimal. Their proof is based on a similar eluder dimension argument. On the other hand, the algorithm of Jiang et al. (2017) is restricted to the case when a complexity measure called the Bellman rank is low. In fact, our agnostic guarantee (see Definition 3.2.1) is related to a similar agnostic guarantee of their algorithm (see their Appendix A.2), where optimism at the initial state allows them to compete with the best policy whose state-value function is realizable. Despite the similarities, neither the algorithm nor the analysis applies to our setting.

## 1.4. Summary of contributions for $q^\star$ and $v^\star$-realizability

Our first contribution, published as (Weisz et al., 2021b), proves an exponential lower bound that arises due to the stochasticity of the MDP's rewards, while the transitions are deterministic (such a result would not be possible if the rewards were also deterministic, see Theorem 1.4.5). To state this result, let $\mathcal{M}^{\text{Pdet}}$ be the class of featurized MDPs with deterministic transitions:

**Theorem 1.4.1** (Weisz et al., 2021b, Theorem 9, lower bound for exponentially many actions)**.** *There exists $\delta > 0$ and $B > 0$, such that as $H \wedge d \to \infty$, for $A = 2^{\Omega(d \wedge H)}$,*[3]

$$\mathcal{C}^{\star}_{\mathrm{GA}}(\mathcal{M}^{q^{\star}}_{B,d,H,A} \cap \mathcal{M}^{\mathrm{Pdet}}, \delta) = 2^{\Omega(d \wedge H)}.$$

According to this result, as long as there are exponentially many actions, planning remains intractable even for featurized MDPs where the features provided realize the optimal action-value function of the associated MDP and even if the MDP's transitions are deterministic. Note that the exponential lower bound in this theorem is nontrivial since the query complexity of finding a good approximation to a function that lies in the span of $d$ features from input-output examples is polynomial in the number of features regardless the cardinality of the input domain of the function, hence, the intractability in the above result cannot be solely attributed to the presence of a large action set. Indeed, by setting $H = 1$ we arrive at the linear bandits problem, which has a polynomial query complexity (Lattimore and Szepesvári, 2020). The same approach fails when a sequential element is introduced (by setting $H \gg 1$), as the transition observation of $(R', S') \sim Q(\cdot|s, a)$ is missing the crucial information $q^{\star}(S', \cdot)$. This leads to a chicken-and-egg problem, where to find a near-optimal policy from some stage $h$ it would suffice to know the optimal policy from stage $h+1$ onwards, but this information is exponentially difficult in $H$ to obtain.

*Theorem 1.4.1 uncovers the fundamental reason why function approximation in RL and planning is necessarily much harder than in bandits.*

A limitation of this result is that the hardness result only applies to MDPs with exponentially many actions, where even knowing the entire $q^{\star}$ function would not necessarily lead to a computationally efficient implementation of a near-optimal policy (as even solving $\pi(s) = \arg\max_{a \in \mathcal{A}} q^{\star}(s, a)$ may be computationally inefficient).

*An intriguing question is whether planning for the same setting as considered by that of Theorem 1.4.1 but with polynomial number of actions is tractable.*

This question is answered by our next contribution, published as (Weisz et al., 2022b), which states that even with an action count that is polynomial in $d$ and $H$, planning remains intractable for the same class of MDPs:

3. $a \wedge b = \min(a, b)$.

**Theorem 1.4.2** (Weisz et al., 2022b, Theorem 1.1, lower bound with global access, at least poly$(d \wedge H)$ actions)**.** *There exists $\delta > 0$ and $B > 0$, such that as $H \wedge d \to \infty$, for $A \geq d^{1/4} \wedge H^{1/2}$,*

$$\mathcal{C}^\star_{\mathrm{GA}}(\mathcal{M} \cap \mathcal{M}^{\mathrm{Pdet}}, \delta) = 2^{\Omega(d^{1/4} \wedge H^{1/2})}$$
$$\text{for} \quad \mathcal{M} \in \{\mathcal{M}^{v^\star}_{B,d,H,A}, \mathcal{M}^{q^\star}_{B,d,H,A}, \mathcal{M}^{v^\star/q^\star\mathrm{reach}}_{B,d,H,A}\}.$$

Together with Eq. (6) this result also gives a lower bound on the query complexity of planning when only local access is available to the featurized MDP. Apart from the larger exponential in the lower bound of Theorem 1.4.1, Theorem 1.4.2 is a more general result, with its proof including all of the mechanics of the proof of the former result. Thus, in this thesis we only give proof of Theorem 1.4.2 (see Chapter 2), and refer the reader to (Weisz et al., 2022b) for a proof of Theorem 1.4.1.

The remaining problem is whether planning is tractable when $2 \leq A = o(d^{1/4} \wedge H^{1/2})$. For a constant number of actions, our next contribution published as (Weisz et al., 2021a) answers this question in the positive when $\mathcal{M} = \mathcal{M}^{v^\star}_{B,d,H,A}$. We present this result and its proof in Chapter 3 as Theorems 3.3.2, 3.3.4 and 3.3.5. We call the planner achieving this result TensorPlan. It learns directly from the Bellman equations as the transitions are observed. At the core of the proof lies an argument that the number of learning steps before the near-optimal policy is discovered can be bounded by the solution to the following self-contained and elegant linear algebraic problem:[4]

**Problem 1.4.3.** *Given an integer $d$ and a $d$-dimensional real vector space $V$, what is the largest positive integer $T$ with the property that we can find $T$ pairs of vectors $(e_1, f_1), \ldots, (e_T, f_T)$ in $V$ such that for all $1 \leq t \leq T$,there are $x_i \in \{e_i, f_i\}$ for $i \leq t$ such that neither $e_t$ nor $f_t$ is in the span of $x_1, \ldots, x_{t-1}$?*

Weisz et al. (2022b) extends the polynomial upper bounds to two additional settings, for a fixed number of actions. As the extension is fairly straight-forward but heavy on notation, for simplicity we only present the resulting Theorem 1.4.4 in this thesis, and refer the reader to Weisz et al. (2022b) for its proof:

**Theorem 1.4.4** (Weisz et al., 2022b, Theorem 1.2, upper bound with local access)**.** *For $\mathcal{M} \in \{\mathcal{M}^{v^\star}_{B,d,H,A}, \mathcal{M}^{q^\star}_{B,d,H,A} \cap \mathcal{M}^{\mathrm{Pdet}}, \mathcal{M}^{v^\star/q^\star\mathrm{reach}}_{B,d,H,A}\}$, arbitrary positive reals $\delta, B$ and arbitrary positive integers $d, H$,*

$$\mathcal{C}^\star_{\mathrm{LA}}(\mathcal{M}, \delta) = O\left(\mathrm{poly}\left(\left(\tfrac{dH}{\delta}\right)^A, B\right)\right).$$

4. The version of the problem presented here is relevant for the $A = 2$. The solution of this problem is a contribution due to Barnabás Janzer, a co-author of the paper.

While the aforementioned query cost upper bound is polynomial when $A$ is a small constant, a corresponding negative result shown by Liu et al. (2023) is that the computational cost of any $\delta$-sound planner has to be exponentially large in $d$ or $H$ under the Randomized Exponential Time Hypothesis (rETH) (Dell et al., 2014).

Intriguingly, even the new results leave open whether planning with local access is statistically tractable under $q^\star$ realizability when the MDPs involved have **stochastic transition dynamics and rewards** while the number of actions is constant:

*Open problem: for $A$ constant, what is $\mathcal{C}^\star_{\mathrm{LA}}(\mathcal{M}^{q^\star}_{B,d,H,A}, \delta)$ (the query complexity of the class of $q^\star$-realizable featurized MDPs)?*

In fact, even though our upper bound holds generally for the $v^\star$-realizable and reachable-$v^\star/q^\star$-realizable classes of MDPs, for the $q^\star$-realizable class our upper bound only holds for MDPs with deterministic transitions. While our lower bounds (Theorems 1.4.1 and 1.4.2) also only require deterministic-transition MDPs, the difficulty of the planning or online RL problem immediately collapses when the last remaining source of stochasticity, the rewards, is also set to be deterministic. To present this result, denote by $\mathcal{M}^{\mathrm{PRdet}}$ the class of MDPs where in addition to transitions, rewards are also deterministic. For such MDPs, Optimistic Constraint Propagation (Wen and Van Roy, 2013) can be used to show a polynomial query (and even computational) complexity for online RL:

**Theorem 1.4.5** (Wen and Van Roy (2013))**.** *For any $B, d, H, A$,*

$$\mathcal{C}^\star_{\mathrm{RL}}(\mathcal{M}^{q^\star}_{B,d,H,A} \cap \mathcal{M}^{\mathrm{PRdet}}, 0) \le dH\,.$$

*Proof sketch.* Let $\Theta$ be the set of possible values of $\theta^\star$ that based on the transitions observed so far may satisfy $q^\star(s,a) = \varphi(s,a)^\top \theta^\star$ for all $s, a$, and initialize this to $\mathcal{B}^d(B)$. At the start of each episode choose $\theta = \arg\max_{\theta' \in \Theta} \max_{a \in \mathcal{A}} \varphi(s_0, a)^\top \theta'$. For this episode, play the policy that in each state $S_h$ chooses the action $A_h = \arg\max_{a \in \mathcal{A}} \varphi(S_h, a)^\top \theta$, and observe $R_{h+1}, S_{h+1}$. When the episode is over, take the largest stage $0 \le i < H$ such that $\sum_{H \ge j > i} R_j \ne \varphi(S_i, A_i)^\top \theta$. If such an $i$ does not exist, stop and return the current policy, which will be an optimal policy. Otherwise, for this deterministic-reward MDP write the reward as $r(s,a)$ and observe the fact that for $s \in \mathcal{S}_{H-1}$, $q^\star(s,a) = r(s,a)$. By an inductive argument using optimism and this property, $\sum_{H \ge j > i} R_j = q^\star(S_i, A_i)$. Therefore the dimensionality of $\Theta$ reduces by 1 when intersecting it with the subspace that satisfies $\sum_{H \ge j > i} R_j = q^\star(S_i, A_i) = \varphi(S_i, A_i)^\top \theta$. This can happen at most $d - 1$ times, so this algorithm returns with the optimal policy after at most $d$ episodes. ∎

| Publications | Action count | MDP class | poly$(\cdot)$ query complexity? |
|---|---|---|---|
| Wen and Van Roy (2013) | any | $\mathcal{M}^{q^\star}_{B,d,H,A} \cap \mathcal{M}^{\text{det}}$ | ✓ |
| Du et al. (2021) | any | $\mathcal{M}^{v^\star/q^\star}_{B,d,H,A}$ | ✓ |
| Weisz et al. (2021a) | $\mathcal{O}(1)$ | $\mathcal{M}^{v^\star}_{B,d,H,A}$ | ✓ |
| Weisz et al. (2021b) | $2^{\Omega(d \wedge H)}$ | $\mathcal{M}^{q^\star}_{B,d,H,A} \cap \mathcal{M}^{\text{Pdet}}$ | ✗ |
| Weisz et al. (2022b) | $\Omega(d^{1/4} \wedge H^{1/2})$ | $\mathcal{M}^{q^\star}_{B,d,H,A} \cap \mathcal{M}^{\text{Pdet}}$ | ✗ |
| —"— | —"— | $\mathcal{M}^{v^\star}_{B,d,H,A} \cap \mathcal{M}^{\text{Pdet}}$ | ✗ |
| —"— | —"— | $\mathcal{M}^{v^\star/q^{\star\text{reach}}}_{B,d,H,A} \cap \mathcal{M}^{\text{Pdet}}$ | ✗ |
| —"— | $\mathcal{O}(1)$ | $\mathcal{M}^{q^\star}_{B,d,H,A} \cap \mathcal{M}^{\text{Pdet}}$ | ✓ |
| —"— | —"— | $\mathcal{M}^{v^\star}_{B,d,H,A}$ | ✓ |
| —"— | —"— | $\mathcal{M}^{v^\star/q^{\star\text{reach}}}_{B,d,H,A}$ | ✓ |

**Table 1:** Comparison of various query complexity results for planning with global access, and features realizing the optimal value or action-value function. The symbol $\mathcal{M}^{\text{det}}$ stands for the class of finite MDPs with deterministic transitions and rewards. ✓ indicates the existence of a sound planner with query cost polynomial in relevant parameters (excluding $S$ and $A$); ✗ indicates that such a planner does not exist.

Given that our upper bound is polynomial when the number of actions is fixed, one may speculate that when the number of actions is large, perhaps one should replace each stage of an episode with $\log_2(A)$ stages, where actions would be chosen by determining their "bits" one by one, in a sequential fashion. The difficulty then is that this calls for an extension of the state space and a new, suitable feature-map.

**Corollary 1.4.6** (of Theorem 1.4.2; informal)**.** *Let $B, H, d, A$ be as in Theorem 1.4.2 (in particular, $A \geq d^{1/4} \wedge H^{1/2}$), let $\tilde{d}$ be some polynomial of $H, d, A$, and let $\mathcal{M} \in \{\mathcal{M}^{v^\star}_{B,d,H,A}, \mathcal{M}^{q^\star}_{B,d,H,A} \cap \mathcal{M}^{\text{Pdet}}, \mathcal{M}^{v^\star/q^{\star\text{reach}}}_{B,d,H,A}\}$. For any featurized MDP $M$ in $\mathcal{M}$, apply the above action binarization process to derive a corresponding 2-action MDP $M'$. Consider the task of deriving a corresponding $\tilde{d}$-dimensional feature-map, together with which $M'$ belongs to 2-action version of the class $\mathcal{M}$. The local access query complexity of this task is $2^{\Omega(d^{1/4} \wedge H^{1/2})}$.*

*Proof sketch.* By contradiction, if the query complexity of the featurization task is asymptotically smaller than the lower bounds of Theorem 1.4.2, one could take any featurized MDP in $\mathcal{M}$, apply the action binarization and featurization tasks, and apply Theorem 1.4.4 with $A = 2$ to solve the resulting featurized MDP with query cost $O\left(\text{poly}\left(\left(\frac{dH}{\delta}\right)^2, B\right)\right)$. This approach leads to a total query cost that contradicts the query complexity lower bound of Theorem 1.4.2. ■

Recently, the topic of planning and online RL with good features has also seen many new results. We would like to emphasize two results in this topic closely related to Theorem 1.4.4. For

the first result, we define the class $\mathcal{M}_{B,d,H,A}^{v^\star/q^\star}$ like the class $\mathcal{M}_{B,d,H,A}^{v^\star/q^{\star\text{reach}}}$, except that realizability is required to hold over the **entire state-space**, and not only for states reachable from the initial states. Then, as all the features are given in advance in the case of **global access**,

**Theorem 1.4.7** (Du et al. (2021))**.**

$$\mathcal{C}_{\text{GA}}^\star(\mathcal{M}_{B,d,H,A}^{v^\star/q^\star},\delta) = \text{poly}(B,d,H,\delta^{-1})\,.$$

Note that while according to Theorem 1.4.2, planning with **global (and thus also local) access** over $\mathcal{M}_{B,d,H,A}^{v^\star/q^{\star\text{reach}}}$ is **intractable**, the result just mentioned implies that planning with **global access** over $\mathcal{M}_{B,d,H,A}^{v^\star/q^\star}$ is **tractable**. Thus, while it is immediate from the definitions that

$$\mathcal{M}_{B,d,H,A}^{v^\star/q^\star} \subset \mathcal{M}_{B,d,H,A}^{v^\star/q^{\star\text{reach}}},$$

the two results together imply that the class on the right-hand side (RHS) is substantially larger than the one on the left-hand side (LHS). However, the difference is irrelevant when it comes to **local access**, as the following corollary shows:

**Corollary 1.4.8** (of Theorem 1.4.2)**.** *There exists $\delta > 0$ and $B > 0$, such that as $H \wedge d \to \infty$, for $A \geq d^{1/4} \wedge H^{1/2}$,*

$$\mathcal{C}_{\text{LA}}^\star(\mathcal{M}_{B,d,H,A}^{v^\star/q^\star} \cap \mathcal{M}^{\text{Pdet}},\delta) = 2^{\Omega(d^{1/4}\wedge H^{1/2})}$$

*Sketch proof.* For any featurized MDP $M$ in $\mathcal{M}_{B,d,H,A}^{v^\star/q^{\star\text{reach}}} \cap \mathcal{M}^{\text{Pdet}},\delta)$, we can derive a featurized MDP $M'$ in $\mathcal{M}_{B,d,H,A}^{v^\star/q^\star} \cap \mathcal{M}^{\text{Pdet}},\delta)$ by removing the states unreachable from the initial state. As only the unreachable states are removed, any local access simulator for $M$ is also a local access simulator for $M'$. Furthermore, every policy in $M$ has the same value in $M'$. Therefore a $\delta$-sound local access planner for $\mathcal{M}_{B,d,H,A}^{v^\star/q^\star} \cap \mathcal{M}^{\text{Pdet}}$ is also a $\delta$-sound local access planner for $\mathcal{M}_{B,d,H,A}^{v^\star/q^{\star\text{reach}}} \cap \mathcal{M}^{\text{Pdet}}$, and the lower bound of Theorem 1.4.2 applies. ■

When combined with Theorem 1.4.7, we arrive at the first exponential information theoretic separation result in the literature between local and global access:

**Corollary 1.4.9.** *There exists $\delta > 0$ and $B > 0$, such that as $H \wedge d \to \infty$, for $A \geq d^{1/4} \wedge H^{1/2}$,*

$$\text{poly}(B,d,H,\delta^{-1}) = \mathcal{C}_{\text{GA}}^\star(\mathcal{M}_{B,d,H,A}^{v^\star/q^\star},\delta) \ll \mathcal{C}_{\text{LA}}^\star(\mathcal{M}_{B,d,H,A}^{v^\star/q^\star},\delta) = 2^{\Omega(d^{1/4}\wedge H^{1/2})}\,.$$

For convenience, we summarize the results discussed so far in Table 1.

For some $\rho > 0$ constant, let the class of featurized MDPs where in each state there is a gap of at least $\rho$ between the $q^\star$-values of the best and second-best actions be $\mathcal{M}^{\rho-\text{gap}}$. The second result of interest is in online RL, due to Wang et al. (2021), who give an exponential lower bound in the flavor of Theorem 1.4.1 on $\mathcal{C}^\star_{\text{RL}}(\mathcal{M}^{q^\star}_{B,d,H,A} \cap \mathcal{M}^{\rho-\text{gap}}, \delta)$. They achieve this by adapting the hard MDP construction of Weisz et al. (2021b) to satisfy the **constant suboptimality gap** $\rho$. Instead of exponentially downscaling the values in the more advanced stages, such a result is possible by implementing this reduction effect through zero-reward transitions to the episode-over stage, such that the probability of reaching an advanced stage (instead of the value at such a stage) is exponentially small. This does not lead to a hardness result in our planning setup, at least under global access, showing an exponential query cost separation result between global access and online RL:

**Theorem 1.4.10** (Du et al., 2019a, Theorem C.1.)**.** *For any $B, d, H, A, \rho > 0$,*

$$\mathcal{C}^\star_{\text{GA}}(\mathcal{M}^{q^\star}_{B,d,H,A} \cap \mathcal{M}^{\rho-\text{gap}}, 0) \le \text{poly}\left(d, H, \frac{1}{\rho}\right).$$

On the other hand, we note that we expect similar modifications to the hard MDP class underlying our Theorem 1.4.2 to lead to a similar, constant suboptimality gap version of the theorem in the online RL case.

## 1.5. Summary of contributions for $q^\pi$-realizability

Recall that the $q^\pi$-realizable class, $\mathcal{M}^{q^\pi}_{B,d,H,A}$, is the class of finite-state-space featurized MDPs with actions $\mathcal{A} = [A] := \{1, \dots, A\}$, where the feature-vectors are $d$-dimensional, the length of the episodes is $H$. For any $(M, \varphi)$ in this class, $M$ is an MDP with some state space $\mathcal{S}$ and random rewards confined to (say) $[0,1]$, the associated feature-map $\varphi : \mathcal{S} \times [A] \to \mathbb{R}^d$ with $\sup_{s\in\mathcal{S}, a\in[A]} \|\varphi(s,a)\|_2 \le 1$ is such that for **any memoryless policy** $\pi$, there is a $\theta^\pi \in \mathbb{R}^d$ with $\|\theta^\pi\|_2 \le B$, such that for all states $s \in \mathcal{S}$ and actions $a \in [A]$ $q^\pi_M(s,a) = \varphi(s,a)^\top \theta^\pi$ holds. A relaxation of this class with some *misspecification* $\varepsilon \ge 0$ is denoted by $\mathcal{M}^{q^\pi}_{B,d,H,A,\varepsilon}$. Here, a maximum difference of $\varepsilon$ is allowed between the left and right hand sides, over all states and actions:

$$\left|q^\pi(s,a) - \varphi(s,a)^\top \theta^\pi\right| \le \varepsilon \qquad \text{for all } \pi \text{ memoryless policies}, s \in \mathcal{S}, a \in \mathcal{A}.$$

**Table 2:** Comparison of best achievable suboptimality and corresponding query complexity guarantees of various planners with misspecification $\varepsilon > 0$, for the class $\mathcal{M}^{q^\pi}_{B,d,H,A,\varepsilon}$. Drawbacks are highlighted with red, the best bounds with blue.

| Algorithm (Publication) | Query cost | Suboptimality | Access model |
|---|---|---|---|
| MC-LSPI (Lattimore et al., 2020) | $\tilde{\mathcal{O}}(dH^4\varepsilon^{-2})$ | $\tilde{\mathcal{O}}(\varepsilon\sqrt{d}H^2)$ | global access |
| CONFIDENT MC-LSPI (Yin et al., 2022) | $\tilde{\mathcal{O}}(d^2H^4\varepsilon^{-2})$ | $\tilde{\mathcal{O}}(\varepsilon\sqrt{d}H^2)$ | local access |
| CONFIDENT MC-POLITEX (Yin et al., 2022) | $\tilde{\mathcal{O}}(dH^5\varepsilon^{-4})$ | $\tilde{\mathcal{O}}(\varepsilon\sqrt{d}H)$ | local access |
| CAPI-QPI-PLAN (Weisz et al., 2022a) | $\tilde{\mathcal{O}}(dH^4\varepsilon^{-2})$ | $\tilde{\mathcal{O}}(\varepsilon\sqrt{d}H)$ | local access |

It is easy to see that $\mathcal{M}^{q^\pi}_{B,d,H,A}$ is a strictly smaller class than the ones previously considered: $\mathcal{M}^{v^\star}_{B,d,H,A}$, $\mathcal{M}^{q^\star}_{B,d,H,A}$, and $\mathcal{M}^{v^\star/q^\star\text{reach}}_{B,d,H,A}$. As such, we naturally expect more positive results, such as a stronger upper bound or upper bounds that holds in less permissive settings. On the other hand, it is also known that $\mathcal{M}^{q^\pi}_{B,d,H,A}$ is a strictly larger class than the class of linear MDPs (Zanette et al., 2020b, Proposition 4), for which there are efficient algorithms to find a near-optimal policy in the online setting (without a simulator) (Jin et al., 2020b), even in the more challenging reward-free setting where the rewards are only revealed after exploration (Wagenmaker et al., 2022).

In this thesis we also present a planner, called CAPI-QPI-PLAN, for the class of $\mathcal{M}^{q^\pi}_{B,d,H,A,\varepsilon}$, originally published as Weisz et al. (2022a). This planner (1) works in the most permissive local access planning setting, and (2) improves on the state of the art query cost and suboptimality guarantees. Its guarantees are compared to existing solutions in Table 2. This result is summarized in Theorem 1.5.1, which is a consequence of the more general theorems Theorems 4.1.2 and 4.1.3 presented in Chapter 4.[5]. Note the graceful degradation with the aforementioned misspecification $\varepsilon$, in that effectively it only puts a bound on the best suboptimality $\delta$ achievable by the planner.

**Theorem 1.5.1.** *For arbitrary non-negative reals $B, \varepsilon$, arbitrary positive integers $d, H$, and $\delta \geq \varepsilon\sqrt{d}H$,*

$$\mathcal{C}^\star_{\text{LA}}(\mathcal{M}^{q^\pi}_{B,d,H,A,\varepsilon}, \delta) = \tilde{\mathcal{O}}\left(d^2H^6\delta^{-2}\right),$$

*and there is a planner called* CAPI-QPI-PLAN *achieving this query cost while using a computational and memory cost that scales polynomially in the relevant parameters.*

There is a corresponding lower bound due to Weisz et al. (2022a), presented in this thesis as Theorem 4.1.4, that shows that CAPI-QPI-PLAN enjoys query cost and best achievable suboptimality guarantees that are asymptotically optimal in all parameters except $H$.

5. In Chapter 4, for generality, we switch to the $\gamma$-discounted infinite horizon objective. Theorems 4.1.2 and 4.1.3 together imply Theorem 1.5.1 for the fixed, finite horizon case by letting $H = \tilde{\mathcal{O}}(1/(1-\gamma))$ be the effective horizon and noting that a $\delta$-suboptimal policy for the discounted setting is a $2\delta$-suboptimal policy in the $H$-horizon setting.

| MDP class | Online RL | | Local access planning | |
|---|---|---|---|---|
| | poly$(\cdot)$ sample | poly$(\cdot)$ compute | poly$(\cdot)$ sample | poly$(\cdot)$ compute |
| Linear MDP | Jin et al. (2020a) | | | |
| $q^\pi$-realizable MDP | Weisz et al. (2023) | Open problem | Yin et al. (2022) | |

**Table 3:** Comparison of efficiency results for linear MDPs and $q^\pi$-realizable MDPs under online RL and local access planning. Weisz et al. (2023) establishes that $q^\pi$-realizable MDPs are also sample efficiently solvable under online RL. This result is presented in this thesis as Theorem 1.5.2 and Chapter 5. The computational complexity of this problem remains open.

The largest outstanding gap between linear MDPs and $q^\pi$-realizable MDPs (i.e., $\mathcal{M}^{q^\pi}_{B,d,H,A}$) is that while all previous provably efficient algorithms tackling the latter case required the planning setting with local or global access, it is known that linear MDPs can be efficiently (with polynomial query complexity) solved in the more challenging online RL setting too (Jin et al., 2020a). It has been unclear whether having at least a local access simulator is crucial for achieving an efficient (polynomial query cost) solution for $q^\pi$-realizable MDPs. This was perhaps one of the most important differences between the power of the local access planning and online RL regimes (summarized in Table 3). The final contribution presented in this thesis closes these gaps by showing that efficient learning of $q^\pi$-realizable MDPs is possible even in the online RL setting (see Theorem 5.4.1 for the formal version of the theorem, and Chapter 5 for the proof):

**Theorem 1.5.2** (consequence of Theorem 5.4.1, published as Weisz et al., 2023, Theorem 4.1)**.** *For arbitrary positive reals $\delta, B$ and arbitrary positive integers $d, H$, for any $0 \leq \varepsilon \leq \mathrm{poly}(\delta, d, H, \log B)^{-1}$ (for some fixed polynomial),*

$$\mathcal{C}^\star_{\mathrm{RL}}(\mathcal{M}^{q^\pi}_{B,d,H,A,\varepsilon}, \delta) = \tilde{\mathcal{O}}\left(d^7 H^{11} \delta^{-2}\right).$$

The main difficulty of the online RL setting compared to local access is that while in the latter, any (previously observed) state can be reset to in a single step, the query complexity required to reach some specific state in the MDP with online RL must scale at least with the inverse of the maximum reaching probability of the state over any policy. This quantity may be arbitrarily small. As an illustrative example, take an MDP with a sequence of states, each reachable from the previous state only, with $1/2$-probability by any policy. Under the local access model, each of these can be discovered in polynomial query cost, while in general the discovery of such a sequence of states might take at least an exponential (in the length of the sequence) number of samples. This fundamental challenge renders any method relying on a local access simulator unsuitable to tackle the online RL regime. Instead, our approach relies on discovering rich structure in $q^\pi$-realizable

MDPs that allows for techniques similar to those applied on linear MDPs to be adaptable to this regime.

Theorem 1.5.2 leaves two notable open questions: first, Theorem 1.5.2 proves a query complexity result that, while polynomial in the relevant parameters, is less efficient than existing methods for linear MDPs under online RL or for $q^\pi$-realizable MDPs under local access planning. Second, perhaps most notably, it is unknown whether there exists a method that achieves both polynomial sample and computational complexity for $q^\pi$-realizable MDPs under online RL.

## Chapter 2

# Exponential lower bound under $q^\star$-realizability

In this section we give the proof of Theorem 1.4.2. We start by introducing the high level ideas underlying the proof.

## 2.1. Overview

We prove the lower bound by designing a class of MDPs where by traversing the MDP, the agent effectively has to pick corners of a $p$-dimensional hypercube, in sequence, until either $K$ picks were made or a pick was sufficiently close to the secret "solution" corner $w^\star$. Here, $p \approx H^{1/2} \wedge d^{1/4}$ (large if both $H$ and $d$ are large) and $K \approx H/p$ (large if $H$ is large). If the agent picks a corner close to the solution, the episode is effectively terminated and the agent receives the highest possible reward achievable from that state. Otherwise, the agent's next pick has to substantially differ from the previously picked corner. After each choice, the highest reward achievable shrinks by a penalty factor that is governed by how different the subsequent picks are: picking dissimilar corners results in a larger penalty (i.e., a smaller penalty factor). Since subsequent picks need to be substantially different, this means that $q^\star$ (or $v^\star$) reduces at an exponential rate throughout the episode until a guess is sufficiently close to the solution or all $K$ picks are exhausted, in which case the agent receives a Bernoulli reward with expectation $\exp(-\Omega(K))$. Without additional information, guessing sufficiently close to the solution is a needle-in-a-haystack problem with an exponentially large haystack: with probability above (say) $3/4$, the secret corner will not be found within $\exp(\Omega(p))$ guesses. Additional information is not provided to the agent as long as the final reward is 0. Since the probability that this Bernoulli outcome is identically zero for the first $\exp(\Omega(K))$ guesses can be made to be $3/4$ or larger, if a planner uses at most $\exp(\Omega(p \wedge K))$ guesses, with probability at least $1/2$, neither blind guessing nor the Bernoulli outcomes will lead to success. Thus, in expectation, any sound planner has to query more than $\exp(\Omega(p \wedge K))$ times.

To achieve realizability of $q^\star$ (or $v^\star$), it is sufficient if the value of the optimal policy is a low-order polynomial of the $p$-dimensional secret solution at any state in the MDP. To achieve this, the mechanics of choosing a guess and the penalty factor are carefully chosen in such a way that the optimal policy has a simple "greedy" structure that moves any guess as close as possible to the solution. The value of this greedy optimal policy is then proved to be a $4^{\text{th}}$-order polynomial of $w^\star$, which gives rise to a $d \approx p^4$ dimensional feature-map that can realize the optimal values.

For the sake of simplicity and modularity, rather than defining the MDP, we first define a simplified "abstract game" where an "abstract planner" has to guess the above-mentioned secret parameter. This abstract game is essentially what has been described in the previous paragraph. This construction focuses on the information theoretic aspect of the proof, leaving the construction of the MDP with the required realizability properties to the subsequent sections.

## 2.2. Abstract game

The abstract game has a length parameter $K \in \mathbb{N}_+$ and an integer dimensionality parameter $p \geq 2$, which are known to the abstract planner. Let $W = \{-1, 1\}^p$. Let $\mathbf{0}$ and $\mathbf{1}$ indicate the $p$-dimensional vectors of all zeros and all ones, respectively. For vectors $x$ and $y$ from $W$, define $\mathrm{diff}(x, y)$ as the Hamming distance between $x$ and $y$, i.e., the number of components where $x$ and $y$ are different. We will use the property of the Hamming distance that it can be written as an (affine) bilinear function of its arguments: for $w_1, w_2 \in W$,

$$\mathrm{diff}(w_1, w_2) = \frac{1}{2}\left(p - \langle w_1, w_2 \rangle\right) . \tag{7}$$

Note that $\mathrm{diff}(\cdot, \cdot)$ is a metric on the set $W$. Let

$$W^\star = \{w \in W \,:\, p/4 \leq \mathrm{diff}(\mathbf{1}, w) \leq 3p/4\} \tag{8}$$

be the set that will hold the game's secret parameter: $w^\star \in W^\star$. For any $k \in \mathbb{N}$, let

$$W^{\circ k} = \{(w_i)_{i \in [k]} \in W^k \,:\, \mathrm{diff}(w_{i-1}, w_i) \geq p/4 \text{ for } i \in [k]\}, \tag{9}$$

with $w_0 := \mathbf{1}$ defined for convenience, be the subset of $k$-length sequences of $W$ where the elements are "sufficiently far" from each other.

The union of these over $k \leq K$ is the action set of the bandit-like game. Given $w^\star$, the reward function $f_{w^\star} : \{()\} \cup \bigcup_{k \in [K]} W^{\circ k} \to \mathbb{R}$ (index dropped when clear from context) is defined as

follows (again $w_0 := \mathbf{1}$):[6] $f(()) = g(\mathrm{diff}(w_0, w^\star))$, and for $k \in [K]$,

$$f_{w^\star}\left((w_i)_{i\in[k]}\right) = \left(\prod_{i\in[k]} g(\mathrm{diff}(w_{i-1}, w_i))\right) g(\mathrm{diff}(w_k, w^\star)) \qquad \text{where}$$
$$g(x) = 1 - \frac{x}{p} + \frac{(x-1)x}{2p^2}\,.$$

The game is sequential. It proceeds in steps where the abstract planner performs a query and receives a corresponding response (both the query and the response may be randomized). At each step $t \in \mathbb{N}_+$, the abstract planner randomly chooses whether to continue or not, and what its output or next query (correspondingly) is. If it continues, it chooses a sequence length $L_t \in [K]$, and a sequence $S_t = (w_i^t)_{i\in[L_t]} \in W^{\circ L_t}$. Otherwise, if it returns, it chooses its output $S_t = (w_i^t)_{i\in[8]} \in W^{\circ 8}$. Note that the output is confined to have the fixed length of 8.[7] To distinguish this from the case when the planner continues, we let $L_t = 0$ denote that the planner wants to return an output. Let $N = \min\{t \in \mathbb{N}_+ : L_t = 0\}$ indicate the step at which the planner returns. Thus, the planner's output is $S_N$.

At step $t$, denote the choice of the planner by $X_t = (L_t, S_t)$. If the planner is not done yet ($L_t > 0$, and thus $t < N$) then, in response to the planner's query, a random response $Y_t = (U_t, V_t, Z_t) \in \{0,1\} \times \{0,1\} \times [0,1]$ is generated as follows:

- $U_t$ indicates whether the penultimate component of $S_t$ is close to $w^\star$ (for convenience define $w_0^t = \mathbf{1}$):
$$U_t = \mathbb{I}\{\mathrm{diff}(w^t_{L_t-1}, w^\star) < p/4\}\,.$$

- $V_t$ indicates whether the last component of $S_t$ is close to $w^\star$:
$$V_t = \mathbb{I}\{\mathrm{diff}(w^t_{L_t}, w^\star) < p/4\}\,.$$

6. The reason for this form of $f$ will become clear only when the MDP corresponding to the abstract game is defined. For now, let us only note that (1) as the input sequence grows in size, their elements being sufficiently far ensures an exponential rate of reduction of $f$, and (2) $g(x)$ is the second-order Taylor expansion of $(1-1/p)^x$, which ensures through some inequalities that the optimal strategy for maximizing $f$ is to greedily move towards $w^\star$ in the MDP as fast as possible. A simple optimal policy with a low-order polynomial expression for $f$ allows deriving linear features for the MDP's value function.
7. The constant 8 here is sufficiently small to prove that planners cannot guess close enough to $w^\star$ with any of these 8 attempts, yet large enough so that to achieve a small suboptimality in the MDP problem (that will be derived later), it will be crucial to guess a vector among these 8 vectors that is close to $w^\star$.

- $Z_t$ is distributed as $\mathrm{Ber}(f_{w^\star}(S_t))$ if either $V_t = 1$ (the last component of $S_t$ is close to $w^\star$) or $L_t = K$ (all components are used in $S_t$), else $Z_t = 0$. Here, Ber denotes the Bernoulli distribution. This is well-defined as $f_{w^\star}(S_t) \in [0,1]$ by Lemma 2.2.2.

If, on the other hand, the planner indicates that it is done ($L_t = 0$, and thus $t = N$) then there is no feedback, but the payoff (reward) to the planner is

$$R = f_{w^\star}\left((w_i^N)_{i\in[k^\star]}\right) \tag{10}$$

where $k^\star = k^\star(S_t; w^\star)$ denotes the first component of $S_t = (w_k^N)_{k\in[8]}$ that is sufficiently close to $w^\star$, or 8 if none of them are:

$$k^\star = \min\{k \in [8] : k = 8 \text{ or } \mathrm{diff}(w_k^N, w^\star) < p/4\}\,.$$

For future reference, it will be useful to introduce $\tau_{w^\star}(s)$ to denote the first $k^\star(s, w^\star)$ components of $s = (w_i)_{i\in[8]}$ so that $R = f_{w^\star}(\tau_{w^\star}(S_t))$. While the interaction is over at this stage, for simplifying notation, we introduce $Y_t$ and define it as $Y_t = (0,0,0)$.

This finishes the description of the abstract game; for a given value of $w^\star$ we will refer it as "abstract game $w^\star$". To summarize, in this game, the planner can choose actions from a combinatorially structured action set to collect information for the final round where it needs to choose an action from a smaller (but still combinatorially large) subset of the action set. The feedback is non-linear. The essence of the information theoretic argument that will follow will be that good planners essentially need to find $w^\star$.

For these information theoretic arguments, as well as the statement of the main result of this section, some extra definitions are necessary. For $t \in \mathbb{N}_+$, let $F_t = (X_i, Y_i)_{i\in[t-1]}$. For each step $t$ sequentially, if the game is not over yet, i.e., $t - 1 < N$, the planner $\mathcal{A}$ defines the distribution of $X_t$ given $F_t$. Given $F_t$ and $X_t$, the distribution of $Y_t$ is defined as above. Together, $\mathcal{A}$ and $w^\star$ define $\mathbb{P}_{w^\star}^{\mathcal{A}}$, the probability distribution over interaction sequences $(X_t, Y_t)_{t\in[N]}$ between the planner and the game, where the sequence needs to satisfy that $L_t > 0$ for $t < N$ and $L_N = 0$.[8] The planner is well-defined if $\mathbb{P}_{w^\star}^{\mathcal{A}}[N < \infty] = 1$. Let $\mathbb{E}_{w^\star}^{\mathcal{A}}$ be the expectation operator corresponding to $\mathbb{P}_{w^\star}^{\mathcal{A}}$. The abstract planner is sound with worst-case query cost $\bar{N}$ if for all $w^\star \in W^\star$, $\mathbb{E}_{w^\star}^{\mathcal{A}}[N-1] \le \bar{N}$,

8. Luckily for us, $F_t$ takes values in a finite set, which makes it trivial to show that $\mathbb{P}_{w^\star}^{\mathcal{A}}$ with the required properties exist.

and $\mathbb{E}^{\mathcal{A}}_{w^\star}[R] \geq \max_{s\in\mathcal{W}^{\circ 8}} f_{w^\star}(\tau_{w^\star}(s)) - 0.01$. We note in passing that $\max_{s\in W^{\circ 8}} f_{w^\star}(\tau_{w^\star}(s)) = f_{w^\star}(())$, i.e., the maximizing sequence is the empty sequence.

The main result of this section is the following claim, which states that the abstract game is hard:

**Theorem 2.2.1.** *For any abstract planner that is sound with query cost $\bar{N}$,*

$$\bar{N} = 2^{\Omega(p\wedge K)}\,.$$

The proof is given in a number of lemmas. We start with some elementary properties of $f_{w^\star}$:

**Lemma 2.2.2** (Properties of $f_{w^\star}$)**.** *For any $w^\star \in W^\star$, $k \in \mathbb{N}_+$, $s = (w_{k'})_{k'\in[k]} \in W^{\circ k}$, the following hold:*

$$\frac{11}{32} \leq f_{w^\star}(()) \leq \frac{25}{32}\,, \tag{11}$$

$$0 < f_{w^\star}(s) \leq \left(\frac{25}{32}\right)^{k+\mathbb{I}\{\mathrm{diff}(w_k,w^\star)\geq p/4\}} \tag{12}$$

*Proof.* We prove Eq. 12 by first showing that

$$0 < f_{w^\star}(s) \leq \left(\frac{25}{32}\right)^{k}\,. \tag{13}$$

This follows since $f$ is the product of $k+1$ terms, each defined using the function $g$. Now, notice that $g(x)$ decreases as $x$ increases in the range $0 \leq x \leq p$, so for all $k' \in [k]$, thanks to $\mathrm{diff}(w_{k'-1}, w_{k'}) \geq p/4$ which holds since by assumption $s \in W^{\circ k}$, we have

$$0 < g(p) \leq g(\mathrm{diff}(w_{k'-1}, w_{k'})) \leq g(p/4) < \frac{25}{32}\,.$$

This, together with $0 < g(0) \leq 1$ proves Eq. 13. To finish the proof of Eq. 12, note that if $\mathrm{diff}(w_k, w^\star) \geq p/4$ then, similarly to the previous case, we have $0 < g(\mathrm{diff}(w_k, w^\star)) \leq g(p/4) < \frac{25}{32}$, which implies Eq. 12. As $w^\star \in W^\star$, $\frac{1}{4}p \leq \mathrm{diff}(\mathbf{1}, w^\star) \leq \frac{3}{4}p$. Hence, $f_{w^\star}(()) = g(\mathrm{diff}(\mathbf{1}, w^\star)) \geq g(\frac{3}{4}p) \geq \frac{11}{32}$ and $f_{w^\star}(()) \leq g(\frac{1}{4}p) \leq \frac{25}{32}$. ∎

Let

$$n = \left\lfloor \min\left(\frac{e^{\frac{p}{8}}}{16} - 5, \frac{\frac{1}{\varepsilon} - 1}{7.5}\right)\right\rfloor\,, \tag{14}$$

where

$$\varepsilon = \left(\frac{25}{32}\right)^{K+1}.$$

For any $w^\star \in W^\star$, let $E_n^{w^\star}$ be the event when in the first $n$ steps the planner does not hit on any vector that is close to $w^\star$:

$$E_n^{w^\star} = \bigcap_{t\in[n]} \left\{ t > N \text{ or } \left( t = N \text{ and } \min_{i\in[8]} \mathrm{diff}(w_i^N, w^\star) \ge \frac{p}{4} \right) \right.$$
$$\left. \text{or } \left( t < N \text{ and } \mathrm{diff}(w_{L_t-1}^t, w^\star) \ge p/4 \text{ and } \mathrm{diff}(w_{L_t}^t, w^\star) \ge p/4 \right) \right\}.$$

We define the "abstract game 0" (and, for any planner $\mathcal{A}$, the associated probability distribution $\mathbb{P}_0^{\mathcal{A}}$) to be a variant of the game where the responses are $Y_t \equiv (0,0,0)$ for all $t \in \mathbb{N}_+$ (irrespective of the choices of the planner).

Our next lemma claims that the bad event $E_n^{w^\star}$ happens with large probability in abstract game $w^\star$ whenever it happens with large probability in abstract game 0. The reason for this is that the probability of ever receiving nonzero feedback on the bad event is a small value, which in fact can be bounded by $\varepsilon$ (the only way to receive nonzero feedback is by playing to the end, hence $\varepsilon$ appears). From here it will follow that since the number of steps is at most $n$ (bad events are defined for interactions of length at most $n$), the probability of $E_n^{w^\star}$ in game $w^\star$ is at least the probability of this event in game 0 times $(1-\varepsilon)^n$, and the latter is lower bounded by an absolute constant because $n$ is chosen to be not too large compared to $1/\varepsilon$.

**Lemma 2.2.3.** *Take $n$ as defined in Eq. 14. Then, for any abstract planner $\mathcal{A}$ and for any $w^\star \in W$,*

$$\mathbb{P}_{w^\star}^{\mathcal{A}}(E_n^{w^\star}) \ge \frac{7}{8}\mathbb{P}_0^{\mathcal{A}}(E_n^{w^\star}).$$

*Proof.* We prove that

$$\mathbb{P}_{w^\star}^{\mathcal{A}}(E_n^{w^\star}) \ge (1-\varepsilon)^n \mathbb{P}_0^{\mathcal{A}}(E_n^{w^\star}). \tag{15}$$

Since by its choice, $n$ satisfies $n \le \left(\frac{1}{\varepsilon} - 1\right)/7.5$, or, equivalently, $1-\varepsilon \ge 1 - \frac{1}{1+7.5n}$, it follows that

$$(1-\varepsilon)^n \ge \left(1 - \frac{1}{1+7.5n}\right)^n \ge \lim_{n\to\infty}\left(1 - \frac{1}{1+7.5n}\right)^n = e^{-1/7.5} > 7/8,$$

which shows that it suffices to prove Eq. 15.

Let $(X_t, Y_t)_{t\in[N]}$ be a complete interaction history and let $H$ denote the first $n \wedge N$ components of this (thus, $H$ is shorter than the complete sequence when $n < N$). Let $\mathcal{H}$ be the set of all possible values that $H$ can take. For $h \in \mathcal{H}$, let $E_h = E_n^{w^\star} \cap \{H = h\}$. Clearly, $E_n^{w^\star}$ is the disjoint union of the sets $\{E_h\}_{h\in\mathcal{H}}$. Let $\mathcal{H}^+ = \{h \in \mathcal{H} : \mathbb{P}_0^{\mathcal{A}}(E_h) > 0\}$. Then, $\mathbb{P}_0^{\mathcal{A}}(E_n^{w^\star}) = \sum_{h\in\mathcal{H}^+} \mathbb{P}_0^{\mathcal{A}}(E_h)$, and we prove Eq. 15 by showing that for any $h \in \mathcal{H}$,

$$\rho = \frac{\mathbb{P}_{w^\star}^{\mathcal{A}}(E_h)}{\mathbb{P}_0^{\mathcal{A}}(E_h)} \geq (1-\varepsilon)^n \,. \tag{16}$$

Fix $h \in \mathcal{H}^+$ and let $h = (x_t, y_t)_{t\in[n']}$ for some $0 < n' \leq n$. Further, let $x_t = (l_t, s_t)$. Note that for $t < n'$, $l_t > 0$ and either $n' = n$ or $l_{n'} = 0$.

As $\mathbb{P}_0^{\mathcal{A}}(E_h) > 0$, $y_t = (0,0,0)$ for all $t \in [n']$. By definition of $\mathbb{P}_{w^\star}^{\mathcal{A}}$ and $\mathbb{P}_0^{\mathcal{A}}$, both the numerator and denominator factorizes into the product of $n'$ terms. Given the same history, the distribution of $X_t$ under both $\mathbb{P}_{w^\star}^{\mathcal{A}}$ and $\mathbb{P}_0^{\mathcal{A}}$ are identical, so the terms that do not cancel remain:

$$\rho = \prod_{t=1}^{n'} \frac{\mathbb{P}_{w^\star}^{\mathcal{A}}(Y_t = (0,0,0) \,|\, X_t = x_t)}{\mathbb{P}_0^{\mathcal{A}}(Y_t = (0,0,0) \,|\, X_t = x_t)} = \prod_{t=1}^{n'} \mathbb{P}_{w^\star}^{\mathcal{A}}(Z_t = 0 \,|\, X_t = x_t) \,,$$

where $Y_t = (U_t, V_t, Z_t)$. Here, the last equality follows since $\mathbb{P}_0^{\mathcal{A}}[Y_t = (0,0,0) \,|\, X_t = x_t] = 1$ by definition and $\mathbb{P}_{w^\star}^{\mathcal{A}}(Y_t = (0,0,0) \,|\, X_t = x_t) = \mathbb{P}_{w^\star}^{\mathcal{A}}(Z_t = 0 \,|\, X_t = x_t)$ because on $E_h \subset E_n^{w^\star}$, $U_t = V_t = 0$ holds $\mathbb{P}_{w^\star}^{\mathcal{A}}$ almost surely. Now, by definition, $\mathbb{P}_{w^\star}^{\mathcal{A}}(Z_t = 1 \,|\, X_t = x_t) = f_{w^\star}(s_t)\mathbb{I}\{\mathrm{diff}(w_{l_t}^t, w^\star) \leq p/4 \text{ or } l_t = K\}$. Since $E_h \subset E_n^{w^\star}$, $\mathrm{diff}(w_{l_t}^t, w^\star) \leq p/4$ does not hold. Hence, $\mathbb{P}_{w^\star}^{\mathcal{A}}(Z_t = 1 \,|\, X_t = x_t) = f_{w^\star}(s_t)\mathbb{I}\{l_t = K\} \leq (25/32)^{K+1} = \varepsilon$, where the inequality follows from Lemma 2.2.2 using again that $E_h \subset E_n^{w^\star}$ and thus the last component of $s_t$ must be "far" from $w^\star$. Putting things together and using that $n' \leq n$ gives that $\rho \geq (1-\varepsilon)^n$, as required. ■

We plan to argue that the bad event happens with large probability in game 0. In this game, by definition, the planner needs to guess $w^\star$ blindly (as there is no feedback ever). Hence, the success of the planner depends on whether they can without any feedback stumble upon $w^\star$. To bound this success rate, it will be useful to bound the number of vectors close to a given vector in the hypercube $W$:

**Lemma 2.2.4.** *For any $\tilde{w} \in W$, let $W_{\mathrm{close}}(\tilde{w}) = \{w \in W \,|\, \mathrm{diff}(w, \tilde{w}) < p/4\}$. Then,*

$$|W_{\mathrm{close}}(\tilde{w})| \leq 2^p \exp\left(-\frac{p}{8}\right)$$

*Proof.* By symmetry of the $p$-dimensional hypercube, without loss of generality, let $\tilde{w} = \mathbf{1}$ and $W_{\text{close}} = W_{\text{close}}(\tilde{w})$. Let $X = (X_i)_i \in W$ be a uniformly distributed random variable on $W$. Note that the components $X_i$ of $X$ are independent Rademacher random variables. We have

$$\begin{aligned}|W_{\text{close}}| &= \sum_{w \in W} \mathbb{I}\{\langle w, \mathbf{1}\rangle > p/2\} = |W|\,\mathbb{P}(\langle X, \mathbf{1}\rangle > p/2) \\ &= 2^p\,\mathbb{P}\left(\sum_{i \in p} X_i > p/2\right) \leq 2^p \exp\left(\frac{-2(p/2)^2}{4p}\right) = 2^p \exp\left(-\frac{p}{8}\right),\end{aligned}$$

where the second inequality holds by Hoeffding's inequality. ■

Our next lemma shows that for any planner the probability of a bad event has an absolute lower bound. We use the previous lemma to show that for any planner there exists a $w^\star$ such that the probability of the corresponding bad event is lower bounded in game 0, and then we apply Lemma 2.2.3 to get a lower bound for the same event in game $w^\star$.

**Lemma 2.2.5.** *For any abstract planner $\mathcal{A}$ there exists $w^\star \in W^\star$ such that*

$$\mathbb{P}^{\mathcal{A}}_{w^\star}(E_n^{w^\star}) \geq \left(\frac{7}{8}\right)^2.$$

*Proof.* For any $\hat{w} \in W^\star$, under event $\left(E_n^{\hat{w}}\right)^c$, either there exists $t \in [n \wedge (N-1)]$ such that $\text{diff}(w^t_{L_t-1}, \hat{w}) < p/4$ or $\text{diff}(w^t_{L_t}, \hat{w}) < p/4$, or for some $i \in [8]$, $\text{diff}(w_i^N, \hat{w}) < p/4$. That is, $\left(E_n^{\hat{w}}\right)^c \subset \{\hat{w} \in Z\}$ where

$$Z := \bigcup_{t \in [n \wedge (N-1)]} \left(W_{\text{close}}(w^t_{L_t-1}) \cup W_{\text{close}}(w^t_{L_t})\right) \bigcup \left(\bigcup_{i \in [8]} W_{\text{close}}(w_i^N)\right).$$

By Lemma 2.2.4,

$$|Z| \leq (2n+8)2^p \exp\left(-\frac{p}{8}\right). \tag{17}$$

We also have that $W^\star = W \setminus W_{\text{close}}(\mathbf{1}) \setminus W_{\text{close}}(-\mathbf{1})$, so $|W^\star| \geq 2^p\left(1 - 2\exp\left(-\frac{p}{8}\right)\right)$. As $w^\star \in Z$ is the good event for the planner, we define

$$w^\star = \arg\min_{\hat{w} \in W^\star} \mathbb{P}^{\mathcal{A}}_0(\hat{w} \in Z). \tag{18}$$

Putting things together and using that $Z \subseteq W$, we get

$$2^p\left(1-2\exp\left(-\frac{p}{8}\right)\right)\mathbb{P}_0^{\mathcal{A}}\left(w^\star \in Z\right) \le |W^\star|\mathbb{P}_0^{\mathcal{A}}\left(w^\star \in Z\right)$$
$$\le \sum_{\hat{w}\in W^\star}\mathbb{P}_0^{\mathcal{A}}\left(\hat{w}\in Z\right) \le \sum_{\hat{w}\in W}\mathbb{P}_0^{\mathcal{A}}\left(\hat{w}\in Z\right) = \mathbb{E}_0^{\mathcal{A}}[|Z|] \le (2n+8)2^p\exp\left(-\frac{p}{8}\right),$$

Rearranging and using $(E_n^{w^\star})^c \subset \{w^\star \in Z\}$, we get

$$\mathbb{P}_0^{\mathcal{A}}\left(\left(E_n^{w^\star}\right)^c\right) \le \mathbb{P}_0^{\mathcal{A}}\left(w^\star \in Z\right) \le \frac{(2n+8)2^p\exp\left(-\frac{p}{8}\right)}{2^p\left(1-2\exp\left(-\frac{p}{8}\right)\right)} \le 2(n+5)\exp\left(-\frac{p}{8}\right) \le \frac{1}{8},$$

where the last two inequalities follow by our choice of $n$. Combining this with Lemma 2.2.3 finishes the proof. ■

With this, we are ready to prove Theorem 2.2.1. In fact, all that is left to show is that if the planner is sound, then the probability of the bad event cannot be too high. That is, connecting the bad event to poor performance.

*Proof of Theorem 2.2.1.* Take a sound abstract planner $\mathcal{A}$ with query cost $\bar{N}$. Let $w^\star$ be the vector whose existence is guaranteed by the previous lemma. By Markov's inequality,

$$\mathbb{P}_{w^\star}^{\mathcal{A}}\left[N-1 \ge n\right] \le \frac{1}{n}\bar{N}.$$

Let $E'$ be the event under which both $N-1<n$ and $E_n^{w^\star}$ hold: $E' = \{N-1<n\}\cap E_n^{w^\star}$. By the union bound and Lemma 2.2.5,

$$\mathbb{P}_{w^\star}^{\mathcal{A}}\left[E'\right] \ge \left(\frac{7}{8}\right)^2 - \frac{1}{n}\bar{N}. \tag{19}$$

Under the event $E'$, the output of the planner $(w_i^N)_{i\in[8]}$ satisfies $\mathrm{diff}(w_i^N, w^\star) \ge p/4$ for $i\in[8]$, and therefore $k^\star = 8$ and, by Lemma 2.2.2, the reward $R$ of the game satisfies $R < \left(\frac{25}{32}\right)^9$. Therefore, combined with the soundness of $\mathcal{A}$, we get

$$\frac{11}{32} - 0.01 \le f_{w^\star}(()) - 0.01 \le \mathbb{E}_{w^\star}^{\mathcal{A}}\left[R\right] \le \left(\frac{25}{32}\right)^9 + (1-\mathbb{P}_{w^\star}^{\mathcal{A}}\left[E'\right])\frac{25}{32}$$
$$\le \left(\frac{25}{32}\right)^9 + \left(1-\left(\frac{7}{8}\right)^2\right)\frac{25}{32} + \frac{\bar{N}}{n}\frac{25}{32},$$

where we used Lemma 2.2.2 to bound $f_{w^\star}(())$, and the maximum value of $R$ (maximum value of $f$) by $\frac{25}{32}$. To satisfy this inequality, we must have $\bar{N} > 0.05n$, and thus by substituting Eqs. 14 and simplifying we get

$$\bar{N} = \Omega\left(\min\left(\frac{e^{\frac{p}{8}}}{16} - 5, \frac{\frac{1}{\varepsilon} - 1}{7.5}\right)\right) = \min\left(2^{\Omega(p)}, \Omega\left(\left(\frac{32}{25}\right)^{K+1}\right)\right) = 2^{\Omega(p \wedge K)}\,.$$

■

## 2.3. Description of the hard MDP class

Given a large enough horizon $H$ and a large enough dimension $d$, in this section we construct a class of featurized MDPs with horizon $H$ and feature-space dimension $d$, such that *(i)* each featurized MDP in the class corresponds to an abstract game with parameters $(K, p)$ such that $H \approx Kp$, $A = p \approx d^{1/4} \wedge H^{1/2}$ *(ii)* each MDP $M_{w^\star}$ is associated with some abstract game $w^\star \in W^\star \subset W = \{-1, 1\}^p$; *(iii)* the feature-maps associated with the MDPs do not depend on $w^\star$; *(iv)* the respective realizability assumptions are satisfied by the featurized MDPs in the class; *(v)* a planner that is guaranteed to achieve a high value in the MDPs can be used to achieve high values in the associated abstract game, which also means that *(vi)* for every $w^\star \in W^\star$, one should be able to emulate the queries in the featurized MDP associated with $w^\star$ using queries that are available in the abstract game with $w^\star$, while the MDP planner should not get any information about $w^\star$ by any other means than through these queries.

In the abstract game, at the end the planner needs to choose a sequence $(w_i)_{i\in[8]} \in W^{\circ 8}$. This will correspond to the first $8p$ steps of the path that the MDP planner traverses in the MDP, which will have deterministic dynamics. To guarantee that the number of actions is small, choosing such a weight sequence will be implemented in the MDP by first choosing $w_1$ in $p$ steps, then choosing $w_2$ in another $p$ steps, etc. In each of the $p$ steps of these rounds, choosing an action $a \in [p]$ will allow the MDP planner to flip component $a$ of the weight associated with the round. In particular, in the first $p$ steps, the components of $w_1$ are chosen this way, starting from the weight vector $w_0 = \mathbf{1}$. In the next $p$ steps, the components of $w_2$ are chosen this way, but this time starting with $w_1$. The process is identical for choosing $w_k$ based on $w_{k-1}$, where we let $1 \le k \le K$ go up to $K$ to support arbitrary queries in the abstract game. To guarantee that the path chosen is in $\cup_k W^{\circ k}$, further rules are necessary. In particular, since we need to guarantee that $w_k$ differs from $w_{k-1}$ by at least $p/4$ positions, the dynamics is chosen so that in the first $\lceil p/4 \rceil$ steps within the $k$th round, if an action is repeated then it is called illegal, and leads to the end-state $\perp$, while in the remaining

$p - \lceil p/4 \rceil \approx 3p/4$ steps an action repeat is called legal and leads to a "frozen" weight, i.e., starting from the first such repeated action the weight associated with the path cannot be changed until the round is over. These rules guarantee that if a path of length $kp$ does not end up in $\perp$, the path uniquely determines an element of $W^{\circ k}$ (in fact, the last state alone uniquely determines such an element). We associate with every action sequence subject to the constraints just described a unique state, which can be seen as a node on the action tree. We will say that a state $s \neq \perp$ belongs to some round $k \in [0 : K-1]$, if the length $l$ of the associated action sequence $(a_i)_{i \leq l}$ satisfies $kp \leq l < (k+1)p$. We say that the state is in step $i$ of round $k$ if also $l = kp + i$.

Normally, the transitions of the MDP follow the path in the action tree just described, and the rewards are zero. However, there are two exceptions that depend on $w^\star$. To describe them, note that a state $s \neq \perp$ that is in step $p-1$ of some round $k$ is one step away from finalizing the choice of weight vector $w_{k+1}$. Indeed, such a state, together with the action performed in that state, defines the weight sequence $(w_i)_{i \in [k+1]} \in W^{\circ k+1}$, while a shorter sequence $(w_i)_{i \in [k]} \in W^{\circ k}$ is defined by all states $s \neq \perp$ that are in any step $i$ of some round $k$.

For a state that is in some step $i$ of some round $k$, the aforementioned exceptions to the MDP dynamics are: *(i)* if $k > 0$ and $\mathrm{diff}(w_k, w^\star) < p/4$; *(ii)* else if $i = p-1$, and either $k = K-1$ (last step of episode), or $\mathrm{diff}(w_{k+1}, w^\star) < p/4$. In case *(i)*, the next state is $\perp$, and the reward is deterministically set to $\langle \varphi, \theta^\star \rangle$, where $\varphi$ is the feature-vector associated with the state or the state-action pair (depending on which class of featurized MDPs are considered), and $\theta^\star$ is a hidden weight vector corresponding to $w^\star$. In case *(ii)*, a Bernoulli reward with parameter $f_{w^\star}((w_i)_{i \in [k+1]})$ is generated, while also transitioning to $\perp$. Note that the states associated with case *(i)* are unreachable from the initial state as any path to such state goes through a state that satisfies *(ii)*. While there is much information to be gained from any query where the state is of this type, planners with local access can never issue such queries, while planners with global access still have very little chance of encountering such a state (the proportion of these states is exponentially small as can be seen from, e.g., the result of Lemma 2.2.4). We refer the reader to Figure 1 for an illustration of the MDP dynamics and the associated reward structure, and to Eq. 23 for a more precise definition.

The next step is to show that one can define appropriate feature-maps such that the respective realizability conditions hold, which also means that we will need to compute the optimal value (or action-value) functions and then we will also need to show that a sound MDP planner for the appropriate class of MDPs can be used to derive a sound planner for the abstract game.

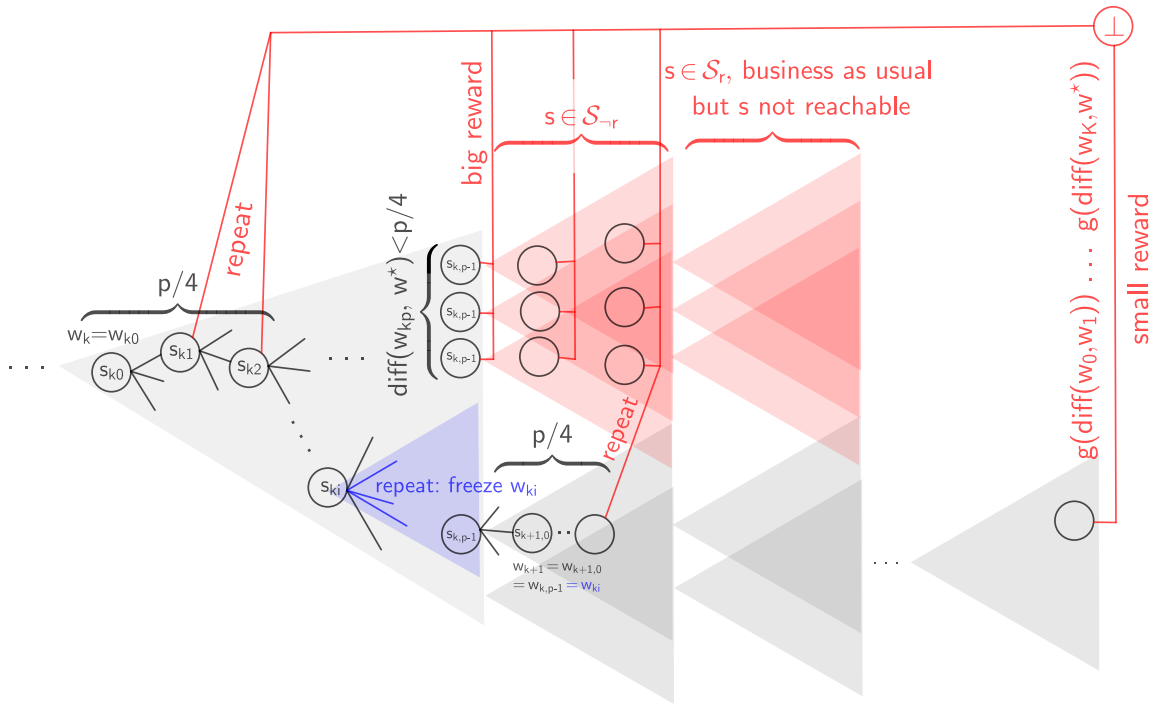


**Figure 1:** Illustration of an MDP associated with a weight vector $w^\star$. The nodes represent states, which are members of the action tree. Subtrees are illustrated with triangles. Edges represent actions, red edges transit to the episode-over state $\perp$. Unless the action was illegal, there are next-states that an MDP with some other $w^\star$ would have transited to. These states still exist in $M$, but are unreachable, and illustrated with a red triangle. The blue triangle represents a part of the action tree where a legal repeated action freezes the weight corresponding the round. Unless written on the edge, there is no reward for the action. In the figure, $(s_{ki})_{ki}$ represents a path through the state space, while for $k$, $i$ fixed, $w_{ki}$ represents the weight vector of step $i$ of round $k$.

Here, the main idea is that the optimal value corresponding to a state $s$ that is in some step $0 \le i \le p-1$ of round $0 \le k \le K-1$ takes the form

$$v^*(s) = f_{w^\star}((w_i)_{i\in[k]}) = g(\mathrm{diff}(w_0,w_1)) \ldots g(\mathrm{diff}(w_{k-1},w_k))g(\mathrm{diff}(w_k,w^\star)),$$

where $w_0 = \mathbf{1}$ by convention, $w_i$ for $1 \le j < k$ is the weight vector for the corresponding round, while $w_k$ is obtained by performing the component manipulations on $w_{k-1}$ prescribed by the action in round $k$ until step $i$, after which, the weight obtained is moved as much as possible towards $w^\star$. Note that $w_k$ here depends on both $s$ and $w^\star$, while the other weight vectors only depend on $s$. In fact, one can write $w_k = A(s)w^\star + b(s)$ for some matrix $A(s)$ and vector $b(s)$ that depend on $s$. Therefore,

$$v^*(s) = h(s)g(\mathrm{diff}(w_{k-1}(s), A(s)w^\star + b(s)))g(\mathrm{diff}(A(s)w^\star + b(s), w^\star)) \tag{20}$$

where $h(s) = g(\mathrm{diff}(w_0, w_1)) \dots g(\mathrm{diff}(w_{k-2}, w_{k-1}))$ is a scalar that depends only on $s$. The expression in Eq. 20 is a fourth-order expression of $w^\star$ since $g$ is a quadratic function, and while diff is (affine) bilinear in its two arguments and so it appears that $\mathrm{diff}(A(s)w^\star + b(s), w^\star)$ could be quadratic itself, due to the special structure, this expression is still linear in $w^\star$. As such, $v^*(s)$ is (roughly[9]) a linear function of $(w^\star)^{\otimes 4}$, the fourth-order tensor product of this vector with itself, which gives rise to the definition of $\varphi_v$ and $\theta^\star$, which is (roughly) the flattening of $(w^\star)^{\otimes 4}$. Of course, it remains to verify that $\varphi_v(s)$ and $\theta^\star$ have small norms as required and also that this definition extends to states that are not reachable from the initial state (to prove the result with global accessibility). In fact, it is exactly this second requirement that made us define the deterministic rewards of $\langle \varphi_v(s), \theta^\star \rangle$ and the associated transitions to $\perp$. (In this case it will be necessary to show that this reward is indeed in the $[0, 1]$ interval.)

A similar argument can be used for $q^*$ realizability, and also for $v^*/q^*$ reachable realizability (in which case the reward at unreachable states could be arbitrary). To finish, one needs to show that a sound MDP planner can be used to implement a sound abstract planner. For this, note that the steps that an MDP planner makes in the first 8 rounds of an episode can be directly translated into an admissible weight sequence of length 8. Further, by construction, the value achieved with this weight sequence is at least as high as the value that the MDP planner would achieve by completing the episode (the function $f_{w^\star}$ and the MDP are such that cutting short a weight sequence obtained from a path in the MDP increases the value of the sequence).

In the remainder of this section, we fill in the gaps of this argument.

## 2.4. The MDP construction

We start with defining $A, p$ and $K$ as a function of the horizon $H \geq 81$ and dimension $d \geq 31$:

$$\begin{aligned} A = p &= \min\left(\max\{x \in \mathbb{N}_+ : x^4 + x^3 + x^2 + x + 1 \leq d\}, \left\lfloor H^{1/2} \right\rfloor\right), \\ K &= \lfloor H/p \rfloor, \\ H' &= Kp. \end{aligned} \tag{21}$$

By our definition of $H$-horizon MDPs, any $H'$-horizon MDP for $H' \leq H$ is also a $H$-horizon MDP (cf. Definition 1.2.2). Hence, we shall construct a $H'$-horizon MDP with $H'$ defined above. For

9. The precise argument will also include lower-order tensor products.

future reference, it will be useful to note that

$$\begin{aligned} A(=p) &= \Theta\left(H^{1/2} \wedge d^{1/4}\right), \qquad K = \Theta(H^{1/2} \vee H/(d^{1/4})), \\ p &\geq 2 \qquad \text{and} \qquad K \geq 9. \end{aligned} \tag{22}$$

Similarly to the abstract game, we fix some $w^\star \in W^\star$ (Eq. 8). In what follows, we define two MDPs $M^v_{w^\star} = (\mathcal{S}, \mathcal{A}, Q^v_{w^\star})$ and $M^q_{w^\star} = (\mathcal{S}, \mathcal{A}, Q^q_{w^\star})$.

The state and action spaces are the same for all these MDPs. The superscript $v$ and $q$ indicates which realizability setting the MDP is tailored for. Together with the indices, we drop them and just use $M$ and $Q$ to minimize clutter. The difference between $M^v_{w^\star}$ and $M^q_{w^\star}$ is minuscule (see Case 23a). As noted beforehand, $\mathcal{A} = [p]$.

Apart from $\perp$, states in $\mathcal{S}$ are uniquely identifiable with an action sequence of length at most $Kp-1$. Of all action sequences, we need to remove any action sequence that has a "repeated" action in the critical first $\lceil p/4 \rceil$ steps of any round. For $k \geq 0$, let $U_k \subset \mathcal{A}^k$ be those sequences of in $\mathcal{A}^k$ which do not have any repeated elements. Then, letting $r = \lceil p/4 \rceil$, $V = (\bigcup_{i\in[r]} U_i) \cup (\bigcup_{i\in[p-r]} U_r \times \mathcal{A}^i)$, we define

$$\mathcal{S} = \{\perp, ()\} \cup \bigcup_{0 \leq k \leq K-1} (U_r \times \mathcal{A}^{p-r})^k \times V,$$

where $()$ denotes the empty sequence. The elements of $\mathcal{S}$ (other than $\perp$) can thus be uniquely identified with a sequence of actions $(a_{00}, \ldots, a_{0,p-1}, \ldots, a_{k0}, \ldots, a_{ki})$ with $0 \leq k \leq K-1$ and $0 \leq i \leq p-1$, where the double indexing emphasizes that the steps are grouped into rounds of length $p$, and commas between indices are often dropped to minimize clutter. For convenience, we let $[< k, i] = \{(n,m) : n \in [0 : K-1], m \in [0 : p-1], np+m < kp+i\}$ denote the index set in this double indexing, so that we can write $(a_{nm})_{(n,m)\in[<k,i]}$ for the above action sequence. Here, we can think of $a_{nm}$ as the action performed in step $m$ of round $n$.

As described beforehand, we associate a "weight", an element of $W$, to each state $s \neq \perp$ that corresponds to all the "flips" described by the action sequence for $s$. Let $w : \mathcal{S} \to W$ be the corresponding map, where we let $w(\perp) = \mathbf{1}$. We will also find it useful to introduce $w : \mathcal{S} \times \mathcal{A} \to W$, where for $(s,a) \in \mathcal{S} \times \mathcal{A}$, $w(s,a)$ is the weight sequence where component $a$ of the last weight vector of $w(s)$ is flipped, except when $s$ is a frozen state or $s = \perp$, in which case $w(s,a) = w(s)$ ($s$ is a frozen state when there is a legal repeated action in the actions that correspond to the current round of the state).

In what follows, we will often find it useful to fix a path, i.e., a complete action sequence of the form $(a_{ki})_{(k,i)\in[0:K-1]\times[0:p-1]} \in \mathcal{A}^{Kp}$. Note that here we allow all action sequences. We then describe the behavior of the MDP in terms of its transitions and rewards encountered during this fixed action sequence. Notationally, we refer to the state (deterministically) reached in round $k$, step $i$ for the fixed action sequence as $s_{ki}$. This means that $s_{00} = ()$, and for $0 \le i \le p-1$, $s_{k,i+1} = \gamma_{w^\star}(s_{ki}, a_{ki})$ and for $0 \le k \le K-1$, $s_{k+1,0} = s_{kp}$, where $\gamma_{w^\star}$ is the transition function of the MDP. Note that the state sequence has extra elements, to help with the notation. In particular, $s_{K0} = s_{K-1,p} = \bot$. By a slight abuse of notation, for the fixed action sequence, we also let $w_{ki} = w(s_{k-1,i}, a_{k-1,i})$ (if $s_{ki} \neq \bot$, $w_{ki} = w(s_{ki})$). To disambiguate, the notation $w_{ki}$ always uses two indices for $w$, while the notation in the abstract game always uses one. To match the weight values of the MDP with those of the abstract game, we introduce the shorthand $w_k = w_{k0}$. To complete the definition of $w_{ki}$, we define $w_{00} = \mathbf{1}$ (similarly to the abstract game's definition of $w_0 = \mathbf{1}$). We will also find it useful to introduce the function $w_{\text{last}} : \mathcal{S} \to W$ which to a given state $s = s_{ki} \neq \bot$ at step $i$ or round $k$ assigns the "last complete weight" $w_k = w_{k0}$ while $w_{\text{last}}(\bot) = \mathbf{1}$.

> *The $(k,i)$-indexed notation, such as $s_{ki}$ and $w_{ki}$ (along with other similarly indexed quantities introduced later) is designed to avoid clutter by hiding the implicit dependence on the action sequence, which is assumed to be fixed whenever we use such notations. The action sequence that is fixed should always be clear from the context. Whenever we state a result concerning these symbols, the result is meant to hold for an arbitrary action sequence.*

For a state $s \in \mathcal{S}$, $s \neq \bot$ that is in step $i$ of round $k$, and an action $a \in [A]$, the transition and reward of taking action $a$ in state $s$ leads to the following reward-next state pair $(R', S')$ (which specifies the kernel $Q$ of the MDP):

$$(R', S') = \begin{cases} \left(\left\langle \varphi, \theta^\star \right\rangle, \bot\right), & \text{if } k > 0 \text{ and } \operatorname{diff}(w_k, w^\star) < p/4 & \text{(23a)} \\ (Z, \bot), & \text{else if } i = p-1, \operatorname{diff}(w_{k+1}, w^\star) < p/4 & \text{(23b)} \\ (Z, \bot), & \text{else if } k = K-1,\ i = p-1 \text{ (last step)} & \text{(23c)} \\ (0, s_{k,i+1}), & \text{otherwise}, & \text{(23d)} \end{cases}$$

Here, the symbols not yet introduced beforehand are defined as follows: *(i)* $(w_{k'})_{k'\in[k+\mathbb{I}\{i=p-1\}]}$ is the sequence of round-start weights $(w_{k',0})_{k'\in[k+\mathbb{I}\{i=p-1\}]}$ that correspond to state $s$ and action $a$. If $i = p-1$, this sequence also includes the newly "compiled" weight $w_{k+1,0} = w(s,a)$. *(ii)* $Z$ has distribution $\mathrm{Ber}(f_{w^\star}((w_{k'})_{k'\in[k+\mathbb{I}\{i=p-1\}]}))$. *(iii)* $\theta^\star$ will be defined in Eq. 31. *(iv)* for feature-maps

$\varphi_v$ and $\varphi_q$ (defined in Eqs. 32, 37), $\varphi = \varphi_v(s_{ki})$ if we are in the $v^\star$-realizable setting (MDP $M^v_{w^\star}$) and $\varphi = \varphi_q(s_{ki}, a)$ otherwise. In either case, the reward in Case 23a is in $[0,1]$ by Eq. 35 and Eq. 39.

Later in the proof, the following lemma will be useful to convert a sound planner for the MDP into a sound planner for the abstract game:

**Lemma 2.4.1.** *We can simulate an outcome of $(R', S')$ in the MDP using at most one query to the abstract game, if the length, dimensionality, and secret parameters of the game are $K$, $p$, and $w^\star$, respectively.*

*Proof.* For $k = 0$, $i < p-1$, we fall under Case 23d and no query to the abstract game is required. Otherwise, let $l = k + \mathbb{I}\{i = p-1\} > 0$, and query the abstract game with $(l, (w_{k'})_{k' \in [l]})$. This is a valid query as $(w_{k'})_{k' \in [l]} \in W^{\circ k}$. The result to this query allows to determine which case the transition falls under, and it also contains $Z$ (with the required distribution) when the case calls for it. ∎

As alluded to before, Case 23a is somewhat pathological: the transitions are such that if at the end of round $k$ the newly "compiled" weight $w_{k+1,0}$ is close to $w^\star$ ($\mathrm{diff}(w_{k+1,0}, w^\star) < p/4$) then the next state is $\perp$. This means that by following the transitions, it is impossible to arrive at a state $s \in \mathcal{S}$, where Case 23a would apply.

**Lemma 2.4.2** (Case 23a is unreachable in $M$)**.** *In MDP $M$, for all $s \in \mathcal{S}_{\mathrm{r}}$ and $s' \in \mathcal{S}_{\neg \mathrm{r}}$,*

$$s' \notin \mathrm{Reach}_M(s)$$

*where*

$$\mathcal{S}_{\neg \mathrm{r}} = \{s \in \mathcal{S} : s \neq \perp \text{ and } \mathrm{diff}(w_{\mathrm{last}}(s), w^\star) < p/4\}, \qquad \mathcal{S}_{\mathrm{r}} = \mathcal{S} \setminus \mathcal{S}_{\neg \mathrm{r}}. \tag{24}$$

We will find some further notation useful to describe essential properties of the MDP states. Take any path in the MDP and the corresponding states $(s_{ki})$. Pick $k$ and $i$ such that $s_{ki} \neq \perp$. Let the "bit mask" $\mathrm{fix}_{ki} \in \{0,1\}^p$ indicate for each component of $w_{ki}$ whether it is fixed (1) in round $k$ at step $i$ or not (0). Recall that a component is fixed if either the corresponding action is performed in round $k$ before step $i$, or there was a legal repeated action, in which case all the components are frozen. Let $\mathrm{ct}^{\mathrm{flip}}_{ki}$ be the number of components flipped in round $k$ by step $i$. Because each component

can only be flipped at most once in a round, this satisfies

$$\mathrm{ct}_{ki}^{\mathrm{flip}} = \mathrm{diff}(w_{k0}, w_{ki})\,.$$

Let $\mathrm{e}_{ki}^{\mathrm{fix}}$ (and $\mathrm{e}_{ki}^{\neg\mathrm{fix}}$) be the number of components that are fixed (and not fixed, respectively) at step $i$ and have the opposite sign of the respective components of $w^\star$. These are "error counts". (As opposed to $\mathrm{ct}_{ki}^{\mathrm{flip}}$ and $\mathrm{fix}_{ki}$, the error counts obviously depend on $w^\star$). Let the operator $\cdot : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ return the componentwise product of its inputs. For $i \in \{0,1\}$, let $\neg i = 1 - i$, which is also extended to binary-valued vectors in a componentwise manner. The definitions imply the following identities:

$$\mathrm{e}_{ki}^{\mathrm{fix}} = \frac{1}{2}\left(\langle \mathbf{1}, \mathrm{fix}_{ki}\rangle - \left\langle \mathrm{fix}_{ki} \cdot w_{ki}, w^\star \right\rangle\right), \tag{25}$$

$$\mathrm{e}_{ki}^{\neg\mathrm{fix}} = \frac{1}{2}\left(\langle \mathbf{1}, \neg\mathrm{fix}_{ki}\rangle - \left\langle \neg\mathrm{fix}_{ki} \cdot w_{ki}, w^\star \right\rangle\right). \tag{26}$$

Consider the case when $\mathrm{fix}_{ki} \neq \mathbf{1}$. Thanks to $s_{ki} \neq \bot$, the first $i$ actions of round $k$ are unique. Therefore, in this case, $\mathrm{ct}_{ki}^{\mathrm{flip}} = i$. Furthermore, each unique action adds 1 to $\langle \mathbf{1}, \mathrm{fix}_{ki}\rangle$, thus $\mathrm{e}_{ki}^{\mathrm{fix}} \leq \langle \mathbf{1}, \mathrm{fix}_{ki}\rangle = i = \mathrm{ct}_{ki}^{\mathrm{flip}}$. Similarly, $\mathrm{e}_{ki}^{\neg\mathrm{fix}} \leq \langle \mathbf{1}, \neg\mathrm{fix}_{ki}\rangle = p - i = p - \mathrm{ct}_{ki}^{\mathrm{flip}}$. If on the other hand, $\mathrm{fix}_{ki} = \mathbf{1}$, then $\mathrm{e}_{ki}^{\neg\mathrm{fix}} = 0$. This leads to the following result, which will be useful for our calculations:

**Lemma 2.4.3.** *Assuming $s_{ki} \neq \bot$, $\mathrm{e}_{ki}^{\neg\mathrm{fix}} \leq p - \mathrm{ct}_{ki}^{\mathrm{flip}}$, and $\mathrm{e}_{ki}^{\neg\mathrm{fix}} \leq p - i$. Furthermore, if $\mathrm{fix}_{ki} \neq \mathbf{1}$, then the following also hold: $\mathrm{ct}_{ki}^{\mathrm{flip}} = i = \langle \mathbf{1}, \mathrm{fix}_{ki}\rangle$, and $\mathrm{e}_{ki}^{\mathrm{fix}} \leq \mathrm{ct}_{ki}^{\mathrm{flip}}$.*

## 2.5. Defining a policy and calculating its value function

We now define a deterministic policy $\pi_{w^\star} : \mathcal{S} \to [A]$, which later will be shown to be the optimal policy. The purpose of the current section is merely to compute the value function of this policy. The policy is defined as follows: Let $s_{ki} \in \mathcal{S}_{\mathrm{r}}$ be a state along step $i$ of round $k$ and assume that $s_{ki} \neq \bot$. Then $\pi_{w^\star}$ greedily flips all the components of $w_{ki}$ that have the wrong sign and are not fixed yet. Once this is done, $\pi_{w^\star}$ freezes the round by repeating an action. Ties are resolved in a systematic fashion.

More formally, let $\mathcal{A}_1$ be the set of actions where the component of $w_{ki}$ has not been fixed yet and where $w_{ki}$ disagrees in sign with $w^\star$; let $\mathcal{A}_2$ be the set of actions where the component has been

fixed:

$$\begin{aligned}\mathcal{A}_1 &= \{a \in [A] \,:\, (\mathrm{fix}_{ki})_a = 0 \text{ and } (w_{ki})_a \neq w^\star_a\} \\ \mathcal{A}_2 &= \{a \in [A] \,:\, (\mathrm{fix}_{ki})_a = 1\}\end{aligned} \tag{27}$$

Then,

$$\pi_{w^\star}(s_{ki}, \cdot) = \begin{cases} 1\,, & \text{if } s_{ki} = \bot\,; & \text{(28a)} \\ \arg\max_{a \in [p]} \left\langle \varphi_q(s_{ki}, a), \theta^\star \right\rangle, & \text{else if } \mathrm{diff}(w_{k0}, w^\star) > p/4\,; & \text{(28b)} \\ \min \mathcal{A}_1\,, & \text{else if } |\mathcal{A}_1| = \mathrm{e}^{\neg\mathrm{fix}}_{ki} > 0\,; & \text{(28c)} \\ \min \mathcal{A}_2\,, & \text{else if } |\mathcal{A}_2| = \mathrm{ct}^{\mathrm{flip}}_{ki} > 0\,, & \text{(28d)} \end{cases}$$

where $\varphi_q$ is the state-action feature-map defined in Eq. 37, and $\theta^\star$ is defined in Eq 31. Note that $s_{ki} \in \mathcal{S}_{\mathrm{r}}$ and $s_{ki} \neq \bot$ implies that either Case 28c or 28d must apply.

With this, the promised result of the section is as follows.

**Lemma 2.5.1.** *Assuming $s_{ki} \in \mathcal{S}_{\mathrm{r}}$ and $s_{ki} \neq \bot$, we have*

$$v^{\pi_{w^\star}}(s_{ki}) = \left( \prod_{k' \in [k]} g(\mathrm{diff}(w_{k'-1,0}, w_{k',0})) \right) g(\mathrm{ct}^{\mathrm{flip}}_{ki} + \mathrm{e}^{\neg\mathrm{fix}}_{ki}) g(\mathrm{e}^{\mathrm{fix}}_{ki})\,.$$

The high level argument underlying this lemma is that the policy reaches the end state $\bot$, either after reaching the last step of the current, or the next round. In either cases, the only reward incurred from the current state to the end is when the transition to the end state happens. The definition of this reward can then be invoked to show the result. The detailed proof is as follows:

*Proof.* Starting from round $k$ step $i$ and letting $\mathcal{A}_1$ be as in Eq. 27, the policy $\pi_{w^\star}$ flips all the components in $\mathcal{A}_1$ (that have the wrong sign and are not fixed yet). We note that $\mathcal{A}_1 = \{\}$ if there was a repeated action in this round (which freezes the components). In this case, $\mathrm{e}^{\neg\mathrm{fix}}_{ki} = 0$ and $s_{ki} \neq \bot$ implies the repeated action was legal, i.e., $i = i + \mathrm{e}^{\neg\mathrm{fix}}_{ki} \geq \lceil p/4 \rceil$, and therefore $w_{ki}$ is frozen, thus regardless of $\pi_{w^\star}$, $w_{k+1,0} = w_{ki}$, so $\mathrm{diff}(w_{k0}, w_{k+1,0}) = \mathrm{ct}^{\mathrm{flip}}_{ki} = \mathrm{ct}^{\mathrm{flip}}_{ki} + \mathrm{e}^{\neg\mathrm{fix}}_{ki}$.

Otherwise, by definition the first $i + |\mathcal{A}_1| = i + \mathrm{e}^{\neg\mathrm{fix}}_{ki} = \mathrm{ct}^{\mathrm{flip}}_{ki} + \mathrm{e}^{\neg\mathrm{fix}}_{ki} \leq p$ actions in round $k$ are unique (noting the inequality comes from Lemma 2.4.3). Furthermore, in this case observe that all components where $w_{k0}$ differs in sign from $w^\star$ are flipped in round $k$ by step $i + \mathrm{e}^{\neg\mathrm{fix}}_{ki}$: either because it was flipped in the first $i$ steps (and thus setting the relevant component of $\mathrm{fix}_{ki}$ to 1), or because the action corresponding to the component is in $\mathcal{A}_1$, and thus flipped by $\pi_{w^\star}$. Therefore

$i + \mathrm{e}_{ki}^{\neg\mathrm{fix}} \geq \mathrm{diff}(w_{k0}, w^\star)$ As $\perp \neq s_{ki} \in \mathcal{S}_\mathrm{r}$, $\mathrm{diff}(w_{k0}, w^\star) \geq p/4$. As $i + \mathrm{e}_{ki}^{\neg\mathrm{fix}}$ is an integer, $i + \mathrm{e}_{ki}^{\neg\mathrm{fix}} \geq \lceil p/4 \rceil$. At step $i + \mathrm{e}_{ki}^{\neg\mathrm{fix}} \geq \lceil p/4 \rceil$, all the actions in $\mathcal{A}_1$ are exhausted, and if there are any remaining steps in the round, $\pi_{w^\star}$ freezes the round by repeating an action (Case 28d). This is a legal action as $i + \mathrm{e}_{ki}^{\neg\mathrm{fix}} \geq \lceil p/4 \rceil$. Therefore $w_{k+1,0} = w_{k,i+\mathrm{e}_{ki}^{\neg\mathrm{fix}}}$.

Regardless of whether $w_{ki}$ is fixed at step $i$, the number of components that have the wrong sign that are not flipped in round $k$ is exactly $\mathrm{e}_{ki}^{\mathrm{fix}}$, and therefore

$$\mathrm{diff}(w_{k0}, w_{k+1,0}) = \mathrm{ct}_{ki}^{\mathrm{flip}} + \mathrm{e}_{ki}^{\neg\mathrm{fix}}$$
$$\mathrm{diff}(w_{k+1,0}, w^\star) = \mathrm{e}_{ki}^{\mathrm{fix}}$$

At the end of round $k$, at step $p-1$, either Case 23b or 23c applies and the expectation of the reward is

$$\begin{aligned} f_{w^\star}\left((w_{k'0})_{k' \in [k+1]}\right) &= \left(\prod_{k' \in [k+1]} g(\mathrm{diff}(w_{k'-1,0}, w_{k',0})\right) g(\mathrm{diff}(w_{k+1,0}, w^\star)) \\ &= \left(\prod_{k' \in [k]} g(\mathrm{diff}(w_{k'-1,0}, w_{k',0})\right) g(\mathrm{ct}_{ki}^{\mathrm{flip}} + \mathrm{e}_{ki}^{\neg\mathrm{fix}}) g(\mathrm{e}_{ki}^{\mathrm{fix}}), \end{aligned}$$

or Case 23d applies and the episode continues with round $k+1$. In this latter case, $\mathrm{fix}_{k+1,0} = \mathbf{0}$, $\mathrm{ct}_{k+1,0}^{\mathrm{flip}} = 0$, $\mathrm{e}_{k+1,0}^{\neg\mathrm{fix}} = \mathrm{diff}(w_{k+1,0}, w^\star) = \mathrm{e}_{ki}^{\mathrm{fix}}$, and so in round $k+1$, $\pi_{w^\star}$ sets all the remaining components to match $w^\star$, i.e., $w_{k+2,0} = w^\star$. The transition at the end of round $k+1$, at step $p-1$, then falls either under Case 23b or 23c, and the expectation of the reward is the same as before as $g(0) = 1$:

$$\begin{aligned} f_{w^\star}\left((w_{k'0})_{k' \in [k+2]}\right) &= \left(\prod_{k' \in [k+2]} g(\mathrm{diff}(w_{k'-1,0}, w_{k',0})\right) g(\mathrm{diff}(w_{k+2,0}, w^\star)) \\ &= \left(\prod_{k' \in [k]} g(\mathrm{diff}(w_{k'-1,0}, w_{k',0})\right) g(\mathrm{ct}_{ki}^{\mathrm{flip}} + \mathrm{e}_{ki}^{\neg\mathrm{fix}}) g(\mathrm{e}_{ki}^{\mathrm{fix}}) g(0), \end{aligned}$$

As in MDP $M$ any transition with a positive reward expectation transitions to state $\perp$, the value of $\pi_{w^\star}$, the expected sum of rewards along the episode, reduces to the expectation of this single reward in the episode.

■

## 2.6. Showing that $\pi_{w^\star}$ is an optimal policy

We start with a lemma that will be used to optimize the attainable reward, given the constraints of the MDP.

**Lemma 2.6.1.** *For $p \geq 2$, $l \geq 2$ integer, let $(x_j)_{j\in[l]}$ be integers $0 \leq x_j \leq p$, and let $0 \leq c_1, c_2, c_3 \leq p$ be further integers such that the following all hold:*

- $c_2 \leq c_1$ ***or*** $c_3 = 0$*;*
- $c_1 + c_3 \leq p$*;*
- $c_1 + c_2 + c_3 \leq \sum_{j\in[l]} x_j$*;*
- $c_2 \leq \sum_{j\in[2:l]} x_j$*;*
- $c_1 \leq x_1$.

Then,

$$\prod_{j\in[l]} g(x_j) \leq g(c_1 + c_3)g(c_2)\,.$$

*Proof.* Note that $g(x) > 0$ and decreases monotonically for $x \in [0, p]$. First we prove for integers $x \geq y$ such that $1 \leq x, y \leq p-1$, it holds that

$$g(x)g(y) \leq g(x+1)g(y-1)\,. \tag{29}$$

Note that $g(x)g(y) - g(x+1)g(y-1) = -\frac{x-y+1}{2p^4}\,(p(x-2) + y(p-x) + x)$, and as $x \geq y$, it only remains to prove that $p(x-2) + y(p-x) + x \geq 0$. If $x = 1$ then $y = 1$ and the above holds with equality. Otherwise $x \geq 2$ and all terms are non-negative, finishing the proof of Eq. 29.

We now claim that for any $0 \leq y \leq x \leq p$ integers, $g(x)g(y) \leq g((x+y) \wedge p)$. Since over $[0, p]$, $g$ takes values in $[0, 1]$, this clearly holds when either $y = 0$ or when $x = p$. Furthermore, if $1 \leq y \leq x \leq p-1$, then from Eq. 29 it follows that $g(x)g(y) \leq g(x+1)g(y-1) \leq g(x+2)g(y-2) \leq g((x+y) \wedge p)g((x+y-p) \vee 0) \leq g((x+y) \wedge p)$ where the last inequality follows again because $g(u) \in [0, 1]$ when $u \in [0, p]$.

Now, $g(x_2)g(x_3)g(x_4) \leq g((x_2+x_3) \wedge p)g(x_4) \leq g((((x_2+x_3) \wedge p) + x_4) \wedge p) = g((x_2+x_3+x_4) \wedge p)$. Continuing this way, letting $x_{\geq 2} = \sum_{j\in[2:l]} x_j$, we get

$$\prod_{j\in[2:l]} g(x_j) \leq g(x_{\geq 2} \wedge p)\,.$$

Thus, $\prod_{j\in[l]} g(x_j) \le g(x_1)g(x_{\ge 2} \wedge p)$.

Consider first the case when $c_3 = 0$. Then, by monotonicity of $g$, as $x_1 \ge c_1 = c_1 + c_3$ and $c_2 \le x_{\ge 2}$, $\prod_{j\in[l]} g(x_j) \le g(c_1+c_3)g(c_2)$ and we are done.

Now, if $c_3 > 0$, by assumption $c_2 \le c_1$. In this case, $c_1+c_2+c_3-(x_1 \wedge (c_1+c_3)) \le x_{\ge 2} \wedge p$, as (1) $c_1 \le (x_1 \wedge (c_1+c_3))$ and thus $c_1+c_2+c_3-(x_1 \wedge (c_1+c_3)) \le c_2+c_3 \le c_1+c_3 \le p$, while (2) by our assumptions, $x_{\ge 2} \ge c_2$ and $x_1 + x_{\ge 2} \ge c_1+c_2+c_3$, and therefore $(x_1 \wedge (c_1+c_3)) + x_{\ge 2} \ge c_1+c_2+c_3$. By the monotonicity of $g$, we can then conclude that

$$\prod_{j\in[l]} g(x_j) \le g(x_1 \wedge (c_1+c_3))g(c_1+c_2+c_3-(x_1 \wedge (c_1+c_3)))\,.$$

Let $x'_1$ and $x'_2$ be the above arguments of $g$ in decreasing order, i.e., $x'_1 = (x_1 \wedge (c_1+c_3)) \vee (c_1 + c_2+c_3-(x_1 \wedge (c_1+c_3)))$ and $x'_2 = (x_1 \wedge (c_1+c_3)) \wedge (c_1+c_2+c_3-(x_1 \wedge (c_1+c_3)))$, so that we have $\prod_{j\in[l]} g(x_j) \le g(x'_1)g(x'_2)$ with $x'_1 \le c_1+c_3$ and $x'_1+x'_2 = c_1+c_2+c_3$. Applying Eq. 29 on this product $c_1+c_3-x'_1$ times, we get that

$$\prod_{j\in[l]} g(x_j) \le g(x'_1)g(x'_2) \le g(c_1+c_3)g(c_2)\,.$$

■

We now show that $\pi_{w\star}$ is an optimal policy by arguing that its value function matches the optimal value function.

**Lemma 2.6.2** ($\pi_{w\star}$ is an optimal policy)**.** *In MDP $M$,*

$$\forall s \in \mathcal{S}, a \in [A], \quad v^{\pi_{w\star}}(s) = v^\star(s)\,.$$

*Proof.* For $s = \perp$, the claim holds by definition as $v^{\pi_{w\star}}(\perp) = v^\star(\perp) = 0$. Otherwise, let $s = s_{ki}$ be a state along step $i$ of round $k$. Let us first consider the case when $s_{ki} \in \mathcal{S}_{\neg \mathrm{r}}$. For any action $a$ performed, the transition will happen under Case 23a, and the deterministic reward given equals $q^\star(s_{ki}, a)$. If we are in the $v^\star$-realizable setting (for MDP $M^v_{w\star}$), this reward does not depend on the action and therefore $v^{\pi_{w\star}}(s) = v^\star(s)$ regardless of $\pi_{w\star}$. Otherwise, $\pi_{w\star}$ chooses an action under Case 28b, which by definition maximizes the reward, so again $v^{\pi_{w\star}}(s) = v^\star(s)$ in this case as well.

Let us turn to the case where $s_{ki} \in \mathcal{S}_{\mathrm{r}}$. There is at most one reward with positive expectation in any round (or none, if an illegal action is taken). As no state in $\mathcal{S}_{\neg \mathrm{r}}$ is reachable from $s_{ki}$ (by Lemma 2.4.2), this reward is collected at the end of some round $K' \in [0 : K-1]$, at step $p-1$, and

has expectation

$$f_{w^\star}\left((w_{k'0})_{k'\in[K'+1]}\right) = \left(\prod_{k'\in[K'+1]} g(\mathrm{diff}(w_{k'-1,0}, w_{k',0}))\right) g(\mathrm{diff}(w_{K'+1,0}, w^\star)) \\ = \prod_{k'\in[K+1]} g(\mathrm{diff}(w_{k'-1,0}, w_{k',0}))\,,$$

where, for convenience, we let $w_{k'0} = w^\star$ for $k' \geq K'+2$ (as $g(0) = 0$). This reward expectation is strictly positive (by Lemma 2.2.2), so the optimal policy will never take an illegal action.

At round $k$, $g(\mathrm{diff}(w_{k'-1,0}, w_{k',0}))$ is fixed for $k' \in [k]$, and the policy can only influence the terms $g(\mathrm{diff}(w_{k'-1,0}, w_{k',0}))$ for $k' \in [k+1:K+1]$. We have by definition that $0 \leq \mathrm{diff}(\cdot,\cdot) \leq p$. In any round, once a component is flipped it cannot be flipped back in the same round. This implies that

$$\mathrm{diff}(w_{k0}, w_{k+1,0}) = \mathrm{diff}(w_{k0}, w_{ki}) + \mathrm{diff}(w_{ki}, w_{k+1,0}) \geq \mathrm{diff}(w_{k0}, w_{ki}) = \mathrm{ct}^{\mathrm{flip}}_{ki}\,.$$

On top of this, $\mathrm{e}^{\mathrm{fix}}_{ki} + \mathrm{e}^{\neg\mathrm{fix}}_{ki}$ components differ in sign between $w_{ki}$ and $w^\star$. By the triangle inequality, as $w_{K+1,0} = w^\star$, this implies that

$$\sum_{k'\in[k+1:K+1]} \mathrm{diff}(w_{k'-1,0}, w_{k',0}) \geq \mathrm{diff}(w_{k0}, w_{ki}) + \mathrm{diff}(w_{ki}, w^\star) = \mathrm{ct}^{\mathrm{flip}}_{ki} + \mathrm{e}^{\mathrm{fix}}_{ki} + \mathrm{e}^{\neg\mathrm{fix}}_{ki}\,.$$

Finally, $\mathrm{e}^{\mathrm{fix}}_{ki}$ of these have already been flipped in round $k$ by step $i$. These cannot be flipped again in the same round $k$, so they need to be included in some future round, i.e., in $\mathrm{diff}(w_{k'-1,0}, w_{k',0}))$ for $k' \geq k+2$:

$$\sum_{k'\in[k+2:K+1]} \mathrm{diff}(w_{k'-1,0}, w_{k',0}) \geq \mathrm{diff}(w_{k+1,1}, w^\star) \geq \mathrm{e}^{\mathrm{fix}}_{ki}\,.$$

By Lemma 2.4.3,

$$\mathrm{e}^{\neg\mathrm{fix}}_{ki} \leq p - \mathrm{ct}^{\mathrm{flip}}_{ki}\,,$$

and either $\mathrm{fix}_{ki} = \mathbf{1}$, implying $\mathrm{e}^{\neg\mathrm{fix}}_{ki} = 0$, or $\mathrm{e}^{\mathrm{fix}}_{ki} \leq \mathrm{ct}^{\mathrm{flip}}_{ki}$:

$$\mathrm{e}^{\mathrm{fix}}_{ki} \leq \mathrm{ct}^{\mathrm{flip}}_{ki} \qquad \text{or} \qquad \mathrm{e}^{\neg\mathrm{fix}}_{ki} = 0\,.$$

Therefore, we can apply Lemma 2.6.1 with $c_1 = \mathrm{ct}_{ki}^{\mathrm{flip}}$, $c_2 = \mathrm{e}_{ki}^{\mathrm{fix}}$, $c_3 = \mathrm{e}_{ki}^{\neg\mathrm{fix}}$ to optimize the parameters $(x_j)_{j\in[K-k+1]}$ where $x_j = \mathrm{diff}(w_{j+k-1}, w_{j+k})$, to get that

$$\prod_{k'\in[k+1:K+1]} g(\mathrm{diff}(w_{k'-1,0}, w_{k',0})) \le g(\mathrm{ct}_{ki}^{\mathrm{flip}} + \mathrm{e}_{ki}^{\neg\mathrm{fix}}) g(\mathrm{e}_{ki}^{\mathrm{fix}}) .$$

Therefore, the optimal policy's expected value (which equals the expectation of the only reward in the episode) is upper bounded as:

$$v^\star(s_{ki}) \ge \left( \prod_{k'\in[k]} g(\mathrm{diff}(w_{k'-1,0}, w_{k',0}) \right) g(\mathrm{ct}_{ki}^{\mathrm{flip}} + \mathrm{e}_{ki}^{\neg\mathrm{fix}}) g(\mathrm{e}_{ki}^{\mathrm{fix}}) = v^{\pi_{w^\star}}(s_{ki}) ,$$

by Lemma 2.5.1. Therefore $v^{\pi_{w^\star}}(s_{ki}) = v^\star(s_{ki})$.

■

## 2.7. Defining $\theta^\star$, $\varphi_v$, and $\varphi_q$, and showing realizability

By Lemma 2.4.2, and because $M^v_{w^\star}$ and $M^q_{w^\star}$ have the same transitions and rewards for any state $s \in \mathcal{S}_{\mathrm{r}}$, we do not notationally distinguish between $M^v_{w^\star}$ and $M^q_{w^\star}$ when describing the value or action-value functions of these MDPs on states $s \in \mathcal{S}_{\mathrm{r}}$, as these are the same in the two MDPs.

We define the feature-map $\varphi_v : \mathcal{S} \to \mathcal{B}_d(1)$ and $\varphi_q : \mathcal{S} \times [A] \to \mathcal{B}_d(1)$. For state $\bot$, let $\varphi_v(\bot) = \mathbf{0}$ and for all actions $a \in [A]$, $\varphi_q(\bot, a) = \mathbf{0}$. Realizability immediately holds as $v^\star(\bot) = q^\star(\bot, a) = 0 = \langle \mathbf{0}, \theta^\star \rangle$. For any state $s \in \mathcal{S}$, $s \neq \bot$, let $s = s_{ki}$ be a state along step $i$ of round $k$. Let us introduce the function

$$v'(s_{ki}) = \left( \prod_{k'\in[k]} g(\mathrm{diff}(w_{k'-1,0}, w_{k',0}) \right) g(\mathrm{ct}_{ki}^{\mathrm{flip}} + \mathrm{e}_{ki}^{\neg\mathrm{fix}}) g(\mathrm{e}_{ki}^{\mathrm{fix}}) . \tag{30}$$

By Lemmas 2.6.2 and 2.5.1, it holds that $v'(s_{ki}) = v^\star(s_{ki})$ if $s_{ki} \in \mathcal{S}_{\mathrm{r}}$. Observe that out of the terms above, only $\mathrm{e}_{ki}^{\neg\mathrm{fix}}$ and $\mathrm{e}_{ki}^{\mathrm{fix}}$ depends on $w^\star$, and this dependence is linear. In particular, recall that $\mathrm{ct}_{ki}^{\mathrm{flip}}$ depends only on the actions, and not on $w^\star$. Combined with the fact that $g$ is a second-order polynomial, $v'(s_{ki})$ is a fourth-order expression in $w^\star$, which can thus be linearized in $1 + p + p^2 + p^3 + p^4 \le d$ dimensions. Let $\bar{w}^\star = w^\star / \|w^\star\|_2 = w^\star/\sqrt{p}$, and

$$\theta^\star = 63 \left[ 1, \bar{w}^\star, (\bar{w}^\star)^{\otimes 2}, (\bar{w}^\star)^{\otimes 3}, (\bar{w}^\star)^{\otimes 4}, \mathbf{0}^{d-(1+p+p^2+p^3+p^4)} \right] , \tag{31}$$

where $\mathbf{0}^{d-(1+p+p^2+p^3+p^4)}$ is a vector of zeros of dimensionality $d-(1+p+p^2+p^3+p^4)$, serving the purpose to pad the vector to exactly $d$ dimensions, as required by the definition. As $\|\bar{w}^\star\|_2 = 1$, we have that

$$\left\|\theta^\star\right\|_2 \le 63 \cdot 5 = 315 := B\,.$$

Finally, for $Z_{(0)}, Z_{(1)}, Z_{(2)}, Z_{(3)}, Z_{(4)}$ calculated in Appendix 2.A.1, if we let

$$\varphi_v(s_{ki}) = \frac{1}{63}\left(\prod_{k'\in[k]} g(\mathrm{diff}(w_{k'-1,0}, w_{k',0}))\right)\left[Z_{(0)}, Z_{(1)}, Z_{(2)}, Z_{(3)}, Z_{(4)}, \mathbf{0}^{d-(1+p+p^2+p^3+p^4)}\right], \tag{32}$$

then by Eq. 41,

$$\left\langle \varphi_v(s_{ki}), \theta^\star \right\rangle = \left(\prod_{k'\in[k]} g(\mathrm{diff}(w_{k'-1,0}, w_{k',0})\right) g(\mathrm{ct}_{ki}^{\mathrm{flip}} + \mathrm{e}_{ki}^{\neg\mathrm{fix}}) g(\mathrm{e}_{ki}^{\mathrm{fix}}) = v'(s_{ki}) \tag{33}$$

$$\tag{34}$$

Eq. 32 completes the definition of $\varphi_v$, while Eq. 33 implies that

$$0 \le \left\langle \varphi_v(s_{ki}), \theta^\star \right\rangle \le 1\,, \tag{35}$$

as $v'(s_{ki})$ is a product of $g(\cdot) \in [0,1]$ terms (as $\mathrm{diff}(\cdot,\cdot) \in [0,p]$). Furthermore, combining this with $\left\|\left[Z_{(0)}, Z_{(1)}, Z_{(2)}, Z_{(3)}, Z_{(4)}\right]\right\|_2 \le 63$ (by Eq. 41), we have that

$$\|\varphi_v(s)\|_2 \le 1 \qquad \text{for all } s \in \mathcal{S}\,,$$

which ensures that $\varphi_v : \mathcal{S} \to \mathcal{B}_d(1)$. We stress that, as required, $\varphi_v(s_{ki})$ does not depend on $w^\star$.

To show $v^\star$-realizability with these features, i.e., that $v^\star(s_{ki}) = \langle \varphi_v(s_{ki}), \theta^\star \rangle$, we start by pointing out that if $s_{ki} \in \mathcal{S}_{\neg\mathrm{r}}$ then this immediately holds:

**Lemma 2.7.1.** *For any state $s \in \mathcal{S}_{\neg\mathrm{r}}$ and action $a \in [A]$, regardless of the values of $\varphi_v(s)$, $\varphi_q(s,a)$, and $\theta^\star$, $v^\star$-realizability for $M^v_{w^\star}$ and $q^\star$-realizability for $M^q_{w^\star}$ immediately holds as the transition*

*falls under Case 23a:*

$$v^\star_{M^v_{w^\star}}(s) = \langle \varphi_v(s), \theta^\star \rangle$$
$$q^\star_{M^q_{w^\star}}(s,a) = \langle \varphi_q(s,a), \theta^\star \rangle$$

Otherwise $s_{ki} \in \mathcal{S}_{\mathrm{r}}$, and $v^\star$-realizability follows from Eq. 33 by recalling that $v'(s_{ki}) = v^\star(s_{ki})$ in this case. We conclude the following lemma from this:

**Lemma 2.7.2.** *$M^v_{w^\star}$ is $v^\star$-realizable with features $\varphi_v$: $(M^v_{w^\star}, \varphi_v) \in \mathcal{M}^{v^\star}_{B,d,H,A} \cap \mathcal{M}^{\mathrm{Pdet}}$.*

We move on to defining $\varphi_q$ and showing $q^\star$-realizability for $M^q_{w^\star}$. For any state $s \in \mathcal{S}$, $s \neq \perp$ and action $a \in [A]$, let $s = s_{ki}$ be a state along step $i$ of round $k$. Let $s^a_{k,i+1}$ denote the value taken by $s_{k,i+1}$ if $a_{ki} = a$, and similarly for $w^a_{k,i+1}$. For $i = p-1$ only, let us introduce

$$q'(s_{k,p-1}, a) = \left( \prod_{k' \in [k]} g(\mathrm{diff}(w_{k'-1,0}, w_{k',0}) \right) g(\mathrm{diff}(w_{k0}, w^a_{k+1,0})) g(\mathrm{diff}(w^a_{k+1,0}, w^\star)) \,. \tag{36}$$

Let

$$c(s_{ki}, a) = \frac{1}{63} \left( \prod_{k' \in [k]} g(\mathrm{diff}(w_{k'-1,0}, w_{k',0}) \right) g(\mathrm{diff}(w_{k0}, w^a_{k+1,0})) \,.,$$

which is a scalar that does not depend on $w^\star$. The only remaining term in $q'$ has a second-order dependence on $w^\star$. For $X_{(0)}, X_{(1)}, X_{(2)}$ calculated in Appendix 2.A.2, we let

$$\varphi_q(s_{ki}, a) = \begin{cases} \varphi_v(s^a_{k,i+1}) & \text{else if } i < p; \qquad (37a) \\ c(s_{ki}, a) \left[ X_{(0)}, X_{(1)}, X_{(2)}, \mathbf{0}^{d-(1+p+p^2)} \right] & \text{otherwise,} \qquad (37b) \end{cases}$$

where $\mathbf{0}^{d-(1+p+p^2)}$ is a vector of zeros of dimensionality $d-(1+p+p^2)$. Then by Eq. 44, for $\theta^\star$ set according to Eq. 31,

$$\langle \varphi_q(s_{k,p-1}, a), \theta^\star \rangle = q'(s_{k,p-1}, a) \,. \tag{38}$$

Eq 37 completes the definition of $\varphi_q$, while Eq. 38 together with Eq. 35 implies that

$$0 \leq \langle \varphi_q(s_{ki}, a), \theta^\star \rangle \leq 1 \qquad \text{for all } a \in [A],\, s_{ki} \in \mathcal{S},\, s_{ki} \neq \perp, \tag{39}$$

as $q'(s_{k,p-1}, a)$ is a product of $g(\mathrm{diff}(\cdot,\cdot)) \in [0,1]$ terms. Furthermore, combining this with $\left\|\left[X_{(0)}, X_{(1)}, X_{(2)}\right]\right\|_2 \le 8$ (by Eq. 44), we have that.

$$\left\|\varphi_q(s,a)\right\|_2 \le 1 \qquad \text{for all } s,a \in \mathcal{S}\times[A],$$

which ensures that $\varphi_q : \mathcal{S}\times[A] \to \mathcal{B}_d(1)$, as required. Again we stress that $\varphi_v(s_{ki})$ does not depend on $w^\star$.

To show $q^\star$-realizability, we first consider the case when $s_{ki} \in \mathcal{S}_{\mathrm{r}}$ and $i = p-1$ i.e., $\varphi_q(s_{ki}, a)$ falls under Case 37b. In this case,

$$\begin{aligned}
\left\langle \varphi_q(s_{k,p-1},a), \theta^\star \right\rangle &= q'(s_{k,p-1},a) \\
&= \left( \prod_{k'\in[k]} g(\mathrm{diff}(w_{k'-1,0}, w_{k',0}) \right) g(\mathrm{diff}(w_{k0}, w^a_{k+1,0})) g(\mathrm{diff}(w^a_{k+1,0}, w^\star)) \\
&= q^\star(s_{k,p-1},a),
\end{aligned}$$

where the first equality comes from Eq 38. The last equality holds by definition if the transition and reward follows Case 23b or 23c; otherwise under Case 23d, it holds since

$$q^\star(s_{k,p-1},a) = v^\star(s^a_{k+1,0}) = q'(s_{k,p-1},a),$$

where the second equality follows from Lemmas 2.5.1 and 2.6.2.

Turning to the case where $s_{ki} \in \mathcal{S}_{\mathrm{r}}$ and $i < p-1$, we note that $\varphi_q(s_{ki},a)$ falls under Case 37a, while the transition and reward follows Case 23d. Therefore

$$q^\star(s_{ki},a) = v^\star(s^a_{k,i+1}) = \left\langle \varphi_v(s^a_{k,i+1}), \theta^\star \right\rangle = \left\langle \varphi_q(s_{k,i+1},a), \theta^\star \right\rangle,$$

where the second equality follows from Lemma. 2.7.2.

Together with Lemma 2.7.1 that proves $q^\star$-realizability for the case of $s_{ki} \in \mathcal{S}_{\neg\mathrm{r}}$, we conclude that the following holds:

**Lemma 2.7.3.** *$M^q_{w^\star}$ is $q^\star$-realizable with features $\varphi_q$: $(M^q_{w^\star}, \varphi_q) \in \mathcal{M}^{q^\star}_{B,d,H,A} \cap \mathcal{M}^{\mathrm{Pdet}}$.*

Recall that $\mathrm{Reach}(s_{00}) \subseteq \mathcal{S}_{\mathrm{r}}$ under either MDP $M^v_{w^\star}$ or $M^q_{w^\star}$ (by Lemma 2.4.2), and that value and action-value functions on such states take the same value for the two MDPs. Then, combining Lemmas 2.7.2 and 2.7.3, we have the following result:

**Lemma 2.7.4.** *$M^{v}_{w^\star}$ is reachable-$v^\star/q^\star$-realizable with features $\varphi_v$ and $\varphi_q$: $(M^{v}_{w^\star}, \varphi_v, \varphi_q) \in \mathcal{M}^{v^\star/q^\star\text{reach}}_{B,d,H,A} \cap \mathcal{M}^{\text{Pdet}}$.*

## 2.8. Reduction to planning in the abstract game

*Proof of Theorem 1.4.2.* Let $\delta \geq 0.01, B \geq 315, d \geq 31, H \geq 81$. In what follows, we prove the theorem only for $A$ (and $p$) set according to Eq. 21. This is sufficient to prove the result for $A \geq \left\lfloor \sqrt{H} \right\rfloor \wedge 0.8d^{14} \geq p$ (Eq. 21) as soundness with a lower action count cannot be harder to achieve, since it is always possible to duplicate some actions without changing the difficulty of the problem.

Let $\mathcal{P}$ be any $\delta$-sound planner with worst-case query cost $\bar{N}$ for some class class $\mathcal{M} \cap \mathcal{M}^{\text{Pdet}}$, where

$$\mathcal{M} \in \{\mathcal{M}^{v^\star}_{B,d,H,A}, \mathcal{M}^{q^\star}_{B,d,H,A}, \mathcal{M}^{v^\star/q^\star\text{reach}}_{B,d,H,A}\}\,.$$

We show that $\mathcal{P}$ gives rise to a sound abstract planner for the abstract game of Section 2.2 and therefore, by Theorem 2.2.1, it must use exponentially many queries.

Lemmas 2.7.2, 2.7.3, and 2.7.4 show that MDPs $\left(M^{v}_{w^\star}\right)_{w^\star \in W^\star}$, $\left(M^{q}_{w^\star}\right)_{w^\star \in W^\star}$, and $\left(M^{v}_{w^\star}\right)_{w^\star \in W^\star}$ respectively, together with feature-maps $\varphi_v$ and $\varphi_q$, belong to these classes. Therefore, the $\delta$-sound planner $\mathcal{P}$ satisfies, for any MDP $M$ with parameter $w^\star$ in its class:

$$v^{\pi_M}_{M}(s_{00}) \geq v^{\star}_{M}(s_{00}) - 0.01\,,$$

where $s_{00}$ is the initial state in $M$ and $\pi_M$ is the policy induced by the interconnection of $\mathcal{P}$ and MDP $M$ in a closed-loop fashion, according to Section 1.2.3. Let $\mathbb{P}$ and $\mathbb{E}$ be the probability measure and expectation, respectively, induced by this interconnection. Then,

$$v^{\pi_M}_{M}(s_{00}) = \mathbb{E}\left[\sum_{t=1}^{H} R_t \,\middle|\, S_0 = s_{00}\right] \geq v^\star(s_{00}) - 0.01$$
$$\mathbb{E}\left[\sum_{t=1}^{8p} R_t + v^\star(S_{8\cdot p}) \,\middle|\, S_0 = s_{00}\right] \geq \mathbb{E}\left[\sum_{t=1}^{H} R_t \,\middle|\, S_0 = s_{00}\right] \geq v^\star(s_{00}) - 0.01\,,$$

where we put $\cdot$ in the index of $S$ to signify multiplication: as opposed to $s$, $S$ only has a single index. It is valid to refer to the state $S_{8\cdot p}$ as $K \geq 9$ by Eq. 22. Let us map any partial trajectory $S_0, A_0, S_1, A_1, \ldots, S_{8\cdot p-1}, A_{8\cdot p-1}$ to the sequence $(\tilde{w}_i)_{i\in[8]} \in W^{\circ 8}$ as follows. Let $j \in [8]$ be the smallest index for which $S_{j\cdot p-1} = \perp$, or let $j = 9$ if no such index exist in $[8]$. For $i \in [j-1]$,

let $\tilde{w}_i = w(S_{i\cdot p-1}, A_{i\cdot p-1})$; for $i \in [j:8]$, let $\tilde{w}_i$ be any values such that $(\tilde{w}_i)_{i\in[8]} \in W^{\circ 8}$ (which is always possible as $(\tilde{w}_i)_{i\in[j-1]} \in W^{\circ j-1}$ when $j > 1$). Let

$$k^\star = \min\{i \in [8] : i = 8 \text{ or } \operatorname{diff}(\tilde{w}_i, w^\star) < p/4\}\,.$$

Let $R$ be the final reward of an abstract game (with the same parameters $K, p, w^\star$) for this sequence $(\tilde{w}_i)_{i\in[8]}$. By Eq. 10,

$$R = f_{w^\star}(w(S_{i\cdot p}))_{i\in[k^\star]}\,.$$

Observe that if there is an illegal action in the sequence $A_0, \ldots, A_{8\cdot p-1}$, then $\sum_{t=1}^{8p} R_t + v^\star(S_{8\cdot p}) = 0$. Otherwise, if $\operatorname{diff}(\tilde{w}_i, w^\star) \geq p/4$ for all $i \in [8]$, then all transitions leading to $S_{8\cdot p}$ fall under Case 23d as $K \geq 9$, and by Lemmas 2.5.1 and 2.6.2, $R = v^\star(S_{8\cdot p}) = \sum_{t=1}^{8p} R_t + v^\star(S_{8\cdot p})$. Finally, if $\operatorname{diff}(\tilde{w}_{k^\star}, w^\star) < p/4$, then $S_{k^\star\cdot p} = \perp$, $v^\star(S_{k^\star\cdot p}) = 0$, $R = R_{k^\star\cdot p}$, and the rest of the rewards are zero. Therefore, either way,

$$\mathbb{E}[R] \geq \mathbb{E}\left[\sum_{t=1}^{8p} R_t + v^\star(S_{8\cdot p}) \,\middle|\, S_0 = s_{00}\right] \geq v^\star(s_{00}) - 0.01 = f_{w^\star}(()) \,. \tag{40}$$

Recall that each response to $\mathcal{P}$'s query to the MDP's simulator, as well as the transitions $(R_{t+1}, S_{t+1}) \sim Q(\cdot|S_t, A_t)$ (for $t \in [0 : H-1]$) can be implemented with at most one simulator call (respectively) to the abstract game (with the same parameters $K, p, w^\star$; see Lemma 2.4.1). In expectation, this results in at most $8p\bar{N} + 8p$ such queries to the abstract game simulator. Together with Eq. 40, and noting that the choice of $w^\star \in W^\star$ was arbitrary, we see that $\mathcal{P}$ can be used to construct an abstract planner $\mathcal{A}$ that is sound with worst-case query cost $8p\bar{N} + 8p$. Therefore, by Theorem 2.2.1, and using Eq. 22,

$$\begin{aligned} 8p\bar{N} + 8p &= 2^{\Omega(p \wedge K)} \\ \bar{N} &= 2^{\Omega(H^{1/2} \wedge d^{1/4})}\,. \end{aligned}$$

■

# Appendix

## 2.A. Calculating the linear features

### 2.A.1. Calculating feature components of $\varphi_v$

We follow the notation of Section 2.7. In particular, for any state $s \in \mathcal{S}$, $s \neq \perp$, let $s = s_{ki}$ be a state along step $i$ of round $k$. We intend to linearize the expression $g(\mathrm{ct}_{ki}^{\mathrm{flip}} + \mathrm{e}_{ki}^{\neg\mathrm{fix}}) g(\mathrm{e}_{ki}^{\mathrm{fix}})$.

Let $x = \mathrm{ct}_{ki}^{\mathrm{flip}} + \mathrm{e}_{ki}^{\neg\mathrm{fix}}$ and $y = \mathrm{e}_{ki}^{\mathrm{fix}}$. Then, $x$ and $y$ can be written according to Eqs. 25, 26 as:

$$
\begin{aligned}
y &= \frac{1}{2}\left(\langle \mathbf{1}, \mathrm{fix}_{ki}\rangle - \left\langle \mathrm{fix}_{ki} \cdot \bar{w}_{ki}, \bar{w}^{\star}\right\rangle\right) = \left\langle y_{(1,0)}, 1\right\rangle + \left\langle y_{(1,1)}, \bar{w}^{\star}\right\rangle \\
&\quad \text{for } y_{(1,0)} = \frac{1}{2}\langle \mathbf{1}, \mathrm{fix}_{ki}\rangle \text{ and } y_{(1,1)} = -\frac{\sqrt{p}}{2}\mathrm{fix}_{ki} \cdot \bar{w}_{ki} \\
&\quad \text{with } \left\|y_{(1,0)}\right\|_2, \left\|y_{(1,1)}\right\|_2 \leq p \\
x &= \mathrm{ct}_{ki}^{\mathrm{flip}} + \frac{1}{2}\left(\langle \mathbf{1}, \neg\mathrm{fix}_{ki}\rangle - \left\langle \neg\mathrm{fix}_{ki} \cdot \bar{w}_{ki}, \bar{w}^{\star}\right\rangle\right) = \left\langle x_{(1,0)}, 1\right\rangle + \left\langle x_{(1,1)}, \bar{w}^{\star}\right\rangle \\
&\quad \text{for } x_{(1,0)} = \mathrm{ct}_{ki}^{\mathrm{flip}} + \frac{1}{2}\langle \mathbf{1}, \neg\mathrm{fix}_{ki}\rangle \text{ and } x_{(1,1)} = -\frac{\sqrt{p}}{2}\neg\mathrm{fix}_{ki} \cdot \bar{w}_{ki} \\
&\quad \text{with } \left\|x_{(1,0)}\right\|_2, \left\|x_{(1,1)}\right\|_2 \leq p
\end{aligned}
$$

Notice that $x_{(\cdot,\cdot)}$ and $y_{(\cdot,\cdot)}$ do not depend on $w^\star$, only on the current state $s_{ki}$. Furthermore, using Lemma 1.2.1

$$
\begin{aligned}
x^2 &= \left\langle x_{(1,0)}^2, 1\right\rangle + \left\langle 2x_{(1,0)}x_{(1,1)}, \bar{w}^\star\right\rangle + \left\langle \flat(x_{(1,1)}\otimes x_{(1,1)}), \flat(\bar{w}^\star\otimes\bar{w}^\star)\right\rangle \\
&= \left\langle x_{(2,0)}, 1\right\rangle + \left\langle x_{(2,1)}, \bar{w}^\star\right\rangle + \left\langle x_{(2,2)}, (\bar{w}^\star)^2\right\rangle \\
&\qquad \text{for } x_{(2,0)} = x_{(1,0)}^2,\ x_{(2,1)} = 2x_{(1,0)}x_{(1,1)},\ x_{(2,2)} = \flat(x_{(1,1)}\otimes x_{(1,1)}), \\
&\qquad\quad \text{and } (\bar{w}^\star)^{\otimes 2} = \flat(\bar{w}^\star\otimes\bar{w}^\star) \\
&\qquad \text{with } \left\|x_{(2,0)}\right\|_2, \left\|x_{(2,1)}\right\|_2, \left\|x_{(2,2)}\right\|_2 \le 2p^2 \\
g(x) &= 1 + x\frac{-2p-1}{2p^2} + x^2\frac{1}{2p^2} = \left\langle X_{(0)}, 1\right\rangle + \left\langle X_{(1)}, \bar{w}^\star\right\rangle + \left\langle X_{(2)}, (\bar{w}^\star)^2\right\rangle \\
&\qquad \text{for } X_{(0)} = 1 + \frac{-2p-1}{2p^2}x_{(1,0)} + \frac{1}{2p^2}x_{(2,0)}, \\
&\qquad\quad X_{(1)} = \frac{-2p-1}{2p^2}x_{(1,1)} + \frac{1}{2p^2}x_{(2,1)}, \text{ and } X_{(2)} = \frac{1}{2p^2}x_{(2,2)} \\
&\qquad \text{with } \left\|X_{(0)}\right\|_2 \le 4, \left\|X_{(1)}\right\|_2 \le 3, \left\|X_{(2)}\right\|_2 \le 1.
\end{aligned}
$$

and by a similar calculation,

$$
\begin{aligned}
y^2 &= \left\langle y_{(2,0)}, 1\right\rangle + \left\langle y_{(2,1)}, \bar{w}^\star\right\rangle + \left\langle y_{(2,2)}, (\bar{w}^\star)^2\right\rangle \\
&\qquad \text{for } y_{(2,0)} = y_{(1,0)}^2,\ y_{(2,1)} = 2y_{(1,0)}y_{(1,1)},\ y_{(2,2)} = \flat(y_{(1,1)}\otimes y_{(1,1)}) \\
&\qquad \text{with } \left\|y_{(2,0)}\right\|_2, \left\|y_{(2,1)}\right\|_2, \left\|y_{(2,2)}\right\|_2 \le 2p^2 \\
g(y) &= \left\langle Y_{(0)}, 1\right\rangle + \left\langle Y_{(1)}, \bar{w}^\star\right\rangle + \left\langle Y_{(2)}, (\bar{w}^\star)^2\right\rangle \\
&\qquad \text{for } Y_{(0)} = 1 + \frac{-2p-1}{2p^2}y_{(1,0)} + \frac{1}{2p^2}y_{(2,0)}, \\
&\qquad\quad Y_{(1)} = \frac{-2p-1}{2p^2}y_{(1,1)} + \frac{1}{2p^2}y_{(2,1)}, \text{ and } Y_{(2)} = \frac{1}{2p^2}y_{(2,2)} \\
&\qquad \text{with } \left\|Y_{(0)}\right\|_2 \le 4, \left\|Y_{(1)}\right\|_2 \le 3, \left\|Y_{(2)}\right\|_2 \le 1.
\end{aligned}
$$

Therefore, again using Lemma 1.2.1,

$$
\begin{aligned}
g(\mathrm{ct}_{ki}^{\mathrm{flip}} + \mathrm{e}_{ki}^{\neg\mathrm{fix}}) g(\mathrm{e}_{ki}^{\mathrm{fix}}) &= g(x) g(y) \\
&= \left\langle \flat\left(X_{(0)} \otimes Y_{(0)}\right), 1 \right\rangle + \left\langle \flat\left(X_{(0)} \otimes Y_{(1)} + X_{(1)} \otimes Y_{(0)}\right), \bar{w}^\star \right\rangle \\
&\quad + \left\langle \flat\left(X_{(0)} \otimes Y_{(2)} + X_{(1)} \otimes Y_{(1)} + X_{(2)} \otimes Y_{(0)}\right), (\bar{w}^\star)^{\otimes 2} \right\rangle \\
&\quad + \left\langle \flat\left(X_{(1)} \otimes Y_{(2)} + X_{(2)} \otimes Y_{(1)}\right), (\bar{w}^\star)^{\otimes 3} \right\rangle + \left\langle \flat\left(X_{(2)} \otimes Y_{(2)}\right), (\bar{w}^\star)^{\otimes 4} \right\rangle \\
&= \left\langle Z_{(0)}, 1 \right\rangle + \left\langle Z_{(1)}, \bar{w}^\star \right\rangle + \left\langle Z_{(2)}, (\bar{w}^\star)^{\otimes 2} \right\rangle + \left\langle Z_{(3)}, (\bar{w}^\star)^{\otimes 3} \right\rangle + \left\langle Z_{(4)}, (\bar{w}^\star)^{\otimes 4} \right\rangle \\
&\quad \text{for } Z_{(0)} = \flat\left(X_{(0)} \otimes Y_{(0)}\right), \\
&\qquad Z_{(1)} = \flat\left(X_{(0)} \otimes Y_{(1)} + X_{(1)} \otimes Y_{(0)}\right), \\
&\qquad Z_{(2)} = \flat\left(X_{(0)} \otimes Y_{(2)} + X_{(1)} \otimes Y_{(1)} + X_{(2)} \otimes Y_{(0)}\right), \\
&\qquad Z_{(3)} = \flat\left(X_{(1)} \otimes Y_{(2)} + X_{(2)} \otimes Y_{(1)}\right), \\
&\qquad (\bar{w}^\star)^{\otimes 3} = \flat\left(\bar{w}^\star \otimes \bar{w}^\star \otimes \bar{w}^\star\right), \\
&\qquad (\bar{w}^\star)^{\otimes 4} = \flat\left(\bar{w}^\star \otimes \bar{w}^\star \otimes \bar{w}^\star \otimes \bar{w}^\star\right) \\
&\quad \text{with } \left\|Z_{(0)}\right\|_2 \le 16, \left\|Z_{(1)}\right\|_2 \le 24, \left\|Z_{(2)}\right\|_2 \le 17, \left\|Z_{(3)}\right\|_2 \le 6.
\end{aligned}
\tag{41}
$$

### 2.A.2. Calculating feature components of $\varphi_q$

We follow the notation of Section 2.7. In particular, for any state $s \in \mathcal{S}$, $s \neq \perp$ and action $a \in [A]$, let $s = s_{ki}$ be a state along step $i$ of round $k$. Let $s^a_{k,i+1}$ denote the value taken by $s_{k,i+1}$ if $a_{ki} = a$, and similarly for $w^a_{k,i+1}$. We intend to linearize the expression $g(\mathrm{diff}(w^a_{k+1,0}, w^\star))$.

Let $x = \mathrm{diff}\left(w^a_{k+1,0}, w^\star\right)$. By Eq. 7,

$$
\begin{aligned}
x = \frac{1}{2}\left(p - \left\langle w^a_{k+1,1}, w_\star \right\rangle\right) &= \left\langle x_{(1,0)}, 1 \right\rangle + \left\langle x_{(1,1)}, w^\star \right\rangle \\
&\quad \text{for } x_{(1,0)} = \frac{1}{2}p \text{ and } x_{(1,1)} = -\frac{1}{2} w^a_{k+1,1} \\
&\quad \text{with } \left\|x_{(1,0)}\right\|_2, \left\|x_{(1,1)}\right\|_2 \le p
\end{aligned}
\tag{42}
$$

By a similar calculation to the previous case,

$$
\begin{aligned}
x^2 &= \left\langle x_{(1,0)}^2, 1 \right\rangle + \left\langle 2x_{(1,0)}x_{(1,1)}, w^\star \right\rangle + \left\langle \flat(x_{(1,1)} \otimes x_{(1,1)}), \flat(w^\star \otimes w^\star) \right\rangle \\
&= \left\langle x_{(2,0)}, 1 \right\rangle + \left\langle x_{(2,1)}, w^\star \right\rangle + \left\langle x_{(2,2)}, (w^\star)^2 \right\rangle \\
&\qquad \text{for } x_{(2,0)} = x_{(1,0)}^2,\ x_{(2,1)} = 2x_{(1,0)}x_{(1,1)},\ x_{(2,2)} = \flat(x_{(1,1)} \otimes x_{(1,1)}), \\
&\qquad\quad \text{and } (w^\star)^{\otimes 2} = \flat(w^\star \otimes w^\star) \\
&\qquad \text{with } \left\|x_{(2,0)}\right\|_2, \left\|x_{(2,1)}\right\|_2, \left\|x_{(2,2)}\right\|_2 \le 2p^2
\end{aligned} \tag{43}
$$

$$
\begin{aligned}
g(x) &= 1 + x\frac{-2p-1}{2p^2} + x^2\frac{1}{2p^2} = \left\langle X_{(0)}, 1 \right\rangle + \left\langle X_{(1)}, w^\star \right\rangle + \left\langle X_{(2)}, (w^\star)^2 \right\rangle \\
&\qquad \text{for } X_{(0)} = 1 + \frac{-2p-1}{2p^2}x_{(1,0)} + \frac{1}{2p^2}x_{(2,0)}, \\
&\qquad\quad X_{(1)} = \frac{-2p-1}{2p^2}x_{(1,1)} + \frac{1}{2p^2}x_{(2,1)}, \text{ and } X_{(2)} = \frac{1}{2p^2}x_{(2,2)} \\
&\qquad \text{with } \left\|X_{(0)}\right\|_2 \le 4, \left\|X_{(1)}\right\|_2 \le 3, \left\|X_{(2)}\right\|_2 \le 1.
\end{aligned} \tag{44}
$$

## Chapter 3

# TensorPlan: efficient planning for few actions

## 3.1. Introduction

This chapter focuses on proving the upper bound of Theorem 1.4.4. We start by considering the MDP class $\mathcal{M}^{v^\star}_{B,d,H,A}$. Despite the fact that linear function approximation reduces the number of unknowns to $d$ (from the unbounded size of the state space), it is not clear at all whether this setting is tractable.

In fact, the lower bound (Theorem 1.4.2) proved in the previous chapter establishes that the related problem when the action-value function of the optimal policy is linearly realizable requires an exponential number of queries, either in $H$ (the horizon of the MDP) or $d$ (the dimension of the feature mapping). The construction however crucially relies on having an action set that scales as a polynomial in the relevant parameters. In contrast, in this chapter, we establish that $\text{poly}(H, d)$ planning *is* possible with state value function realizability whenever the action set has a constant size. In particular, we present the TensorPlan algorithm which uses $\text{poly}((dH/\delta)^A)$ simulator queries to find a $\delta$-optimal policy relative to *any* deterministic policy for which the value function is linearly realizable with some bounded parameter (with a known bound).

This is the first algorithm to give a polynomial query complexity guarantee using only linear-realizability of a single competing value function. We extend the upper bound to the near-realizable case, to the infinite-horizon discounted MDP setup, and finally to the MDP classes $\mathcal{M}^{q^\star}_{B,d,H,A} \cap \mathcal{M}^{\text{Pdet}}$ and $\mathcal{M}^{v^\star/q^\star\text{reach}}_{B,d,H,A}$.

To summarize, the central question we address first in this chapter is the following:

*Is a polynomial query complexity achievable under linear realizability of $v^\star$, when the number of actions is $A = \mathcal{O}(1)$?*

We provide a positive result to this question in the *fixed-horizon* setting, where our algorithm TensorPlan enjoys a per-call query complexity poly$((dH/\delta)^A)$, where $H$ is the horizon and $\delta$ is the suboptimality target that the policy induced by continuously running the planning algorithm at every state encountered needs to satisfy. Given an input state at the beginning of the horizon, in its initialization phase, TensorPlan uses simulations to estimate the parameters of $v^\star$. In this and subsequent calls, given an input state, the estimated $v^\star$ is used by another procedure that uses additional simulations to compute one-step lookahead action-value estimates. We prove that the resulting policy loses at most $\delta$ total expected reward compared to optimality, regardless of the choice of the initial state, while the number of queries both for the initialization and the subsequent steps stays below the quoted polynomial bound.

In fact, TensorPlan enjoys a stronger guarantee – it will automatically compete with the *best deterministic policy* whose value function is realizable by the features. This recovers the previously mentioned "classic" setting: when $v^\star$ is realizable the best deterministic policy is an optimal policy $\pi^\star$.

Loosely, the initialization phase of our algorithm works in the following way: The algorithm keeps track of a list of critical data that is used to refine a hypothesis set that contains those $d$-dimensional parameter vectors that (may) induce a value function for some deterministic policy. Call these parameter vectors consistent. The algorithm refines its hypothesis set in a number of phases. For this, at the beginning of a phase, it chooses a parameter vector from the hypothesis set that maximizes the total predicted value at the initial state; an "optimistic choice". Next, the algorithm runs a fixed number of tests to verify that the parameter vector chosen gives a value function of some policy. If this consistency is satisfied, it also follows that the predicted value is almost as high as the actual value of the parameter-induced policy. As such, by its optimistic choice, the parameter vector gives rise to the policy whose value function is linearly realizable and whose value is the highest in the initial state. When the test fails, the hypothesis set is shrunk by expanding the list of critical data with data from the failed test. To show that the hypothesis set shrinks rapidly, we introduce a novel tensorization device that lifts the consistency checking problem to a $d^A$-dimensional Euclidean space where the tests become linear. This tensorization device allows us to prove that at most $\mathcal{O}(d^A)$ constraints can be added (in the noise-free case) if there exist a deterministic policy with linearly realizable value function.

The rest of the chapter is structured as follows. The upcoming section (Section 3.2) introduces notations, definitions, and the formal problem definition. This is a slight variation of the definitions

introduced in Chapter 1 that better suits the specific problem class of $\mathcal{M}^{v^\star}_{B,d,H,A}$. Then, Section 3.3 presents the TensorPlan algorithm for efficient planning in the finite-horizon setting, and states the query complexity guarantee (Theorem 3.3.2), as well as an extension of this result to the near-realizable case (Theorem 3.3.4) and infinite-horizon discounted case (Theorem 3.3.5). Finally, we conclude in Section 3.4.

## 3.2. Preliminaries

We recall the most important facts about MDPs and introduce a slight variation of our previous notation that allows infinite state spaces. Given a measurable space $(\mathcal{X}, \Sigma)$, we write $\mathcal{M}_1(\mathcal{X})$ for the set of probability measures on that space (the $\sigma$-algebra will be understood from context). Here, an MDP is given by a tuple $\mathscr{M} = (\mathcal{S}, \Sigma, \mathcal{A}, Q)$, where $(\mathcal{S}, \Sigma)$ is a measurable state space, $\mathcal{A}$ is a set of actions and for each $(s,a) \in \mathcal{S} \times \mathcal{A}$, $Q_{s,a} \in \mathcal{M}_1([0,1] \times \mathcal{S})$ is a probability measure on rewards and next-state transitions received upon taking action $a$ at state $s$. Note that it follows that the random rewards are bounded in $[0,1]$. We denote by $r_{sa}$ the expected reward when using action $a$ in state $s$: $r_{sa} = \mathbb{E}_{(R',S') \sim Q(\cdot|s,a)} R'$. Further, we let $P_{sa}$ denote the distribution of the next-state: $P_{sa}(s') = \mathcal{P}_{(R',S') \sim Q(\cdot|s,a)}(S' = s')$. We assume that $\mathcal{A}$ is finite, and thus without loss of generality we let $\mathcal{A} = [A]$ for some integer $A \geq 2$. For notational simplicity, in this chapter, for any two tensors $A$, $B$ of compatible shapes, let $\langle A, B \rangle$ denote their flattened inner product, i.e., $\langle A, B \rangle = \langle \flat(A), \flat(B) \rangle = \flat(A)^\top \flat(B)$.

In the **fixed-horizon setting** with horizon $H \geq 1$ the agent (a decision maker) interacts with the MDP in an $H$-step sequential process as follows: The process is initialized at a random initial state $S_1 \in \mathcal{S}$. In step $h \in [H]$, the agent first observes the current state $S_h \in \mathcal{S}$, then chooses an action $A_h \in \mathcal{A}$ based on the information available to it. The MDP then gives a reward $R_h$ and transitions to a next-state $S_{h+1}$, where $(R_h, S_{h+1}) \sim Q_{S_h, A_h}$. After time-step $H$, the episode terminates.

The goal of the agent is to maximize the total expected reward $\sum_{h \in [H]} R_h$ for the episode by choosing the actions based on the observed past states and actions in the episode. A *(memoryless) policy* $\pi$ takes the form $(\pi^{(h)})_{h \in [H]}$ where $\forall h \in [H]$, $\pi^{(h)} : \mathcal{S} \to \mathcal{M}_1(\mathcal{A})$. A *deterministic policy* $\pi$ further satisfies that for any $h \in [H]$ and $s \in \mathcal{S}$ there exists $a \in \mathcal{A}$ such that $\pi^{(h)}(s) = \delta_a$ where $\delta$ is the Dirac delta distribution. Given a memoryless policy $\pi$, a state $s \in \mathcal{S}$ and step $h \in [H]$ within an episode, the value $v^\pi_h(s)$ is defined as the total expected reward incurred until the end of the episode when the MDP is started from $s$ in step $h$ and $\pi$ is followed throughout. Writing $\mu f = \int f(s') \mu(ds')$ for the expected value of a measurable function $f : \mathcal{S} \to \mathbb{R}$ with respect to $\mu \in \mathcal{M}_1(\mathcal{S})$, these values

are known to satisfy

$$v_h^\pi(s) = r_\pi(s) + P_\pi(s) v_{h+1}^\pi, \qquad s \in \mathcal{S},$$

where $v_{H+1}^\pi = 0$, $r_\pi(s) = \sum_{a\in\mathcal{A}} \pi(a|s) r_{sa}$, and $P_\pi(s)(ds') = \sum_{a\in\mathcal{A}} \pi(a|s) P_{sa}(ds')$. The maximum value achievable from a state $s \in \mathcal{S}$ when in step $h \in [H]$ is denoted by $v_h^\star(s)$. We also define $v_{H+1}^\star(s) = 0$, for convenience. We let $v^\star = (v_h^\star)_{h\in[H+1]}$ and call $v^\star$ the optimal value function. It is known that $v^\star$ satisfies the recursive *Bellman optimality equations*:

$$v_h^\star(s) = \max_{a\in\mathcal{A}} \left\{ r_{sa} + P_{sa} v_{h+1}^\star \right\}, \qquad s \in \mathcal{S}. \tag{45}$$

As is well known, the policy that in state $s \in \mathcal{S}$ chooses an action that maximizes the right-hand side of Eq. (45), is optimal. It also follows that there is always at least one optimal deterministic memoryless policy.

### 3.2.1. Featurized MDPs, feature map compatible optimal values

As noted earlier, we provide the planner with a feature mapping which captures the optimal value function. In the finite-horizon setting this translates to the existence of some $\theta^\star$ such that

$$v_h^\star(s) = \left\langle \varphi_h(s), \theta^\star \right\rangle, \qquad \text{for all } h \in [H] \text{ and } s \in \mathcal{S}. \tag{46}$$

We also consider the nearly-realizable case, where for some "misspecification" parameter $\eta \geq 0$ there exists some $\theta^\star$ such that

$$\left| v_h^\star(s) - \left\langle \varphi_h(s), \theta^\star \right\rangle \right| \leq \eta, \qquad \text{for all } h \in [H] \text{ and } s \in \mathcal{S}. \tag{47}$$

The parameter $\theta^\star$ is unknown to the planner in both cases. Here, $\varphi_h : \mathcal{S} \to \mathbb{R}^d$ is the so-called feature map. As will be described in more details in the next section, the planner is given *local access* to the feature map. That is, the planner can access $\varphi_h(s)$ for all the states $s \in \mathcal{S}$ that it has previously encountered while interacting with the simulator, but has no access to the features of other states. For convenience, in the finite horizon-setting we will also define $\varphi_{H+1}(s) = \mathbf{0}$ for all $s \in \mathcal{S}$, regardless of the other maps. An MDP together with a feature map $\varphi = (\varphi_h)_{h\in[H]}$ on its state-space is called a ***featurized MDP***. When Eq. (46) holds we say that $v^\star$ ***is (linearly) realizable by the feature map*** $\varphi$.

In this chapter we consider a setting that relaxes linear realizability of the optimal value function. To define this setting we need the notion of ***v*-linearly realizable policies**:

**Definition 3.2.1** ($v$-linearly realizable policies)**.** *We say that a policy $\pi$ is $v$-linearly realizable with misspecification $\eta \geq 0$ under the feature map $\varphi = (\varphi_h)_{h\in[H]}$ if there exists some $\theta \in \mathbb{R}^d$ such that its value function satisfies $\left|v_h^\pi(s) - \langle \varphi_h(s), \theta\rangle\right| \leq \eta$ for all $h \in [H]$ and $s \in \mathcal{S}$. Furthermore, if $\theta$ satisfies $\|\theta\|_2 \leq B$ we say that $\pi$ is $B$-boundedly $v$-linearly realizable with misspecification $\eta$ under $\varphi$.*

In what follows we will be concerned with designing a planning algorithm that, given local access to a feature map, competes with the best $v$-linearly realizable memoryless deterministic (MLD) policy under that feature map (if one exists) in the following sense: For $B > 0$ and $\eta \geq 0$, define the function $v^\circ_{B,\eta} : \mathcal{S} \to \mathbb{R}$ as

$$v^\circ_{B,\eta}(s) = \sup\left\{v_1^\pi(s) \;:\pi \text{ is MLD and is } B\text{-boundedly } v\text{-linearly realizable with misspecification } \eta \text{ given } \varphi\right\}. \tag{48}$$

We call will $v^\circ_{B,\eta}$ the ***φ*-compatible optimal value function** *at scale $B$ and misspecification $\eta$*. Note that if there are no $v$-linearly realizable policies with misspecification $\eta$ in an MDP, $v^\circ_{B,\eta}(s) \equiv -\infty$ for each state $s \in \mathcal{S}$ of the MDP. Competing with the best $v$-linearly realizable MLD policy (at scale $B > 0$ and misspecification $\eta \geq 0$) means the ability to generate actions of a policy whose value function is close to $v^\circ_{B,\eta}$ (for the fully formal definition, see the next section). Note that if the optimal value function of an MDP is linearly realizable with parameter vector $\theta^\star$ and misspecification $\eta'$ then for any $B \geq \|\theta^\star\|_2$, $v^\circ_{B,\eta} = v^\star$ for any $\eta \geq \eta'$. Hence, the setting we introduce generalizes the one where the optimal value function is exactly or near-realizable with $B$-bounded parameter vectors.

### 3.2.2. Local Planning

In the ***fixed-horizon local planning*** problem, a ***planner*** is given an input state and is tasked with computing a near-optimal action for that state while interacting with a black-box that simulates the MDP. In the ***(state-)featurized local planning*** problem, the black-box also returns the feature-vector of the next state. Access to the black-box is provided by means of calling a function SIMULATE, whose semantics is essentially as just described, but will be further elaborated on below.

More formally, the planner needs to "implement" a function, which we call `GetAction` and whose semantics, in the context of fixed-horizon MDPs, is as follows:

**Definition 3.2.2** (`GetAction`$(d, A, H, \text{SIMULATE}, s, h, \varphi_h(s), \delta, B)$)**.** *The meaning of inputs is as follows: $d$ is the dimension of the underlying feature map, $A$ is the number of actions, $H$ is the episode length,* SIMULATE *is a function that provides access to the oracle that simulates the MDP, $s$ is the state where an action is needed at stage $h \in [H]$, $\delta > 0$ is a suboptimality target, and $B$ is the parameter vector bound. This function needs to return an action in $\mathcal{A}$ with the intent that this is a "good action" to be used at stage $h$ when the state is $s$.*

Given a featurized MDP and a planner as described above, the planner ***induces a (randomized, possibly memoryful) policy***, which is the policy that results from calling `GetAction` along a trajectory and following its recommended actions. If the initial state is $S_1 = s_0 \in \mathcal{S}$, the first action taken by this policy is $A_1 =$ `GetAction`$(\ldots, S_1, 1, \ldots)$, the second is $A_2 =$ `GetAction`$(\ldots, S_2, 2, \ldots)$ where $S_2 \sim P_{S_1,A_1}$, etc. If `GetAction` does not save data between the calls, the resulting policy would be memoryless, but this is not a requirement. *In fact, we require that* `GetAction` *is first called with $h = 1$ and then $h = 2$, etc.* A practical planner which is used across multiple episodes can also save data between episodes. In this case `GetAction` can be called with $h = 1$ after being called with $h = H$, designating the start of a new episode. For now, we assume that this is not the case, as this allows for cleaner definitions.[10]

Inside `GetAction` the planner can issue any number of calls to SIMULATE. The function SIMULATE takes as inputs a state-stage-action triplet $(s, h, a)$. In response, SIMULATE returns a triplet $(R, S', \varphi_{h+1}(S'))$ where $(R, S')$ is a "fresh" random draw from $Q(s, a)$. For generality the simulator is also allowed some inaccuracy, in the sense that it returns $(\text{clip}_{[0,1]}(R + \Lambda_{sa}), S', \varphi_{h+1}(S'))$ where $\Lambda_{sa} \in \mathbb{R}$ is a constant satisfying $|\Lambda_{sa}| \le \lambda$, for some $\lambda \ge 0$ that we call the simulator's accuracy, and $\text{clip}_{[0,1]}(x) = \max(0, \min(1, x))$ (ie. inaccurate rewards are clipped in $[0, 1]$). Neither $\Lambda_{sa}$ nor $\lambda$ are known to the planner. The planner can only access states that it is given access to either when `GetAction` is called, or returned by a call to SIMULATE. The same holds for the features of the states. We note that this is essentially the same setting as what is called *sampling with state revisiting* by Li et al. (2021).

The ***quality of a planner*** is, on one hand, assessed based on the quality of the policy that it induces and, on the other hand, by its ***worst-case (per-episode) query-cost***, which is defined as the largest total query-cost (ie. number of calls to SIMULATE made by `GetAction`) encountered while running the planner for the $H$ stages of an episode, starting at stage $h = 1$.

10. Jumping a bit ahead of ourselves, if we cared about long-run average per-state query-complexity, one could perhaps do better by allowing planners to save data between episodes.

**Definition 3.2.3** (Sound planner)**.** *Let $B, \delta > 0$, $\lambda, \eta \geq 0$, $H \geq 1$. A planner is $(\delta, B)$-**sound** with simulator accuracy $\lambda$ and misspecification $\eta$ if for any featurized $H$-horizon MDP $(\mathscr{M}, \varphi)$ with rewards bounded in $[0,1]$ and with 1-bounded feature maps (i.e. for all $h \in [H]$, $s' \in S$, $\|\varphi_h(s')\|_2 \leq 1$), the (random) $H$-horizon policy $\pi$ that the planner induces while interacting with the $\lambda$-accurate simulation oracle satisfies*

$$v_1^\pi(s) \geq v_1^\circ(s) - \delta \quad \text{for all } s \in \mathcal{S}, \tag{49}$$

*where $v^\pi$ is the $H$-horizon value function of $\pi$ in $\mathscr{M}$ and $v^\circ = v^\circ_{B,\eta}$ is the $H$-horizon $\varphi$-compatible optimal value function of $\mathscr{M}$ (cf. Equation (48)).*

**Further notations** For $v \in \mathbb{R}^d$, and $a \leq b \leq d$ positive integers, let $v_{a:b} \in \mathbb{R}^{b-a+1}$ be the vector corresponding to the entries with indices in $\{a, a+1, \ldots, b\}$, i.e., $(v_{a:b})_i = v_{a+i-1}$.

## 3.3. Efficient planning for the finite-horizon setting

In this section, we present TensorPlan (Algorithm 1) and prove its soundness (cf. Definition 3.2.3) and efficiency (Theorem 3.3.2). We start with a high-level description of the main ideas underlying the planner. Initially, we only prove soundness for exact realizability (ie. $\eta = 0$), which we later generalize in Theorem 3.3.4.

The planner belongs to the family of generate-and-test algorithms. To describe it, let $\mathscr{M} = (\mathcal{S}, \Sigma, \mathcal{A}, Q)$ denote the MDP that the planner interacts with and let $\varphi = (\varphi_h)_{h \in [H]}$ be the underlying feature map. Further, let $\Theta^\circ \subset \mathbb{R}^d$ be the set which collects the parameter vectors of the value functions of $B$-boundedly $v$-linearly realizable DML policies with misspecification $\eta = 0$ (Definition 3.2.1). That is, $\Theta^\circ$ is such that for any $\theta \in \Theta^\circ$, $\|\theta\|_2 \leq B$ and for some DML policy $\pi$ of $\mathscr{M}$,

$$v_h^\pi(s) = \langle \varphi_h(s), \theta \rangle \quad \text{for all } h \in [H] \text{ and } s \in \mathcal{S}. \tag{50}$$

Let $s_0$ be the state which the planner is called for. The algorithm will maintain a subset $\Theta$ of $\mathbb{R}^d$ such that, with high probability, $\Theta^\circ \subset \Theta$. The set is initialized to the $\ell^2$-ball of radius $B$, which obviously satisfies this constraint. Given the set $\Theta$ of admissible parameter vectors and $s_0 \in \mathcal{S}$, the planner finds the *optimistic* parameter vector $\theta^+ = \arg\max_\Theta \langle \varphi_1(s_0), \theta \rangle$ from the set $\Theta$. Let us write $v_h(s;\theta) \coloneqq \langle \varphi_h(s), \theta \rangle$. If $\theta \in \Theta^\circ$ then for any $h \in [H]$ and $s \in \mathcal{S}$, since the policies defining $\Theta^\circ$ are

deterministic, it follows that there exists an action $a \in \mathcal{A}$ such that

$$v_h(s;\theta) = r_{sa} + P_{sa} v_{h+1}(\cdot;\theta)\,. \tag{51}$$

For any $\theta$, let $\pi_\theta$ denote the policy which chooses the action satisfying the above equation when in state $s$ and stage $h$ (when there is no action that satisfies the consistency condition Eq. (51), the policy can choose any action).

To test whether $\theta^+ \in \Theta^\circ$, the algorithm aims to "roll out" $\pi_{\theta^+}$. By this, we mean that upon encountering a state $s$ in stage $h$ in such a rollout, the algorithm checks whether there is an action $a$ that satisfies Eq. (51). If such an action is found, it is sent to the simulator, which responds with the next state. If no such action is found, the test fails – this means that $\theta^+ \notin \Theta^\circ$. When this happens, the data corresponding to the transition where the test failed is used to refine the set of admissible parameter vectors and a new admissible set $\Theta'$ is established. Assuming that the test failed at stage $h^\star$ and state $s^\star$, this new set is

$$\Theta' = \{\theta \in \Theta : \exists a \in \mathcal{A} \text{ s.t. } Eq.\ (51) \text{ holds with } s = s^\star \text{ and } h = h^\star\}\,.$$

Then the testing of $\theta^+$ is abandoned, $\Theta$ is updated to $\Theta'$, and the process is repeated. Clearly, $\Theta^\circ \subset \Theta'$ still holds, so $\Theta^\circ \subset \Theta$ also holds after the update.

When a rollout continues up to the end of the episode without failure, the algorithm is given some evidence that $\theta^+ \in \Theta^\circ$, but this evidence is weak. This is because the states encountered in a rollout are random, and the trajectory generated may just happen to avoid the "tricky" states where the consistency test would fail. Luckily though, if the algorithm keeps testing with multiple rollouts and the tests do not fail for a sufficiently large (but not too large) number of such rollouts, this can be taken as evidence that $\pi_{\theta^+}$ is indeed a good policy in starting state $s_0$. It may happen that $\theta^+$ is still not in $\Theta^\circ$, but the value of $\pi_{\theta^+}$ cannot be low.

This is easy to see, if for the moment we add a further, (seemingly) stronger test. This test checks whether $v_1(s_0;\theta^+)$ correctly predicts the value of $\pi_{\theta^+}$ in state $s_0$. To this end, the test simply takes the average sum of rewards along the rollouts. If we detect that $v_1(s_0;\theta^+)$ is not sufficiently close to the measured average value, the test fails. If this strengthened test does not fail either then this is strong evidence that $v_1^{\pi_{\theta^+}}(s_0)$ is as high as $v_1(s_0;\theta^+)$. Now, since $\Theta^\circ \subset \Theta$ holds throughout the execution of the algorithm, $v_1(s_0;\theta^+) \geq \max_{\theta\in\Theta^\circ} v_1^{\pi_\theta}(s_0) = v_B^\circ(s)$ (since we pick $\theta^+$ optimistically),

and hence policy $\pi_{\theta^+}$ can successfully compete with the best $v$-linearly realizable policy in $\mathscr{M}$ under $\varphi$ and at scale $B$ (Eq. (48)).

To complete the description of the algorithm, there are three outstanding issues. The first is that due to the randomizing simulation oracle, for any given state $s \in \mathcal{S}$, one can only check whether Eq. (51) holds up to some fixed accuracy and only with high probability. Luckily, this does not cause any issues – when the tests fail, the parameters can be set so that $\Theta^\circ \subset \Theta$ is still maintained.

The second issue is whether the algorithm is efficient. (So far we have been concerned only with soundness.) This is addressed by "tensorizing" the consistency test. For $\psi : \mathcal{S} \to \mathbb{R}^d$, we let $P_{sa}\psi = \int \psi(s')P_{sa}(ds')$. Using Lemma 1.2.1 we then observe that the existence of an action such that Eq. (51) holds is equivalent to:

$$\begin{aligned} 0 &= \prod_{a\in\mathcal{A}} r_{sa} + \langle P_{sa}\varphi_{h+1} - \varphi_h(s), \theta\rangle = \prod_{a\in\mathcal{A}} \langle [r_{sa}, P_{sa}\varphi_{h+1} - \varphi_h(s)], [1,\theta]\rangle \\ &= \langle \otimes_{a\in\mathcal{A}}[r_{sa}, P_{sa}\varphi_{h+1} - \varphi_h(s)], \otimes_{a\in\mathcal{A}}[1,\theta]\rangle \,. \end{aligned}$$

Now, defining $M_\theta = \otimes_{a\in\mathcal{A}}[1,\theta]$ and $T_s = \otimes_{a\in\mathcal{A}}[r_{sa}, P_{sa}\varphi_{h+1} - \varphi_h(s)]$, we see that $\theta \in \Theta^\circ$ is equivalent to that $\langle T_s, M_\theta\rangle = 0$ holds for all $s \in \mathcal{S}$. Testing a parameter vector at some state is equivalent to checking whether $M_\theta$ is orthogonal to $T_s$. Clearly, the maximum number of tests that can fail before identifying an element of $\Theta^\circ$ is at most $d^A$, the dimension of $M_\theta$. Since our tests are noisy, we use an argument based on eluder dimensions (which allow imperfect measurements) to complete our efficiency proof (Russo and Van Roy, 2014).

The final issue is really an optimization opportunity. In our proposed algorithm we do not separately test if the value estimates at $s_0$ are close to the empirical return over the rollouts, and instead rely only on the consistency tests. This can be done since, when consistency holds, the expected total reward in an episode is close to the predicted value. This follows from a telescoping argument. Let $S_1 = s_0, A_1, S_2, A_2, \ldots, S_H, A_H, S_{H+1}$ be the state-action pairs in a rollout where the tests do not fail, and note that

$$v_1^{\pi_{\theta^+}}(s_0) = \mathbb{E}_{\pi_{\theta^+}}\left[\sum_{h=1}^{H} r_{S_h,A_h}\right] = \mathbb{E}_{\pi_{\theta^+}}\left[\sum_{h=1}^{H} v_h(S_h;\theta^+) - v_{h+1}(S_{h+1};\theta^+)\right] = v_1(s_0;\theta^+)\,,$$

where the first equality uses the definition of $v^{\pi_{\theta^+}}$, the second equality uses $r_{S_h,A_h} = v_h(S_h;\theta^+) - P_{S_hA_h}v_{h+1}(\cdot;\theta^+)$, and the last equality uses that $v_{H+1} \equiv 0$. When measurements are noisy, a similar

telescoping argument gives that with high probability, $v_1^{\pi_{\theta^+}}(s_0)$ is almost as high as $v_1(s_0;\theta^+)$ when consistency tests do not fail for a number of rollouts.

### 3.3.1. The TensorPlan algorithm

The pseudocode of `GetAction` of TensorPlan is shown in Algorithm 1.

**Algorithm 1** TensorPlan.`GetAction`

**Inputs:** $d, A, H, \text{SIMULATE}, s, h, \varphi_h(s), \delta, B$
**if** $h = 1$ **then** ▷ Initialize global $\theta^+$
    TensorPlan.`Init`$(d, A, H, \text{SIMULATE}, s, \varphi_1(s), \delta)$
**end if**
$\overline{\Delta}. \leftarrow \text{ApproxTD}(s, h, \varphi_h(s), A, n_2, \text{SIMULATE})$
Access $\theta^+$ saved by TensorPlan.`Init`
**return** $\arg\min_{a\in[A]} \left|\left\langle \overline{\Delta}_a, [1, \theta^+]\right\rangle\right|$

**Algorithm 2** ApproxTD

**Inputs:** $s, h, \varphi_h(s), A, n, \text{SIMULATE}$
**for** $a = 1$ to $A$ **do**
    **for** $l = 1$ to $n$ **do**
        $(R_l, S'_l, \varphi_{h+1}(S'_l)) \leftarrow (\text{SIMULATE}(s, h, a))$
        $\tilde{\Delta}_l \leftarrow [R_l, \varphi_{h+1}(S'_l) - \varphi_h(s)]$
    **end for**
    $\Delta_a := \frac{1}{n}\sum_{l\in[n]} \tilde{\Delta}_l$
**end for**
**return** $(\Delta_a)_{a\in[A]}$

The main workhorse of TensorPlan is the initialization routine, TensorPlan.`Init` (Algorithm 3), which generates a global variable $\theta^+ \in \mathbb{R}^d$ that is an estimate for the parameter of the best realizable value function $v_B^\circ$. Within an episode, this parameter is used by the current and subsequent calls to `GetAction`. In particular, given $\theta^+$, `GetAction` approximately implements $\pi_{\theta^+}$ of the previous section. For this, `GetAction` calls ApproxTD[11] (Algorithm 2), which produces an estimate of $[r_{sa}, P_{sa}\varphi_{h+1} - \varphi_h(s)]$ for all actions $a \in \mathcal{A}$.

The `Init` function uses

$$\text{Sol}(\Delta_1, \ldots, \Delta_\tau) = \left\{\theta \in \mathbb{R}^d \,:\, \|\theta\|_2 \le B, \forall i \in [\tau] \,:\, \left|\left\langle \Delta_i, \otimes_{a\in[A]}[1, \theta]\right\rangle\right| \le \frac{H^A \varepsilon}{2\sqrt{E_d}}\right\}. \tag{52}$$

where $\varepsilon$ is a function of the target suboptimality and $E_d = \tilde{\mathcal{O}}\left(d^A A\right)$, defined in Eq. (58), is an upper bound on the the eluder dimension of a tensorized clipped-linear function class (cf. Eq. (59)). The $\text{Sol}(\cdot)$ set stands for the successfully refined sets $\Theta$ of the previous section and its arguments $\Delta_i \in \mathbb{R}^{(d+1)^A}$ correspond to estimates of $\otimes_{a\in\mathcal{A}}[r_{sa}, P_{sa}\varphi_{h+1} - \varphi_h(s)]$ for the various states $s$ and stages $h$ where the algorithm detects a failure of the consistency test it runs. Estimates of these in `Init` are obtained by calls to ApproxTD.

11. Thusly named since $\langle [r_{sa}, P_{sa}\varphi_{h+1} - \varphi_h(s)], [1, \theta]\rangle$ corresponds to the "temporal difference" error of value function $v_\theta$ at state-action pair $(s, a)$ (Sutton, 1988).

**Algorithm 3** TensorPlan.`Init`

1: **Inputs:** $d, A, H, \text{SIMULATE}, s_0, \varphi_1(s_0), \delta$
2: $X \leftarrow \{\}$ ▷ $X$ is a list
3: Initialize $\zeta, \varepsilon, n_1, n_2, n_3$ via equations (53), (54), (55), (56), (57), respectively.
4: **for** $\tau = 1$ to $E_d + 2$ **do**
5: Choose any $\theta_\tau \in \arg\max_{\theta \in \text{Sol}(X)} \langle \varphi_1(s_0), \theta \rangle$ ▷ Optimistic choice
6: `CleanTest` $\leftarrow$ `true`
7: **for** $t = 1$ to $n_1$ **do** ▷ $n_1$ rollouts with $\theta_\tau$-induced policy
8: $S_{\tau t 1} = s_0$ ▷ Initialize rollout
9: **for** $j = 1$ to $H$ **do** ▷ Stages in episode
10: $\overline{\Delta}_{\tau t j, \cdot} \leftarrow \text{ApproxTD}(S_{\tau t j}, j, \varphi_j(S_{\tau t j}), A, n_2, \text{SIMULATE})$
11: **if** `CleanTest` and $\min_{a \in [A]} \left| \left\langle \overline{\Delta}_{\tau t j a}, [1, \theta_\tau] \right\rangle \right| > \frac{\delta}{4H}$ **then** ▷ Consistency failure?
12: $\hat{\Delta}_{\tau t j, \cdot} \leftarrow \text{ApproxTD}(S_{\tau t j}, j, \varphi_j(S_{\tau t j}), A, n_3, \text{SIMULATE})$ ▷ Refined data
13: $X.\text{append}\left( \otimes_{a \in [A]} \hat{\Delta}_{\tau t j a} \right)$ ▷ Save failure data
14: `CleanTest` $\leftarrow$ `false` ▷ Not clean anymore
15: **end if**
16: $A_{\tau t j} \leftarrow \arg\min_{a \in [A]} \left| \left\langle \overline{\Delta}_{\tau t j a}, [1, \theta_\tau] \right\rangle \right|$ ▷ Find most consistent action
17: $(R_{\tau t j}, S_{\tau t j+1}, \varphi_{j+1}(S_{\tau t j+1})) \leftarrow \text{SIMULATE}(S_{\tau t j}, j, A_{\tau t j})$ ▷ Roll forward
18: **end for**
19: **end for**
20: **if** `CleanTest` **then break** ▷ Success?
21: **end for**
22: Save into global memory $\theta^+ \leftarrow \theta_\tau$

Note that `Init` as described continues to generate rollout data even after a consistency test fails. This is clearly superfluous and in an optimized implementation one could break out of the test loop to generate the next candidate immediately after a failure happens. The only reason the algorithm is described in the way it is done here is because this allows for a cleaner analysis: every policy will have access to data from $n_1$ rollouts, even if the policy fails a consistency test.

**Remark 3.3.1.** *The reader might wonder why* TensorPlan *follows the most consistent action in Line 7 of* `GetAction`*, instead of the best action according to its* $\theta^+$*, which would be* $\arg\max_{a \in [A]} \left\langle \overline{\Delta}_a, [1, \theta^+] \right\rangle$*. Indeed, a practical implementation might adopt this, together with the same change to Line 16 of* `Init`*, and a strengthening of the consistency test of* `Init`*'s Line 11 to require that* the best *action (according to* $\theta_\tau$*) be consistent, instead of* any *action. This test would fail if* $\left| \max_{a \in [A]} \left\langle \overline{\Delta}_{\tau t j a}, [1, \theta_\tau] \right\rangle \right| > \frac{\delta}{4H}$*. One might hope that this strengthened consistency test improves sample efficiency, and indeed the proofs go through (giving the same query complexity bounds), albeit with a significant weakening of the final guarantee: this version of* TensorPlan *could* only *compete with optimal policies that are realizable, instead of the best of* all *realizable*

*DML policies.* TensorPlan, *as presented, is able to compete with the latter, with its only source of pressure to do well coming from the optimistic choice of* $\theta_\tau$ *in Line 5 of* `Init`.

The following theorem gives a query complexity guarantee on using TensorPlan to find a near-optimal policy. The precise values of $\zeta, \varepsilon, n_1, n_2$, and $n_3$ mentioned in the theorem can be found in Section 3.A. For the theorem statement recall that $B$ is the bound on the 2-norm of value-function parameter vectors that the algorithm competes with.

**Theorem 3.3.2.** *[Weisz et al., 2021a, Theorem 4.2] For any* $\delta > 0$ *and* $B > 0$*, there exists values of* $\zeta, \varepsilon, n_1, n_2$*, and* $n_3$ *such that the* TensorPlan *algorithm (Algorithm 1) is* $(\delta, B)$*-sound (Definition 3.2.3) with misspecification* $\eta = 0$ *and simulator accuracy* $\lambda \leq \varepsilon/(4\sqrt{E_d}) = \tilde{\mathcal{O}}\left(\left(\frac{\delta}{12\sqrt{d}H^2}\right)^A / \sqrt{A}\right)$ *for the* $H$*-horizon planning problem with worst-case per-episode query-cost*

$$\tilde{\mathcal{O}}\left(d^A A^4 B^2/\delta^2 \left(H^5 B^2 d/\delta^2 + d^A A H^{4(A+1)} 12^{2A}/\delta^{2A}\right)\right) = \text{poly}\left((dH/\delta)^A, B\right).$$

**Corollary 3.3.3** (Weisz et al., 2021a, Corollary 4.3)**.** *When the optimal value function* $v^\star$ *is linearly realizable with the given feature map with misspecification* $\eta = 0$*, then* TensorPlan*, given access to a simulator with accuracy* $\lambda \leq \varepsilon/(4\sqrt{E_d})$ *induces a policy* $\pi$ *within the budget constraints of Theorem 3.3.2 for which* $v_1^\pi(s_0) \geq v_1^\star(s_0) - \delta$.

*Proof (of Theorem 3.3.2).* We provide here a very brief sketch, and defer the full proof to Appendix 3.A. The proof proceeds in a few steps. First, fix any starting state $s_0 \in \mathcal{S}$ and any $\theta^\circ \in \Theta^\circ$. Section 3.A.1 establishes that despite the simulator's inaccuracy, the estimates $\hat{\Delta}$ and $\overline{\Delta}$ are close to their respective expected values (Lemma 3.A.1) and that $\left\langle \overline{\Delta}, \theta^\circ \right\rangle$ is close to its expected value (Lemma 3.A.2). This entails that $\theta^\circ$ does not get eliminated from the solution set (Lemma 3.A.3). In Section 3.A.2, we use the eluder dimension to bound the maximal length of $X$ (essentially, the list of states where consistency is broken). It follows that, with high probability, the iteration over $\tau$ will be exited in Line 20 with `CleanTest` being true for $\tau \leq E_d + 1$. The last subsection (Section 3.A.3) bounds the suboptimality of the policy induced by $\theta^+$ in terms of the inner product between $\theta^+$ and the measured TD vectors (Lemma 3.A.6). We then bound these suboptimalities by the desired sub-optimality (Corollary 3.A.7) and finally establish in Corollary 3.A.8 that the policy induced by the planner is $\delta$-optimal compared to $v_1(s_0; \theta^\circ)$. Since this argument holds for any $s_0 \in \mathcal{S}$ and $\theta^\circ \in \Theta^\circ$, the planner is $(\delta, B)$-sound according to Definition 3.2.3. ■

Our next theorem generalizes the previous results to the misspecified case (ie. $\eta > 0$) by trading off simulator accuracy for misspecification. Formally, we provide a reduction to the realizable

case and run TensorPlan with a slightly modified simulation oracle SIMULATE′ which requires no additional information beyond that provided by the original simulator. The proof is deferred to Section 3.B. The main idea of the proof is to define an alternate MDP with an expanded state space where states are indexed by which stage they belong to so that the misspecification error of a target policy can be "pushed" into the rewards of the new MDP. This way, the target policy will not have misspecification errors. The simulator for the new MDP still reports the rewards from the original MDP, but this is allowed since the previous result was stated for the case when the simulator introduces (small) errors when reporting the rewards.

**Theorem 3.3.4.** *[Weisz et al., 2021a, Theorem 4.4] For any $\delta, B > 0$, TensorPlan is $(\delta, B)$-sound with misspecification $\eta \le \varepsilon/(12\sqrt{E_d})$ and simulator accuracy $\lambda \le \varepsilon/(12\sqrt{E_d})$ with worst-case per-episode query-cost* $\mathrm{poly}\left((dH/\delta)^A, B\right)$*, when run with input $\delta' = 0.98\delta$ and simulation oracle* SIMULATE′.

### 3.3.2. Discounted MDPs

In the discounted MDP setting, instead of maximizing the expected value of the reward $\sum_{h\in[H]} R_h$ over a horizon $H$, the goal of the agent is to maximize the expected value of the discounted total reward, $\sum_{h\in\mathbb{N}_+} \gamma^{h-1} R_h$, over an infinite horizon, where $0 \le \gamma < 1$ is a fixed discount factor, given to the agent. The value function for a policy $\pi$, $v^\pi : \mathcal{S} \to \mathbb{R}$ is defined as $v^\pi(s) = r_\pi(s) + \gamma P_{sa} v^\pi$. The stage index $h$ is dropped from the feature mapping ($\varphi : \mathcal{S} \to \mathbb{R}^d$), and the definition of v-linearly realizable policies (Definition 3.2.1) changes from requiring $\left|v_h^\pi(s) - \langle \varphi_h(s), \theta\rangle\right| \le \eta$ to requiring

$$|v^\pi(s) - \langle \varphi(s), \theta\rangle| \le \eta \qquad \text{for all } s \in \mathcal{S}\,.$$

Soundness is otherwise defined identically to the $H$-horizon case, except for swapping the value function to $v^\pi$. Importantly, value guarantees are only required for the initial state the planner is called with, and not for every state that the planner ever encounters. As the episodes are infinitely long in this setting, we use the per-state (instead of per-episode) query-cost.

We use a reduction of the discounted case to the finite-horizon case with an "effective horizon" $H_{\gamma,\delta}$. Our next theorem shows that the guarantees of TensorPlan in the $H_{\gamma,\delta}$-horizon setting transfer to the discounted setting if it is run with a slightly modified simulation oracle $\textsc{Simulate}^{\gamma,\delta}$, which once again does not require any additional information beyond that of the original simulation oracle. As this is a reduction, the input $h$ given to TensorPlan's `GetAction` should be incremented

for each transition, exactly as in the finite-horizon case. The definition of $H_{\gamma,\delta}$ and $\textsc{Simulate}^{\gamma,\delta}$, as well as the proof can be found in Section 3.C.

**Theorem 3.3.5.** *[Weisz et al., 2021a, Theorem 4.5] For any $\delta, B > 0$, TensorPlan is $(\delta, B)$-sound for discounted MDPs with discount factor $0 \le \gamma < 1$, with misspecification $\eta \le \varepsilon/(24\sqrt{E_d})$ and simulator accuracy $\lambda \le \varepsilon/(12\sqrt{E_d})$, with worst-case per-state query-cost* $\text{poly}\left(\left(dH_{\gamma,\delta}/\delta\right)^A, B\right)$, *when run with input $\delta' = 0.98\delta$ and simulation oracle $\textsc{Simulate}^{\gamma,\delta}$.*

## 3.4. Conclusions and discussion

We presented TensorPlan, a provably efficient algorithm for local planning in finite-horizon MDPs which only requires linear realizability of $v^\star$. When the action set is small (i.e. $\mathcal{O}(1)$), TensorPlan is the first algorithm that enjoys polynomial query complexity without further assumptions. Our results are also complemented by an extension of the positive result to the near-realizable as well as the discounted setting.

In contrast to ADP-type algorithms (Schweitzer and Seidmann, 1985), our algorithm does not use value fitting. In fact, without stronger assumptions such as a core set, ADP algorithms appear to be susceptible to an exponential blow-up of errors (Tsitsiklis and Van Roy, 1996; Dann et al., 2018; Zanette et al., 2019; Wang et al., 2020a; Weisz et al., 2021b). For the same reason, our algorithm works with a weaker simulation oracle that provides access only to states that have been encountered previously. Learning via local consistency ("bootstrapping") also allows us to provide a more agnostic guarantee, which automatically matches the best realizable value function.

# Appendix

## 3.A. Proof of Theorem 3.3.2

To prove that TensorPlan (Algorithm 1) is $(\delta, B)$-sound (Definition 3.2.3) for the $H$-dimensional planning problem, we fix $\delta > 0$, $B > 0$, $H > 1$, a featurized MDP $(\mathscr{M}, \varphi)$ with 1-bounded feature maps, a suboptimality target $0 < \delta < H$, and a (starting) state state $s_0 \in \mathcal{S}$. We assume that $\delta < H$ as otherwise, for $\delta \geq H$, Eq. (49) trivially holds due to the rewards being bounded in $[0,1]$ (and therefore the values in $[0,H]$).

The precise values of hyperparameters used in TensorPlan will be set to:

$$\zeta = \frac{1}{4H}\delta \tag{53}$$

$$\varepsilon = \left(\frac{\delta}{12H^2}\right)^A \Big/ \left(1 + \frac{1}{2\sqrt{E_d}}\right) \tag{54}$$

$$n_1 = \left\lceil \frac{32H^2(1+2B)^2}{\delta^2} \log \frac{E_d+1}{\zeta} \right\rceil \tag{55}$$

$$n_2 = \left\lceil \frac{1867H^2(B+1)^2(d+1)}{2\delta^2} \log(4(E_d+1)n_1HA(d+1)/\zeta) \right\rceil \tag{56}$$

$$n_3 = \left\lceil \max\left\{ n_2, \frac{32(H+1)^2E_d}{\varepsilon^2} \log((2(E_d+1)n_1HA))/\zeta \right\} \right\rceil \tag{57}$$

We assume $H > 1$ for simplicity of presentation, as for $H = 1$ the same analysis will apply, replacing $H$ with $H+1$ in the above display for $\varepsilon$.

Denote by $\tau^+$ the final value of $\tau$ at the end of TensorPlan.`Init`. For the proof let $\mathbb{P}$ denote the probability distribution induced by the interconnection of TensorPlan with the MDP when the initial state of the episode is $s_0$ and the planner is used for the $H$ steps. In particular, $\mathbb{P}$ is defined over some measurable space $(\Omega, \mathbb{P})$ that carries the random variables $S_1$, $A_1$, $S_2$, $A_2$, …, $A_H$, $S_{H+1}$, where $S_1 = s_0$, $S_i \sim P_{A_{i-1}}(S_{i-1})$ for $i > 1$, and for $j \in [H]$, $A_j$ is the action returned by `GetAction` when called with $S_j$ and $h = j$. $(\Omega, \mathbb{P})$ also carries the random variables $\hat{\Delta}$, $\bar{\Delta}$, $\tilde{\Delta}$, and

$(S_{\tau tj}, A_{\tau tj})_{\tau \le E_d+2, t\in[n_1], j\in[H]}$ of the TensorPlan algorithm. For the latter, assume for now that TensorPlan.`Init` does not break out from the loop over $\tau$ when the test fails, but that it keeps running, so that we can refer to $(S_{\tau tj}, A_{\tau tj})$ even for $\tau > \tau^+$. Note that all other quantities that appear in TensorPlan can be written as a function of these. We denote the expectation operator underlying $\mathbb{P}$ by $\mathbb{E}$.

### 3.A.1. Concentration bounds

This section establishes concentration bounds on the estimated difference vectors $\hat{\Delta}$ and $\overline{\Delta}$, and then establishes that the true parameter is unlikely to be eliminated from the solution set.

**Lemma 3.A.1.** *If the simulator's accuracy $\lambda \le \frac{\varepsilon}{4\sqrt{E_d}}$, then with $n_2$ samples for $\overline{\Delta}$ and $n_3$ samples for $\hat{\Delta}$, with probability greater than $1-\zeta$, for all $\theta \in \mathbb{R}^d$ with $\|\theta\|_2 \le B$, for all $\tau \in [E_d+1]$, $t \in [n_1]$, $j \in [H]$ and action $a \in [A]$, $\overline{\Delta}_{\tau tja}$ and $\hat{\Delta}_{\tau tja}$ satisfy*

$$\left|\left\langle \overline{\Delta}_{\tau tja} - \Delta(S_{\tau tj}, a), [1, \theta] \right\rangle\right| \le \delta/(12H) \quad \text{and} \quad \left|\left\langle \hat{\Delta}_{\tau tja} - \Delta(S_{\tau tj}, a), [1, \theta] \right\rangle\right| \le \delta/(12H),$$

*where $\Delta(S_{\tau tj}, a) = [r_{S_{\tau tj}, a}, P_{S_{\tau tj}a}\varphi_{j+1} - \varphi_j(S_{\tau tj})]$.*

*Proof.* We show this for $\overline{\Delta}_{\tau tja}$, i,e., that the first inequality holds with probability at least $1-\zeta/2$. As $n_3 \ge n_2$, by a similar argument this statement holds for $\hat{\Delta}_{\tau tja}$ too, and a union bound on the failure probability finishes the proof. Let us refer here to the measurements $\tilde{\Delta}_l$ done by ApproxTD called in Line 10 in Algorithm 3 as $(\tilde{\Delta}_{\tau tjal})_{l\in[n_2]}$. By the bounded rewards (the simulator's rewards are clipped in $[0,1]$ despite its inaccuracy), triangle inequality, and the assumption that $\forall h \in [H+1], s \in \mathcal{S}$, $\|\varphi_h(s)\|_2 \le 1$, we have that $\left\|\tilde{\Delta}_{\tau tjal}\right\|_\infty \le \left\|\tilde{\Delta}_{\tau tjal}\right\|_2 \le 3$.

Since $\overline{\Delta}_{\tau tja}$ is the average of $n_2$ independent identically distributed bounded samples of the distribution of $\tilde{\Delta}_{\tau tjal}$, which has expectation $\Delta'(S_{\tau tj}, a) = [\mathrm{clip}_{[0,1]}(r_{S_{\tau tj},a} + \Lambda_{S_{\tau tj},a}), P_{S_{\tau tj}a}\varphi_{j+1} - \varphi_j(S_{\tau tj})]$, we can apply Hoeffding's inequality for each component $i \in [d+1]$ of the vector:

$$\mathbb{P}\left(\left|\left(\overline{\Delta}_{\tau tja} - \Delta'(S_{\tau tj}, a)\right)_i\right| > \delta/\left(\frac{72}{5}H(B+1)\sqrt{d+1}\right)\right) \le 2\exp\left(-\frac{2n_2\delta^2}{\left(\frac{72}{5}\right)^2 H^2(B+1)^2(d+1)3^2}\right)$$

Setting $n_2 = \left\lceil \frac{1867H^2(B+1)^2(d+1)}{2\delta^2} \log(4(E_d+1)n_1HA(d+1)/\zeta) \right\rceil$ allows this probability to be bounded by $\zeta/(2(E_d+1)n_1HA(d+1))$. A union bound over $\tau \in [E_d+1]$, $t \in [n_1]$, $j \in [H]$, $a \in [A]$, and $i \in [d+1]$ achieves the $\zeta/2$ failure probability bound. Under this high-probability event we have that $\left\|\overline{\Delta}_{\tau tja} - \Delta'(S_{\tau tj}, a)\right\|_\infty \le \delta/\left(\frac{72}{5}H(B+1)\sqrt{d+1}\right)$, so $\left|\left\langle \overline{\Delta}_{\tau tja} - \Delta'(S_{\tau tj}, a), [1, \theta] \right\rangle\right| \le$

$\left\|\overline{\Delta}_{\tau tja} - \Delta'(S_{\tau tj}, a)\right\|_\infty \|[1,\theta]\|_1 \le \left\|\overline{\Delta}_{\tau tja} - \Delta'(S_{\tau tj}, a)\right\|_\infty \|[1,\theta]\|_2 \sqrt{d+1} \le \delta/(\frac{72}{5}H)$. By the triangle inequality:

$$\left|\left\langle \overline{\Delta}_{\tau tja} - \Delta(S_{\tau tj}, a), [1,\theta]\right\rangle\right| \le \left|\left\langle \overline{\Delta}_{\tau tja} - \Delta'(S_{\tau tj}, a), [1,\theta]\right\rangle\right| + \lambda \le \delta/H\left(\frac{5}{72} + \frac{1}{72}\right) = \delta/(12H),$$

as $\lambda \le \frac{\varepsilon}{4\sqrt{E_d}} \le \delta/(12H)/4/(1+\frac{1}{2})$. ■

**Lemma 3.A.2.** *If the simulator's accuracy $\lambda \le \frac{\varepsilon}{4\sqrt{E_d}}$, then with $n_3$ samples for $\hat{\Delta}$, with probability at least $1-\zeta$, for all $\tau \in [E_d+1]$, $t \in [n_1]$, $j \in [H]$ and action $a \in [A]$,*

$$\left|\left\langle \hat{\Delta}_{\tau tja} - \Delta(S_{\tau tj}, a), [1,\theta^\circ]\right\rangle\right| \le \frac{\varepsilon}{2\sqrt{E_d}}$$

*where $\Delta(S_{\tau tj}, a) = [r_{S_{\tau tj},a}, P_{S_{\tau tja}}\varphi_{j+1} - \varphi_j(S_{\tau tj})]$.*

*Proof.* Let us refer here to the measurements $\tilde{\Delta}_l$ done by ApproxTD called in Line 12 in Algorithm 3 as $(\tilde{\Delta}_{\tau tjal})_{l\in[n_3]}$. Since $\theta^\circ \in \Theta^\circ$, $\theta^\circ$ satisfies Eq. (50) for some policy. Furthermore, due to the bounded rewards, horizon $H$, and the simulator's clipping of rewards into $[0,1]$ (despite its inaccuracy), and the bounded values (of any state for any policy) in $[0,H]$, we have that $\left\langle \tilde{\Delta}_{\tau tjal}, [1,\theta^\circ]\right\rangle \in [-(H+1),(H+1)]$. Since $\hat{\Delta}_{\tau tja}$ is the average of $n_3$ independent identically distributed bounded samples of the distribution of $\tilde{\Delta}_{\tau tjal}$, which has expectation $\Delta'(S_{\tau tj}, a) = [\mathrm{clip}_{[0,1]}(r_{S_{\tau tj},a} + \Lambda_{S_{\tau tj},a}), P_{S_{\tau tja}}\varphi_{j+1} - \varphi_j(S_{\tau tj})]$, we can apply Hoeffding's inequality:

$$\mathbb{P}\left(\left|\left\langle \hat{\Delta}_{\tau tja}, [1,\theta^\circ]\right\rangle - \left\langle \Delta'(S_{\tau tj}, a), [1,\theta^\circ]\right\rangle\right| > \frac{\varepsilon}{4\sqrt{E_d}}\right) \le 2\exp\left(-\frac{n_3\varepsilon^2}{32(H+1)^2 E_d}\right).$$

Setting $n_3 = \left\lceil \max\left\{n_2, \frac{32(H+1)^2 E_d}{\varepsilon^2}\log((2(E_d+1)n_1HA))/\zeta\right\}\right\rceil$ allows this probability to be bounded by $\zeta/((E_d+1)n_1HA)$. By the triangle inequality, under the high-probability event, the desired bound with $\Delta$ instead of $\Delta'$ is guaranteed as:

$$\left|\left\langle \hat{\Delta}_{\tau tja}, [1,\theta^\circ]\right\rangle - \left\langle \Delta(S_{\tau tj}, a), [1,\theta^\circ]\right\rangle\right| \le \left|\left\langle \hat{\Delta}_{\tau tja}, [1,\theta^\circ]\right\rangle - \left\langle \Delta'(S_{\tau tj}, a), [1,\theta^\circ]\right\rangle\right| + |\Lambda_{S_{\tau tj},a}| \le 2\frac{\varepsilon}{4\sqrt{E_d}}$$

A union bound over $\tau \in [E_d+1]$, $t \in [n_1]$, $j \in [H]$, and $a \in [A]$ achieves the desired probability bound. ■

**Lemma 3.A.3** ($\theta^\circ \in \mathrm{Sol}(X)$)**.** *For $\tau \in [E_d+1]$, let $X_{\le\tau}$ denote the first $\tau$ elements of $X$, where $X$ is defined in Line 2 of Algorithm 3. Then, with probability at least $1-\zeta$ we have that $\forall\tau \in [E_d+1]$, $\theta^\circ \in \mathrm{Sol}(X_{\le\tau})$.*

*Proof.* As in Lemma 3.A.2, by MDP reward boundedness, $\left|\left\langle \Delta(S_{\tau tj}, a), [1, \theta^\circ]\right\rangle\right| \le H$ for any $\Delta(S_{\tau tj}, a)$. Let $A^\circ_{\tau tj}$ be the action satisfying Eq. (51) for $\theta^\circ$ in state $S_{\tau tj}$. Then we have that $\left\langle \Delta(S_{\tau tj}, A^\circ_{\tau tj}), [1, \theta^\circ]\right\rangle = 0$. Thus, using Lemma 3.A.2, with probability at least $1-\zeta$, for all $\tau \in [E_d+1]$, $t \in [n_1]$, $j \in [H]$, $a \in [A]$,

$$\begin{aligned}\left|\left\langle \hat{\Delta}_{\tau tja}, [1, \theta^\circ]\right\rangle\right| &= \left|\left\langle \Delta(S_{\tau tj}, a), [1, \theta^\circ]\right\rangle + \left\langle \hat{\Delta}_{\tau tja} - \Delta(S_{\tau tj}, a), [1, \theta^\circ]\right\rangle\right| \\ &\le \left|\left\langle \Delta(S_{\tau tj}, a), [1, \theta^\circ]\right\rangle\right| + \left|\left\langle \hat{\Delta}_{\tau tja} - \Delta(S_{\tau tj}, a), [1, \theta^\circ]\right\rangle\right| \\ &\le \mathbb{I}\{a \ne A^\circ_{\tau tj}\} H + \frac{\varepsilon}{2\sqrt{E_d}},\end{aligned}$$

where $\mathbb{I}\{S\}$ is the indicator of a set $S$. We can then bound the product across $a \in [A]$ as

$$\prod_{a\in[A]} \left\langle \hat{\Delta}_{\tau tja}, [1, \theta^\circ]\right\rangle \le \left(H + \frac{\varepsilon}{2\sqrt{E_d}}\right)^{A-1} \frac{\varepsilon}{2\sqrt{E_d}} = \left(1 + \frac{\varepsilon}{2\sqrt{E_d}H}\right)^{A-1} H^{A-1} \frac{\varepsilon}{2\sqrt{E_d}},$$

and

$$\begin{aligned}\left(1 + \frac{\varepsilon}{2\sqrt{E_d}H}\right)^{A-1} &\le 1 + (2^{A-1} - 1)\frac{\varepsilon}{2\sqrt{E_d}H} < 1 + 2^A\varepsilon \\ &< 1 + 2^A \frac{\delta^A}{(12H^2)^A} = 1 + \left(\frac{2\delta}{12H^2}\right)^A < 2 \le H,\end{aligned}$$

so $\prod_{a\in[A]} \left\langle \hat{\Delta}_{\tau tja}, [1, \theta^\circ]\right\rangle < H^A \frac{\varepsilon}{2\sqrt{E_d}}$. Let $\tau \in [E_d+1]$. The $\tau^{\text{th}}$ element added to $X$ will be $\otimes_{a\in[A]} \hat{\Delta}_{\tau tja}$ computed in Line 12 of Algorithm 3 for some $\tau \in [E_d+1]$, $t \in [n_1]$, $j \in [H]$, so $\theta^\circ \in \mathrm{Sol}(X_{\le\tau})$ according to Eq. (52). ∎

### 3.A.2. Eluder dimension

This subsection uses the eluder dimension to bound the maximal number of iterations. For $\Theta \in \mathbb{R}^{(d+1)^A}$ and $x \in \mathbb{R}^{(d+1)^A}$, let

$$f_\Theta(x) = \langle \mathrm{truncate}(x), \Theta\rangle,$$

where $\mathrm{truncate}(x) = \frac{x}{\|x\|_2} \min\{\|x\|_2, 3^A\}$. Notice the similarity between these functions and the form of the constraints we use in Eq. (52) to define the set of parameter vectors $\mathrm{Sol}(\cdot)$ consistent with our

observations. Let

$$\mathcal{F}^+ = \{f_\Theta : \Theta \in \mathbb{R}^{(d+1)^A}, \|\Theta\|_2 \le (B+1)^A\}$$

and

$$E_d = \left\lfloor 3(d+1)^A \frac{e}{e-1} \ln\left\{3 + 3\left(\frac{2(B+1)^A 3^A}{H^A\varepsilon}\right)^2\right\} + 1 \right\rfloor = \tilde{\mathcal{O}}\left(d^A A\right). \tag{58}$$

By Russo and Van Roy (2014), $\dim_E(\mathcal{F}^+, H^A\varepsilon)$, the **eluder dimension** of $\mathcal{F}^+$ at scale $H^A\varepsilon$ is the length $\tau$ of the longest **eluder sequence** $x_1, \dots, x_\tau$, such that for some $\varepsilon' \ge H^A\varepsilon$, for each $l \in [\tau]$,

$$w_l := \sup\left\{|f_1(x_l) - f_2(x_l)| : \sqrt{\sum_{i=1}^{l-1}(f_1(x_i) - f_2(x_i))^2} \le \varepsilon',\ f_1, f_2 \in \mathcal{F}^+\right\} > \varepsilon'.$$

Also by Russo and Van Roy (2014) (Appendix C.2), $\dim_E(\mathcal{F}^+, H^A\varepsilon) \le E_d$. Now let

$$\mathcal{F} = \{f_\Theta : \exists\theta \in \mathbb{R}^d.\ \|\theta\|_2 \le B, \Theta = \flat(\otimes_{a\in[A]}[1,\theta])\}. \tag{59}$$

Since $\|\theta\|_2 \le B$ implies $\left\|\flat(\otimes_{a\in[A]}[1,\theta])\right\|_2 \le (B+1)^A$, $\mathcal{F} \subseteq \mathcal{F}^+$, and so $\dim_E(\mathcal{F}, H^A\varepsilon) \le \dim_E(\mathcal{F}^+, H^A\varepsilon) \le E_d$.

**Lemma 3.A.4.** *With probability at least $1-2\zeta$, at any point in the execution of Algorithm 3, the sequence $X_{\le E_d+1}$ is an eluder sequence for $\mathcal{F}$ at scale $H^A\varepsilon$.*

*Proof.* Let us assume the event under which $\theta^\circ \in \mathrm{Sol}(X_{\le\tau})$ for $\tau \in [E_d+1]$, which has probability at least $1-\zeta$ by Lemma 3.A.3. Let us also assume the high-probability event of Lemma 3.A.1. Let $\varepsilon' = H^A\varepsilon$. The empty sequence is trivially an eluder sequence. By induction, assume for some $\tau \in [E_d+1]$ that $X_{\le\tau-1}$ is an eluder sequence. Let $\bar\theta^\circ = \flat(\otimes_{a\in[A]}[1,\theta^\circ])$ and let $\bar\theta_j = \flat(\otimes_{a\in[A]}[1,\theta_j])$.

$$\begin{aligned}
w_\tau &= \sup\left\{|f_1(X_\tau) - f_2(X_\tau)| : \sqrt{\sum_{i=1}^{\tau-1}(f_1(X_i) - f_2(X_i))^2} \le H^A\varepsilon,\ f_1, f_2 \in \mathcal{F}\right\} \\
&\ge \sup\left\{|f_1(X_\tau) - f_2(X_\tau)| : \forall i \in [\tau-1].\ |(f_1(X_i) - f_2(X_i))| \le \frac{H^A\varepsilon}{\sqrt{E_d}},\ f_1, f_2 \in \mathcal{F}\right\} \\
&\ge \left|f_{\bar\theta_\tau}(X_\tau) - f_{\bar\theta^\circ}(X_\tau)\right| > (\delta/(4H))^A - \left|f_{\bar\theta^\circ}(X_\tau)\right| > H^A\varepsilon\left(1 + \frac{1}{2\sqrt{E_d}}\right) - \frac{H^A\varepsilon}{2\sqrt{E_d}} = H^A\varepsilon,
\end{aligned}$$

where the first line expands the definition of $w_\tau$, the second comes from proving that $\forall i \in [\tau-1].\ |(f_1(X_i) - f_2(X_i))| \le \frac{H^A\varepsilon}{\sqrt{E_d}}$ implies $\sqrt{\sum_{i=1}^{\tau-1}(f_1(X_i) - f_2(X_i))^2} \le H^A\varepsilon$. We show this by assuming

the former and letting $v \in \mathbb{R}^{\tau-1}$ be $v_i = f_1(X_i) - f_2(X_i)$, and then $\|v\|_2 \le \|v\|_\infty \sqrt{\tau-1} \le H^A\varepsilon$ as $\tau - 1 \le E_d$ by the induction assumption.

The last line comes from substituting $f_1 = f_{\bar\theta_\tau}$ and $f_2 = f_{\bar\theta^\circ}$. For this we have to show that $f_{\bar\theta_\tau}, f_{\bar\theta^\circ} \in \mathcal{F}$, and that $\forall i \in [\tau-1], \left|(f_{\bar\theta_\tau}(X_i) - f_{\bar\theta^\circ}(X_i))\right| \le \frac{H^A\varepsilon}{\sqrt{E_d}}$. The former holds by definition (as $\|\theta^\circ\|_2 \le B$ and $\|\theta_\tau\|_2 \le B$ as $\theta_\tau \in \mathrm{Sol}(X_{\le\tau-1})$). For the latter, we use that $\theta^\circ, \theta_\tau \in \mathrm{Sol}(X_{\le\tau-1})$, so for either $\bar\theta \in \{\bar\theta^\circ, \bar\theta_\tau\}$, $\forall i \in [\tau-1]\,.\,\left|f_{\bar\theta}(X_i)\right| \le \left|\left\langle X_i, \bar\theta\right\rangle\right| \le \frac{H^A\varepsilon}{2\sqrt{E_d}}$, so by the triangle inequality, $\forall i \in [\tau-1]\,.\,\left|f_{\bar\theta_\tau}(X_i) - f_{\bar\theta^\circ}(X_i)\right| \le \frac{H^A\varepsilon}{\sqrt{E_d}}$. Finally, it is left to show that $\left|f_{\bar\theta_\tau}(X_\tau) - f_{\bar\theta^\circ}(X_\tau)\right| > H^A\varepsilon$. For some $t \in [n_1]$, $j \in [H]$, $X_\tau = \otimes_{a\in[A]}\hat\Delta_{\tau tja}$. Since $\left\|\hat\Delta_{\tau tja}\right\|_2 \le 3$, $\|X_\tau\|_2 \le 3^A$, so $\forall f_{\bar\theta} \in \mathcal{F}$, $f_{\bar\theta}(X_\tau) = \left\langle \mathrm{truncate}(X_\tau), \bar\theta\right\rangle = \left\langle X_\tau, \bar\theta\right\rangle$. Furthermore, because the algorithm added $(X_{\tau a})_a = (\hat\Delta_{\tau tja})_a$ in Line 13, $\min_{a\in[A]}\left|\left\langle\overline{\Delta}_{\tau tja}, [1,\theta]_\tau\right\rangle\right| > \delta/(4H) = 3H\varepsilon^{1/A}\left(1 + \frac{1}{2\sqrt{E_d}}\right)^{1/A}$. Under the assumed high-probability event of Lemma 3.A.1, for $a \in [A]$, since $\tau \in [E_d+1]$ and $t \in [n_1]$, by Lemma 3.A.1 and the triangle inequality, $\left|\left\langle\overline{\Delta}_{\tau tja}, [1,\theta]_\tau\right\rangle\right| - \left|\left\langle\hat\Delta_{\tau tja}, [1,\theta]_\tau\right\rangle\right| \le 2\delta/(12H)$, so $\min_{a\in[A]}\left|\left\langle\hat\Delta_{\tau tja}, [1,\theta]_\tau\right\rangle\right| > \delta/(12H) = H\varepsilon^{1/A}\left(1 + \frac{1}{2\sqrt{E_d}}\right)^{1/A}$, therefore $\prod_{a\in[A]}\left|\left\langle\hat\Delta_{\tau tja}, [1,\theta]_\tau\right\rangle\right| > H^A\varepsilon\left(1 + \frac{1}{2\sqrt{E_d}}\right)$. We finish by bounding $\left|f_{\bar\theta^\circ}(X_\tau)\right| \le \frac{H^A\varepsilon}{2\sqrt{E_d}}$ as $\theta^\circ \in \mathrm{Sol}(X_{\le\tau})$ by our high-probability assumption, so by the triangle inequality, and noting that $f_{\bar\theta_\tau}(X_\tau) = \prod_{a\in[A]}\left\langle\hat\Delta_{\tau tja}, [1,\theta]_\tau\right\rangle$, we have that $\left|f_{\bar\theta_\tau}(X_\tau) - f_{\bar\theta^\circ}(X_\tau)\right| \ge \prod_{a\in[A]}\left|\left\langle\hat\Delta_{\tau tja}, [1,\theta]_\tau\right\rangle\right| - \left|f_{\bar\theta^\circ}(X_\tau)\right| > H^A\varepsilon\left(1 + \frac{1}{2\sqrt{E_d}}\right) - \frac{H^A\varepsilon}{2\sqrt{E_d}}$. ■

By definition of the eluder dimension, we then have:

**Corollary 3.A.5.** *With probability at least* $1-2\zeta$, $\tau^+ \le \dim_E(\mathcal{F}, H^A\varepsilon) + 1 \le E_d + 1$.

*Proof.* Assume the high-probability statements of Lemma 3.A.4 hold and that $\tau^+ > \dim_E(\mathcal{F}, H^A\varepsilon) + 1$. Take $X_{\le\dim_E(\mathcal{F},H^A\varepsilon)+1}$ which is of length $\dim_E(\mathcal{F}, H^A\varepsilon) + 1$. Also, $\dim_E(\mathcal{F}, H^A\varepsilon) + 1 \le E_d + 1$. Therefore, by Lemma 3.A.4, $X_{\le\dim_E(\mathcal{F},H^A\varepsilon)+1}$ is an eluder sequence for $\mathcal{F}$ at scale $H^A\varepsilon$ of length $> \dim_E(\mathcal{F}, H^A\varepsilon)$, which is a contradiction. ■

### 3.A.3. Value bound

Denote by $\pi_{\mathrm{TP}}$ the policy induced by TensorPlan.

**Lemma 3.A.6.** *With probability* $1-2\zeta$, *if* $\tau^+ \in [E_d+1]$, $v_1^{\pi_{\mathrm{TP}}}(s_0) \ge \langle\varphi_1(s_0), \theta^+\rangle - \frac{1}{n_1}\sum_{t\in[n_1]}\sum_{j\in[H]}\left\langle\overline{\Delta}_{\tau^+tjA_{\tau^+tj}}, [1,\theta^+]\right\rangle - \frac{1}{2}\delta$.

*Proof.* Let us denote the state we reach after $H$ steps (once the episode is over) by $S_{H+1}$ in the following. For $\psi : \mathcal{S} \to \mathbb{R}^d$, we let $P_{sa}\psi = \int \psi(s')P_{sa}(ds')$.

Recall that under $\mathbb{P}$, the random variables $S_1 = s_0$, $A_1$, $S_2$, $A_2$, …, $A_H$, $S_{H+1}$ have the distribution of an episode in the MDP that starts from $s_0$ and follows the policy $\pi_{\mathrm{TP}}$ induced by TensorPlan.

$$
\begin{aligned}
v_1^{\pi_{\mathrm{TP}}}(s_0) &= \mathbb{E}\sum_{j\in[H]} r_{S_j,A_j} = \mathbb{E}\left\langle [\sum_{j\in[H]} r_{S_j,A_j}, \varphi_{H+1}(S_{H+1})], [1,\theta^+]\right\rangle \\
&= \mathbb{E}\left[\left\langle \varphi_1(s_0),\theta^+\right\rangle + \sum_{j\in[H]} \left\langle [r_{S_j,A_j}, \varphi_{j+1}(S_{j+1}) - \varphi_j(S_j)], [1,\theta^+]\right\rangle\right] \\
&= \left\langle \varphi_1(s_0),\theta^+\right\rangle + \sum_{j\in[H]} \mathbb{E}\left\langle [r_{S_j,A_j}, \varphi_{j+1}(S_{j+1}) - \varphi_j(S_j)], [1,\theta^+]\right\rangle \\
&= \left\langle \varphi_1(s_0),\theta^+\right\rangle + \sum_{j\in[H]} \mathbb{E}\left[\left\langle [r_{S_j,A_j}, P_{S_jA_j}\varphi_{j+1} - \varphi_j(S_j))], [1,\theta^+]\right\rangle\right] \\
&\geq \left\langle \varphi_1(s_0),\theta^+\right\rangle + \frac{1}{n_1}\sum_{t\in[n_1]}\sum_{j\in[H]} \left[\left\langle [r_{S_{\tau^+tj},A_{\tau^+tj}}, P_{S_{\tau^+tj}A_{\tau^+tj}}\varphi_{j+1} - \varphi_j(S_{\tau^+tj})], [1,\theta^+]\right\rangle\right] - \frac{1}{4}\delta \\
&\geq \left\langle \varphi_1(s_0),\theta^+\right\rangle + \frac{1}{n_1}\sum_{t\in[n_1]}\sum_{j\in[H]} \left\langle \overline{\Delta}_{\tau^+tjA_{\tau^+tj}}, [1,\theta^+]\right\rangle - \frac{1}{2}\delta,
\end{aligned}
$$

where in the first line we used that $\varphi_{H+1}(S_{H+1}) = \mathbf{0}$, in the second that $s_0 = S_1$, in the third that $s_0$ is fixed so can be moved out of the expectation, and in the fourth we used the tower rule for expectations. In the fifth line we replace the outer expectation with an average of rollouts by the algorithm that is close to the expectation with high probability, while we also switched to the variable notation used in Algorithm 3. More specifically, we use the fact that for all $h \in [H+1]$, $s \in \mathcal{S}$, and $\tau \in [E_d+1]$, we have that $\|\varphi_h(s)\|_2 \leq 1$ and $\|\theta^+\|_2 \leq B$, $\left|\left\langle [r_{S_{\tau tj},A_{\tau tj}}, P_{S_{\tau tj}A_{\tau tj}}\varphi_{j+1} - \varphi_j(S_{\tau tj}))], [1,\theta^+]\right\rangle\right| \leq 1+2B$ (as rewards are bounded in $[0,1]$). We can therefore apply Hoeffding's inequality on the $n_1$ independent rollouts:

$$
\begin{aligned}
&\mathbb{P}\left(\frac{1}{n_1}\sum_{t\in[n_1]}\left[\sum_{j\in[H]}\left[\left\langle [r_{S_{\tau tj},A_{\tau tj}}, P_{S_{\tau tj}A_{\tau tj}}\varphi_{j+1} - \varphi_j(S_{\tau tj})], [1,\theta^+]\right\rangle\right.\right.\right. \\
&\qquad \left.\left.\left. - \mathbb{E}\left\langle [r_{S_{\tau tj},A_{\tau tj}}, P_{S_{\tau tj}A_{\tau tj}}\varphi_{j+1} - \varphi_j(S_{\tau tj})], [1,\theta^+]\right\rangle\right]\right] > \delta/4\right) \\
&\leq \exp\left(-\frac{n_1\delta^2}{32H^2(1+2B)^2}\right) \leq \frac{\zeta}{E_d+1},
\end{aligned}
$$

if $n_1 = \left\lceil 32H^2(1+2B)^2/\delta^2 \log\frac{E_d+1}{\zeta}\right\rceil$. With an union bound, the probability that any of these bounds fail for any $\tau \in [E_d+1]$ is upper bounded by $\zeta$. We can therefore apply this bound for $\tau = \tau^+$, noting

that

$$\mathbb{E}\left\langle [r_{S_{\tau^+tj},A_{\tau^+tj}}, P_{S_{\tau^+tj}A_{\tau^+tj}}\varphi_{j+1} - \varphi_j(S_{\tau^+tj})], [1,\theta^+]\right\rangle = \mathbb{E}\left\langle [r_{S_j,A_j}, P_{S_jA_j}\varphi_{j+1} - \varphi_j(S_j)], [1,\theta^+]\right\rangle .$$

This is because $\theta^+ = \theta_{\tau^+}$, so for all $t \in [n_1]$, the episode $\left(S_{\tau^+t1}, A_{\tau^+t1}, \ldots, A_{\tau^+tH}, S_{\tau^+t,H+1}\right)$ is distributed identically to the episode $(S_1, A_1, S_2, A_2, \ldots, A_H, S_{H+1})$. Finally, in the sixth line we replace the remaining expectation with the average measured by the algorithm, which is close to the expectation with high probability (Lemma 3.A.1) for $\tau \in [E_d+1], t \in [n_1], j \in [H], a \in [A]$. By a union bound, this adds another $\zeta$ to the probability that our bound does not hold. ■

**Corollary 3.A.7.** *With probability at least* $1-3\zeta$, $v_1^{\pi_{\mathrm{TP}}}(s_0) \geq \langle \varphi_1(s_0), \theta^+\rangle - \frac{3}{4}\delta$.

*Proof.* Under the high-probability event of Corollary 3.A.5, $\tau^+ \leq E_d + 1$. From the proof of Lemma 3.A.6:

$$v_1^{\pi_{\mathrm{TP}}}(s_0) \geq \left\langle \varphi_1(s_0), \theta^+\right\rangle + \frac{1}{n_1}\sum_{t\in[n_1]}\sum_{j\in[H]}\left\langle \overline{\Delta}_{\tau tjA_{\tau tj}}, [1,\theta^+]\right\rangle - \frac{1}{2}\delta \geq \left\langle \varphi_1(s_0), \theta^+\right\rangle - \frac{3}{4}\delta$$

where we use the fact that, since $\tau^+ \leq E_d + 1$, we exited the $\tau$ loop as `CleanTest` was true in Line 20, so for $\tau^+$, all $t \in [n_1]$ the path in Line 16 was chosen (otherwise we would have finished with a larger $\tau^+$). This directly bounds the inner product of interest. Taking a union bound over the underlying high-probability events, $v_1^{\pi_{\mathrm{TP}}}(s_0) \geq \langle \varphi_1(s_0), \theta^+\rangle - \frac{3}{4}\delta$ holds with probability at least $1-3\zeta$. ■

**Corollary 3.A.8.** $v_1^{\pi_{\mathrm{TP}}}(s_0) \geq v_1(s_0;\theta^\circ) - \delta$.

*Proof.* Assume all high-probability events introduced so far, which hold with probability at least $1-3\zeta$. By Corollary 3.A.5, $\tau^+ \leq E_d + 1$. By Lemma 3.A.3, $\theta^\circ \in \mathrm{Sol}(X_{\leq\tau^+})$. Since $\theta^+$ was chosen optimistically in Line 5, $\langle \varphi_1(s_0), \theta^+\rangle \geq \langle \varphi_1(s_0), \theta^\circ\rangle = v_1(s_0;\theta^\circ)$. By Corollary 3.A.7, $v_1^{\pi_{\mathrm{TP}}}(s_0) \geq \langle \varphi_1(s_0), \theta^+\rangle - \frac{3}{4}\delta \geq v_1(s_0;\theta^\circ) - \frac{3}{4}\delta$. Therefore, with probability at least $1-3\zeta = 1-\frac{1}{4H}\delta$, $v_1^{\pi_{\mathrm{TP}}}(s_0) \geq v_1(s_0;\theta^\circ) - \frac{3}{4}\delta$, so $v_1^{\pi_{\mathrm{TP}}}(s_0) \geq \left(1-\frac{1}{4H}\delta\right)\left(v_1(s_0;\theta^\circ) - \frac{3}{4}\delta\right) \geq v_1(s_0;\theta^\circ) - \delta$ (using that due to bounded rewards, $v_1^{\pi_{\mathrm{TP}}}(s_0) \leq H$).

■

### 3.A.4. Final bound

We can now combine all the ingredients together to get the final result.

**Theorem 3.3.2.** *[Weisz et al., 2021a, Theorem 4.2] For any $\delta > 0$ and $B > 0$, there exists values of $\zeta, \varepsilon, n_1, n_2$, and $n_3$ such that the* TensorPlan *algorithm (Algorithm 1) is $(\delta, B)$-sound (Definition 3.2.3) with misspecification $\eta = 0$ and simulator accuracy $\lambda \leq \varepsilon/(4\sqrt{E_d}) = \tilde{\mathcal{O}}\left(\left(\frac{\delta}{12\sqrt{d}H^2}\right)^A/\sqrt{A}\right)$ for the $H$-horizon planning problem with worst-case per-episode query-cost*

$$\tilde{\mathcal{O}}\left(d^A A^4 B^2/\delta^2\left(H^5 B^2 d/\delta^2 + d^A A H^{4(A+1)} 12^{2A}/\delta^{2A}\right)\right) = \mathrm{poly}\left((dH/\delta)^A, B\right).$$

*Proof.* Fix $\delta > 0$ and $B > 0$. By Corollary 3.A.8, $v_1^{\pi_{\mathrm{TP}}}(s_0) \geq v_1(s_0;\theta^\circ) - \delta$ for any $\theta^\circ \in \Theta^\circ$. Denoting by $v^\circ = v_B^\circ$ the H-horizon $\varphi$-compatible optimal value function of $\mathscr{M}$, $v_1^{\pi_{\mathrm{TP}}}(s_0) \geq \sup_{\theta^\circ \in \Theta^\circ} v_1(s_0;\theta^\circ) - \delta = v_1^\circ(s_0) - \delta$ by definition, proving soundness. In each episode, *Line 5* in TensorPlan.`GetAction` is called $H$ times, and TensorPlan.`Init` is called once. The former results in $Hn_2A$ calls to the simulator. We turn our attention to the query complexity of TensorPlan.`Init`. The loop variable $\tau$ of `Init` goes up to $E_d + 2$ so $\tau^+ \leq E_d + 2$. Line 10 can therefore be called at most $(E_d+2)n_1HA$ times, each performing $n_2$ interactions with the simulator. Line 17 can be called at most $(E_d+2)n_1H$ times, each performing 1 interaction with the simulator. Line 12 can be called at most $(E_d+2)n_1A$ times, each performing $n_3$ interactions with the simulator. Using that $E_d = \tilde{\mathcal{O}}\left(d^A A\right)$, $\lambda = \tilde{\mathcal{O}}\left(\left(\frac{\delta}{12\sqrt{d}H^2}\right)^A/\sqrt{A}\right)$. Furthermore, using that $n_1 = \tilde{\mathcal{O}}\left(H^2B^2A/\delta^2\right)$, $n_2 = \tilde{\mathcal{O}}\left(H^2B^2dA/\delta^2\right)$, $n_3 = \tilde{\mathcal{O}}\left(d^A A^2 H^2/\varepsilon^2 + H^2B^2dA/\delta^2\right) = \tilde{\mathcal{O}}\left(d^A A^2 H^{4A+2} 12^{2A}/\delta^{2A} + H^2B^2dA/\delta^2\right)$, the (worst-case per-episode) query-cost of TensorPlan (along any episode) is

$$\begin{aligned}\tilde{\mathcal{O}}\left(Hn_2A + E_d n_1 A\left(Hn_2 + n_3\right)\right) &= \tilde{\mathcal{O}}\left(E_d n_1 A\left(Hn_2 + n_3\right)\right) = \tilde{\mathcal{O}}\left(d^A A^3 H^2 B^2/\delta^2\left(Hn_2 + n_3\right)\right) \\ &= \tilde{\mathcal{O}}\left(d^A A^4 B^2/\delta^2\left(H^5 B^2 d/\delta^2 + d^A A H^{4(A+1)} 12^{2A}/\delta^{2A}\right)\right).\end{aligned}$$

■

## 3.B. Proof of Theorem 3.3.4

**Theorem 3.3.4.** *[Weisz et al., 2021a, Theorem 4.4] For any $\delta, B > 0$,* TensorPlan *is $(\delta, B)$-sound with misspecification $\eta \leq \varepsilon/(12\sqrt{E_d})$ and simulator accuracy $\lambda \leq \varepsilon/(12\sqrt{E_d})$ with worst-case per-episode query-cost* $\mathrm{poly}\left((dH/\delta)^A, B\right)$*, when run with input $\delta' = 0.98\delta$ and simulation oracle* SIMULATE′.

*Proof.* Fix $\delta > 0$, $H \geq 1$, $\eta = \varepsilon/(12\sqrt{E_d})$ and $\lambda = \varepsilon/(12\sqrt{E_d})$. We assume that $\delta < H$ as soundness trivially holds otherwise. Let $(\mathscr{M}, \varphi)$ be any featurized MDP with 1-bounded feature maps and rewards bounded in $[0,1]$. Let SIMULATE be the $\lambda$-accurate simulation oracle for $(\mathscr{M}, \varphi)$. We

will shortly define a slightly modified simulation oracle SIMULATE′ corresponding to a featurized MDP $(\mathscr{M}', \varphi')$ derived from $(\mathscr{M}, \varphi)$. This oracle will simply use the data returned from calls to SIMULATE while we will claim that it is a simulator for $(\mathscr{M}', \varphi')$ with inaccuracy not more than $\varepsilon/(4\sqrt{E_d})$.

Denote by $\pi_{\mathrm{TP}}$ the policy while TensorPlan interacts with the simulator SIMULATE′. By the correspondence between the two MDPs, $\pi_{\mathrm{TP}}$ can be interpreted as a policy of $\mathscr{M}$. We will then prove that for all states $s \in \mathcal{S}$ of $\mathscr{M}$,

$$v_1^{\pi_{\mathrm{TP}}}(s) \geq v_1^{\circ}(s) - \delta\,,$$

where $v^{\pi_{\mathrm{TP}}}$ is the $H$-horizon value function of TensorPlan's policy $\pi_{\mathrm{TP}}$ in $\mathscr{M}$ and $v^{\circ} = v^{\circ}_{B,\eta}$ is the $H$-horizon $\varphi$-compatible optimal value function of $\mathscr{M}$ (cf. Equation (48)).

Let $\Pi^{\circ}_{B,\eta}$ be the set of memoryless, deterministic (MLD) policies that are $B$-boundedly $v$-linearly realizable with misspecification $\eta$ and features $\varphi$. Then, by definition, $v^{\circ}_{B,\eta}(s) = \sup_{\pi \in \Pi^{\circ}_{B,\eta}} v_1^{\pi}(s)$. It is enough to prove that for any $\pi \in \Pi^{\circ}_{B,\eta}$,

$$v_1^{\pi_{\mathrm{TP}}}(s) \geq v_1^{\pi}(s) - \delta\,.$$

Fix a $\theta \in \mathbb{R}^d$ such that $\left|v_h^{\pi}(s) - \langle \varphi_h(s), \theta \rangle\right| \leq \eta$ for all $h \in [H]$ and $s \in \mathcal{S}$. Such a $\theta$ exists by definition. We now construct an alternative featurized MDP $(\mathscr{M}', \varphi')$ that will mimic $\mathscr{M}$, but with slightly different rewards and an expanded state-space. The main point of introducing this MDP is that the value function of $\pi$ (when "used" in $\mathscr{M}'$) will be realizable with $\eta = 0$. The function SIMULATE′ will be defined to act as a simulator for $(\mathscr{M}', \varphi')$. Then we will use an extension Theorem 3.3.2 to argue that TensorPlan induces a policy that can compete with $\pi$ in $\mathscr{M}'$ and hence, by the correspondence between the two MDPs, it also competes with $\pi$ in $\mathscr{M}$. The required extension of Theorem 3.3.2 is as follows:

**Claim 3.B.1.** *The conclusions of Theorem 3.3.2 remain valid with the following two changes:*

*(i) The rewards in the MDP are allowed to belong to* $[-2, 2]$*;*

*(ii) A set* $\mathcal{S}_1 \subset \mathcal{S}$ *is fixed and the requirement of soundness is redefined so that the initial state chosen at the beginning of an episode must belong to* $\mathcal{S}_1$ *while v-realizability (cf. Definition 3.2.1) of a policy* $\pi$ *is redefined so that instead of* $\max_{h \in [H]} \sup_{s \in \mathcal{S}} \left|v_h^{\pi}(s) - \langle \varphi_h(s), \theta \rangle\right| \leq \eta$ *we require* $\max_{h \in [H]} \sup_{s \in \mathcal{S}_h} \left|v_h^{\pi}(s) - \langle \varphi_h(s), \theta \rangle\right| \leq \eta$ *where* $\mathcal{S}_h \subset \mathcal{S}$ *is defined as the set of states*

*that can be reached with positive probability in $\mathscr{M}$ from some state in $\mathcal{S}_1$ and action sequence of length $h-1$.*

*Proof.* For *(i)* note that shifting the rewards does not impact the proof, while the range of rewards scales the query cost quadratically (this comes from the use of Hoeffding's inequality, where ranges of temporal difference errors appear, which scale linearly with the range of rewards). For *(ii)* we only need to check that if $\theta^\circ$ is a parameter vector of a policy with the modified definition, this parameter vector will not be eliminated by TensorPlan. A quick look at the proof of Lemma 3.A.3 confirms that this is the case. Indeed, TensorPlan constructs data for checking consistency at stage $h$ only with states that it reaches through $h-1$, or $h$ actions from the initial state it is given. Therefore the states that appear with $\varphi_h$ always belong to $\mathcal{S}_h$. As such, Lemma 3.A.3 continues to hold true, and the result follows. ■

Let us now return to the definition of $\mathscr{M}' = (\mathcal{S}', \Sigma', [A], Q')$ and $\varphi'$. We let the states of $\mathscr{M}'$ be $\mathcal{S}' = \mathcal{S} \times [H] \cup \{\bot\}$, that is, the state space of $\mathscr{M}'$ contains $H$ copies of each state, and a final absorbing state $\bot$. The intention is that only states of the form $(s, h+1)$ are accessible from states of the form $(s,h)$. We let $\Sigma'$ to be the smallest $\sigma$ algebra under which $\{\bot\}$ and all the sets of the form $S \times \{h\}$ are measurable where $S \in \Sigma$ and $h \in [H]$. We let $\varphi'_h((s,\cdot)) = \varphi_h(s)$ and $\varphi'_h(\bot) = \mathbf{0}$, a $d$-dimensional vector of all zeros.

The transition kernel $Q'$ in $\mathscr{M}'$ will follow that in $\mathscr{M}$, with the appropriate modification to create the promised "levelled" structure, while the rewards are modified to "cancel out the misspecification" of policy $\pi$. That is, for $h < H$, from state $(s,h) \in \mathcal{S}'$ taking action $a \in \mathcal{A}$, kernel $Q'$ gives $(R + z(s,h), (S', h+1))$ where $(R, S') \sim Q_{sa}$ and

$$z(s,h) = \mathbb{E}_{a' \sim \pi^{(h)}(s)} \left[ \langle \varphi_h(s) - P_{sa'} \varphi_{h+1}, \theta \rangle - r_{sa'} \right] .$$

From state $(s,H) \in \mathcal{S}'$ or $\bot$, any action leads deterministically to $\bot$ while incurring zero reward.

Notice that any $(s',h) \in \mathcal{S}'$ can only be reached after exactly $h$ steps when starting from some other state $(s,1)$, $s \in \mathcal{S}$. Furthermore, denoting by $r'$ the immediate rewards in $\mathscr{M}'$, we have $r'_{(s,h),a} = r_{sa} + z(s,h)$. Note that $|z(s,h)| \le 2\eta$, since $\left| v_h^\pi(s) - \langle \varphi_h(s), \theta \rangle \right| \le \eta$ for all $h \in [H]$ and $s \in \mathcal{S}$, and $\mathbb{E}_{a' \sim \pi_h(s)} \left[ v_h^\pi(s) - P_{sa'} v_{h+1}^\pi - r_{sa'} \right] = 0$ by the Bellman equation. Hence, the rewards in $\mathscr{M}'$ are supported on $[-2\eta, 1+2\eta] \subset [-2,2]$ (as $\eta < 1/2$).

For any $(s,h) \in \mathcal{S}'$, let $\bar{v}'_h((s,h)) = \left\langle \varphi'_h((s,h)), \theta \right\rangle = \langle \varphi_h(s), \theta \rangle$. We claim that $\bar{v}'_h$ satisfies the Bellman equation of $\pi$ when policy $\pi$ in $\mathscr{M}'$ is taken as a policy of $\mathscr{M}'$ with the understanding that

in state $(s,h)$ and stage $h$, following $\pi$ means using $\pi_h(s)$, while in stage $h' \neq h$, an arbitrary action can be taken. Indeed, for any $(s,h) \in \mathcal{S}'$ we have

$$\begin{aligned}
\bar{v}'_h((s,h)) = \langle \varphi_h(s), \theta \rangle &= E_{a\sim\pi^{(h)}(s)} \left[ r_{sa} + \langle \varphi_h(s) - P_{sa}\varphi_{h+1}, \theta \rangle - r_{sa} + \langle P_{sa}\varphi_{h+1}, \theta \rangle \right] \\
&= E_{a\sim\pi^{(h)}(s)} \left[ r_{sa} + E_{a'\sim\pi^{(h)}(s)} \left[ \langle \varphi_h(s) - P_{sa'}\varphi_{h+1}, \theta \rangle - r_{sa'} \right] + \langle P_{sa}\varphi_{h+1}, \theta \rangle \right] \\
&= E_{a\sim\pi^{(h)}(s)} \left[ r'_{(s,h),a} + \langle P_{sa}\varphi_{h+1}, \theta \rangle \right] \\
&= E_{a\sim\pi^{(h)}(s)} \left[ r'_{(s,h),a} + P'_{(s,h),a}\bar{v}'_{h+1} \right] \\
&= r'_\pi((s,h)) + P'_\pi((s,h))\bar{v}'_{h+1},
\end{aligned}$$

where $P'$ is the transition kernel in $\mathscr{M}'$ and $P'_\pi$ ($r'_\pi$) is the corresponding kernel (respectively, reward function) induced by $\pi$. Since $v'^\pi$ also satisfies this equation and $\bar{v}'_{H+1} = v'^\pi_{H+1} = \mathbf{0}$, it follows that for any $(s,h) \in \mathcal{S}'$, $v'^\pi_h((s,h)) = \bar{v}'_h((s,h)) = \left\langle \varphi'_h((s,h)), \theta \right\rangle$. Now, define $\mathcal{S}'_1 = \mathcal{S} \times \{1\}$. Then, $\mathcal{S}'_h$, the set of states reachable in $\mathscr{M}'$ with positive probability from $\mathcal{S}'_1$ with an action sequence of length $h-1$, is easily seen to be a subset of $\mathcal{S} \times \{h\}$. Therefore, policy $\pi$ is v-realizable with $\eta' = 0$ in the sense of the definition of v-realizability given in Part *(ii)* of Claim 3.B.1.

For state and action $s \in \mathcal{S}, a \in [A]$, recall that SIMULATE$(s,h,a)$ is implemented by a $\lambda$-accurate simulator for $(\mathscr{M}, \varphi)$, and that the state transitions of $\mathscr{M}$ and $\mathscr{M}'$ are the same apart from that in the latter the stage counter is incremented in each transition. Hence, we define SIMULATE$'$ as follows: SIMULATE$'((s,h),h',a)$ for $(s,h) \in \mathcal{S}'$ calls $(R, S', \varphi_{h'+1}(S')) \leftarrow$ SIMULATE$(s,h',a)$ and returns $(R, (S', h+1), \varphi_{h'+1}(S'))$ for $h < H$ and $(R, \perp, \mathbf{0})$ otherwise. We also let SIMULATE$'(\perp,\cdot,\cdot)$ deterministically returns $(0, \perp, \mathbf{0})$.

Let $\pi'$ be a policy of $\mathscr{M}'$ that is induced by a planner interacting with $\mathscr{M}'$ using SIMULATE$'$ where the episode starts in $\mathscr{M}'$ are restricted to $\mathcal{S}'_1$. Then, on the one hand, $\pi'$ can be seen as a policy in $\mathscr{M}$: For a history in $\mathscr{M}$, one just needs to add the respective stage counters to the states in the history and then use $\pi'$ to return an action.

Now note that the reward distribution of $\mathscr{M}'$ is shifted by up to $2\eta$ compared to the reward distribution of $\mathscr{M}$. The distribution of the simulator's rewards $\mathrm{clip}_{[0,1]}(R_{sa} + \Lambda_{sa})$ are shifted by up to $\Lambda_{sa} \leq \lambda$ compared to the reward distribution of $\mathscr{M}$, so by the triangle inequality it is shifted by up to $2\eta + \lambda$ compared to the reward distribution of $\mathscr{M}'$. Since $2\eta + \lambda = \varepsilon/(4\sqrt{E_d})$, using the reward of the simulator call SIMULATE$(s,h',a)$ as the output of SIMULATE$'((s,h),h',a)$ ensures SIMULATE$'$ is a simulator for $(\mathscr{M}', \varphi')$ with inaccuracy $\varepsilon/(4\sqrt{E_d})$.

Therefore, applying Claim 3.B.1 with $\eta' = 0$, $\lambda' = \varepsilon/(4\sqrt{E_d})$, and $\delta' = 0.98\delta$, TensorPlan is $(\delta', B)$-sound for $\mathscr{M}'$ and initial states from $\mathcal{S}'$ when run with the simulator SIMULATE′, with worst-case per-episode query-cost poly$\left((dH/\delta)^A, B\right)$. Thus, for all $(s,1) \in \mathcal{S}'$ (ie. all $s \in \mathcal{S}$), $v_1'^{\pi_{\mathrm{TP}}}((s,1)) \geq v_1'^{\circ}((s,1)) - 0.98\delta$, where $v'^{\circ} = v'^{\circ}_{B,0}$ is the $H$-horizon $\varphi$-compatible optimal value function of $\mathscr{M}'$. As $\pi$ is $v$-linearly realizable in MDP $\mathscr{M}'$ with no misspecification, $v_1'^{\circ}((s,1)) \geq v_1'^{\pi}((s,1))$, so $v_1'^{\pi_{\mathrm{TP}}}((s,1)) \geq v_1'^{\pi}((s,1)) - 0.98\delta$. As the state transition distributions of $\mathscr{M}$ and $\mathscr{M}'$ are the same except for the stage counter incrementation in $\mathscr{M}'$, the distribution of any policy $\pi$ in $\mathscr{M}$ producing an episode $(S_1, A_1, S_2, A_2 \ldots, S_H, A_H)$ is the same as the distribution of $\pi$ in $\mathscr{M}'$ producing the episode $((S_1,1), A_1, (S_2,2), A_2, \ldots, (S_H,H), A_H)$. Furthermore, the rewards of $\mathscr{M}'$ are shifted by up to $2\eta$. Therefore, the $H$-horizon value functions $v_1^{\pi'}(s)$ and $v_1'^{\pi'}((s,1))$ for any $\pi'$ differ by at most $2H\eta$, and thus by treating $\pi_{\mathrm{TP}}$ as a policy of both $\mathscr{M}$ and $\mathscr{M}'$, we have

$$v_1^{\pi_{\mathrm{TP}}}(s) \geq v_1'^{\pi}((s,1)) - 0.98\delta - 2H\eta \geq v_1^{\pi}(s) - 0.98\delta - 4H\eta \geq v_1^{\pi}(s) - \delta,$$

as $4H\eta = \frac{H\varepsilon}{3\sqrt{E_d}} \leq \frac{H\frac{\delta}{12H^2}/(1+0.5)}{3\sqrt{E_d}} \leq \frac{\delta}{18H}/3 \leq 0.02\delta$. ■

We note in passing that the result as stated could be (slightly) strengthened and simplified: Since SIMULATE′ generates the same data (with some redundancy) as SIMULATE, using TensorPlan on $(\mathscr{M}', \varphi')$ via SIMULATE′ produces the same policy in $\mathscr{M}$ as using it directly on $(\mathscr{M}, \varphi)$ via SIMULATE. Thus, SIMULATE′ is only needed for the proof; the conclusion of the result applies when TensorPlan directly uses SIMULATE with a near-realizable featurized MDP.

By reiterating the arguments of Claim 3.B.1 in the context of Theorem 3.3.4, we get the following claim, which will be needed in the next section:

**Claim 3.B.2.** *The conclusions of Theorem 3.3.4 remain valid with the following two changes:*

*(i) The rewards in the MDP are allowed to belong to* $[-2, 2]$*;*

*(ii) A set* $\mathcal{S}_1 \subset \mathcal{S}$ *is fixed and the requirement of soundness is redefined so that the initial state chosen at the beginning of an episode must belong to* $\mathcal{S}_1$ *while v-realizability (cf. Definition 3.2.1) of a policy* $\pi$ *is redefined so that instead of* $\max_{h\in[H]} \sup_{s\in\mathcal{S}} \left|v_h^{\pi}(s) - \langle \varphi_h(s), \theta\rangle\right| \leq \eta$ *we require* $\max_{h\in[H]} \sup_{s\in\mathcal{S}_h} \left|v_h^{\pi}(s) - \langle \varphi_h(s), \theta\rangle\right| \leq \eta$ *where* $\mathcal{S}_h \subset \mathcal{S}$ *is defined as the set of states that can be reached with positive probability in* $\mathscr{M}$ *from some state in* $\mathcal{S}_1$ *and action sequence of length* $h-1$*.*

## 3.C. Proof of Theorem 3.3.5

**Theorem 3.3.5.** *[Weisz et al., 2021a, Theorem 4.5] For any $\delta, B > 0$, TensorPlan is $(\delta, B)$-sound for discounted MDPs with discount factor $0 \le \gamma < 1$, with misspecification $\eta \le \varepsilon/(24\sqrt{E_d})$ and simulator accuracy $\lambda \le \varepsilon/(12\sqrt{E_d})$, with worst-case per-state query-cost* $\text{poly}\left(\left(dH_{\gamma,\delta}/\delta\right)^A, B\right)$, *when run with input $\delta' = 0.98\delta$ and simulation oracle* $\textsc{Simulate}^{\gamma,\delta}$.

*Proof.* Fix a suboptimality target $\delta > 0$. We assume that $\delta < H$ as soundness trivially holds otherwise. Fix $\eta = \varepsilon/(24\sqrt{E_d})$ and $\lambda = \varepsilon/(12\sqrt{E_d})$; proving soundness and the query-cost bound for these values implies the same results for smaller $\eta$ or $\lambda$. Let $(\mathscr{M}, \varphi)$ be a featurized MDP in the discounted setting with 1-bounded feature maps and rewards bounded in $[0,1]$. Take a $\lambda$-accurate simulation oracle $\textsc{Simulate}$ for $(\mathscr{M}, \varphi)$. Let

$$H_{\gamma,\delta} = \left\lceil \frac{\log\left((1-\gamma)\eta\right)/\log\gamma}{1-\gamma} \right\rceil.$$

In the remainder of the proof we shorten $H_{\gamma,\delta}$ and will just use $H$ (i.e., in what follows $H = H_{\gamma,\delta}$). We now construct a featurized, fixed-horizon MDP $(\mathscr{M}', \varphi^{\gamma,\delta})$ with horizon $H$. Let the states of $\mathscr{M}'$ be $\mathcal{S}' = \mathcal{S} \times [H] \cup \{\perp\}$, that is, the state space contains $H$ copies of each state, and an additional state $\perp$, which will play the role of a final, absorbing state. The $\sigma$ algebra for $\mathcal{S}'$ is constructed as in the proof of Theorem 3.3.4 (we omit the definition). The action set of $\mathscr{M}'$ remains $[A]$. The kernel $Q'$ is inherited from $\mathscr{M}$, again, with the appropriate modification to create the promised "levelled" structure, while the rewards are modified to accommodate discounting: That is, for $h < H$, from state $(s,h) \in \mathcal{S}'$ taking action $a \in \mathcal{A}$, kernel $Q'$ gives $(\gamma^{h-1}R, (S', h+1))$ where $(R, S') \sim Q_{sa}$. From state $(s,H) \in \mathcal{S}'$ or $\perp$, any action leads deterministically to $\perp$ while incurring zero reward. In words, states with associated stage $h < H$ lead to respective states with associated stage $h+1$, and the episode is terminated after $H$ steps by transitioning to the absorbing state $\perp$. By letting $r'$ denote the immediate expected rewards in $\mathscr{M}'$, for state $(s,h) \in \mathcal{S}'$ and action $a$ we have $r'_{(s,h),a} = \gamma^{h-1} r_{sa}$.

Let $\varphi_h^{\gamma,\delta}((s,\cdot)) = \gamma^{h-1}\varphi(s)$ and $\varphi_h^{\gamma,\delta}(\perp) = \mathbf{0}$, a $d$-dimensional vector of all zeros. We define $\textsc{Simulate}^{\gamma,\delta}$ as follows: $\textsc{Simulate}^{\gamma,\delta}$ is a simulation oracle for $(\mathscr{M}', \varphi^{\gamma,\delta})$ so that for $(s,h) \in \mathcal{S}'$ with $h < H$, $h' \in [H]$ and $a \in [A]$, $\textsc{Simulate}^{\gamma,\delta}((s,h), h', a)$ first gets $(R, S, \varphi(S)) \leftarrow \textsc{Simulate}(s, h', a)$ to return $\left(\gamma^{h-1}R, (S, h+1), \varphi_{h'+1}^{\gamma,\delta}((S, h+1))\right)$, while it returns $\left(\gamma^{H-1}R, \perp, \mathbf{0}\right)$ when $h = H$. Finally, $\textsc{Simulate}^{\gamma,\delta}(\perp,\cdot,\cdot)$ deterministically returns $(0, \perp, \mathbf{0})$. As $\gamma < 1$, the inaccuracy of $\textsc{Simulate}^{\gamma,\delta}$ is at most the inaccuracy of $\textsc{Simulate}$, which is at most $\lambda$, by assumption.

Next, we prove that the value function of the discounted MDP $\mathscr{M}$ is close to the corresponding values of its $H$-horizon counterpart $\mathscr{M}'$. For this, we first need to agree on a way of transporting policy between $\mathscr{M}$ and $\mathscr{M}'$. This is done as follows: Let $\alpha$ be a function that maps histories in $\mathscr{M}$

to histories in $\mathscr{M}'$ by adding stage counters to them. Let $\alpha^{-1}$ be the "inverse", which simply drops stage indices from histories of $\mathscr{M}'$. For any $h$ history of $\mathscr{M}$, $\alpha^{-1}(\alpha(h)) = h$, while $\alpha(\alpha^{-1}(h')) = h'$ holds for all histories $h'$ of $\mathscr{M}'$ whose start state is from $\mathcal{S}'_1 = \mathcal{S} \times \{1\}$ and where the states in the history do not include $\bot$. If $\pi'$ is any (possibly memoryful) policy of $\mathscr{M}'$, following $\pi'$ in $\mathscr{M}$ means that given some history $h$ of $\mathscr{M}$, the action $A \sim \pi'(\cdot|\alpha(h))$ should be taken. Conversely, using a policy $\pi$ of $\mathscr{M}$ in $\mathscr{M}'$ means that given a history $h'$, $A \sim \pi(\cdot|\alpha^{-1}(h'))$ should be taken. This way, we can view a policy of either $\mathscr{M}$ or $\mathscr{M}'$ as a policy of the other MDP.

Now take any policy $\pi$ of $\mathscr{M}$ and take any $(s,h) \in \mathcal{S}'$. As $\pi$ is also a policy of $\mathscr{M}'$, we can talk about its value function in $\mathscr{M}'$, which we denote by $v'^{\pi}$. By definition, $v'^{\pi}_h((s,h_0)) = \mathbb{E}'_{\pi,(s,h_0)}[\sum_{h'=1}^{H-h+1} r'_{(S_{h'},h_0+h'-1),A_{h'}}]$, where $\mathbb{E}'_{\pi,s'}$ denotes the expectation operator underlying the distribution $\mathbb{P}'_{\pi,s'}$ over state-action trajectories induced by the interconnection of $\pi$ and $\mathscr{M}'$ given the initial state $s' \in \mathcal{S}'$. Similarly, we will use $\mathbb{E}_{\pi,s}$ to denote this operator when the MDP is $\mathscr{M}$ and the initial state is $s \in \mathcal{S}$, and we let $\mathbb{P}_{\pi,s}$ denote the underlying distribution. With this note that

$$\mathbb{P}'_{\pi,(s,h)}(U \times V) = \mathbb{P}_{\pi,s}(\alpha(U \times V)) \tag{60}$$

holds for any measurable subset $U$ of $(\mathcal{S} \times [H] \times [A])^{H-h+1}$ and where $V = (\mathcal{S} \times [H] \times [A])^{\mathbb{N}_+}$ is the set of all histories. We claim that the following holds:

$$\left|v'^{\pi}_h((s,h)) - \gamma^{h-1} v^{\pi}(s)\right| \le \eta\,. \tag{61}$$

We calculate

$$\begin{aligned}
&\left|v'^{\pi}_h((s,h)) - \gamma^{h-1} v^{\pi}(s)\right| \\
&\quad= \left|\mathbb{E}'_{\pi,(s,h)}\left[\sum_{h'=1}^{H-h+1} r'_{(S_{h'},h+h'-1),A_{h'}}\right] - \gamma^{h-1}\mathbb{E}_{\pi,s}\left[\sum_{h'=1}^{\infty} \gamma^{h'-1} r_{S_{h'},A_{h'}}\right]\right| \\
&\quad= \left|\gamma^{h-1}\mathbb{E}'_{\pi,(s,h)}\left[\sum_{h'=1}^{H-h+1} \gamma^{h'-1} r_{S_{h'},A_{h'}}\right] - \gamma^{h-1}\mathbb{E}_{\pi,s}\left[\sum_{h'=1}^{H-h+1} \gamma^{h'-1} r_{S_{h'},A_{h'}}\right]\right. \\
&\qquad\left. - \gamma^{h-1}\mathbb{E}_{\pi,s}\left[\sum_{h'=H-h+2}^{\infty} \gamma^{h'-1} r_{S_{h'},A_{h'}}\right]\right| \\
&\quad= \left|-\gamma^{H}\mathbb{E}_{\pi,s}\left[\sum_{h'=H-h+2}^{\infty} \gamma^{h'-(H-h+2)} r_{S_{h'},A_{h'}}\right]\right| \qquad \text{(by Eq. (60))} \\
&\quad\le \gamma^{H}\sum_{i=0}^{\infty}\gamma^{i} = \frac{\gamma^{H}}{1-\gamma} \le \frac{\gamma^{\log((1-\gamma)\eta)/\log\gamma}}{1-\gamma} = \eta\,,
\end{aligned}$$

where in the last line used the fact that rewards are bounded in $[0,1]$. Now, notice that if $\pi$ was a policy of $\mathscr{M}'$, Eq. (60) would still hold true, and as such, Eq. (61) also holds for $\pi$.

Take any policy $\pi$ that is $v$-linearly realizable in $\mathscr{M}$ with misspecification $\eta$ under the feature map $\varphi$. By definition, there exists a $\theta \in \mathbb{R}^d$ such that $|v^\pi(s) - \langle \varphi(s), \theta \rangle| \le \eta$ for all $s \in \mathcal{S}$ (ie. for all $(s,h) \in \mathcal{S}'$). By Eq. (61) and the triangle inequality, for all $(s,h) \in \mathcal{S}'$,

$$\begin{aligned}\left| v'^{\pi}_h((s,h)) - \left\langle \varphi^{\gamma,\delta}_h((s,h)), \theta \right\rangle \right| &= \left| v'^{\pi}_h((s,h)) - \gamma^{h-1} \langle \varphi(s), \theta \rangle \right| \\ &\le \left| v'^{\pi}_h((s,h)) - \gamma^{h-1} v^\pi(s) \right| + \gamma^{h-1} |v^\pi(s) - \langle \varphi(s), \theta \rangle| \le 2\eta\,.\end{aligned}$$

Therefore any such policy $\pi$ is $v$-linearly realizable in MDP $\mathscr{M}'$ with misspecification $2\eta$ under the feature map $\varphi^{\gamma,\delta}$ for the respective stage $h$ for each state $(s,h) \in \mathcal{S}'$.

Therefore we can apply Claim 3.B.2 for featurized MDP $(\mathscr{M}', \varphi^{\gamma,\delta})$, initial set $\mathcal{S} \times \{1\}$, and $\lambda$-accurate simulator $\textsc{Simulate}^{\gamma,\delta}$, with misspecification $\eta' = 2\eta$ and $\delta' = 0.98\delta$, which guarantees that TensorPlan is $(\delta', B)$-sound for MDP $\mathscr{M}'$ when run with this simulator and features. Furthermore, it has a worst-case per-state query-cost $\operatorname{poly}\left((dH/\delta)^A, B\right)$. Denote by $\pi_{\text{TP}}$ the policy induced by TensorPlan while interacting with the simulator $\textsc{Simulate}^{\gamma,\delta}$. We then have that $\pi_{\text{TP}}$ satisfies

$$v'^{\pi_{\text{TP}}}_1((s,1)) \ge v'^{\circ}_1((s,1)) - 0.98\delta\,,$$

where $v'^{\pi_{\text{TP}}}$ is the $H$-horizon value function of $\pi_{\text{TP}}$ in $\mathscr{M}'$ and $v'^{\circ} = v'^{\circ}_{B,2\eta}$ is the $H$-horizon $\varphi^{\gamma,\delta}$-compatible optimal value function of $\mathscr{M}'$ under misspecification $2\eta$ (cf. Equation (48)). Similarly, let $v^\circ = v^\circ_{B,\eta}$ be the discounted $\varphi$-compatible optimal value function of $\mathscr{M}$ under misspecification $\eta$. Let $\Pi'^{\circ}_{B,2\eta}$ be the set of MLD policies that are $B$-boundedly $v$-linearly realizable in MDP $\mathscr{M}'$ with misspecification $2\eta$ and features $\varphi^{\gamma,\delta}$, and let $\Pi^\circ_{B,\eta}$ be the set of MLD policies that are $B$-boundedly $v$-linearly realizable in $\mathscr{M}$ with misspecification $\eta$ and features $\varphi$. Then, by definition, $v'^{\circ}_{B,2\eta}(s) = \sup_{\pi \in \Pi'^{\circ}_{B,2\eta}} v'^{\pi}_1((s,1))$ and $v^\circ_{B,\eta}(s) = \sup_{\pi \in \pi'^{\circ}_{B,\eta}} v^\pi(s)$.

As we have seen, $\pi \in \Pi^\circ_{B,\eta}$ implies $\pi \in \Pi'^{\circ}_{B,2\eta}$, in other words, $\Pi^\circ_{B,\eta} \subseteq \Pi'^{\circ}_{B,2\eta}$. For any policy $\pi$ Eq. (61) applies with any $(s,1) \in \mathscr{M}'$, and therefore

$$\begin{aligned}v^\circ_{B,\eta}(s) &= \sup_{\pi \in \Pi^\circ_{B,\eta}} v^\pi(s) \le \sup_{\pi \in \Pi'^{\circ}_{B,2\eta}} v^\pi(s) \le \sup_{\pi \in \Pi'^{\circ}_{B,2\eta}} v'^{\pi}_1((s,1)) + \eta \\ &= v'^{\circ}_{B,2\eta}((s,1)) + \eta \le v'^{\pi_{\text{TP}}}_1((s,1)) + 0.98\delta + \eta \le v'^{\pi_{\text{TP}}}(s) + 0.98\delta + 2\eta\,,\end{aligned}$$

where the last inequality used again Eq. (61) with $\pi_{\mathrm{TP}}$. Lastly we use that $\eta = \frac{\varepsilon}{24\sqrt{E_d}} \le \frac{\frac{\delta}{12H^2}/(1+0.5)}{24\sqrt{E_d}} \le \frac{\delta}{18H}/24 < 0.01\delta$ to obtain $v^{\circ}_{B,\eta}(s) \le v_1'^{\pi_{\mathrm{TP}}}((s,1)) + \delta$, which establishes that TensorPlan's policy $\pi_{\mathrm{TP}}$ is $(\delta, B)$-sound for the featurized MDP $(\mathscr{M}, \varphi)$ in the discounted setting, with misspecification $\eta \le \frac{\varepsilon}{24\sqrt{E_d}}$ and simulator accuracy $\lambda \le \frac{\varepsilon}{12\sqrt{E_d}}$. ■

## Chapter 4

# Planning with $q^\pi$-realizability

Switching to $q^\pi$-realizability, this chapter focuses on proving Theorem 1.5.1. For generality, for this chapter only, we switch to the *discounted, infinite horizon* MDP setting. This requires a slightly different notational framework, which we introduce in this chapter shortly. The improvement in generality comes from the fact that in discounted MDPs, the same state can be entered multiple times in an episode (unlike in our finite horizon setup, where the state space is disjoint for each stage). In our discounted MDP setup however, the only policies that we require to be $q^\pi$-realizable are the stationary ones, that behave the same way when seeing the same state, regardless of where in the episode they encounter that state. As a result, $q^\pi$-realizability becomes a more permissive setting for featurized MDPs (as fewer policies need to be realizable), consequently making this a more challenging setting for planners. Thus, the planner and corresponding guarantees we present for this discounted setting (Theorems 4.1.2 and 4.1.3) are more powerful than if we had continued with the finite horizon setting.

We start by considering approximate dynamic programming in $\gamma$-discounted Markov decision processes and apply it to approximate planning with linear value-function approximation. Our first contribution is a new variant of APPROXIMATE POLICY ITERATION (API), called CONFIDENT APPROXIMATE POLICY ITERATION (CAPI), which computes a deterministic stationary policy with an optimal error bound scaling linearly with the product of the effective horizon $H$ and the worst-case approximation error $\varepsilon$ of the action-value functions of stationary policies. This improvement over API (whose error scales with $H^2$) comes at the price of an $H$-fold increase in memory cost. Unlike Scherrer and Lesner (2012), who recommended computing a non-stationary policy to achieve a similar improvement (with the same memory overhead), we are able to stick to stationary policies. This allows for our second contribution, the application of CAPI to planning with local access to a simulator and $d$-dimensional linear function approximation. As such, we design a planning algo-

rithm that applies CAPI to obtain a sequence of policies with successively refined accuracies on a dynamically evolving set of states. The algorithm outputs an $\tilde{\mathcal{O}}(\sqrt{d}H\varepsilon)$-optimal policy after issuing $\tilde{\mathcal{O}}(dH^4/\varepsilon^2)$ queries to the simulator, simultaneously achieving the optimal accuracy bound and the best known query complexity bound, while earlier algorithms in the literature achieve only one of them. This query complexity is shown to be tight in all parameters except $H$. These improvements come at the expense of a mild (polynomial) increase in memory and computational costs of both the algorithm and its output policy.

## 4.1. Introduction

In this chapter we focus on the problem of planning with linear function approximation. Our goal is to design a planner and prove that it finds and outputs a near-optimal policy with high probability in response to any call, when given query access to a simulator. Instead of the global access, we adopt the more practical and challenging *local access* setting. The efficiency of a planner is measured in four ways: the *suboptimality* of the policy found, that is, how far its value is from that of the optimal policy; the *query cost*, that is, the number of queries issued to the simulator; the *computational cost*, which is the number of operations used; and the *memory cost*, which is the amount of memory used (we adopt the real computation model for these costs).

In this chapter, for featurized MDPs, we consider a feature-map to be a "good fit" to an MDP if the worst-case error of using the feature-map to approximate value functions of all *stationary, deterministic, memoryless* policies of the MDP is small:

**Definition 4.1.1** ($q^\pi$-realizability: uniform policy value-function approximation error)**.** *Given an MDP, the uniform policy value-function approximation error induced by a feature map $\varphi$, which maps state-action pairs $(s,a)$ to the Euclidean ball of radius $L$ centered at zero in $\mathbb{R}^d$, over a set of parameters belonging to the $d$-dimensional centered Euclidean ball of radius $B$ is*

$$\varepsilon = \sup_{\pi} \inf_{\theta:\|\theta\|_2 \le B} \sup_{(s,a)} |q^\pi(s,a) - \langle \varphi(s,a), \theta \rangle| ,$$

*where the outermost supremum is over all possible stationary deterministic memoryless policies (i.e., maps from states to actions) of the MDP.*

Our goal is to design algorithms that scale gracefully with the uniform approximation error $\varepsilon$ at the expense of controlled computational cost. To achieve nontrivial guarantees, the uniform approximation error needs to be "small". As we recall from Chapter 1, this (implicit) assumption is stronger than the $q^\star$-realizability assumption (where the approximation error is only considered for

optimal policies), which Weisz et al. (2021b) showed an exponential query complexity lower bound for. At the same time, it is (strictly) weaker than the linear MDP assumption (Zanette et al., 2020b), for which there are efficient algorithms to find a near-optimal policy in the online setting (without a simulator) (Jin et al., 2020b), even in the more challenging reward-free setting where the rewards are only revealed after exploration (Wagenmaker et al., 2022).

In the *local access* setting, the planner learns the features $\varphi(s,a)$ of a state-action pair *only* for those states $s$ that have already been encountered. In contrast, in the *global access* setting, the whole feature map $\varphi(\cdot,\cdot)$, of (possibly infinite) size $d|\mathcal{S}||\mathcal{A}|$ (where $\mathcal{S}$ and $\mathcal{A}$ are the state and action sets, resp.), is given to the planner as input. In the latter setting, when only the query cost is counted, Du et al. (2019a) and Lattimore et al. (2020) proposed algorithms (the latter working in the misspecified, $\varepsilon > 0$ regime) that issue a number of queries that is polynomial in the relevant parameters, but require a barycentric spanner or near-optimal design of the input features. In the worst case, computing any of these sets scales polynomially in $|\mathcal{S}|$ and $|\mathcal{A}|$, which can be prohibitive.

In the case of *local access*, considered in this chapter, the best known bound on the suboptimality of the computed policy is achieved by CONFIDENT MC-POLITEX (Yin et al., 2022). In the more permissive *global access* setting, the best known query cost is achieved by Lattimore et al. (2020). Our algorithm, CAPI-QPI-PLAN (given in Algorithm 6), achieves the *best of both* while only assuming *local access*. This is shown in the next theorem; in the theorem $\varepsilon$ is as defined in Definition 4.1.1, $\gamma$ is the discount factor, and $v^\star$ and $v^\pi$ are the state value functions associated with the optimal policy and policy $\pi$, respectively (precise definitions of these quantities are given in the next section). A comparison to other algorithms in the literature is given in Table 2; there the accuracy parameter $\omega$ of the algorithms is set to $\varepsilon$, but a larger $\omega$ can be used to trade off suboptimality guarantees for an improved query cost.

**Theorem 4.1.2** (Weisz et al., 2022a, Theorem 1.2)**.** *For any confidence parameter $\delta \in (0,1]$, accuracy parameter $\omega > 0$, and initial state $s_0 \in \mathcal{S}$, with probability at least $1-\delta$,* CAPI-QPI-PLAN *(Algorithm 6) finds a policy $\pi$ with*

$$v^\star(s_0) - v^\pi(s_0) = \tilde{\mathcal{O}}\left((\varepsilon+\omega)\sqrt{d}(1-\gamma)^{-1}\right), \tag{62}$$

*while executing at most $\tilde{\mathcal{O}}\left(d(1-\gamma)^{-4}\omega^{-2}\right)$ queries in the* local access *setting.*

CAPI-QPI-PLAN is based on CONFIDENT MC-LSPI, another algorithm of Yin et al. (2022), which relies on policy iteration from a *core set* of informative state-action pairs, but achieves inferior

performance both in terms of suboptimality and query complexity. However, CAPI-QPI-PLAN's improvements come at the expense of increased memory and computational costs, as shown in the next theorem: compared to CONFIDENT MC-LSPI, the memory and computational costs of our algorithm increase by a factor of the effective horizon $H = \tilde{\mathcal{O}}(1/(1-\gamma))$, and the policy computed by CAPI-QPI-PLAN uses a $dH$ factor more memory for storage and a $d^2H$ factor more computation to evaluate.

**Theorem 4.1.3** (Weisz et al., 2022a, Theorem 1.3, memory and computational cost)**.** *The memory and computational cost of running* CAPI-QPI-PLAN *(Algorithm 6) are* $\tilde{\mathcal{O}}\left(d^2/(1-\gamma)\right)$ *and* $\tilde{\mathcal{O}}\left(d^4|\mathcal{A}|(1-\gamma)^{-5}\omega^{-2}\right)$*, respectively, while the memory and computational costs of storing and evaluating the final policy outputted by* CAPI-QPI-PLAN*, respectively, are* $\tilde{\mathcal{O}}\left(d^2/(1-\gamma)\right)$ *and* $\tilde{\mathcal{O}}\left(d^3|\mathcal{A}|/(1-\gamma)\right)$.

Next we present a lower bound corresponding to Theorem 4.1.2 that holds even in the more permissive *global access* setting, and shows that CAPI-QPI-PLAN trades off the query cost and the suboptimality of the returned policy asymptotically optimally up to its dependence on $1/(1-\gamma)$. See Weisz et al. (2022a) for the proof.

**Theorem 4.1.4** (Weisz et al., 2022a, Theorem 1.4, query cost lower bound)**.** *Let* $\alpha \in (0, \frac{0.05\gamma}{(1-\gamma)(1+\gamma)^2})$, $\delta \in (0, 0.08]$, $\gamma \in [\frac{7}{12}, 1]$, $d \geq 3$, *and* $\varepsilon \geq 0$. *Then there is a class* $\mathcal{M}$ *of MDPs with uniform policy value-function approximation error at most* $\varepsilon$ *such that any planner that guarantees to find an* $\alpha$*-optimal policy* $\pi$ *(i.e.,* $v^\star(s_0) - v^\pi(s_0) \leq \alpha$*) with probability at least* $1-\delta$ *for all* $M \in \mathcal{M}$ *when used with a simulator for* $M$ *with* global access*, the worst-case (over* $\mathcal{M}$*) expected number of queries issued by the planner is at least*

$$\max\left(\exp\left(\Omega\left(\frac{d\varepsilon^2}{\alpha^2(1-\gamma)^2}\right)\right), \Omega\left(\frac{d^2}{\alpha^2(1-\gamma)^3}\right)\right). \tag{63}$$

If $\omega$ is set to $\varepsilon$ for CAPI-QPI-PLAN, the first term of Eq. (63) implies that any planner with an asymptotically smaller (apart from logarithmic factors) suboptimality guarantee than Eq. (62) executes exponentially many queries in expectation. The second term of Eq. (63), which is shown to be a lower bound in Weisz et al. (2022a, Theorem H.3) even in the more general setting of linear MDPs with zero misspecification ($\varepsilon = 0$), matches the query complexity of Theorem 4.1.2 up to an $\tilde{\mathcal{O}}((1-\gamma)^2)$ factor. Thus, the lower bound implies that the suboptimality and query cost bounds of Theorem 4.1.2 are tight up to logarithmic factors in all parameters except the $1/(1-\gamma)$-dependence of the query cost bound.

At the heart of our method is a new algorithm, which we call CONFIDENT APPROXIMATE POLICY ITERATION (CAPI). This algorithm, which belongs to the family of approximate dynamic programming algorithms (Bertsekas, 2012; Munos, 2003, 2005), is a novel variant of APPROXIMATE POLICY ITERATION (API) (Bertsekas and Tsitsiklis, 1996): in the policy improvement step, CAPI only updates the policy in states where it is confident that the update will improve the performance. This simple modification allows CAPI to avoid the problem of "classical" approximate dynamic programming algorithms (approximate policy and value iteration) of inflating the value function evaluation error by a factor of $H^2$ where $H = \tilde{\mathcal{O}}(1/(1-\gamma))$ (for discussions of this problem, see also the papers by Scherrer and Lesner, 2012 and Russo, 2020), and reduce this inflation factor to $H$. A similar result has already been achieved by Scherrer and Lesner (2012), who proposed to construct a non-stationary policy that strings together all policies obtained while running either approximate value or policy iteration. However, applying this result to our planning problem is problematic, since the policies to be evaluated are non-stationary, and hence including them in the policy set we aim to approximate may drastically increase the error $\varepsilon$ as compared to Definition 4.1.1, which only considers stationary memoryless policies.

While the improvements provided by CAPI allows CAPI-QPI-PLAN to match the performance of CONFIDENT MC-POLITEX in terms of suboptimality, it is unlikely that a simple modification of CONFIDENT MC-POLITEX would lead to an algorithm which matches CAPI-QPI-PLAN's performance in terms of query cost (see Table 2): Both methods evaluate a sequence of policies at an $\tilde{\mathcal{O}}(\varepsilon)$ accuracy each (requiring $\tilde{\mathcal{O}}(1/\varepsilon^2)$ queries, omitting the dependence on other parameters). However, while CAPI-QPI-PLAN (and CONFIDENT MC-LSPI) evaluates $\mathcal{O}(\log(1/\varepsilon))$ (again in terms of $\varepsilon$ only) policies to find one which is $\tilde{\mathcal{O}}(\varepsilon)$-optimal, CONFIDENT MC-POLITEX needs to compute $\tilde{\mathcal{O}}(1/\varepsilon^2)$ policies to achieve the same. As a consequence, CONFIDENT MC-POLITEX only achieves $\tilde{\mathcal{O}}(1/\varepsilon^4)$ query complexity, and to match CAPI-QPI-PLAN's $\tilde{\mathcal{O}}(1/\varepsilon^2)$ complexity, one would need to come up with either significantly better policy evaluation methods (potentially using the similarity in the subsequent policies) or a much faster (exponential vs. square-root) convergence rate in the suboptimality of the policy sequence produced by CONFIDENT MC-POLITEX.

The rest of the chapter is organized as follows: The model and notation are introduced in Section 4.2. CAPI is introduced and analyzed in Section 4.3. Planning with $q^\pi$-realizability is introduced in Section 4.4, with CAPI-QPI-PLAN being built-up and analyzed in Sections 4.4.1 and 4.4.2. In particular, the proof of Theorem 4.1.2 is given in Section 4.4.2. Several proofs are

relegated to appendices, in particular, Theorem 4.1.3 is proved and implementation details of CAPI-QPI-PLAN are discussed in Section 4.G.

## 4.2. Notation and preliminaries

We recall the most important facts about MDPs and introduce a slight variation of our previous notation. In particular, we switch to the discounted infinite horizon objective. This allows us to present the advantage of our method that we can stick to stationary policies only. This improves both the strength of our modification to API, and leads to a more general result for $q^\pi$-realizable MDPs, where only the stationary policies need to be realizable.

For some integer $i$, let $[i] = \{0, \dots, i-1\}$. For $x \in \mathbb{R}$, let $\lceil x \rceil$ denote the smallest integer i such that $i \geq x$. For a positive definite $V \in \mathbb{R}^{d\times d}$ and $x \in \mathbb{R}^d$, let $\|x\|_V^2 = x^\top V x$. For matrices $A$ and $B$, we say that $A \succeq B$ if $A - B$ is positive semidefinite. Let $\mathbf{I}$ be the $d$-dimensional identity matrix. Let $\mathcal{M}_1(X)$ denote the space of probability distributions supported on the set $X$ (throughout, we assume that the $\sigma$-algebra is implicit). We write $a \approx_\varepsilon b$ for $a, b, \varepsilon \in \mathbb{R}$ if $|a - b| \leq \varepsilon$. We denote by $\tilde{\mathcal{O}}(\cdot)$ and $\tilde{\Theta}(\cdot)$ the variants of the big-O notation that hide polylogarithmic factors.

A Markov Decision Process (MDP) is a tuple $M = (\mathcal{S}, \mathcal{A}, \mathcal{Q})$, where $\mathcal{S}$ is a measurable state space, $\mathcal{A}$ is a finite action space, and $\mathcal{Q} : \mathcal{S}\times\mathcal{A} \to \mathcal{M}_1(\mathcal{S}\times[0,1])$ is the transition-reward kernel. We define the transition and reward distributions $P : \mathcal{S}\times\mathcal{A} \to \mathcal{M}_1(\mathcal{S})$ and $\mathcal{R} : \mathcal{S}\times\mathcal{A} \to \mathcal{M}_1([0,1])$ as the marginals of $\mathcal{Q}$. By a slight abuse of notation, for any $s \in \mathcal{S}$ and $a \in \mathcal{A}$, let $P(\cdot|s,a)$ and $\mathcal{R}(\cdot|s,a)$ denote the distributions $P(s,a)$ and $\mathcal{R}(s,a)$, respectively. We further denote by $r(s,a) = \int_0^1 x \,\mathrm{d}\mathcal{R}(x|s,a)$ the expected reward for an action $a \in \mathcal{A}$ taken in a state $s \in \mathcal{S}$. Without loss of generality, we assume that there is a designated initial state $s_0 \in \mathcal{S}$.

Starting from any state $s \in \mathcal{S}$, a stationary memoryless policy $\pi : \mathcal{S} \to \mathcal{M}_1(\mathcal{A})$ interacts with the MDP in a sequential manner for time-steps $t \in \mathbb{N}$, defining a probability distribution $\mathcal{P}_{\pi,s}$ over the episode trajectory $\{S_i, A_i, R_i\}_{i\in\mathbb{N}}$ as follows: $S_0 = s$ deterministically, $A_i \sim \pi(S_i)$, and $(S_{i+1}, R_i) \sim \mathcal{Q}(S_i, A_i)$. By a slight variation, let $\mathcal{P}_{\pi,s,a}$ denote (for some $a \in \mathcal{A}$) the distribution of the trajectory when $A_0 = a$ deterministically, while the distribution of the rest of the trajectory is defined analogously.

This allows us to conveniently define the expected state-value and action-value functions in the discounted setting we consider, for some discount factor $0 < \gamma < 1$, respectively, as

$$v^\pi(s) = \mathbb{E}_{\pi,s}\left[\sum_{t\in\mathbb{N}} \gamma^t R_t\right] \quad \text{and} \quad q^\pi(s,a) = \mathbb{E}_{\pi,s,a}\left[\sum_{t\in\mathbb{N}} \gamma^t R_t\right] \qquad \text{for all } (s,a) \in \mathcal{S}\times\mathcal{A}, \quad (64)$$

where throughout the chapter we use the convention that $\mathbb{E}_\bullet$ is the expectation operator corresponding to a distribution $\mathcal{P}_\bullet$ (e.g., $\mathbb{E}_{\pi,s}$ is the expectation with respect to $\mathcal{P}_{\pi,s}$). It is well known (see, e.g., Puterman, 1994) that there exists an optimal stationary deterministic memoryless policy $\pi^\star$ such that

$$\sup_\pi v^\pi(s) = v^{\pi^\star}(s) \qquad \text{and} \qquad \sup_\pi q^\pi(s,a) = q^{\pi^\star}(s,a) \qquad \text{for all } (s,a) \in \mathcal{S}\times\mathcal{A}.$$

Let $v^\star = v^{\pi^\star}$ and $q^\star = q^{\pi^\star}$. For any policy $\pi$, $v^\pi$ and $q^\pi$ are known to satisfy the Bellman equations (Puterman, 1994):

$$v^\pi(s) = \sum_{a\in\mathcal{A}} \pi(a|s) q^\pi(s,a) \text{ and } q^\pi(s,a) = r(s,a) + \gamma \int_{s'\in\mathcal{S}} v^\pi(s')\, \mathrm{d}P(s'|s,a) \text{ for all } (s,a)\in\mathcal{S}\times\mathcal{A}. \quad (65)$$

Finally, we call a policy $\pi$ deterministic if for all states, $\pi(s)$ is a distribution that assigns unit weight to one action and zero weight to the others. With a slight abuse of notation, for a deterministic policy $\pi$, we denote by $\pi(s)$ the action $\pi$ chooses (deterministically) in state $s \in \mathcal{S}$.

## 4.3. Confident Approximate Policy Iteration

In this section we introduce Confident Approximate Policy Iteration (CAPI), our new approximate dynamic programming algorithm. In approximate dynamic programming, the methods are designed around oracles that return either an approximation to the application of the Bellman optimality operator to a value function ("approximate value iteration"), or an approximation to the value function of some policy ("approximate policy iteration"). Our setting is the second. The novelty is that we assume access to the accuracy of the approximation and use this knowledge to modify the policy update, which leads to improved guarantees on the suboptimality of the computed policy.

We present the pseudocodes of API (Bertsekas and Tsitsiklis, 1996) and CAPI jointly in Algorithm 4: starting from an arbitrary (deterministic) policy $\pi_0$, the algorithm iterates a policy estimation (Line 2) and a policy update step (Line 3) $I$ times. The policy update for API is greedy with respect to the action-value estimates $\hat{q}$ and is defined as $\pi_{\hat{q}}(s) = \arg\max_{a\in\mathcal{A}} \hat{q}(s,a)$. We assume that $\arg\max_{a\in\mathcal{A}}$ breaks ties in a consistent manner by ordering the actions (using the notation $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_{|\mathcal{A}|})$) and always choosing action $\mathcal{A}_i$ with the lowest index $i$ that achieves the maximum. For CAPI, the policy update further relies on a global estimation-accuracy parameter $\omega$, and a set of fixed-states $\mathcal{S}_{\text{fix}} \subseteq \mathcal{S}$. For the purposes of this section, it is enough to keep $\mathcal{S}_{\text{fix}} = \{\}$.

**Algorithm 4** APPROXIMATE POLICY ITERATION (API) and CONFIDENT APPROXIMATE POLICY ITERATION (CAPI)

**for** $i = 1$ to $I$ **do**
  $\hat{q} \leftarrow \text{ESTIMATE}(\pi_{i-1})$
  $\pi_i \leftarrow \begin{cases} \pi_{\hat{q}} & \text{API} \\ \pi_{\hat{q},\pi_{i-1},\mathcal{S}_{\text{fix}}} & \text{CAPI} \end{cases}$
**end for**
**return** $\pi_I$

CAPI updates the policy to one that acts greedily with respect to $\hat{q}$ *only* on states that are not in $\mathcal{S}_{\text{fix}}$ and where it is confident that this leads to an improvement over the previous policy (Case 66a); otherwise, the new policy will return the same action as the previous one (Case 66b). To decide, $\hat{q}(s,\pi(s))+\omega$ is treated as the upper bound on the previous policy's value, and $\max_{a\in\mathcal{A}}\hat{q}(s,a)-\omega$ as the lower bound of the action-value of the greedy action (Eq. 66):

$$\pi_{\hat{q},\pi,\mathcal{S}_{\text{fix}}}(s) = \begin{cases} \arg\max_{a\in\mathcal{A}} \hat{q}(s,a), & \text{if } s \notin \mathcal{S}_{\text{fix}} \text{ and } \hat{q}(s,\pi(s))+\omega < \max_{a\in\mathcal{A}}\hat{q}(s,a)-\omega\,; \quad (66a) \\ \pi(s), & \text{otherwise.} \quad (66b) \end{cases}$$

Note that $\pi_{\hat{q},\pi,\mathcal{S}_{\text{fix}}}$ also depends on $\omega$, however, this dependence is omitted from the notation (as $\omega$ is kept fixed throughout).

CAPI can also be seen as a refinement of CONSERVATIVE POLICY ITERATION (CPI) of Kakade and Langford (2002) with some important differences: While CPI introduces a global parameter to ensure the update stays close to the previous policy, CAPI has no such parameter, and it dynamically decides when to stay close to (more precisely, use) the previous policy, individually for every state, based on whether there is evidence for a guaranteed improvement.

Let $\pi$ be any stationary deterministic memoryless policy, $\hat{q}^\pi : \mathcal{S}\times\mathcal{A}\to\mathbb{R}$ be any function, $\omega\in\mathbb{R}_+$, and $\mathcal{S}_{\text{fix}}\subseteq\mathcal{S}$. First, we show that as long as $\hat{q}^\pi$ is an $\omega$-accurate estimate of $q^\pi$, the CAPI policy update only improves the policy's values:

**Lemma 4.3.1** (No deterioration)**.** *Let $\pi' = \pi_{\hat{q}^\pi,\pi,\mathcal{S}_{fix}}$. Assume that for all $s\in\mathcal{S}\setminus\mathcal{S}_{fix}$ and $a\in\mathcal{A}$, $\hat{q}^\pi(s,a)\approx_\omega q^\pi(s,a)$. Then, for any $s\in\mathcal{S}$, $v^{\pi'}(s)\geq v^\pi(s)$.*

*Proof.* Fix any $s\in\mathcal{S}$. If $s\in\mathcal{S}_{\text{fix}}$ or $\hat{q}^\pi(s,\pi(s))+\omega\geq\max_{a\in\mathcal{A}}\hat{q}^\pi(s,a)-\omega$, then $\pi'(s)=\pi(s)$ and therefore $q^\pi(s,\pi'(s))=v^\pi(s)$. Otherwise, $s\notin\mathcal{S}_{\text{fix}}$ and $\hat{q}^\pi(s,\pi(s))+\omega\leq\max_{a\in\mathcal{A}}\hat{q}^\pi(s,a)-\omega$, hence $\pi'(s)=\arg\max_{a\in\mathcal{A}}\hat{q}^\pi(s,a)$, and it follows by our assumptions that $q^\pi(s,\pi'(s))\geq\hat{q}^\pi(s,\pi'(s))-\omega=\max_{a\in\mathcal{A}}\hat{q}^\pi(s,a)-\omega>\hat{q}^\pi(s,\pi(s))+\omega\geq q^\pi(s,\pi(s))=v^\pi(s)$. Therefore, in any case, $q^\pi(s,\pi'(s))\geq v^\pi(s)$. Since this holds for any $s\in\mathcal{S}$, the Policy Improvement Theorem (Sutton and Barto, 2018, Section 4.2) implies that for any $s\in\mathcal{S}$, $v^{\pi'}(s)\geq v^\pi(s)$. ■

Next we introduce two approximate optimality criterion for a policy on a set of states:

**Definition 4.3.2** (Policy optimality on a set of states)**.** *A policy $\pi$ is $\Delta$-optimal (for some $\Delta \geq 0$) on a set of states $\mathcal{S}' \subseteq \mathcal{S}$, if for all $s \in \mathcal{S}'$, $v^\star(s) - v^\pi(s) \leq \Delta$.*

**Definition 4.3.3** (Next-state optimality on a set of states)**.** *A policy $\pi$ is next-state $\Delta$-optimal (for some $\Delta \geq 0$) on a set of states $\mathcal{S}' \subseteq \mathcal{S}$, if for all $s \in \mathcal{S}'$ and all actions $a \in \mathcal{A}$, $\int_{s' \in \mathcal{S}} (v^\star(s') - v^\pi(s'))\, \mathrm{d}P(s'|s,a) \leq \Delta$.*

Note that in the special case of $\mathcal{S}' = \mathcal{S}$ the first property implies the second, that is, if $\pi$ is $\Delta$-optimal on $\mathcal{S}$, then it is also next-state $\Delta$-optimal on $\mathcal{S}$. Next, we show that the suboptimality of a policy updated by CAPI evolves as follows (the proof is relegated to Section 4.A):

**Lemma 4.3.4** (Iteration progress)**.** *Let $\pi' = \pi_{\hat{q}^\pi, \pi, \mathcal{S}_{fix}}$. Assume that for all $s \in \mathcal{S} \setminus \mathcal{S}_{fix}$ and $a \in \mathcal{A}$, $\hat{q}^\pi(s,a) \approx_\omega q^\pi(s,a)$, and that $\pi$ is next-state $\Delta$-optimal on $\mathcal{S} \setminus \mathcal{S}_{fix}$. Then $\pi'$ is $(4\omega + \gamma\Delta)$-optimal on $\mathcal{S} \setminus \mathcal{S}_{fix}$.*

### 4.3.1. CAPI guarantee with accurate estimation everywhere

To obtain a final suboptimality guarantee for CAPI, first consider the ideal scenario in which we assume that we have a mechanism to estimate $q^\pi(s,a)$ up to some $\omega$ accuracy for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$, and for any policy $\pi$:

**Assumption 4.3.5.** *There is an oracle called* ESTIMATE *that accepts a policy $\pi$ and returns $\hat{q}^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ such that for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$, $\hat{q}^\pi(s,a) \approx_\omega q^\pi(s,a)$.*

**Theorem 4.3.6** (CAPI performance)**.** *Assume* CAPI *(Algorithm 4) is run with $\mathcal{S}_{fix} = \{\}$, iteration count to $I = \lceil \log \omega / \log \gamma \rceil$, and suppose that the estimation used in Line 2 satisfies Assumption 4.3.5. Then the policy $\pi_I$ returned by the algorithm is $5\omega/(1-\gamma)$-optimal on $\mathcal{S}$.*

*Proof.* We prove by induction that policy $\pi_i$ is $\Delta_i$-optimal on $\mathcal{S}$ for $\Delta_i = 4\omega \sum_{j \in [i]} \gamma^j + \frac{\gamma^i}{1-\gamma}$. This holds immediately for the base case of $i = 0$, as rewards are bounded in $[0,1]$ and thus $v^\star(s) \leq 1/(1-\gamma)$ for any $s$. Assuming now that the inductive hypothesis holds for $i-1$ we observe that $\pi_{i-1}$ is next-state $\Delta$-optimal on $\mathcal{S} = \mathcal{S} \setminus \mathcal{S}_{\text{fix}}$. Together with Assumption 4.3.5, this implies that the conditions of Lemma 4.3.4 are satisfied for $\pi = \pi_{i-1}$, which yields $v^\star(s) - v^{\pi_i}(s) \leq 4\omega + \gamma\Delta_{i-1} = \Delta_i$, finishing the induction. Finally, by the definition of $I$, $\pi_I$ is $\Delta_I$-optimal with $\Delta_I \leq \frac{4\omega}{1-\gamma} + \frac{\gamma^I}{1-\gamma} \leq \frac{5\omega}{1-\gamma}$. ∎

## 4.4. Local access planning with $q^\pi$-realizability

Our planner, CAPI-QPI-PLAN, is based on the CONFIDENT MC-LSPI algorithm of Yin et al. (2022). This latter algorithm gradually builds a *core set* of state-action pairs whose corresponding features are informative. The $q$-values of the state-action pairs in the core set are estimated using rollouts. The procedure is restarted with an extended core set whenever the algorithm encounters a new informative feature. If such a new feature is not encountered, the estimation error can be controlled, and the estimation is extended to all state-action pairs using the least-squares estimator. Finally, the extended estimation is used in Line 2 of API.

CAPI-QPI-PLAN improves upon CONFIDENT MC-LSPI in two ways. First, using CAPI instead of API improves the final suboptimality bound by a factor of the effective horizon. Second, we apply a novel analysis on a more modular variant of the CONFIDENTROLLOUT subroutine used in CONFIDENT MC-LSPI, which delivers $q$-estimation accuracy guarantees with respect to a large class of policies simultaneously. This allows for a dynamically evolving version of policy iteration, that does not have to restart whenever a new informative feature is encountered. Intuitively, this prevents duplication of work.

### 4.4.1. Estimation oracle

To obtain an algorithm for planning with local access whose performance degrades gracefully with the uniform approximation error, we must weaken Assumption 4.3.5. This is because under local access, we cannot guarantee to cover all states or hope to obtain accurate $q$-value estimates for all states. Instead, we are interested in an accuracy guarantee that holds for $q$-values only on some subset $\mathcal{S}' \subseteq \mathcal{S}$ of states, but holds simultaneously for *any* policy that agrees with $\pi$ on $\mathcal{S}'$ but may take arbitrary values elsewhere. For this, we define the extended set of policies:

**Definition 4.4.1.** *Let $\Pi_{det}$ be the set of all stationary deterministic memoryless policies, $\pi \in \Pi_{det}$, and $\mathcal{S}' \subseteq \mathcal{S}$. For $(\pi, \mathcal{S}')$, we define $\Pi_{\pi,\mathcal{S}'}$ to be the set of policies that agree with $\pi$ on $s \in \mathcal{S}'$:*

$$\Pi_{\pi,\mathcal{S}'} = \{\pi' \in \Pi_{det} : \pi(s) = \pi'(s) \text{ for all } s \in \mathcal{S}'\} .$$

We aim to first accurately estimate $q^\pi(s,a)$ for *some specific* $(s,a)$ pairs, based on which we extend the estimates to other state-action pairs using least-squares. To this end, we first devise a subroutine called MEASURE (Algorithm 5). MEASURE is a modularized variant of the CONFIDENTROLLOUT subroutine of Yin et al. (2022). The modularity of our variant is due to the parameter $\mathcal{S}'$ that corresponds to the set of states on which the planner is confident for CONFIDENTROLLOUT.

**Algorithm 5** MEASURE

**Input:** state $s$, action $a$, deterministic policy $\pi$, set of states $\mathcal{S}' \subseteq \mathcal{S}$, accuracy $\omega > 0$, failure probability $\zeta \in (0,1]$
**Initialize:** $H \leftarrow \lceil \log((\omega/4)(1-\gamma))/\log\gamma \rceil$, $n \leftarrow \lceil (\omega/4)^{-2}(1-\gamma)^{-2}\log(2/\zeta)/2 \rceil$
**for** $i = 1$ to $n$ **do**
  $(S, R_{i,0}) \leftarrow$ SIMULATOR$(s,a)$ ▷ Call to the simulator oracle
  **for** $h = 1$ to $H-1$ **do**
    **if** $S \notin \mathcal{S}'$ **then return** (discover, $S$)
    **end if**
    $A \leftarrow \pi(S)$
    $(S, R_{i,h}) \leftarrow$ SIMULATOR$(S,A)$
  **end for**
**end for**
**return** (success, $\frac{1}{n}\sum_{i=1}^{n}\sum_{h=0}^{H-1}\gamma^h R_{i,h}$)

MEASURE unrolls the policy $\pi$ starting from $(s,a)$ for a number of episodes, each lasting $H$ steps, and returns with the average measured reward. Throughout, we let $H = \lceil \log((\omega/4)(1-\gamma))/\log\gamma \rceil$ be the effective horizon. At the end of this process, MEASURE returns status *success* along with the empirical average $q$-value, where compared to Eq. (64), the discounted summation of rewards is truncated to $H$. If, however, the algorithm encounters a state not in its input $\mathcal{S}'$, it returns with status *discover*, along with that state. This is because in such cases, the algorithm could no longer guarantee an accurate estimation with respect to any member of the extended set of policies. The next lemma, proved in Section 4.B, shows that MEASURE provides accurate estimates of the action-value functions for members of the extended policy set.

**Lemma 4.4.2.** *For any input parameters $s \in \mathcal{S}, a \in \mathcal{A}, \pi \in \Pi_{det}, \mathcal{S}' \subset \mathcal{S}, \omega > 0, \zeta \in (0,1)$,* MEASURE *either returns with $(\mathit{discover}, s')$ for some $s' \notin \mathcal{S}'$ (Line 6), or it returns with $(\mathit{success}, \tilde{q})$ such that with probability at least $1-\zeta$,*

$$q^{\pi'}(s,a) \approx_\omega \tilde{q} \quad \textit{for all} \quad \pi' \in \Pi_{\pi,\mathcal{S}'}. \tag{67}$$

Suppose we have a list of state-action pairs $C = (s_i, a_i)_{i \in [|C|]}$ and corresponding $q$-estimates $\bar{q} = (\bar{q}_i)_{i \in |C|}$. We use the regularized least-squares estimator LSE (Eq. 69) to extend the estimates for all state-action pairs, with regularization parameter $\lambda = \omega^2/B^2$ (recall that $B$ is defined in Definition 4.1.1):

$$V(C) = \lambda \mathbf{I} + \textstyle\sum_{i \in [|C|]} \varphi(s_i,a_i)\varphi(s_i,a_i)^\top, \tag{68}$$

$$\mathrm{LSE}_{C,\bar{q}}(s,a) = \left\langle \varphi(s,a), V(C)^{-1} \textstyle\sum_{i \in [|C|]} \varphi(s_i,a_i)\bar{q}_i \right\rangle. \tag{69}$$

For $C = \bar{q} = ()$ (the empty sequence), we define $\mathrm{LSE}_{C,\bar{q}}(\cdot,\cdot) = 0$. This estimator satisfies the guarantee below.

**Lemma 4.4.3.** *Let $\pi$ be a stationary deterministic memoryless policy. Let $C = (s_i, a_i)_{i\in[n]}$ be sequences of state-action pairs of some length $n \in \mathbb{N}$ and $\bar{q} = (\bar{q}_i)_{i\in[n]}$ a sequence of corresponding reals such that for all $i \in [n]$, $q^\pi(s_i, a_i) \approx_\omega \bar{q}_i$. Then, for all $s, a \in \mathcal{S}\times\mathcal{A}$,*

$$\left|LSE_{C,\bar{q}}(s,a) - q^\pi(s,a)\right| \le \varepsilon + \|\varphi(s,a)\|_{V(C)^{-1}} \left(\sqrt{\lambda}B + (\omega+\varepsilon)\sqrt{n}\right), \tag{70}$$

*where $\varepsilon$ is the uniform approximation error from Definition 4.1.1.*

The proof is given in Section 4.C. The order of the estimation accuracy bound (Eq. 70) is optimal, as shown by the lower bounds of Du et al. (2019a) and Lattimore et al. (2020).

We intend to use the LSE estimator given in Eq. (69) and the bound in Lemma 4.4.3 only for state-action pairs where $\|\varphi(s,a)\|_{V(C)^{-1}} \le 1$ (and $n = \tilde{\mathcal{O}}(d)$). We call these state-action pairs *covered* by $C$, and we call a state $s$ covered by $C$ if for all their corresponding actions $a$, the pair $(s,a)$ is covered by $C$:

$$\mathrm{ActionCover}(C) = \{(s,a) \in \mathcal{S}\times\mathcal{A} : \|\varphi(s,a)\|_{V(C)^{-1}} \le 1\} \tag{71}$$

$$\mathrm{Cover}(C) = \{s \in \mathcal{S} : \forall a \in \mathcal{A}, (s,a) \in \mathrm{ActionCover}(C)\}. \tag{72}$$

We will use the parameter $\mathcal{S}_{\mathrm{fix}}$ of CAPI (see CAPI's update rule in Eq. 66) to ensure policies are only updated on covered states, where the approximation error is well-controlled by Eq. (70).

### 4.4.2. Main algorithm

Finally, we are ready to introduce CAPI-QPI-PLAN, presented in Algorithm 6, our algorithm for planning with local access under approximate $q^\pi$-realizability. For this, we define levels $l = 0, 1, \ldots, H$, and corresponding suboptimality requirements: For any $l \in [H+1]$, let

$$\Delta_l = 8(\varepsilon+\omega)\left(\sqrt{\tilde{d}}+1\right)\sum_{j\in[l]}\gamma^j + \frac{\gamma^l}{1-\gamma},$$

for some $\tilde{d} = \tilde{\Theta}(d)$ defined in Eq. (74). For each level $l$, the algorithm maintains a policy $\pi_l$ and a set of covered states on which it can guarantee that $\pi_l$ is a $\Delta_l$-optimal policy. More specifically, this set is $\mathrm{Cover}(C_l)$, where $C_l$ is a list of state-action pairs with elements $C_{l,i} = (s_l^i, a_l^i)$ for $i \in [|C_l|]$. The

algorithm maintains the following suboptimality guarantee below, which we prove in Section 4.E after showing some further key properties of the algorithm.

**Lemma 4.4.4.** *Assuming that Eq. (67) holds whenever* MEASURE *returns* success*, $\pi_l$ is $\Delta_l$-optimal on Cover$(C_l)$ (Definition 4.3.2) for all $l \in [H+1]$ at the end of every iteration of the main loop of* CAPI-QPI-PLAN.

CAPI-QPI-PLAN aims to improve the policies, while *propagating* the members of $C_l$ to $C_{l+1}$, and so on, all the way to $C_H$. During this, whenever the algorithm discovers a state-action pair with a sufficiently new feature direction, this pair is appended to the sequence $C_0$ corresponding to level 0, as there are no suboptimality guarantees yet available for such a state. However, such a discovery can only happen $\tilde{\mathcal{O}}(d)$ times. When, eventually, all discovered state-action pairs end up in $C_H$, the final suboptimality guarantee is reached, and the algorithm returns with the final policy. Note that in the local access setting we consider, the algorithm cannot enumerate the set Cover$(C_l)$, but can answer membership queries, that is, for any $s \in \mathcal{S}$ it encounters, it is able to decide if $s \in$ Cover$(C_l)$. The algorithm maintains sequences $\bar{q}_l$, corresponding to $C_l$, for each level $l$. Whenever a new $(s,a)$ pair is appended to the sequence $C_l$, a corresponding $\perp$ symbol is appended to the sequence $\bar{q}_l$, to signal that an estimate of $q^{\pi_l}(s,a)$ is not yet known.

After initializing $C_0$ to cover the initial state $s_0$ (Lines 4 to 6), the algorithm measures $q^{\pi_\ell}(s,a)$ for the smallest level $\ell$ for which there still exists a $\perp$ in the corresponding $\bar{q}_\ell$. After a successful measurement, if there are no more $\perp$'s left at this level (i.e., in $\bar{q}_\ell$), the algorithm executes a policy update on $\pi_\ell$ (Line 20) using the least-squares estimate obtained from the measurements at this level, but only for states in Cover$(C_\ell)$ (using $\mathcal{S}_{\text{fix}} = \mathcal{S} \setminus \text{Cover}(C_\ell)$). Next, Line 21 merges this new policy $\pi'$ with the existing policy $\pi_{\ell+1}$ of the next level, setting $\pi_{\ell+1}$ to be the policy $\pi''$ defined as

$$\pi''(s) = \begin{cases} \pi_{\ell+1}(s), & \text{if } s \in \text{Cover}(C_{\ell+1}); \\ \pi'(s), & \text{otherwise.} \end{cases}$$

This ensures that the existing policy $\pi_{\ell+1}$ remains unchanged by $\pi''$ (its replacement) on states that are already covered by $C_{\ell+1}$, and therefore $\pi'' \in \Pi_{\pi_{\ell+1},\text{Cover}(C_{\ell+1})} = \Pi_{\pi'',\text{Cover}(C_{\ell+1})}$. We also observe that $C_l$ can only grow for any $l$ (elements are never removed from these sequences), thus for any update where $C_l$ is assigned a new value $C_l'$ (Lines 5, 13, and 23), $V(C_l') \geq V(C_l)$, and therefore Cover$(C_l') \supseteq$ Cover$(C_l)$ and $\Pi_{\pi_l,\text{Cover}(C_l')} \subseteq \Pi_{\pi_l,\text{Cover}(C_l)}$. Combining these properties yields the following result:

**Algorithm 6** CAPI-QPI-PLAN

1: **Input:** initial state $s_0 \in \mathcal{S}$, dimensionality $d$, parameter bound $B$, accuracy $\omega$, failure probability $\delta > 0$
2: **Initialize:** $H \leftarrow \lceil \log((\omega/4)(1-\gamma))/\log\gamma \rceil$, for $l \in [H+1]$, $C_l \leftarrow ()$, $\bar{q}_l \leftarrow ()$, $\pi_l \leftarrow$ policy that always returns action $\mathcal{A}_1$, $\lambda \leftarrow \omega^2/B^2$
3: **while** True **do** ▷ main loop
4: **if** $\exists a \in \mathcal{A}$, $(s_0, a) \notin \text{ActionCover}(C_0)$ **then**
5: append $(s_0, a)$ to $C_0$, append $\perp$ to $\bar{q}_0$
6: **break**
7: **end if**
8: let $\ell$ be the smallest integer such that $\bar{q}_{\ell,m} = \perp$ for some $m$; set $\ell = H$ if no such $l$ exists
9: **if** $\ell = H$ **then return** $\pi_H$
10: **end if**
11: $(\text{status}, \text{result}) \leftarrow \text{MEASURE}(s^m_\ell, a^m_\ell, \pi_\ell, \text{Cover}(C_\ell), \omega, \delta/(\tilde{d}H))$ ▷ recall $C_{\ell,m} = (s^m_\ell, a^m_\ell)$
12: **if** status = discover **then**
13: append $(\text{result}, a)$ to $C_0$ for some $a$ such that $(\text{result}, a) \notin \text{ActionCover}(C_0)$
14: append $\perp$ to $\bar{q}_0$
15: **break**
16: **end if**
17: $\bar{q}_{\ell,m} \leftarrow \text{result}$
18: **if** $\nexists m'$ such that $\bar{q}_{\ell,m'} = \perp$ **then**
19: $\hat{q} \leftarrow \text{LSE}_{C_\ell, \bar{q}_\ell}$
20: $\pi' \leftarrow \pi_{\hat{q}, \pi_\ell, \mathcal{S} \setminus \text{Cover}(C_\ell)}$
21: $\pi_{\ell+1} \leftarrow (s \mapsto \pi_{\ell+1}(s) \text{ if } s \in \text{Cover}(C_{\ell+1}) \text{ else } \pi'(s))$
22: **for** $(s, a) \in C_\ell$ such that $(s, a) \notin C_{\ell+1}$ **do**
23: append $(s, a)$ to $C_{\ell+1}$, $\perp$ to $\bar{q}_{\ell+1}$
24: **end for**
25: **end if**
26: **end while**

**Lemma 4.4.5.** *If for any $l \in [H]$, $\pi_l$ and $C_l$ take some values $\pi_l^{old}$ and $C_l^{old}$ at any point in the execution of the algorithm, then at any later point during the execution, $\pi_l \in \Pi_{\pi_l, Cover(C_l)} \subseteq \Pi_{\pi_l^{old}, Cover(C_l^{old})}$.*

Any value in $\bar{q}_l$ that is set to anything other than $\perp$ will never change again. Since as long as the sample paths generated by MEASURE in Line 11 of CAPI-QPI-PLAN remain in $\text{Cover}(C_l)$, their distribution is the same under any policy from $\Pi_{\pi_l, \text{Cover}(C_l)}$, the $\bar{q}_l$ estimates are valid for these policies, as well. Combined with Lemma 4.4.5, we get that the accuracy guarantees of Lemma 4.4.2 continue to hold throughout:

**Lemma 4.4.6.** *Assuming that Eq. (67) holds whenever* MEASURE *returns* success*, for any level $l$ and index $m$ such that $\bar{q}_{l,m} \neq \perp$, $q^{\pi'}(s^m_l, a^m_l) \approx_\omega \bar{q}_{l,m}$ for all $\pi' \in \Pi_{\pi_l, Cover(C_l)}$ throughout the execution of* CAPI-QPI-PLAN.

Once $\pi_{\ell+1}$ is updated in Line 21, in Line 23 we append to the sequence $C_{\ell+1}$ all members of $C_\ell$ that are not yet in $C_{\ell+1}$, while adding a corresponding $\perp$ to $\bar{q}_{\ell+1}$ indicating that these $q$-values are not yet measured for policy $\pi_{\ell+1}$. Thus, whenever all $\perp$ values disappear from some level $l \in [H+1]$, by the end of that iteration $C_{l+1} = C_l$, and hence $\mathrm{ActionCover}(C_l) = \mathrm{ActionCover}(C_{l+1})$. Together with the fact that for any $l \in [H+1]$, whenever a new state-action pair is appended to $C_l$, an $\perp$ symbol is appended to $\bar{q}_l$, we have by induction the following result:

**Lemma 4.4.7.** *Throughout the execution of* CAPI-QPI-PLAN*,after Line 8 when $\ell$ is set,*

$$\mathit{ActionCover}(C_0) = \mathit{ActionCover}(C_1) = \cdots = \mathit{ActionCover}(C_\ell)\,.$$

As a result, whenever the MEASURE call of Line 11 outputs $(\text{discover}, s)$ for some state $s$, by Lemma 4.4.2, there is an action $a \in \mathcal{A}$ such that $(s,a) \notin \mathrm{ActionCover}(C_\ell) = \mathrm{ActionCover}(C_0)$. This explains why adding such an $(s,a)$ pair to $C_0$ is always possible in Line 13. Consider the $i^{\text{th}}$ time Line 13 is executed, and denote $s$ by $s_i$ and $a$ by $a_i$, and $V_i = \lambda \mathbf{I} + \sum_{t=1}^{i-1} \varphi(s_t,a_t)\varphi(s_t,a_t)^\top$. Observe that as $V_i = V(C)$, $(s_i,a_i) \notin \mathrm{ActionCover}(C_0)$ implies $\|\varphi(s_i,a_i)\|_{V_i^{-1}} > 1$. Therefore, $\sum_{t=1}^{i} \min\{1, \|\varphi(s_t,a_t)\|_{V_t^{-1}}\} = i$, and thus by the elliptical potential lemma (Lattimore and Szepesvári, 2020, Lemma 19.4), $i \le 2d \log\left(\frac{d\lambda + iL^2}{d\lambda}\right)$. This inequality is satisfied by the largest value of $i$, that is, the total number of times MEASURE returns with *discover*. Since any element of $C_l$ is also an element of $C_0$ for any $l \in [H+1]$, we have that at any time during the execution of CAPI-QPI-PLAN,

$$|C_l| \le 4d \log\left(1 + \frac{4L^2}{\lambda}\right) =: \tilde{d} = \tilde{\mathcal{O}}(d)\,. \tag{74}$$

When CAPI-QPI-PLAN returns at Line 9 with the policy $\pi_H$, it is $\Delta_H$-optimal on $\mathrm{Cover}(C_H)$ by Lemma 4.4.4 when the estimates of MEASURE are correct. Furthermore, $s_0 \in \mathrm{Cover}(C_0)$ is guaranteed by Lines 4 to 6, and hence $s_0 \in \mathrm{Cover}(C_H)$ by Lemma 4.4.7 when the algorithm finishes. Hence, bounding $\Delta_H$ using the definition of $H$ immediately gives the following result:

**Lemma 4.4.8.** *Assuming that Eq. (67) holds whenever* MEASURE *returns* success*, the policy $\pi$ returned by* CAPI-QPI-PLAN *is $\Delta$-optimal on $\{s_0\}$ for*

$$\Delta = 9(\varepsilon+\omega)\left(\sqrt{\tilde{d}}+1\right)(1-\gamma)^{-1} = \tilde{\mathcal{O}}\left((\varepsilon+\omega)\sqrt{d}(1-\gamma)^{-1}\right).$$

To finish the proof of Theorem 4.1.2, we only need to analyze the query complexity and the failure probability (i.e., the probability of Eq. (67) not being satisfied for some MEASURE call that returns *success*) of CAPI-QPI-PLAN:

*Proof of Theorem 4.1.2.* Both the total failure probability and query complexity of CAPI-QPI-PLAN depend on the number of times MEASURE is executed, as this is the only source of randomness and of interaction with the simulator. MEASURE can return *discover* at most $|C_0|$ times, which is bounded by $\tilde{d}$ by Eq. (74). For every $l \in [H]$, MEASURE is executed exactly once with returning *success* for each element of $C_l$. Hence, by Eq. (74) again, MEASURE returns *success* at most $\tilde{d}H$ times, each satisfying Eq. (67) with probability at least $1-\zeta = 1-\delta/(\tilde{d}H)$ by Lemma 4.4.2. By the union bound, MEASURE returns *success* in all occasions with probability at least $1-\delta$. Hence Eq. (67) holds with probability at least $1-\delta$, which, combined with Lemma 4.4.8, proves Eq. (62).

Each successful run of MEASURE executes at most $nH$ queries ($n$ is set in Line 2 of Algorithm 5). Since $H < (1-\gamma)^{-1}\log(4\omega^{-1}(1-\gamma)^{-1}) = \tilde{\mathcal{O}}((1-\gamma)^{-1})$, in total CAPI-QPI-PLAN executes at most $\tilde{\mathcal{O}}\left(d(1-\gamma)^{-4}\omega^{-2}\right)$ queries. As this happens at most $\tilde{d}H$ times, we obtain the desired bound on the query complexity. ■

## 4.5. Conclusions and future work

In this chapter we presented CONFIDENT APPROXIMATE POLICY ITERATION, a confident version of API, which can obtain a stationary policy with a suboptimality guarantee that scales linearly with the effective horizon $H = \tilde{\mathcal{O}}(1/(1-\gamma))$. This scaling is optimal as shown by Scherrer and Lesner (2012).

CAPI can be applied to local planning with approximate $q^\pi$-realizability (yielding the CAPI-QPI-PLAN algorithm) to obtain a sequence of policies with successively refined accuracies on a dynamically evolving set of states, resulting in a final, recursively defined policy achieving simultaneously the optimal suboptimality guarantee and best query cost available in the literature. More precisely, CAPI-QPI-PLAN achieves $\tilde{\mathcal{O}}(\varepsilon\sqrt{d}H)$ suboptimality, where $\varepsilon$ is the uniform policy value-function approximation error. We showed that this bound is the best (up to polylogarithmic factors) that is achievable by any planner with polynomial query cost. We also proved that the $\tilde{\mathcal{O}}\left(dH^4\varepsilon^{-2}\right)$ query cost of CAPI-QPI-PLAN is optimal up to polylogarithmic factors in all parameters except for $H$; whether the dependence on $H$ is optimal remains an open question.

Finally, our method comes at a memory and computational cost overhead, both for the final policy and the planner. It is an interesting question if this overhead necessarily comes with the API-style method we use (as it is also present in the works of Scherrer and Lesner, 2012; Scherrer,

2014), or if it is possible to reduce it by, for example, compressing the final policy into one that is greedy with respect to some action-value function realized with the features.

# Appendix

## 4.A. Proof of Lemma 4.3.4

Take any $s \in \mathcal{S} \setminus \mathcal{S}_{\text{fix}}$.

$$\begin{aligned} v^\star(s) - v^{\pi'}(s) &= v^\star(s) - q^{\pi'}(s, \pi'(s)) \\ &= v^\star(s) - q^\pi(s, \pi'(s)) + q^\pi(s, \pi'(s)) - q^{\pi'}(s, \pi'(s)) \\ &\leq v^\star(s) - q^\pi(s, \pi'(s)), \end{aligned} \tag{75}$$

where the first equality holds because $\pi'$ is deterministic, and the inequality is true because

$$q^\pi(s, \pi'(s)) - q^{\pi'}(s, \pi'(s)) = \gamma \int_{s' \in \mathcal{S}} \left( v^\pi(s') - v^{\pi'}(s') \right) \mathrm{d}P(s' | s, \pi'(s)) \leq 0$$

by Lemma 4.3.1. Next observe that

$$\hat{q}^\pi(s, \pi'(s)) \geq \max_{a \in \mathcal{A}} \hat{q}^\pi(s, a) - 2\omega \tag{76}$$

since, as $s \notin \mathcal{S}_{\text{fix}}$, either $\pi'(s)$ is defined by Case 66a as $\pi'(s) = \arg\max_{a \in \mathcal{A}} \hat{q}^\pi(s, a)$ and so $\hat{q}^\pi(s, \pi'(s)) = \max_{a \in \mathcal{A}} \hat{q}^\pi(s, a)$, or it is defined by Case 66b in which case $\hat{q}^\pi(s, \pi'(s)) = \hat{q}^\pi(s, \pi(s)) \geq \max_{a \in \mathcal{A}} \hat{q}^\pi(s, a) - 2\omega$. Combining Eqs. (75) and (76), we obtain

$$\begin{aligned} v^\star(s) - v^{\pi'}(s) &\leq v^\star(s) - \hat{q}^\pi(s, \pi'(s)) + \hat{q}^\pi(s, \pi'(s)) - q^\pi(s, \pi'(s)) \\ &\leq v^\star(s) - \hat{q}^\pi(s, \pi'(s)) + \omega \\ &\leq v^\star(s) - \max_{a \in \mathcal{A}} \hat{q}^\pi(s, a) + 3\omega, \end{aligned}$$

where in the first line we added and subtracted $\hat{q}^\pi(s, \pi'(s))$, and the second inequality holds as $\hat{q}^\pi(s, a) \approx_\omega q^\pi(s, a)$ for $s \notin \mathcal{S}_{\text{fix}}$ and $a \in \mathcal{A}$ by the assumptions of the lemma.

We continue by adding and subtracting $\max_{a\in\mathcal{A}} q^\pi(s,a)$:

$$\begin{aligned}
v^\star(s) - v^{\pi'}(s) &\le v^\star(s) - \max_{a\in\mathcal{A}} q^\pi(s,a) + \max_{a\in\mathcal{A}} q^\pi(s,a) - \max_{a\in\mathcal{A}} \hat{q}^\pi(s,a) + 3\omega \\
&\le v^\star(s) - \max_{a\in\mathcal{A}} q^\pi(s,a) + 4\omega \\
&= \max_{a\in\mathcal{A}} \left[ r(s,a) + \gamma \int_{s'\in\mathcal{S}} v^\star(s')\, \mathrm{d}P(s'|s,a) \right] \\
&\qquad - \max_{a\in\mathcal{A}} \left[ r(s,a) + \gamma \int_{s'\in\mathcal{S}} v^\pi(s')\, \mathrm{d}P(s'|s,a) \right] + 4\omega \\
&\le \max_{a\in\mathcal{A}} \left[ \gamma \int_{s'\in\mathcal{S}} \big(v^\star(s') - v^\pi(s')\big)\, \mathrm{d}P(s'|s,a) \right] + 4\omega \\
&\le 4\omega + \gamma\Delta,
\end{aligned}$$

where in the fifth line we used that $\pi$ is next-state $\Delta$-optimal by assumption. ■

## 4.B. Proof of Lemma 4.4.2

For an episode trajectory $\{S_h, A_h, R_h\}_{h\in\mathbb{N}}$, let $K$ be the smallest positive integer such that $S_K \notin \mathcal{S}'$. For any $i \in \{1,\dots,n\}$, let $I_i$ denote the indicator of the event that at the $i^{\text{th}}$ iteration of the outer loop of Algorithm 5, the algorithm encounters $S \notin \mathcal{S}'$ in Line 6. Note that $\mathbb{E}_{\pi,s,a}[I_i] = \mathcal{P}_{\pi,s,a}[1 \le K < H]$. Then, by Hoeffding's inequality (see, e.g., Lattimore and Szepesvári (2020)), with probability at least $1-\zeta/2$,

$$\left| \mathcal{P}_{\pi,s,a}[1 \le K < H] - \frac{1}{n}\sum_{i=1}^{n} I_i \right| \le \frac{\omega(1-\gamma)}{4}\,.$$

MEASURE only returns *success* if all indicators are zero; therefore, the above inequality implies that if MEASURE returns *success* then, with probability at least $1-\zeta/2$, we have

$$\mathcal{P}_{\pi,s,a}[1 \le K < H] \le \frac{\omega(1-\gamma)}{4}\,. \tag{77}$$

Recall that if MEASURE returns $(\text{success}, \tilde{q})$, then $\bar{q} = \frac{1}{n}\sum_{i=1}^{n}\sum_{h=0}^{H-1}\gamma^h R_{i,h}$. Since

$$0 \le q^\pi(s,a) - \mathbb{E}_{\pi,s,a}\sum_{h=0}^{H-1}\gamma^h R_h = \mathbb{E}_{\pi,s,a}\sum_{h=H}^{\infty}\gamma^h R_h \le \frac{\gamma^H}{1-\gamma} \le \frac{\omega}{4}\,,$$

another application of Hoeffding's inequality yields that $q^\pi(s,a)$ and $\bar{q}$ are close with high probability: with probability at least $1-\zeta/2$,

$$\begin{aligned}|q^\pi(s,a)-\bar{q}| &= \left|q^\pi(s,a)-\frac{1}{n}\sum_{i=1}^{n}\sum_{h=0}^{H-1}\gamma^h R_{i,h}\right| \\ &\le \omega/4+\left|\mathbb{E}_{\pi,s,a}\sum_{h=0}^{H-1}\gamma^h R_h-\frac{1}{n}\sum_{i=1}^{n}\sum_{h=0}^{H-1}\gamma^h R_{i,h}\right| \le \omega/2\,, \end{aligned} \tag{78}$$

where we also used that the range of the sum of the rewards above for every $i$ is $[0,1/(1-\gamma)]$.

Pick any $\pi' \in \Pi_{\pi,\mathcal{S}'}$. Observe that for any $s \in \mathcal{S}$ and $a \in \mathcal{A}$, the distribution of the trajectory $S_0, A_0, R_0, S_1, A_1, R_1, \ldots, A_{K-1}, R_{K-1}, S_K$ is the same under $\mathcal{P}_{\pi',s,a}$ and $\mathcal{P}_{\pi,s,a}$, as $\pi$ and $\pi'$ select the same actions for states in $\mathcal{S}'$. By Eqs. (64) to (65), we can write

$$\begin{aligned}\left|q^{\pi'}(s,a)-q^\pi(s,a)\right| &= \left|\mathbb{E}_{\pi',s,a}\left[\sum_{t\in[K]}\gamma^t R_t+\gamma^K v^{\pi'}(S_K)\right]-\mathbb{E}_{\pi,s,a}\left[\sum_{t\in[K]}\gamma^t R_t+\gamma^K v^{\pi'}(S_K)\right]\right| \\ &= \left|\mathbb{E}_{\pi,s,a}\left[\gamma^K\left(v^{\pi'}(S_K)-v^\pi(S_K)\right)\right]\right| \le \frac{1}{1-\gamma}\mathbb{E}_{\pi,s,a}\left[\gamma^K\right] \\ &\le \frac{1}{1-\gamma}\mathcal{P}_{\pi,s,a}\left[1\le K<H\right]+\frac{\gamma^H}{1-\gamma} \le \frac{1}{1-\gamma}\mathcal{P}_{\pi,s,a}\left[1\le K<H\right]+\omega/4\,. \end{aligned} \tag{79}$$

Combining Eqs. (77) to (79), it follows by the union bound that if MEASURE returns with $(\text{success},\tilde{q})$, then with probability at least $1-\zeta$,

$$\left|q^{\pi'}(s,a)-\bar{q}\right| \le \left|q^{\pi'}(s,a)-q^\pi(s,a)\right|+|q^\pi(s,a)-\bar{q}| \le \omega\,. \qquad \blacksquare$$

## 4.C. Proof of Lemma 4.4.3

We start the proof by showing that there exists a $\theta \in \mathbb{R}^d$ such that

$$\|\theta\|_2 \le B \text{ and for all } s\in\mathcal{S} \text{ and } a\in\mathcal{A},\ q^\pi(s,a) \approx_\varepsilon \langle\theta,\varphi(s,a)\rangle\,. \tag{80}$$

For any finite set $W \subseteq \mathcal{S}\times\mathcal{A}$, $\max_{(s,a)\in W}|q^\pi(s,a)-\langle\varphi(s,a),\theta'\rangle|$ is a continuous function of $\theta'$, hence it attains its infimum on the compact set $\{\theta'\in\mathbb{R}^d : \|\theta'\|_2 \le B\}$. By Definition 4.1.1, this infimum is at most $\varepsilon$. Therefore, the compact sets $\Theta_{s,a} = \{\theta'\in\mathbb{R}^d : \|\theta'\|_2 \le B \text{ and } |q^\pi(s,a)-\langle\varphi(s,a),\theta'\rangle| \le \varepsilon\}$ are non-empty for all $(s,a)\in\mathcal{S}\times\mathcal{A}$, and any intersection of a finite collection of these sets is also non-empty. Therefore, $\bigcap_{(s,a)\in\mathcal{S}\times\mathcal{A}}\Theta_{s,a}$ is non-empty by (Rudin et al., 1976,

Theorem 2.36), and any element $\theta$ of this set satisfies Eq. (80). For the remainder of this proof, fix such a $\theta$.

For any $i \in [n]$, with a slight abuse of notation, we introduce the shorthand $\varphi_i = \varphi(s_i, a_i)$, and let $\hat{q}_i = \langle \theta, \varphi_i \rangle$ and $\xi_i = \bar{q}_i - \hat{q}_i$. Note that by the triangle inequality, $|\xi_i| \le |\bar{q}_i - q^\pi(s_i, a_i)| + |q^\pi(s_i, a_i) - \hat{q}_i| \le \omega + \varepsilon$. Let $\bar{\theta} = V(C)^{-1} \sum_{i \in [n]} \varphi_i \bar{q}_i$ and $\hat{\theta} = V(C)^{-1} \sum_{i \in [n]} \varphi_i \hat{q}_i$.

For any $v \in \mathbb{R}^d$ by the Cauchy-Schwarz inequality,

$$\left|\left\langle \bar{\theta} - \theta, v \right\rangle\right| \le \left|\left\langle \hat{\theta} - \theta, v \right\rangle\right| + \left|\left\langle \bar{\theta} - \hat{\theta}, v \right\rangle\right| \le \|v\|_{V(C)^{-1}} \left\|\hat{\theta} - \theta\right\|_{V(C)} + \left|\left\langle V(C)^{-1} \sum_{i \in [n]} \varphi_i \xi_i, v \right\rangle\right| .$$

To bound the first term on the right-hand side above, observe that

$$\left\|\hat{\theta} - \theta\right\|_{V(C)} = \left\| V(C)^{-1} \left( \sum_{i \in [n]} \varphi_i \varphi_i^\top \right) \theta - \theta \right\|_{V(C)} = \lambda \|\theta\|_{V(C)^{-1}} \le \lambda \|\theta\|_{\frac{1}{\lambda}\mathbf{I}} \le \sqrt{\lambda} B ,$$

where in the last line we used that $V(C) \ge \lambda \mathbf{I}$.

The second term can be bounded as

$$\begin{aligned}
\left|\left\langle V(C)^{-1} \sum_{i \in [n]} \varphi_i \xi_i, v \right\rangle\right| &\le \sum_{i \in [n]} \left|\left\langle V(C)^{-1} \varphi_i \xi_i, v \right\rangle\right| \\
&\le (\omega + \varepsilon) \sum_{i \in [n]} \left|\left\langle V(C)^{-1} \varphi_i, v \right\rangle\right| \\
&\le (\omega + \varepsilon) \sqrt{n} \sqrt{\sum_{i \in [n]} \left(\left\langle V(C)^{-1} \varphi_i, v \right\rangle\right)^2} \\
&\le (\omega + \varepsilon) \sqrt{n} \sqrt{v^\top V(C)^{-1} \left( \sum_{i \in [n]} \varphi_i \varphi_i^\top \right) V(C)^{-1} v + v^\top V(C)^{-1} \lambda \mathbf{I} V(C)^{-1} v} \\
&= (\omega + \varepsilon) \sqrt{n} \|v\|_{V(C)^{-1}} ,
\end{aligned}$$

where the first inequality holds by the triangle inequality, the second by our bound on $|\xi_i|$, the third by the Cauchy-Schwartz inequality, and the fourth by the positivity of $\lambda$. Putting it all together, for

any $s \in \mathcal{S}$ and $a \in \mathcal{A}$, using the previous bounds with $v = \varphi(s,a)$,

$$\begin{aligned}\left|\mathrm{LSE}_{C,\bar{q}}(s,a) - q^{\pi}(s,a)\right| &\le |q^{\pi}(s,a) - \langle \theta, \varphi(s,a)\rangle| + \left|\left\langle \bar{\theta} - \theta, \varphi(s,a)\right\rangle\right| \\ &\le \varepsilon + \|\varphi(s,a)\|_{V(C)^{-1}} \left(\sqrt{\lambda}B + (\omega+\varepsilon)\sqrt{n}\right),\end{aligned}$$

completing the proof. ■

## 4.D. Deriving next-state optimality of $\pi_\ell$ for Lemma 4.4.4

**Lemma 4.D.1.** *Assume that Eq. (67) holds whenever* MEASURE *returns* success. *At any point of* CAPI-QPI-PLAN *after Line 19 is executed, for any $\pi'' \in \Pi_{\pi_\ell, Cover(C_\ell)}$, $s \in Cover(C_\ell)$, and $a \in \mathcal{A}$,*

$$\left|\hat{q}(s,a) - q^{\pi''}(s,a)\right| \le (\omega+\varepsilon)(\sqrt{\tilde{d}}+1).$$

*Proof.* By Lemma 4.4.6 and Eq. (67), $\bar{q}_{l,m} \approx_\omega q^{\pi''}(C_{l,m})$ for all $m \in [|C_\ell|]$ (recall that $C_{l,m}$ is the $m^{\text{th}}$ state-action pair in $C_l$). Therefore, applying Lemma 4.4.3 with $q^{\pi''}$, $C_\ell$ and $\bar{q}_\ell$, as $\hat{q} = \mathrm{LSE}_{C_\ell,\bar{q}_\ell}$, we get that for any $s \in \mathrm{Cover}(C_\ell)$ and all $a \in \mathcal{A}$,

$$\begin{aligned}\left|\hat{q}(s,a) - q^{\pi''}(s,a)\right| &\le \varepsilon + \|\varphi(s,a)\|_{V(C_\ell)^{-1}} \left(\sqrt{\lambda}B + (\omega+\varepsilon)\sqrt{|C_\ell|}\right) \\ &\le (\omega+\varepsilon)(\sqrt{\tilde{d}}+1),\end{aligned}$$

where the second inequality holds because $\|\varphi(s,a)\|_{V(C_\ell)^{-1}} \le 1$ since $s \in \mathrm{Cover}(C_\ell)$, $|C_\ell| \le \tilde{d}$ by Eq. (74), and the definition of $\lambda$. ■

**Lemma 4.D.2.** *Assume that Eq. (67) holds whenever* MEASURE *returns* success. *Consider a time when Lines 20 to 23 of* CAPI-QPI-PLAN *are run and assume that at this time, for all $l \in [H+1]$, $\pi_l$ is $\Delta_l$-optimal on $Cover(C_l)$. Then, $\pi_\ell$ is next-state $(\Delta_\ell + 4(\omega+\varepsilon)(\sqrt{\tilde{d}}+1)/\gamma)$-optimal on $Cover(C_\ell)$.*

*Proof.* Let $\pi^+_\ell$ be defined as in Eq. (83). As $\pi^+_\ell \in \Pi_{\pi_\ell, \mathrm{Cover}(C_\ell)}$, by Lemma 4.D.1, for any $s \in \mathrm{Cover}(C_\ell)$ and all $a \in \mathcal{A}$,

$$\left|\hat{q}(s,a) - q^{\pi^+_\ell}(s,a)\right| \le (\omega+\varepsilon)(\sqrt{\tilde{d}}+1).$$

Similarly, applying Lemma 4.D.1 with $\pi_\ell$ (which trivially belongs to $\Pi_{\pi_\ell, \mathrm{Cover}(C_\ell)}$), we also have

$$|\hat{q}(s,a) - q^{\pi_\ell}(s,a)| \le (\omega+\varepsilon)(\sqrt{\tilde{d}}+1).$$

Therefore,

$$\left|q^{\pi^+_\ell}(s,a) - q^{\pi_\ell}(s,a)\right| \leq 2(\omega+\varepsilon)(\sqrt{\tilde{d}}+1). \tag{81}$$

Since $\pi_\ell$ is $\Delta_\ell$-optimal on Cover$(C_\ell)$ by assumption, this makes $\pi^+_\ell$ $\Delta$-optimal on Cover$(C_\ell)$ for

$$\Delta = \Delta_\ell + 2(\omega+\varepsilon)(\sqrt{\tilde{d}}+1). \tag{82}$$

For a trajectory in the MDP, let the random variable $\tau$ be the first time the state is in Cover$(C_\ell)$:

$$\tau = \min\{t \in \mathbb{N} \,|\, S_t \in \text{Cover}(C_\ell)\}\,.$$

Since $\pi^+_\ell$ agrees with $\pi^\star$ on states not in Cover$(C_\ell)$, the distribution of the trajectory up to and including $S_\tau$ is the same under both policies, starting from any state $s \in \mathcal{S}$. Therefore, for any $s \in \mathcal{S}$,

$$\begin{aligned} v^\star(s) - v^{\pi^+_\ell}(s) &= \mathbb{E}_{\pi^\star,s}\left[\sum_{t\in\mathbb{N}} \gamma^t R_t\right] - \mathbb{E}_{\pi^+_\ell,s}\left[\sum_{t\in\mathbb{N}} \gamma^t R_t\right] \\ &= \mathbb{E}_{\pi^+_\ell,s}\left[\gamma^\tau \left(v^\star(S_\tau) - v^{\pi^+_\ell}(S_\tau)\right)\right] \\ &\leq \Delta\,, \end{aligned}$$

as $\gamma^\tau \leq 1$ and $\pi^+_\ell$ is $\Delta$-optimal on Cover$(C_\ell)$. That is, $\pi^+_\ell$ is also $\Delta$-optimal on $\mathcal{S}$ (with $\Delta$ defined in Eq. 82). Using this, for any $s \in \text{Cover}(C_\ell)$, and $a \in \mathcal{A}$, we have

$$\begin{aligned} &\int_{s'\in\mathcal{S}} \left(v^\star(s') - v^{\pi_\ell}(s')\right) \mathrm{d}P(s'|s,a) \\ &\leq \int_{s'\in\mathcal{S}} \left(v^\star(s') - v^{\pi^+_\ell}(s')\right) \mathrm{d}P(s'|s,a) + \left|\int_{s'\in\mathcal{S}} \left(v^{\pi^+_\ell}(s') - v^{\pi_\ell}(s')\right) \mathrm{d}P(s'|s,a)\right| \\ &\leq \Delta_\ell + 2(\omega+\varepsilon)(\sqrt{\tilde{d}}+1) + \frac{1}{\gamma}\left|q^{\pi^+_\ell}(s,a) - q^{\pi_\ell}(s,a)\right| \\ &\leq \Delta_\ell + 2(\omega+\varepsilon)(\sqrt{\tilde{d}}+1) + 2(\omega+\varepsilon)(\sqrt{\tilde{d}}+1)/\gamma \\ &= \Delta_\ell + 4(\omega+\varepsilon)(\sqrt{\tilde{d}}+1)/\gamma\,, \end{aligned}$$

where the third inequality holds by Eq. (81). Therefore $\pi_\ell$ is next-state $(\Delta_\ell + 4(\omega+\varepsilon)(\sqrt{\tilde{d}}+1)/\gamma))$-optimal on Cover$(C_\ell)$. ■

## 4.E. Poof of Lemma 4.4.4

*Proof of Lemma 4.4.4.* We prove by induction on the iterations of the main loop of CAPI-QPI-PLAN the *inductive hypothesis:* at the start of iteration $i$, for all $l \in [H+1]$, $\pi_l$ is $\Delta_l$-optimal on $\text{Cover}(C_l)$. We first observe that after initialization, $C_l$ is the empty sequence for every $l$, so we can apply Lemma 4.4.3 with $q^\star$ and empty sequences ($n = 0$) to get that for any $s \in \text{Cover}(())$ and $a \in \mathcal{A}$, $q^\star(s,a) \le \varepsilon + \sqrt{\lambda}B = \varepsilon + \omega$. Then, $v^\star(s) \le \varepsilon + \omega \le \Delta_l$. Therefore, at initialization, any policy is $\Delta_l$-optimal on $\text{Cover}(C_l)$ for any $l \in [H+1]$.

Assuming that the inductive hypothesis holds at the start of some iteration, it is left to prove that it continues to hold at the end of the iteration (assuming Eq. (67) holds whenever MEASURE returns *success*); this implies that the hypothesis also holds at the start of the next iteration and hence also proves the lemma. For any $(s,a)$ appended to $C_0$, the inductive hypothesis trivially continues to hold as $\Delta_0 = 1/(1-\gamma) \ge v^\star(s)$ for any $s \in \mathcal{S}$ because the rewards are bounded in $[0,1]$. The only other case in which $C_l$ or $\pi_l$ changes for any $l$ is in Lines 21 and 23, where the changes happen only for $l = \ell + 1$.

We will use Lemma 4.3.4 to analyze the effect of these updates, thus next we show that the conditions of the lemma are satisfied:

*(a)* In Lemma 4.D.2 we show that $\pi_\ell$ is next-state $(\Delta_\ell + 4(\omega+\varepsilon)(\sqrt{\tilde{d}}+1)/\gamma)$-optimal on $\text{Cover}(C_\ell)$. In the proof of the lemma, we introduce a policy in Eq. (83) that acts as $\pi_\ell$ on states in $\text{Cover}(C_\ell)$, and as an optimal stationary deterministic memoryless policy $\pi^\star$ otherwise:

$$\pi^+_\ell(s) = \begin{cases} \pi_\ell(s) & \text{if } s \in \text{Cover}(C_\ell); \\ \pi^\star(s) & \text{otherwise.} \end{cases} \tag{83}$$

Intuitively, this policy corrects $\pi_\ell$ on the low-confidence states. The proof of Lemma 4.D.2 then uses the fact that this policy is also $q^\pi$-realizable (Definition 4.1.1) and satisfies $\pi^+_\ell \in \Pi_{\pi_\ell, \text{Cover}(C_\ell)}$ to show (i) that the $q$-values of $\pi_\ell$ and $\pi^+_\ell$ are close on the measured state-action pairs (via Lemma 4.4.6 and Lemma 4.D.1); (ii) an optimality guarantee on $\pi^+_\ell$ for all $s \in \mathcal{S}$; and, as a consequence, (iii) the next-state optimality of $\pi_\ell$.

*(b)* Next, to analyze the effect of Line 21, we introduce hypothetical $q$-approximators $\tilde{q}_l$ for $l \in [H+1]$, defined as follows: At initialization, $\tilde{q}_l(s,a) = 0$ for all $l \in [H+1]$, $s \in \mathcal{S}$, and $a \in \mathcal{A}$. It

is updated every time after Line 19 of the algorithm is executed as

$$\tilde{q}_\ell(s,a) \leftarrow \begin{cases} \tilde{q}_\ell(s,a) & \text{if } s \in \text{Cover}(C_{\ell+1}); \quad (84a) \\ \hat{q}(s,a) & \text{otherwise.} \quad (84b) \end{cases}$$

In other words, $\tilde{q}_\ell$ is only updated to the newly computed $\hat{q}$ for states that are not in $\text{Cover}(C_{\ell+1})$, and stays unchanged for other states. We show in Lemma 4.F.2 that the new policy that $\pi_{\ell+1}$ is updated to, which is constructed in two steps (Lines 20–21), can be expressed as the result of a *single* CAPI policy update that uses $\tilde{q}$:

$$\pi_{\ell+1} \leftarrow \pi_{\tilde{q}_\ell, \pi_\ell, \mathcal{S}\setminus\text{Cover}(C_l)}\,.$$

We show in Lemma 4.F.1 that $\tilde{q}_\ell \approx_{\omega'} q^{\pi_\ell}$ with $\omega' = (\omega+\varepsilon)(\sqrt{\tilde{d}}+1)$ on $\text{Cover}(C_\ell)$.

By the above, we can apply Lemma 4.3.4 with policy $\pi_\ell$, $q$-approximation $\tilde{q}_\ell$ (with approximation error guarantee $\omega'$ on $\text{Cover}(C_\ell)$, and $\mathcal{S}_{\text{fix}} = \mathcal{S} \setminus \text{Cover}(C_\ell)$ to get that the new value of $\pi_{\ell+1}$ is a $\Delta_{\ell+1} = (8(\omega+\varepsilon)(\sqrt{\tilde{d}}+1)+\gamma\Delta_\ell)$-optimal policy on $\text{Cover}(C_\ell)$. By the end of the loop in Line 23, $\text{Cover}(C_{\ell+1}) = \text{Cover}(C_\ell)$, so $\pi_{\ell+1}$ is $\Delta_{\ell+1}$-optimal on $\text{Cover}(C_{\ell+1})$. This finishes the proof that the inductive hypothesis continues to hold at the end of the iteration, finishing the proof of the lemma. ■

## 4.F. Auxiliary results for Lemma 4.4.4 about $\tilde{q}_l$

Throughout the execution of CAPI-QPI-PLAN, for $l \in [H+1]$, let $\tilde{q}_l^-$, $\pi_l^-$, $C_l^-$ denote the values of variables $\tilde{q}_\ell$, $\pi_\ell$, $C_\ell$, respectively, at the time when Lines 19–23 were most recently executed with $\ell = l$ in a previous iteration of the main loop of CAPI-QPI-PLAN. If such a time does not exist, let their values be the initialization values. Thus, $C_l^-$ may (only) change at the start of some iteration $i$ if Lines 19–23 were executed with $\ell = l$ in the previous iteration $i-1$. Observe that whenever this happens, Lines 19–23 may also change $C_{\ell+1}$ in iteration $i-1$, and this is the only time $C_{l+1}$ can be changed for any $l \in [H]$. After this, at the beginning of iteration $i$, $C_{l+1}$ always has the same elements as $C_l^-$. Therefore, since it also holds at the initialization of the algorithm, we conclude that at the start of each iteration,

$$\text{Cover}(C_{l+1}) = \text{Cover}(C_l^-)\,. \quad (85)$$

**Lemma 4.F.1.** *Assume that Eq. (67) holds whenever* MEASURE *returns* success. *Then, whenever Line 21 of* CAPI-QPI-PLAN *is executed, for all $s \in Cover(C_\ell)$ and $a \in \mathcal{A}$,*

$$\left|\tilde{q}_\ell(s,a) - q^{\pi''}(s,a)\right| \le (\omega+\varepsilon)(\sqrt{\tilde{d}}+1) \qquad \text{for all } \pi'' \in \Pi_{\pi_\ell, Cover(C_\ell)}\,. \tag{86}$$

*Proof.* We prove this by induction for every time Line 21 is executed with any value of $\ell$. We first observe that after initialization, $C_l$ is the empty sequence for every $l$, so we can apply Lemma 4.4.3 with $q^\star$ and empty sequences ($n = 0$) to get that for any $s \in \text{Cover}(())$ and $a \in \mathcal{A}$, $q^{\pi''}(s,a) \le q^\star(s,a) \le \varepsilon + \sqrt{\lambda}B = \varepsilon + \omega$. Also, $\tilde{q}_l(\cdot,\cdot) = 0$ at initialization, so Eq. (86) holds for any value of $\ell$.

Consider a time when Line 21 is executed and assume the inductive hypothesis holds for the previous time Line 21 was executed with the same value of $\ell$ (or at the initialization if this is the first time), that is,

$$\left|\tilde{q}^-_\ell(s,a) - q^{\pi''}(s,a)\right| \le (\omega+\varepsilon)(\sqrt{\tilde{d}}+1) \qquad \text{for all } \pi'' \in \Pi_{\pi^-_\ell, \text{Cover}(C^-_\ell)},\ s \in \text{Cover}(C^-_\ell)\,.$$

To prove that the statement now holds for any $s \in \text{Cover}(C_\ell)$, first consider any $s \in \text{Cover}(C_{\ell+1}) = \text{Cover}(C^-_\ell)$. For such an $s$, by Lemma 4.4.5 we have that $\Pi_{\pi_\ell, \text{Cover}(C_\ell)} \subseteq \Pi_{\pi^-_\ell, \text{Cover}(C^-_\ell)}$. Also, by definition, $\tilde{q}_\ell(s,\cdot) = \tilde{q}^-_\ell(s,\cdot)$ for $s \in \text{Cover}(C_{\ell+1})$. Combining with the inductive hypothesis, it follows that Eq. (86) holds for $s \in \text{Cover}(C_{\ell+1})$.

It remains to show that Eq. (86) also holds for $s \in \text{Cover}(C_\ell) \setminus \text{Cover}(C_{\ell+1})$. For such an $s$, $\tilde{q}_\ell(s,\cdot) = \hat{q}(s,\cdot)$ by definition, and hence Lemma 4.D.1 implies that Eq. (86) holds in this case.

Combining the two cases, it follows that the inductive hypothesis continues to hold when Line 21 is executed. ■

**Lemma 4.F.2.** *Throughout the execution of* CAPI-QPI-PLAN, *at the start of any iteration, for all $l \in [H]$,*

$$\pi_{l+1} = \pi_{\tilde{q}^-_l, \pi^-_l, \mathcal{S}\setminus Cover(C^-_l)}\,. \tag{87}$$

*Proof.* We prove this by induction for the start of any iteration. Eq. (87) holds at the start of the algorithm due to its initialization (because at initialiaztion, $\tilde{q}^-_l(s,a) = 0$ for all $s,a$, and hence by our tie-breaking rule, the policy on the right-hand side of Eq. (87) always chooses action $\mathcal{A}_1$, which is the initial policy for $\pi_l$).

In what follows, we use the fact that for any $q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, policy $\pi$, and $\mathcal{S}_{\text{fix}} \subseteq \mathcal{S}$, the CAPI policy update $\pi_{q,\pi,\mathcal{S}_{\text{fix}}}$ is a policy whose value at any $s \in \mathcal{S}$ only depends on $q(s,\cdot)$, $\pi(s)$, and whether or not $s \in \mathcal{S}_{\text{fix}}$, by definition (Eq. 66). Therefore, for an alternative $q'$, $\pi'$, $\mathcal{S}'_{\text{fix}}$, for any $s \in \mathcal{S}$, $\pi_{q,\pi,\mathcal{S}_{\text{fix}}}(s) = \pi_{q',\pi',\mathcal{S}'_{\text{fix}}}(s)$ whenever the following three conditions hold: (C1) $q(s,a) = q'(s,a)$ for all $a \in \mathcal{A}$; (C2) $\pi(s) = \pi'(s)$; and (C3) either both or none of $\mathcal{S}_{\text{fix}}$ and $\mathcal{S}'_{\text{fix}}$ include $s$.

Assume the inductive hypothesis holds at the beginning of some iteration. Let $\pi''$ be the policy Line 21 updates $\pi_{\ell+1}$ to, noting that this is the only place where policies are updated. All we need to prove is that $\pi''$ is equal to

$$\tilde{\pi} = \pi_{\tilde{q}_\ell, \pi_\ell, \mathcal{S} \setminus \text{Cover}(C_\ell)} .$$

First, for any $s \notin \text{Cover}(C_{\ell+1})$, $\pi''(s) = \pi'(s) = \pi_{\hat{q},\pi_\ell,\mathcal{S}\setminus\text{Cover}(C_\ell)}(s)$ and $\hat{q}(s,\cdot) = \tilde{q}_\ell(s,\cdot)$ by definition. Hence, $\pi''(s) = \pi_{\hat{q},\pi_\ell,\mathcal{S}\setminus\text{Cover}(C_\ell)}(s) = \pi_{\tilde{q}_\ell,\pi_\ell,\mathcal{S}\setminus\text{Cover}(C_\ell)}(s) = \tilde{\pi}(s)$, as all of conditions (C1)-(C3) are satisfied for $s$ (C2 and C3 hold trivially).

Next, take any $s \in \text{Cover}(C_{\ell+1}) = \text{Cover}(C^-_\ell)$. Then, by Line 21, $\pi''(s) = \pi_{\ell+1}(s)$. By the inductive hypothesis, the current value of $\pi_{\ell+1}$ can be written as $\pi_{\tilde{q}^-_\ell,\pi^-_\ell,\mathcal{S}\setminus\text{Cover}(C^-_\ell)}$. We prove that this policy takes the same value as $\tilde{\pi}$ at $s$, by showing conditions (C1)-(C3). First, by Lemma 4.4.5, $\pi_\ell \in \Pi_{\pi^-_\ell,\text{Cover}(C^-_\ell)}$. Thus, as $s \in \text{Cover}(C^-_\ell)$, $\pi_\ell(s) = \pi^-_\ell(s)$, showing condition (C2). Furthermore, as $s \in \text{Cover}(C_{\ell+1})$, by definition, $\tilde{q}_\ell(s,\cdot) = \tilde{q}^-_\ell(s,\cdot)$, showing condition (C1). Finally, as $s \in \text{Cover}(C_{\ell+1}) = \text{Cover}(C^-_\ell) \subseteq \text{Cover}(C_\ell)$, $s \notin \mathcal{S} \setminus \text{Cover}(C^-_\ell)$ and $s \notin \mathcal{S} \setminus \text{Cover}(C_\ell)$, showing condition (C3).

Combining the two cases, $\pi''(s) = \tilde{\pi}(s)$ for any $s \in \mathcal{S}$, finishing the induction. ■

## 4.G. Efficient implementation and proof of Theorem 4.1.3

In this section we consider the efficient implementation of CAPI-QPI-PLAN in terms of memory and computational costs of both the algorithm itself and the final policy it outputs.

Focusing on the memory cost, first we can observe that throughout the execution of the algorithm, $C_l$ for all $l \in [H+1]$ only stores up to $\tilde{d}$ unique state-action pairs altogether (cf. Eq. (74)), as they use the same pairs; let $W = (s_i, a_i)_{i \in \hat{d}}$ denote these for some $\hat{d} \leq \tilde{d}$. Furthermore, throughout the execution of the algorithm, for any level $l$, the only features that $\pi_l$ depends on are the features associated with members of $W$. Storing all these features takes $d\hat{d}$ memory. Denote all the policies that CAPI-QPI-PLAN constructs in Line 21, in order, as $\pi^{(0)}, \pi^{(1)}, \ldots, \pi^{(n-1)}$, where $n$ is the number of times Line 21 is executed. Recall from the proof of Theorem 4.1.2 that the number of times MEASURE returns *success*, which is an upper bounds on $n$, is itself bounded by $\tilde{d}H$, hence $n \leq \tilde{d}H$.

Together, Lines 20-21 construct a policy that, for an $s \in \mathcal{S}$, decides whether the action should be $\arg\max_{a\in\mathcal{A}} \langle \varphi(s,a), \theta \rangle$ for some $\theta$ given by LSE (Eq. (69)), or the value of the policy should be determined by a recursive call to a previously constructed policy, either $\pi_{\ell+1}$ or $\pi_\ell$ (through $\pi'$). Now there exist some $a, b \in [n]$ such that $\pi^{(a)} = \pi_\ell$ and $\pi^{(b)} = \pi_{\ell+1}$ before the new policy is constructed in Line 21. To implement the new $\pi_{\ell+1}$ constructed policy, it is enough therefore to store, in addition to the existing policies, $\theta$ (from $\hat{q}$), the decision rules, and the indices $a$ and $b$. The decision rules are fully defined by $\theta$, $C_\ell$, and $C_{\ell+1}$. It is therefore enough to further store $C_\ell, C_{\ell+1} \subseteq W$, which can be encoded as $\hat{d}$-dimensional vectors each, storing the bitmask of which state-action pairs are included. We also store the current value of $\ell$ (the level) for the newly constructed policy. Together, a policy thus consumes $3+d+2\hat{d}$ memory. We store all policies constructed, along with the features of $W$, and the final value of $V(C_H)^{-1}$, at a memory cost of $d\hat{d} + \tilde{d}H(3+d+\hat{d}) + d^2 = \tilde{\mathcal{O}}(d^2/(1-\gamma))$. This is the memory cost of the final policy outputted by CAPI-QPI-PLAN. The memory cost of running CAPI-QPI-PLAN itself is of the same order, as additionally storing $C_l$, $\bar{q}_l$, and $V(C_l)^{-1}$ for $l \in [H+1]$ takes $\tilde{\mathcal{O}}(d^2/(1-\gamma))$ memory.

To efficiently implement the final policy found by CAPI-QPI-PLAN with the stored information described above, we start from evaluating the last policy constructed, $\pi^{(i)}$ for $i = n-1$. We introduce auxiliary variables $\tilde{V}(C_l)^{-1}$ and $\tilde{C}_l$ for $l \in [H+1]$ to efficiently track the required values of $V(C_l)^{-1}$ and $C_l$. We keep updating these variables so that for $l \in \{\ell, \ell+1\}$, they match the values of $V(C_l)^{-1}$ and $C_l$, respectively, at the time of construction of the current policy $\pi^{(i)}$ under consideration, where $\ell$ is the (saved) level of $\pi^{(i)}$. For $i = n-1$, observe that when it was constructed, $C_0 = C_1 = \cdots = C_H$ by Lemma 4.4.7. We therefore start by initializing variables $\tilde{V}(C_0)^{-1}, \ldots, \tilde{V}(C_H)^{-1}$ to the saved final value of $V(C_H)^{-1}$, and variables $\tilde{C}_0, \ldots, \tilde{C}_H$ to $W$. Implementing the decisions of a policy takes an order of $|\mathcal{A}|d^2$ computation ($|\mathcal{A}|$ vector and matrix multiplications), after which we recover either the policy output or a previously constructed policy to recurse into. For the latter case, we have to consider the evaluation of this policy, denoted by $\pi^{(i')}$. Let the (saved) level of $\pi^{(i')}$ be $\ell'$. Before we set $i$ to $i'$ and start evaluating it, we need to update the values of $\tilde{V}(C_l)$ and $C_l$ for $l \in \{\ell', \ell'+1\}$. The updates are needed for these two levels only, as the decision rule of policy $i'$ only depends on these levels, as shown before. Let us describe the update procedure for some $l \in \{\ell', \ell'+1\}$: Since $\pi^{(i')}$ was constructed earlier than $\pi^{(i)}$ (i.e., $i' < i$), and $C_{l'}$ can only grow during the algorithm for any $l' \in [H+1]$, we only need to remove members of the variable $\tilde{C}_l$ to match the value of $C_l$ at the time of construction of $\pi^{(i')}$. The members to be removed are given by the difference of the members of $\tilde{C}_l$ and the bitmasks

stored for $\pi^{(i')}$ for level $l$. For each state-action pair $(s,a)$ removed, we also need to update $\tilde{V}(C_l)^{-1}$ to $\left(\tilde{V}(C_l) - \varphi(s,a)\varphi(s,a)^\top\right)^{-1}$, which can be done in order $d^2$ computation using the Sherman-Morrison-Woodbury formula (Max, 1950). The total number of such removal operations for any level $l$ is bounded by the sum of the number of state-action pairs in the initialization of $\tilde{C}_{l'}$ (for $l' \in [H+1]$), that is, by $(H+1)\hat{d}$. As a result, the computational cost of the final policy of CAPI-QPI-PLAN is $\tilde{\mathcal{O}}((H+1)\hat{d}d^2) + n\tilde{\mathcal{O}}(|\mathcal{A}|d^2) = \tilde{\mathcal{O}}(d^3|\mathcal{A}|/(1-\gamma))$.

Finally, we consider the computational cost of running CAPI-QPI-PLAN. The number of iterations of the outer loop is bounded by $\tilde{\mathcal{O}}(dH) = \tilde{\mathcal{O}}(d/(1-\gamma))$, as each iteration involves either a MEASURE call that returns *success*, or a new member added to some $C_l$. For each iteration, Line 4 takes $\tilde{\mathcal{O}}(d^2|\mathcal{A}|)$, Line 8 takes $\tilde{\mathcal{O}}(d/(1-\gamma))$, Line 13 takes $\tilde{\mathcal{O}}(d^2|\mathcal{A}|)$ computation; for Line 19, calculating $\theta$, the second component of the inner product of the least-squares predictor in Eq. (69) takes $\tilde{\mathcal{O}}(d^2)$ computation, and if $C_l$ ever changes for some $l$, updating $V(C_l)^{-1}$ by the Sherman-Morrison-Woodbury takes $\tilde{\mathcal{O}}(d^2)$ computation. Overall, all the operations except those associated to the MEASURE call of Line 11 take $\tilde{\mathcal{O}}(d^3|\mathcal{A}|/(1-\gamma))$ computation in total. We conclude our calculations by considering the computational cost of the MEASURE calls, which will dominate the overall computational cost. Line 6 of Algorithm 5 has a computational cost of order $d^2|\mathcal{A}|$, while the majority of the computational cost comes from evaluating the policy at Line 8. By our previous calculations, this takes $\tilde{\mathcal{O}}(d^3|\mathcal{A}|/(1-\gamma))$ computation and happens (at most) once for each simulator call. Using the query cost bound of Theorem 4.1.2, we conclude that the computational cost of CAPI-QPI-PLAN is $\tilde{\mathcal{O}}(d^4|\mathcal{A}|(1-\gamma)^{-5}\omega^{-2})$. ■

## Chapter 5

# Online RL with $q^\pi$-realizability

In the final chapter of this thesis, we tackle online RL, the most challenging setting considered. The goal of this chapter is to prove Theorem 1.5.2. To do this, we switch back to finite-horizon episodic Markov decision processes (MDPs), and employ the linear $q^\pi$-realizability assumption. Recall that the class of $q^\pi$-realizable MDPs is known to be more general than linear MDPs (where the transition kernel and the reward function are assumed to be linear functions of the feature vectors). As our first contribution, we show that the difference between the two classes is the presence of states in linearly $q^\pi$-realizable MDPs where for any policy, all the actions have approximately equal values, and skipping these states by following an arbitrarily fixed policy in those states transforms the problem to a linear MDP. Based on this observation, we derive a novel learning algorithm for linearly $q^\pi$-realizable MDPs that simultaneously learns what states should be skipped and runs another learning algorithm on the linear MDP hidden in the problem. The new algorithm returns an $\varepsilon$-optimal policy after $\mathrm{polylog}(H,d)/\varepsilon^2$ interactions with the MDP, where $H$ is the time horizon and $d$ is the dimension of the feature vectors, giving the first polynomial-sample-complexity online RL algorithm for this setting. The results are proved for the misspecified case, where the query complexity is shown to degrade gracefully with the misspecification error.

## 5.1. Introduction

There are several sample-efficient algorithms discovering near-optimal policies in linear MDPs under various MDP access models and settings (online RL: Jin et al. (2020a); batch setting: Jin et al. (2021); reward-free setting: Wagenmaker et al. (2022)). The best known sample-complexity bound for the online RL setting is achieved by the computationally inefficient algorithm of Zanette et al. (2020b), called ELEANOR, which serves as a starting point of our work.

As opposed to linear MDPs, before the work of Weisz et al. (2023), sample efficient solutions were only known for $q^\pi$-realizable MDPs when the MDP is accessed through a simulator that

implements some form of a state-reset function (Lattimore et al., 2020; Yin et al., 2022; Weisz et al., 2022a). In this thesis we present the work of Weisz et al. (2023), showing that having access to a state-reset is not essential in this setting. To this end, we present SKIPPYELEANOR (Algorithm 7) and a corresponding theorem (Theorem 5.4.1) that shows that SKIPPYELEANOR, which uses online interactions only, is a provably sample-efficient solution to this problem. The rest of this chapter is organized as follows. In Section 5.2 we introduce the basic definitions. In Section 5.3 we give an insight into the difference between linear $q^\pi$-realizability and linear MDPs, which motivates our approach. In Section 5.4 we describe our algorithm and the most important technical tools we discovered for its analysis. Notably, we develop a novel, modular theory in Section 5.4.2 that establishes a rich structure inherent in $q^\pi$-realizable MDPs, which acts as the technical foundation to the main results of the chapter, and may be of independent interest. Finally, Section 5.5 gives a summary of the proof of the main result (Theorem 5.4.1).

## 5.2. Preliminaries

For a linear subspace $X$ of $\mathbb{R}^d$, let $\text{Proj}_X$ denote the orthogonal projection matrix onto $X$. Throughout we fix $d \in \mathbb{N}_+$. For $L > 0$, let $\mathcal{B}(L) = \{x \in \mathbb{R}^d : \|x\|_2 \le L\}$ denote the $d$-dimensional Euclidean ball of radius $L$ centered at the origin, where $\|\cdot\|_2$ denotes the Euclidean norm. Let PD denote the set of positive definite matrices in $\mathbb{R}^{d\times d}$. We write $a \approx_\varepsilon b$ for $a, b, \varepsilon \in \mathbb{R}$ if $|a - b| \le \varepsilon$. Let $\mathbb{I}\{B\}$ be the indicator function on boolean-valued (possibly random) $B$ taking value 1 if $B$ is true, 0 if false. We let $\mathcal{M}_1(X)$ denote the set of probability distributions supported on set $X$.[12] The rest of our notation is standard, but described in Section 5.A for completeness.

We recall the most important facts about MDPs and introduce a slight variation of our previous notation. For the setting of episodic finite horizon RL, with horizon $H$, a finite-action Markov Decision Process (MDP) describes an environment for sequential decision-making. It is defined by a tuple $(\mathcal{S}, [\mathcal{A}], P, \mathcal{R})$ as follows. The state space $\mathcal{S}$ is split across stages: $\mathcal{S} = (\mathcal{S}_t)_{t\in[H]}$ with $\mathcal{S}_1 = \{s_1\}$ for some designated initial state $s_1$. Without loss of generality, we assume the $(\mathcal{S}_t)_{t\in[H]}$ are disjoint sets. We define the function stage : $\mathcal{S} \to [H]$ as stage$(s) = t$ if $s \in \mathcal{S}_t$. We consider finite action spaces of size $\mathcal{A}$ for some $\mathcal{A} \in \mathbb{N}$, and without loss of generality, define the set of actions to be $[\mathcal{A}] := \{1, \dots, \mathcal{A}\}$. The transition kernel is $P : \left(\bigcup_{t\in[H-1]} \mathcal{S}_t\right) \times [\mathcal{A}] \to \mathcal{M}_1(\mathcal{S})$, with the property that transitions happen between successive stages, that is, for any $t \in [H-1]$, state $s_t \in \mathcal{S}_t$, and action $a \in [\mathcal{A}]$, $P(s_t, a) \in \mathcal{M}_1(\mathcal{S}_{t+1})$. The reward kernel is $\mathcal{R} : \mathcal{S} \times [\mathcal{A}] \to \mathcal{M}_1([0,1])$. An agent interacts sequentially with this environment in an episode lasting $H$ steps by taking some

12. Here, and in what follows, we assume the availability of appropriate measurability structures when necessary.

action $a \in [\mathcal{A}]$ in the current state. The environment responds by transitioning to some next-state according to $P$, and giving a reward in $[0,1]$ according to $\mathcal{R}$.[13]

We describe an agent interacting with the MDP by a *policy* $\pi$, which, to each history of interaction (including states, actions and rewards) assigns a probability distribution over the actions. Policies where this distribution only depend on the last state in the history are called *memoryless*, and these are identified with elements of the set $\Pi = \{\pi : \mathcal{S} \to \mathcal{M}_1([\mathcal{A}])\}$. Using a policy $\pi$, starting at some state $s$ in an MDP induces a probability distribution over histories, which we denote by $\mathcal{P}_{\pi,s}$. For any $a \in [\mathcal{A}]$, $\mathcal{P}_{\pi,s,a}$ is the distribution over the histories when first action $a$ is used in state $s$, after which policy $\pi$ is followed. $\mathbb{E}_\bullet$ is the expectation operator corresponding to a distribution $\mathcal{P}_\bullet$ (e.g., $\mathbb{E}_{\pi,s}$ is the expectation with respect to $\mathcal{P}_{\pi,s}$). The state- and action-value functions $v^\pi$ and $q^\pi$ are defined as the expected total reward within the first episode while $\pi$ is used:

$$v^\pi(s) = \mathbb{E}_{\pi,s} \sum_{u=\mathrm{stage}(s)}^{H} R_u \quad \text{for } s \in \mathcal{S} \quad \text{and} \quad q^\pi(s,a) = \mathbb{E}_{\pi,s,a} \sum_{u=\mathrm{stage}(s)}^{H} R_u \quad \text{for } s \in \mathcal{S}, a \in [\mathcal{A}].$$

Let $\pi^\star \in \Pi$ be an optimal policy, satisfying $q^{\pi^\star}(s,a) = \sup_{\pi\in\Pi} q^\pi(s,a) = \sup_{\pi \in \text{all policies}} q^\pi(s,a)$ for all $(s,a) \in \mathcal{S} \times [\mathcal{A}]$. Let $q^\star(s,a) = q^{\pi^\star}(s,a)$ and $v^\star(s) = \sup_{a'\in[\mathcal{A}]} q^\star(s,a)$ for all $(s,a) \in \mathcal{S} \times [\mathcal{A}]$.

## 5.3. From linear $q^\pi$-realizability to linear MDPs

As described in the introduction, we endow our MDP with a feature map $\varphi : \mathcal{S} \times [\mathcal{A}] \to \mathcal{B}(L)$ for some $L_1 > 0$. For reference, we start with a definition of linear MDPs with a parameter norm bound $B > 0$, formalizing that the transition kernel and the expected rewards are approximately linear functions of the features:[14]

**Definition 5.3.1.** *[$\kappa$-approximately linear MDP] For any $\kappa \le 1$, an MDP is a $\kappa$-approximately linear MDP if* (i) *there exists $\theta_1, \dots, \theta_H \in \mathcal{B}(B)$ such that for any $h \in [H]$ and $(s,a) \in \mathcal{S}_h \times [\mathcal{A}]$, $|\mathbb{E}_{R\sim\mathcal{R}(s,a)} R - \langle \varphi(s,a), \theta_h \rangle| \le \kappa$ and* (ii) *for any $f : \mathcal{S} \to [0,H]$ and $h \in [H-1]$, there exists $\theta'_h \in \mathcal{B}(B)$ such that for all $(s,a) \in \mathcal{S}_h \times [\mathcal{A}]$, $|\mathbb{E}_{S'\sim P(s,a)} f(S') - \langle \varphi(s,a), \theta'_h \rangle| \le \kappa$.*

13. Here, the reward and next-state are independent, given the current state and last action. Independence is nonessential and is assumed only to simplify the presentation.
14. Compared to the definition of Jin et al. (2020b), our definition does not require the existence of a vector-valued measure to represent the transition kernel. This is a generalization that is compatible with all existing algorithms for linear MDPs.

A key consequence of the linear MDP assumption is that the *inherent Bellman error*

$$\sup_{\theta_{h+1}\in\mathcal{B}(B)} \inf_{\theta_h\in\mathcal{B}(B)} \sup_{(s,a)\in\mathcal{S}_h\times[\mathcal{A}]} \left|\mathbb{E}_{R\sim\mathcal{R}(s,a),S'\sim P(s,a)} R(s,a) + \max_{a'\in[\mathcal{A}]} \langle\varphi(S',a'),\theta_{h+1}\rangle - \langle\varphi(s,a),\theta_h\rangle\right|,$$

scales with the misspecification $\kappa$. This property is also referred to as the *closedness to the Bellman operator*, and is a crucial component in the analysis of approximation errors for algorithms tackling linear MDPs.

In this thesis we consider a weaker linearity assumption where we only assume that the action-value functions are approximately linear:

**Definition 5.3.2** ($q^\pi$-realizability: uniform linear function approximation error of value-functions)**.** *Given an MDP, the uniform value-function approximation error (or misspecification) induced by a feature map $\varphi : \mathcal{S}\times[\mathcal{A}] \to \mathcal{B}(L)$, over a set of parameters in $\mathcal{B}(B)$ is*

$$\eta = \sup_{\pi\in\Pi} \max_{h\in[H]} \inf_{\theta^{(h)}\in\mathcal{B}(B)} \sup_{(s,a)\in\mathcal{S}_h\times[\mathcal{A}]} |q^\pi(s,a) - \left\langle\varphi(s,a),\theta^{(h)}\right\rangle|\,.$$

*For the MDP and the corresponding feature map, for all $h\in[H]$ fix any $\theta_h : \Pi \to \mathcal{B}(B)$ mapping each memoryless policy $\pi\in\Pi$ to its "parameter", such that*

$$q^\pi(s,a) \approx_\eta \langle\varphi(s,a),\theta_h(\pi)\rangle \qquad \text{for all } \pi\in\Pi,\ s\in\mathcal{S}_h, \text{ and } a\in[\mathcal{A}]\,. \tag{88}$$

*The set of all parameters $\Theta_h \subseteq \mathcal{B}(B)$ for a stage $h\in[H]$ is given by $\Theta_h = \{\theta_h(\pi) : \pi\in\Pi\}$.*

Note that $\theta_h$ satisfying Eq. (88) always exist (Weisz et al., 2022a, Appendix C). We focus on the feasible regime where $\eta$ is polynomially small in the relevant parameters. Specifically, we assume that $\eta$ is bounded according to Eq. (108). The main problem of interest in this thesis is the following:

**Problem 5.3.3** (informal)**.** *For any $\varepsilon,\zeta > 0$ and any MDP with corresponding uniform value-function approximation error $\eta$, derive an algorithm that, with probability at least $1-\zeta$, will find an $\varepsilon$-optimal policy (i.e., a policy $\pi$ such that $v^\pi(s_1) \geq v^\star(s_1) - \varepsilon$) by interacting with the MDP online for $T$ steps with $T$ bounded by a polynomial function of $(d, H, \varepsilon^{-1}, \log\zeta^{-1}, \log L, \log B)$. That the interaction with the MDP is online means that it is only possible to observe the features corresponding to the current state, and to take an action and subsequently observe the resulting reward and next state, which then becomes the current state. We consider the fixed horizon episodic setting, that is, the next state is reset to the initial state $s_1$ after every $H$ steps.*


**Figure 5.3.1: Left:** MDP with deterministic transitions and rewards (edges are labeled with action/reward). **Right:** The same MDP with the red "low-range" states "skipped" over. $\varphi(s_1,\cdot) = (1), \varphi(s_3,\cdot) = (0.5), \varphi(\cdot,\cdot) = (0)$ otherwise. Both MDPs are $q^\pi$-realizable, but only the right MDP is linear.

Algorithms developed for linear MDPs are not directly applicable to Problem 5.3.3 when the MDP is only $q^\pi$-realizable: While a linear MDP is also $q^\pi$-realizable, a $q^\pi$-realizable MDP may be neither a linear MDP, nor one with a low inherent Bellman error (Zanette et al., 2020b). As an illustrative example, Fig. 5.3.1, left shows an MDP that is $q^\pi$-realizable but not linear. To see this, observe that the features for both actions in $s_1$ are identical, but their transitions and rewards are not. As illustrated in the figure however, if we *skip* over the red states (with identical actions) by taking the first action on them and summing up the rewards received until we reach a black state, we arrive at a linear MDP. This serves as the main intuition behind our work: the red states have no bearing on action-values, so they can be skipped, and the resulting MDP is linear.

More generally, we can define the *range* of any state as the maximum possible difference in action-value that the choice of action in that state can make:

$$\text{range}(s) = \sup_{\theta \in \Theta_{\text{stage}(s)}} \max_{i,j \in [\mathcal{A}]} \langle \varphi(s,i,j), \theta \rangle \text{ for all } h \in [H], s \in \mathcal{S}_h\,, \tag{89}$$

where $\varphi(s,i,j) = \varphi(s,i) - \varphi(s,j)$ is the notation for feature differences. Clearly, the choice of action in low-range states is not too important, as

$$v^\pi(s) - q^\pi(s,a) \le \text{range}(s) + 2\eta \qquad \text{for any } \pi \in \Pi \text{ and all } a \in [\mathcal{A}]. \tag{90}$$

Not only are the action choices in low-range states unimportant for the task of finding a near-optimal policy for the MDP, these choices can affect transitions and rewards in a nonlinear way. Interestingly, the existence of low-range states is the reason why $q^\pi$-realizable MDPs are not necessarily linear, as shown by the next result (proved in Section 5.C), which follows easily from Lemma 5.4.7.

**Proposition 5.3.4.** *Consider an MDP with uniform value-function approximation error $\eta \ge 0$. If there are no states $s \in \mathcal{S}$ with* $\text{range}(s) < \alpha$ *for some $\alpha > 0$, then the transitions and rewards of*

*the MDP are linear (Definition 5.3.1) with misspecification scaling with $\eta$, and parameter norms scaling inversely with $\alpha$.*

**Our approach.** The above result immediately offers a strategy to learn under the (linear) $q^\pi$-realizability assumption. Assuming access to an oracle that can determine whether or not $\text{range}(s) < \alpha$ for any state $s$, the MDP could be "converted" to one that has no low-range states but has near-identical state and action-value functions of any policy (compared to the original MDP), by skipping over low-range states (by executing an arbitrary action) until a state with a range at least $\alpha$ is reached. We will call such a multi-state transition a *skippy step* and refer to such a policy as a *skippy policy*. The reward presented for a skippy step is the cumulative reward over the skipped states. When the oracle is correct, the new MDP is a linear MDP, allowing techniques such as ELEANOR to efficiently learn a near-optimal policy. This conversion argument is part of the intuition of our method, but it is not strictly part of the proof, so we defer the details to Section 5.C. The only missing piece for solving the general case, Problem 5.3.3, is learning an oracle that can suggest when to skip over a state, and combining it with the learning algorithm for the linear MDP. This general approach leads to our algorithm, SKIPPYELEANOR, which runs a modified version of ELEANOR with guessed oracles. During the algorithm, we detect when an incorrect oracle leads to suboptimal results, and refine the oracle accordingly. The details of the algorithm are explained in the next section.

## 5.4. Algorithm

In this section we present our main results following our plan outlined above. We first give Algorithm 7, along with a high-level overview of the algorithm; the details are explained throughout the section. The parameters of the algorithm are presented in Section 5.B.

For every stage $h \in [H]$, the algorithm keeps a progressively refined estimate of the geometry of the parameter space $\Theta_h$, by maintaining an ever shrinking ellipsoid enclosing $\Theta_h$. This ellipsoid is parametrized by an 'inverse covariance matrix'-like quantity $\mathcal{Q}_h$, determined by $\tilde{\mathcal{O}}(d)$ vectors, which guarantees $\max_{\theta_h \in \Theta_h} \|\theta_h\|_{\mathcal{Q}_h^{-2}} = \tilde{\mathcal{O}}(\sqrt{d})$. Looking at the definition of range in Eq. (89), it is clear that the smaller the ellipsoid becomes, the better estimate we can give for the ranges.

Given some data collected so far and $(Q_h)_{h\in[H]}$, SKIPPYELEANOR computes optimistic estimates of the action-values by calculating an optimistic policy parameter $\bar{\theta}$, as well as a guess $\hat{G}$ to a near-optimal design which is used to estimate the range for the states (due to technical reasons, $\hat{G}$ will guess a near-optimal design for the transformed parameter space $Q_h^{-1}\Theta_h$).

**Algorithm 7** SKIPPYELEANOR

1: **Input:** accuracy $\varepsilon > 0$, failure probability $\zeta > 0$
2: Initialize $m \leftarrow 0$, $m' \leftarrow 0$, $\mathcal{Q}_h = BI$ for $h \in [H]$, $\pi^0 = (s \mapsto 1)$
3: **while** $m' \leq m'_{\max}$ **do**
4: $m \leftarrow m+1$, $m' \leftarrow m'+1$ ▷ $m'$ also counts iterations repeated due to Line 15
5: Estimate optimistic problem parameters $\hat{G}, \bar{\theta}$ by solving Optimization Problem 5.4.10
6: **for** $k \in [H]$ **do**
7: Let $\pi^{mk}$ be the policy defined by SKIPPYPOLICY$(\hat{G}, \bar{\theta}, k)$
8: Sample $n$ trajectories by executing $\pi^{mk}$ from $s_1$ for $n$ episodes
9: Record data $(S_h^{mkj}, A_h^{mkj}, R_h^{mkj})_{h\in[H], j\in[n]}$ and stage-mapping functions $(p^{mkj})_{j\in[n]}$
10: **end for**
11: Solve Optimization Problem 5.4.12 with input $(\hat{G}, \bar{\theta})$, ▷ Consistency check
record its value $x$ (maximum discrepancy), and arguments $v$ (direction) and $i$ (stage).
12: Calculate useful component $w \leftarrow \mathrm{Proj}_{Z(\mathcal{Q},i)}\, v$ ▷ Definition 5.4.2
13: **if** $x >$ discrepancy_threshold **then**
14: $\mathcal{Q}_i \leftarrow \left(\mathcal{Q}_i^{-2} + \mathcal{Q}_i^{-1} w w^\top \mathcal{Q}_i^{-1}\right)^{-\frac{1}{2}}$ ▷ append $\mathcal{Q}_i^{-1} w$ to $C_i$ according to Eq. (91)
15: $m \leftarrow m-1$ ▷ redo this iteration
16: **continue**
17: **end if**
18: **if** average_uncertainty $\leq$ uncertainty_threshold **then**
19: **return** policy $\pi^{mH}$
20: **end if**
21: **end while**

Data is collected by running stochastic versions of skippy policies on the MDP, where the states to be skipped over are determined based on the range estimates; when a state is skipped, an action is selected using a deterministic policy $\pi^0$ that always chooses the first action in every state. To ensure that the estimation problem is smooth in terms of $\hat{G}$, we use a smoothed version of skippy policies, where states are skipped randomly, and the probability of skipping is larger for states with lower ranges, while high-range states are never skipped. Similarly to ELEANOR, we aim to estimate the action-value function of a state-action pair by adding the estimated one-step reward to the estimated value-function of the next state. However, unlike ELEANOR, we would like to do this in the reduced MDP, where the low-range states that are skipped over are removed (and the corresponding transitions are replaced by skippy steps). Since we do not know these states in advance, we run exploratory policies that skip over next states starting from any state: namely, we run SKIPPYPOLICY$(\hat{G}, \bar{\theta}, k)$ for all $k \in [H]$ with a maximum number of unskipped states $k$ (Phase I), and once this is skip budget is exhausted, all remaining states are skipped over by rolling out $\pi^0$ (Phase II), which ensures that we collect enough data at every stage of the MDP to be able to estimate the one-skippy-step reward of any skipping mechanism. Compared to ELEANOR, this introduces an additional loop in Line 6 of SKIPPYELEANOR; see Section 5.D for additional details.

**Algorithm 8** SKIPPYPOLICY

Input: $\hat{G}, \bar{\theta}, k$
Initialize $S_1 \leftarrow s_1$, $j \leftarrow 1$, $\pi^0 \leftarrow (s \mapsto 1)$, stage mapping $p$
**for** $i = 1$ to $H$ **do**
    Compute skip probabilities $\tau_i \leftarrow \tau_{\hat{G}}(S_i)$ and non-skip action $a^+ \leftarrow \pi^+_{\bar{\theta}}(S_i)$ from Eq. (95)
    Sample independently $B_i \sim \text{Bernoulli}(\tau_i)$
    **if** $B_i = 0$ **then** $A_i \leftarrow 1$ ▷ skip (follow $\pi^0$) with probability $1 - \tau_i$
    **else**
        $p(j) \leftarrow i$, $j \leftarrow j + 1$
        **if** $j \le k$ **then** $A_i \leftarrow a^+$ (Phase I) **else** $A_i \leftarrow 1$ (Phase II)
        **end if**
    **end if**
    **if** $i = H$ **then**
        $p(j') = H + 1$ for $j' = j, \ldots, H$
    **end if**
**end for**

For any execution, SKIPPYPOLICY maintains a stage-mapping function $p$, which, for any stage $h$ of the trajectory in the reduced MDP gives the stage index in the original MDP. In other words, $p(j)$ is the stage of the landing state of the $j^{th}$ skippy step.

Finally, we check if the data collected is consistent with our estimates $\hat{G}$ and $\bar{\theta}$, by calculating the maximal discrepancy of the estimates of the action-value difference at the last non-skipped state of $\pi^{mk} = \text{SKIPPYPOLICY}(\hat{G}, \bar{\theta}, k)$ and that of the fixed skipping policy $\pi^0$ in different directions in the parameter space. If the discrepancy is too large for any $k$, we add the discrepancy-maximizing direction to $\mathcal{Q}$ and throw away the data collected in this (i.e., the $m^{th}$) iteration; this is achieved by reducing the iteration counter $m$ by 1. On the other hand, if the discrepancy is small enough, we can guarantee that the gap between the value of $\pi^{mH}$ and $v^\star(s_1)$ scales with how much new information we collected, thus the algorithm can terminate returning this policy if this term is sufficiently small (which it eventually has to be).

The following theorem shows that with high probability, SKIPPYELEANOR finds a near-optimal policy after polynomially many interactions with the MDP. Rhe proof sketch is provided in Section 5.5, while our method and proof strategy is explained from the perspective of ELEANOR in Section 5.D.

**Theorem 5.4.1.** *With probability at least* $1 - \zeta$, SKIPPYELEANOR *interacts with the MDP for at most* $\tilde{\mathcal{O}}\left(H^{11} d^7 / \varepsilon^2\right)$ *many steps, before returning a policy* $\pi$ *that satisfies* $v^\star(s_1) \le v^\pi(s_1) + \varepsilon$.

### 5.4.1. Preconditioning: the enclosing ellipsoid

In this section we give the technical details about the effects of using the matrix $Q_h$ describing an enclosing ellipsoid for $\Theta_h$ (see Lemma 5.4.3) as preconditioning the features.

**Definition 5.4.2** (Valid preconditioning)**.** *$Q = (\mathcal{Q}_h)_{h\in[H]}$ is a valid preconditioning matrix sequence if for all $h \in [H]$*

$$\mathcal{Q}_h = \left(B^{-2}I + \textstyle\sum_{v\in C_h} vv^\top\right)^{-1/2} \tag{91}$$

*for some sequence $C_h = (v_1, \dots, v_n)$ of vectors in $\mathbb{R}^d$ such that for all $1 \le i \le n$,*

$$\sup_{\theta\in\Theta_h} |\langle\theta, v_i\rangle| \le 1 \quad \textit{and} \quad \left\|\left(B^{-2}I + \textstyle\sum_{j=1}^{i-1} v_j v_j^\top\right)^{-\frac{1}{2}} v_i\right\|_2^2 \ge \tfrac{1}{2} \quad \textit{and} \quad \|v\|_2 \le L_3\,, \tag{92}$$

*where $L_3$ is some fixed polynomial of the problem parameters $(d, H, \varepsilon^{-1}, \log\zeta^{-1}, \log L, \log B)$. (see Eq. (122) for its precise value).*

*For a valid preconditioning $\mathcal{Q}$ and some $h \in [H]$, let $Z(\mathcal{Q}, h)$ be the linear subspace spanned by those eigenvectors of $Q$ whose corresponding eigenvalues are at least $L_3^{-2}$. Let $\mathrm{Proj}_{Z(\mathcal{Q},h)}$ be the orthogonal projection matrix onto this subspace.*

Sometimes it will be convenient to *precondition* the features and parameters so that the enclosing ellipsoid is transformed to a ball of controlled radius (as Lemma 5.4.3 will show). To this end, introduce for all $h \in [H]$ and $(s, a, b) \in \mathcal{S}_h \times [\mathcal{A}] \times [\mathcal{A}]$ the following:[15]

$$\begin{gathered}
\varphi_{\mathcal{Q}}(s,a) = \mathcal{Q}_h\varphi(s,a), \qquad \varphi_{\mathcal{Q}}(s,a,b) = \mathcal{Q}_h\varphi(s,a,b) \\
\theta_h^{\mathcal{Q}}(\pi) = \mathcal{Q}_h^{-1}\theta_h(\pi), \qquad \Theta_h^{\mathcal{Q}} = \left\{\theta_h^{\mathcal{Q}}(\pi) : \pi \in \Pi\right\} = \left\{\mathcal{Q}_h^{-1}\theta : \theta \in \Theta_h\right\} \\
\hat{q}^\pi(s,a) = \langle\varphi(s,a), \theta_h(\pi)\rangle = \left\langle\varphi_{\mathcal{Q}}(s,a), \theta_h^{\mathcal{Q}}(\pi)\right\rangle \text{ for all } \pi \in \Pi\,.
\end{gathered} \tag{93}$$

The next lemma (proved in Section 5.F) shows that for all $h \in [H]$, $\mathcal{Q}_h$ defines an enclosing ellipsoid for $\Theta_h$; that is, $\Theta_h \subset \{\theta : \|\theta\|_{\mathcal{Q}_h^{-2}} \le \sqrt{d_1+1}\}$.

**Lemma 5.4.3.** *Let $d_1 = 4d\log(1+16L_3^4B^4) = \tilde{\mathcal{O}}(d)$. Then, for any valid preconditioning $\mathcal{Q}$ and $h \in [H]$,*

$$\sup_{\theta\in\Theta_h} \|\theta\|_{\mathcal{Q}_h^{-2}} = \sup_{\theta\in\Theta_h^{\mathcal{Q}}} \|\theta\|_2 \le \sqrt{d_1+1}\,.$$

15. Note that $Q_h, h \in [H]$ is invertible by construction.

Clearly, every time a new vector is added to $C_h$, the enclosing ellipsoid $\{\theta : \|\theta\|_{\mathcal{Q}_h^{-2}} \le \sqrt{d_1+1}\}$ shrinks (as a positive semidefinite matrix is added to $\mathcal{Q}_h^{-2}$). The following lemma (also proved in Section 5.F) uses an elliptical potential argument to bound the number of times this can happen.

**Lemma 5.4.4.** *For any valid preconditioning $\mathcal{Q}$, for all $h \in [H]$, the length of sequence $C_h$ corresponding to $\mathcal{Q}_h$ according to Definition 5.4.2 is at most $d_1$.*

**Near-optimal design for $\Theta_h^{\mathcal{Q}}$.** As $\mathcal{Q}_h$ only provides an enclosing ellipsoid for $\Theta_h$, we introduce an (unknown) ellipsoid that aligns better with $\Theta_h^{\mathcal{Q}}$. For all $h \in [H]$, fix a set $G_h^{\mathcal{Q}}$ of policies of size $d_0 := 4d\log\log(d)+16$, together with a probability distribution $\rho_h^{\mathcal{Q}}$ on $G_h^{\mathcal{Q}}$, such that $(G_h^{\mathcal{Q}}, \rho_h^{\mathcal{Q}})$ is a near-optimal design for $\Theta_h^{\mathcal{Q}}$ (i.e., satisfying Definition 5.F.1). The existence of such a near-optimal design follows from (Todd, 2016, Part (ii) of Lemma 3.9).

We apply $G_h^{\mathcal{Q}}$ to define a cruder version of range that depends only on a small set of policies, and can therefore be succinctly parametrized to inform SKIPPYPOLICY:

$$\mathrm{range}_{\mathcal{Q}}(s) = \max_{\pi \in G_h^{\mathcal{Q}}} \max_{i,j\in[\mathcal{A}]} \langle \varphi(s,i,j), \theta_h(\pi)\rangle \qquad \text{for all } h \in [H], s \in \mathcal{S}_h\,. \tag{94}$$

$\mathrm{range}_{\mathcal{Q}}$ is easy to estimate, and can be used to bound the range function (proved in Section 5.F):

**Proposition 5.4.5.** *For all $s \in \mathcal{S}$ and $\mathcal{Q} \in PD^H$, $\mathrm{range}(s) \le \sqrt{2d}\,\mathrm{range}_{\mathcal{Q}}(s)$.*

### 5.4.2. Linearly realizable functions

$q^\pi$-realizability (Definition 5.3.2) implies the linearity of many more functions than the action-value functions. In this section we characterize an interesting set of such functions, whose (approximate) linearity plays a crucial role in our algorithm and analysis, as their parameters can be conveniently estimated by least squares using the features. We rely on functions $f : \mathcal{S}_h \to \mathbb{R}$ (for some $h \in [H]$) being small for all states, relative to the states' $\mathrm{range}_{\mathcal{Q}}$-value:

**Definition 5.4.6.** *For any $h \in [H]$, $f : \mathcal{S}_h \to \mathbb{R}$ is $\alpha$-admissible for some $\alpha > 0$ if for all $s \in \mathcal{S}_h$, $|f(s)| \le \mathrm{range}_{\mathcal{Q}}(s)/\alpha$.*

The key observation is that expected (admissible) $f$ values are linearly realizable.

**Lemma 5.4.7** (Admissible-realizability)**.** *If $f : \mathcal{S}_h \to \mathbb{R}$ is $\alpha$-admissible then it is realizable, that is, for all $t \in [h-1]$ and $\pi \in \Pi$, there exists some $\tilde\theta \in \mathbb{R}^d$ with $\left\|\tilde\theta\right\|_2 \le 4d_0B/\alpha$ such that for all $(s,a) \in \mathcal{S}_t \times [\mathcal{A}]$,*

$$\mathbb{E}_{\pi,s,a} f(S_h) \approx_{\eta_0} \left\langle \varphi(s,a), \tilde\theta \right\rangle \qquad \text{where } \eta_0 = 5d_0\eta/\alpha.$$

The proof relies on constructing a set of policies that at states $s \in \mathcal{S}_h$ take a higher value action as opposed to a lower one with a certain probability, configured such that the expected action-value difference of some pairs within the set of policies is (approximately) proportional to $f(s)$. Thus, a linear combination of the action-values of policies in this set are also (approximately) proportional to $f(s)$. The statement of the lemma then follows from setting $\tilde{\theta}$ to the corresponding linear combination of the policies' parameters. The full proof is presented in Section 5.G.

Next, we define matrix-valued functions with a special admissibility guarantee even when the underlying scalar-valued function does not satisfy any non-trivial admissibility criterion. We introduce a *guess* on the near-optimal design parameters that define $\text{range}_{\mathcal{Q}}$ (Eq. (94)) for some valid preconditioning $\mathcal{Q}$:

**Definition 5.4.8.** *For $h \in [2:H]$, fix some arbitrary order of the policies in the set $G_h^{\mathcal{Q}}$ (recall that this set is the support of the near-optimal design for $\Theta_h^{\mathcal{Q}}$). Let the parameter of the $i^{th}$ policy in $G_h^{\mathcal{Q}}$ be $\vartheta_h^i$ for $i \in [d_0]$. Call a "guess" of these parameters $\hat{G} = (\hat{G}_h)_{h\in[2:H]} = (\hat{\vartheta}_h^i)_{h\in[2:H],i\in[d_0]}$ "valid", if for all $h \in [2:H], i \in [d_0]$, $\hat{\vartheta}_h^i \in \mathcal{B}(\sqrt{d_1+1})$. Let the set of valid guesses be $\mathbf{G}$.*[16] *By Lemma 5.4.3, $(\vartheta_h^i)_{h\in[2:H],i\in[d_0]} \in \mathbf{G}$, that is, it is a valid guess, and we call this the "correct" guess.*

From a guess $\hat{G} = (\hat{\vartheta}_h^i)_{h\in[2:H],i\in[d_0]}$ we can calculate corresponding guesses of the $\text{range}_{\mathcal{Q}}$-values:

$$\text{range}_{\mathcal{Q}}^{\hat{G}}(s) = \max_{k\in[d_0]} \max_{i,j\in[\mathcal{A}]} \left\langle \varphi_{\mathcal{Q}}(s,i,j), \hat{\vartheta}^k_{\text{stage}(s)} \right\rangle \qquad \text{for all } h \in [2:H], s \in \mathcal{S}_h\,.$$

Note that for any $h \in [2:H]$ and $s \in \mathcal{S}_h$, $\text{range}_{\mathcal{Q}}^{\hat{G}}(s) = \text{range}_{\mathcal{Q}}(s)$ if $\hat{G}$ is the correct guess for stage $h$.

Let $\bar{\varphi}_{\mathcal{Q}}(s)$ be the unit vector in the direction of the largest feature difference between actions in $s$ and the zero vector if all feature vectors are the same (see Eq. (114) for a formal definition). Then, for any $\hat{G} \in \mathbf{G}$, $h \in [2:H]$, and $f : \mathcal{S}_h \to [-H, H]$, let

$$\mathbf{f}(s) = \bar{\varphi}_{\mathcal{Q}}(s)\bar{\varphi}_{\mathcal{Q}}(s)^\top \min\left\{1, \text{range}_{\mathcal{Q}}^{\hat{G}}(s)\frac{\sqrt{2d}H}{\varepsilon}\right\} f(s) \quad \text{for } s \in \mathcal{S}_h\,.$$

For such $\mathbf{f} : \mathcal{S}_h \to \mathbb{R}^{d\times d}$, we adopt the notation $a^\top \mathbf{f} b$ for any $a, b \in \mathbb{R}^d$ to denote the function $s \in \mathcal{S}_h \mapsto a^\top \mathbf{f}(s) b$, and similarly, $\text{Tr}(\mathbf{f})$ to denote the function $s \in \mathcal{S}_h \mapsto \text{Tr}(\mathbf{f}(s))$.

Let $\text{Proj}_{\|(\mathcal{Q},h)}$ be the projection matrix onto the linear subspace spanned by those eigenvectors of the design matrix $V(G_h^{\mathcal{Q}}, \rho_h^{\mathcal{Q}})$ (defined in Eq. (112)) whose corresponding eigenvalues are at

16. Note that while $G_h^{\mathcal{Q}}$ contains policies, $\mathbf{G}$ and its elements (commonly denoted by $\hat{G}$) contain policy parameter vectors.

least $\gamma$ (for some $\gamma > 0$ specified in Section 5.B). Intuitively, this is the subspace where $\Theta_h^{\mathcal{Q}}$ has a sufficiently large width. Let $\mathrm{Proj}_{\perp(\mathcal{Q},h)}$ be the projection to the orthogonal complement subspace. For any $v \in \mathbb{R}^d$, we write $v_{\parallel(\mathcal{Q},h)}$ and $v_{\perp(\mathcal{Q},h)}$ for $\mathrm{Proj}_{\parallel(\mathcal{Q},h)} v$ and $\mathrm{Proj}_{\perp(\mathcal{Q},h)} v$, respectively.

We are now ready to state our special admissibility guarantee, which is proved in Section 5.G. Let $\alpha = \tilde{\mathcal{O}}(\varepsilon/(d^{1.5}H^2))$ be as in Eq. (103).

**Lemma 5.4.9.** *For any $h \in [2:H]$, $\hat{G} \in \mathbf{G}$, any function $\mathbf{f}$ constructed as above from some $f : \mathcal{S}_h \to [-H, H]$, and any $v, w \in \mathcal{B}(1)$, $v_{\parallel(\mathcal{Q},h)}^\top \mathbf{f} w$ is $\alpha$-admissible. Furthermore, if $\hat{G} = (\vartheta_h^i)_{h\in[2:H],i\in[d_0]}$ (the correct guess), $\mathrm{Tr}(\mathbf{f})$ is also $\alpha$-admissible.*

### 5.4.3. Least-squares targets and Optimization Problem 5.4.10

Recall that SkippyEleanor estimates action-values of states by first adding the estimated one-step reward and the estimated value-function of the next state in the reduced MDP (where low-range states are skipped). Due to the linearity of $q^\pi$-values, these can be used as target variables of a least-squares estimator to estimate the policy parameters. This estimator is only guaranteed to be accurate if the right (low-range) states are skipped; otherwise, we will argue in Section 5.4.4 that a discrepancy is detected and it is handled by changing the preconditioning $\mathcal{Q}$. Finally, to ensure optimism, we select parameter estimates that lead to the largest estimated policy values. The whole estimation process leads to Optimization Problem 5.4.10, which we define in this section along with the functions that it uses as least-square targets. Each estimation is for a particular stage $h$ and may use the estimates $\bar{\theta}_i$ of Optimization Problem 5.4.10 for stages $i > h$. In this subsection, we consider the $m^{\text{th}}$ iteration of the optimization called by SkippyEleanor, and consider $Q$ fixed. As a shorthand, we introduce the following notation for $l \in [m], j \in [n], k \in [H]$:

$$\mathrm{p}(lkj) = p^{lkj}(k) \qquad \text{as recorded in Line 9 of Algorithm 7, and}$$
$$S_{\mathrm{p}(k)}^{lkj} = S_{p^{lkj}(k)}^{lkj},\ A_{\mathrm{p}(k)}^{lkj} = A_{p^{lkj}(k)}^{lkj},\ R_{\mathrm{p}(k)}^{lkj} = R_{p^{lkj}(k)}^{lkj},\ \varphi_t^{lkj} = \varphi(S_t^{lkj}, A_t^{lkj}),\ \varphi_{\mathrm{p}(k)}^{lkj} = \varphi(S_{\mathrm{p}(k)}^{lkj}, A_{\mathrm{p}(k)}^{lkj}).$$

We collect the set of $(l,k,j)$ tuples for which the $k^{\text{th}}$ skippy step lands at stage $t$, for $t \in [H]$, as

$$\mathbf{I}^m(t) = \{(l,k,j) : l \in [m-1], j \in [n], k \in [H], \mathrm{p}(lkj) = t\}$$

Note in particular that here $l \in [m-1]$, so $\mathbf{I}^m$ only considers data collected prior to iteration $m$.

To estimate the parameters $\hat{G}$ and $\bar{\theta}$, we consider (simulated) trajectories of SkippyPolicy starting from stage $t$. For simplicity, we suppress the dependence of quantities on $\hat{G}$ and $\bar{\theta}$, which

will be brought back later. The skipping probability $1-\tau$, the policy $\pi^+$ (to be also used in SKIPPY-POLICY), and corresponding clipped action-value estimates are defined as

$$\begin{aligned} \tau(s) &= \min\left\{1, \mathrm{range}_{\mathcal{Q}}^{\hat{G}}(s)\frac{\sqrt{2d}H}{\varepsilon}\right\} && \text{if } \mathrm{stage}(s) > 1, \text{ and } \tau(s_1) = 1; \\ \pi^+(s_i) &= \arg\max_{a\in[\mathcal{A}]} \left\langle \varphi(s_i,a), \bar{\theta}_i \right\rangle, && C(s_i) = \mathrm{clip}_{[0,H]}\left\langle \varphi(s_i,\pi^+(s_i)), \bar{\theta}_i \right\rangle. \end{aligned} \tag{95}$$

Let $s_i{\rightarrow} = (s_i, a_i, r_i, \ldots, s_H, a_H, r_H) \in \mathcal{S}_i \times [\mathcal{A}] \times [0,1] \times \cdots \times [0,1]$ be any ending of a trajectory. For $s_{t+1}{\rightarrow}$, let $I$ be the (random) index of the first state that is *not* skipped by SKIPPYPOLICY with the above $\tau$ (or $H+1$, if such an index does not exist). Then the estimated policy value of SKIPPYPOLICY from stage $t$ is

$$\mathbb{E}_I\left[\textstyle\sum_{u=t}^{I-1} r_u + \mathbb{1}\{I < H+1\} C(s_I)\right],$$

the sum of rewards along the skipped states plus the policy-value estimate from stage $I$. It follows from Corollary 5.4.11 below (proved based on Lemma 5.4.9) that if $\mathrm{range}_{\mathcal{Q}}^{\hat{G}}$ is an accurate estimate of $\mathrm{range}_{\mathcal{Q}}$, then this quantity decomposes into terms that are linearly expressible using the features. Therefore, we use such quantities as least-square targets. Indeed, writing out the expectation, we can re-express the estimated policy value as the sum of all rewards $\sum_{u=t}^{H} r_u$ plus a correction term $E^{\rightarrow}(s_{t+1}{\rightarrow})$ defined as

$$E^{\rightarrow}(s_i{\rightarrow}) = \sum_{j=i}^{H} D(s_j{\rightarrow})\tau(s_j)\prod_{j'=i}^{j-1}(1-\tau(s_{j'})) \text{ where } D(s_i{\rightarrow}) = C(s_i) - \sum_{u=i}^{H} r_u \quad \text{for } i > 1. \tag{96}$$

The next optimization problem aims to find optimistic parameters yielding the largest estimated action-value function for $s_1$, where $\bar{\theta}$ is in the confidence ellipsoid of the least-squares estimates $\hat{\theta}$.

**Optimization Problem 5.4.10** (for iteration $m$)**.** *With $\beta$ defined in Section 5.B (emphasizing the dependence of functions defined above on $\hat{G}$ and $\bar{\theta}$ by adding them as subscripts):*

$$\arg\max_{\hat{G}\in\mathbf{G},\, \bar{\theta}_t \in \mathcal{B}(4d_0HB/\alpha) \text{ for } t\in[H]} C_{\hat{G}\bar{\theta}}(s_1) \qquad \textit{subject to, for all } t \in [H]$$

$$X_{mt} = \lambda I + \sum_{lkj\in\mathbf{I}^m(t)} \varphi_t^{lkj}\varphi_t^{lkj\top},\ \left\|\bar{\theta}_t - \hat{\theta}_t\right\|_{X_{mt}} \le \beta H,\ \hat{\theta}_t = X_{mt}^{-1}\sum_{lkj\in\mathbf{I}^m(t)} \varphi_t^{lkj}\underbrace{\left(E^{\rightarrow}_{\hat{G}\bar{\theta}}(S_{t+1}^{lkj},\ldots,R_H^{lkj}) + \sum_{u=t}^{H} R_u^{lkj}\right)}_{\textit{least-squares target}}$$

Since our realizability results in Section 5.4.2 only apply to functions defined at a given stage (as only memoryless policies are $q^\pi$-realizable), to be able to show that the least-squares targets are

linearly realizable, we first decompose $E^{\rightarrow}(s_i\rightarrow)$ ($i \in [2:H]$) to directly express the effect of each stage in the trajectory (backwards): defining $E(s_i\rightarrow) = E^{\rightarrow}(s_i\rightarrow) - E^{\rightarrow}(s_{i+1}\rightarrow)$ (for convenience, we use the notation $E^{\rightarrow}(s_{H+1}\rightarrow) = 0$), we easily obtain

$$E^{\rightarrow}(s_i\rightarrow) = \textstyle\sum_{j=i}^{H} E(s_j\rightarrow) \qquad \text{and} \qquad E(s_i\rightarrow) = \tau(s_i)\big(D(s_i\rightarrow) - E^{\rightarrow}(s_{i+1}\rightarrow)\big). \tag{97}$$

Next we define matrix-valued functions, whose trace equals $E(s_i\rightarrow)$, that have the same form as $\mathbf{f}$ in Section 5.4.2, for which Lemma 5.4.9 applies. This is crucial in establishing optimism of Optimization Problem 5.4.10, as well as learning from instances where we detect that $E^{\rightarrow}$ is not realizable in Optimization Problem 5.4.12. To this end, let

$$F(s_i\rightarrow) = \bar{\varphi}_{\mathcal{Q}}(s_i)\bar{\varphi}_{\mathcal{Q}}(s_i)^\top E(s_i\rightarrow) \qquad \text{and} \qquad \bar{F}(s_i) = \mathbb{E}_{\pi^0,s_i}[F(s_i, A_i, \ldots, R_H)] \quad \text{for } s_i \in \mathcal{S}_i\,.$$

Let $\bar{\Theta} = (\mathcal{B}(4d_0HB/\alpha))^H$ denote the base set for the variables $\bar{\theta}_t$ in Optimization Problem 5.4.10. As $\bar{F}$ is of the same form as $\mathbf{f}$, we can apply Lemma 5.4.9 and then Lemma 5.4.7 to arrive at the following:

**Corollary 5.4.11.** *For any $\hat{G} \in \mathbf{G}$, $\bar{\theta} \in \bar{\Theta}$, $v, w \in \mathcal{B}(1)$, and for any $t \in [H-1]$, $i \in [t+1:H]$, there exists some $\tilde{\theta}_{ti} \in \mathbb{R}^d$ with $\|\tilde{\theta}_{ti}\|_2 \le 4d_0B/\alpha = 1/\sqrt{\lambda}$ such that for all $(s,a) \in \mathcal{S}_t \times [\mathcal{A}]$, where $\eta_0$ is defined in Lemma 5.4.7.*

$$\mathbb{E}_{\pi^0,s,a}\left[v_{\|(\mathcal{Q},i)}^\top \bar{F}_{\hat{G}\bar{\theta}}(S_i)w\right] \approx_{\eta_0} \left\langle \varphi(s,a), \tilde{\theta}_{ti}\right\rangle. \tag{98}$$

*Furthermore, if $\hat{G}$ is the correct guess, there exists some $\tilde{\theta}'_{ti} \in \mathbb{R}^d$ with $\|\tilde{\theta}'_{ti}\|_2 \le 4d_0B/\alpha$ such that for all $(s,a) \in \mathcal{S}_t \times [\mathcal{A}]$, $\mathbb{E}_{\pi^0,s,a}[E_{\hat{G}\bar{\theta}}(S_i, \ldots, R_H))] = \mathbb{E}_{\pi^0,s,a}[\mathrm{Tr}(\bar{F}_{\hat{G}\bar{\theta}}(S_i))] \approx_{\eta_0} \left\langle \varphi(s,a), \tilde{\theta}'_{ti}\right\rangle$.*

### 5.4.4. Checking consistency

Considering the $m^{\text{th}}$ iteration of SkippyEleanor, we want to verify if the estimated targets of Optimization Problem 5.4.10 are accurate (and learn if a discrepancy is detected), by using Corollary 5.4.11 on the targets' decomposition into $F$-functions. We filter the data collected in the $m^{\text{th}}$ iteration with the indicator $\tilde{c}^j_{ki} = \mathbb{1}\{\mathrm{p}(mkj) < i\}$ for $j \in [n]$, $k \in [H+1]$, $i \in [H+1]$, and further constrain this by another indicator $c^j_{ki}$ (defined in Section 5.B) that requires the data-point's least-squares uncertainty term to be sufficiently low, and the prediction non-negative (the contribution of the rest of the data will be analyzed separately). Next, we define the least-squares solution for estimating the matrix-valued $F$, as well as the empirical average prediction and realization of $F$ on the data collected in the $m^{\text{th}}$ round. For any $i \in [2:H]$, $k \in [i-1]$ (recall that $\otimes$ denotes the tensor

product):

$$\begin{aligned}
\hat{\theta}^{ti}_{\hat{G}\bar{\theta}} &= X^{-1}_{mt} \sum_{lkj \in \mathbf{I}^m(t)} \varphi^{lkj}_t \otimes F_{\hat{G}\bar{\theta}}(S^{lkj}_i, \ldots, R^{lkj}_H) \qquad \text{for } t \in [i-1] \\
y^{ki}_{\hat{G}\bar{\theta}} &= \frac{1}{n} \sum_{j \in [n]} c^j_{ki} \varphi^{mkj\top}_{\mathrm{p}(k)} \hat{\theta}^{\mathrm{p}(mkj),i}_{\hat{G}\bar{\theta}} \qquad \hat{F}^{ki}_{\hat{G}\bar{\theta}} = \frac{1}{n} \sum_{j \in [n]} c^j_{ki} F_{\hat{G}\bar{\theta}}(S^{mkj}_i, \ldots, R^{mkj}_H)
\end{aligned} \tag{99}$$

In Section 5.E.1, it is established via the usual least-squares analysis techniques and covering arguments, that with high probability the norm of the product of the matrix $y^{ki}_{\hat{G}\bar{\theta}} - \hat{F}^{ki}_{\hat{G}\bar{\theta}}$ and the projection matrix $\mathrm{Proj}_{\|(\mathcal{Q},i)}$ is small (Lemmas 5.E.2 and 5.E.3). The next optimization problem tests if this is true in arbitrary directions:

**Optimization Problem 5.4.12** (Consistency check)**.** *Input:* $(\hat{G}, \bar{\theta})$

$$\underset{k \in [H-1], i \in [k+1:H], v \in \mathbb{R}^d : \|v\|_2 = 1}{\arg\max} v^\top \left( y^{ki}_{\hat{G}\bar{\theta}} - \hat{F}^{ki}_{\hat{G}\bar{\theta}} \right) v$$

Lemma 5.E.1 shows that the projection $w = \mathrm{Proj}_{Z(\mathcal{Q},i)} v$ is close to $v$, where $v$ is the outcome of Optimization Problem 5.4.12. Also, Lemmas 5.E.1–5.E.3 imply that if the consistency check fails (i.e., Line 14 is executed because the value of Optimization Problem 5.4.12 is large), then $w$ aligns well with the subspace $\mathrm{Proj}_{\perp(\mathcal{Q},i)}$ projects to, and therefore $\mathcal{Q}$ stays a valid preconditioning after appending $w$ to the list of values $\mathcal{Q}$ is calculated from (Lemma 5.E.4). Thus, $\mathcal{Q}$ is always a valid preconditioning.

## 5.5. Proof overview

The proof of Theorem 5.4.1 is presented in Section 5.E. It is composed of the following main steps: First, we bound the number of times the consistency check can fail (i.e., Line 14 is executed) by Lemma 5.4.4. Combining this with Lemma 5.E.5, an elliptical potential argument bounding the number of times the average uncertainty can be large (these are the only two ways that the main iteration can continue) implies a sample-complexity result for SkippyEleanor (Corollary 5.E.6). Having limited the number of times the consistency check can fail, we derive guarantees regarding the performance of the policy returned by the algorithm: Via an induction argument (Lemma 5.E.8) we show Corollary 5.E.9, which shows that with high probability the difference between the optimization value of Optimization Problem 5.4.10, $C_{\hat{G},\bar{\theta}}(s_1)$ and $v^{\pi^{mH}}$ scales with the average uncertainty term $\sum_{i=1}^{H} \bar{\sigma}^m_k$. Thus, they are close when SkippyEleanor returns in Line 19. This is

complemented with the *optimism* property proved in Lemma 5.E.10, stating that the optimization value $C_{\hat{G},\bar{\theta}}(s_1)$ is close to $v^\star(s_1)$. Combined, this proves Theorem 5.4.1.

## 5.6. Future work

Since we are not aware of a computationally efficient implementation of SKIPPYELEANOR, it remains an open question whether the problem of learning near-optimal policies from online interactions with a $q^\pi$-realizable MDP (Problem 5.3.3) is possible if the computational resources as well as the query complexity are bounded by a polynomial in the relevant parameters. One approach is to replace ELEANOR with LSVI-UCB as the underlying algorithm, as the latter, despite having worse query complexity, has a computationally efficient implementation (Jin et al., 2020b). The challenge is to compute the optimal solution for the parameter $\hat{G}$ in Optimization Problem 5.4.10. This parameter interacts with the least-squares targets in a highly nonlinear way. We have been unable to derive a computationally efficient approximation that has an additive instead of a multiplicative approximation error (additive errors increase linearly in $H$, while multiplicative errors increase exponentially). Alternatively, it may be possible to show a computational hardness result for Problem 5.3.3 by e.g., reducing it to the satisfiability problem. These are left for future work. Our work on the realizability of auxiliary functions (Section 5.4.2) may be of independent interest for designing provably efficient algorithms for related problem settings, e.g., the setting of $q^\pi$-realizability in batch RL, where the data collection is not controlled.

# Appendix

## 5.A. Notation

Let $\mathbb{R}$, $\mathbb{N}$, and $\mathbb{N}^+$ denote the set of reals, non-negative and positive integers, respectively. For $i \in \mathbb{N}^+$, let $[i] = \{1, \ldots, i\}$; for another positive integer $j$, let $[i : j] = \{i, \ldots, j\}$ if $i \leq j$, and $[i : j] = \{\}$ otherwise. For $a, b, x \in \mathbb{R}$, let $\text{clip}_{[a,b]}(x) = \min\{\max\{x, a\}, b\}$ and let $\lceil x \rceil$ denote the smallest integer i such that $i \geq x$. Let $\mathbf{0}$ be the all-0 vector in $\mathbb{R}^d$ and $I$ the $d$-dimensional identity matrix. For a (square) matrix $V$, let $V^\dagger$ denote its Moore-Penrose inverse, and $\text{Tr}(V)$ denote its trace. Let PD (and PSD) denote the set of positive definite (and positive semi-definite, respectively) matrices in $\mathbb{R}^{d \times d}$. For some $A \in \text{PSD}$ let $A^{\frac{1}{2}}$ denote the unique matrix $B \in \text{PSD}$ such that $A = BB$. For $V \in \text{PD}$ and $x \in \mathbb{R}^d$, let $\|x\|_V^2 = x^\top G x$. For matrices $A$ and $B$, we say that $A \succeq B$ (or $A \preceq B$) if $B - A$ (or $A - B$, respectively) is positive semidefinite. $\text{Ker}(A)$ and $\text{Im}(A)$ are the kernel (or null space), and image, respectively, of matrix $A$. For compatible vectors $x, y$, let $\langle x, y \rangle$ be their inner product: $\langle x, y \rangle = x^\top y$. We write $y \otimes A$ for the tensor product between $y$ and matrix $A$, and then $\langle x, y \otimes A \rangle = \langle x, y \rangle A$. Where $\mathcal{Q}$ and $h$ are obvious from the context, we write $v_\parallel$ and $v_\perp$ for $v_{\parallel(\mathcal{Q},h)}$ and $v_{\perp(\mathcal{Q},h)}$, respectively. Throughout the chapter, we omit commas between quantities in subscripts or superscript for clarity of presentation, for example, by writing $A_{bc}$ for $A_{b,c}$.

For the big-Oh notation $\mathcal{O}$, we introduce its counterpart $\tilde{\mathcal{O}}$ that hides logarithmic factors of the problem parameters $(d, H, \varepsilon^{-1}, \zeta^{-1}, L, B)$.

## 5.B. Parameters of Algorithm 7

$$n = \tilde{\mathcal{O}}\left(d^5 H^6/\varepsilon^2\right) \qquad \text{(for precise value see Eq. (129))}$$

$$\omega = 7(d_1+1)+7/3 = \tilde{\mathcal{O}}(d) \tag{100}$$

$$\gamma^{-1} = 8d = \tilde{\mathcal{O}}(d) \tag{101}$$

$$\beta = \tilde{\mathcal{O}}(H^{1.5}d) \qquad \text{(for precise value see Eq. (123))} \tag{102}$$

$$\alpha^{-1} = \frac{\sqrt{2d}\sqrt{d_1+1}H^2}{\sqrt{\gamma}\varepsilon} = \tilde{\mathcal{O}}(d^{1.5}H^2/\varepsilon) \tag{103}$$

$$\lambda^{-1} = (4d_0 B/\alpha)^2 \tag{104}$$

$$m_{\max} = \beta^2 \log\left(1+\frac{HmnL^2}{d\lambda}\right)+1 = \tilde{\mathcal{O}}\left(H^3 d^2\right)$$

$$m'_{\max} = m_{\max} + Hd_1 = \tilde{\mathcal{O}}(H^3 d^2)$$

$$\bar{\sigma}_k^m = \frac{1}{n}\sum_{j\in[n]} \tilde{c}^j_{k,H+1} \min\left\{2(\beta\omega dH)^{-1}, \left\|\varphi^{mkj}_{\mathrm{p}(k)}\right\|_{X^{-1}_{m,\mathrm{p}(mkj)}}\right\} \qquad \text{for } k\in[H] \tag{105}$$

(106)

$$c^j_{ki} = \mathbb{1}\left\{\mathrm{p}(mkj) < i \text{ and } \left\|\varphi^{mkj}_{\mathrm{p}(k)}\right\|_{X^{-1}_{m,\mathrm{p}(mkj)}} < 2(\beta\omega dH)^{-1} \text{ and } \left\langle \varphi^{mkj}_{\mathrm{p}(k)}, \bar{\theta}_{\mathrm{p}(mkj)}\right\rangle \geq 0\right\} \tag{107}$$

$$\text{average\_uncertainty} = \sum_{k=1}^{H} \bar{\sigma}_k^m$$

$$\text{uncertainty\_threshold} = \varepsilon/(dH^2\beta\omega)$$

$$\text{discrepancy\_threshold} = \bar{\sigma}_k^m \beta\omega + 3\frac{\varepsilon}{dH^2}$$

Assumption on the maximum discrepancy:

$$\eta \leq \frac{\alpha}{10d_0}\min\left\{\varepsilon/(dH^3\omega), 1/\sqrt{m'_{\max}nH}\right\} = \tilde{\mathcal{O}}\left(\frac{\varepsilon^2}{d^6H^8}\right) \tag{108}$$

## 5.C. Proof of Proposition 5.3.4

***Proof of Proposition 5.3.4, and the MDP conversion argument.*** First, for *(i)*, we show the linearity of rewards with $\theta_1, \dots, \theta_H$. For this take any $h \in [H]$. Fix any policy $\pi \in \Pi$ and let $\bar{\theta}_h \in \mathcal{B}(B)$ be such that for all $(s,a) \in \mathcal{S}_h \times [\mathcal{A}]$, $q^\pi(s,a) \approx_\eta \left\langle \varphi(s,a), \bar{\theta}_h \right\rangle$ (the existence of such a $\bar{\theta}$ follows from Definition 5.3.2). If $h = H$, $\mathbb{E}_{R \sim \mathcal{R}(s,a)}[R] = q^\pi(s,a)$, so $\theta_H = \bar{\theta}_H$ satisfies Definition 5.3.1. For $h < H$, let $f : \mathcal{S}_{h+1} \to \mathbb{R}$ be defined as $f(s) = v^\pi(s)$. Fix an arbitrary $\mathcal{Q} \in \mathrm{PD}^H$, e.g., $\mathcal{Q} = (I, \dots, I)$. Since $v^\pi(s) \in [0, H]$ and $\mathrm{range}_{\mathcal{Q}}(s) \geq \mathrm{range}(s)/\sqrt{2d} \geq \alpha/\sqrt{2d}$ by Proposition 5.4.5, $f$ is $\alpha/(\sqrt{2d}H)$-admissible, and therefore by Lemma 5.4.7 we can take $\tilde{\theta}_h \in \mathcal{B}(4Hd_0\sqrt{2d}B/\alpha)$ such that for all $(s,a) \in \mathcal{S}_h \times [\mathcal{A}]$,

$$\mathbb{E}(v^\pi(S_{h+1}) \,|\, s,a) \approx_{\sqrt{2d}H\eta_0} \left\langle \varphi(s,a), \tilde{\theta}_h \right\rangle,$$

where, as before, $\eta_0 = 5d_0\eta/\alpha$. Since

$$\mathbb{E}_{R \sim \mathcal{R}(s,a)}(R) = q^\pi(s,a) - \mathbb{E}(v^\pi(S_{h+1}) \,|\, s,a),$$

letting $\theta_h = \bar{\theta}_h - \tilde{\theta}_h$ satisfies *(i)* of Definition 5.3.1 with $\kappa = \eta + \sqrt{2d}H\eta_0 = \eta + 5H\sqrt{2d}d_0\eta/\alpha$.

To show *(ii)*, take any $f : \mathcal{S} \to [0, H]$ and $h \in [H-1]$. As before, $f$ is $\alpha/(\sqrt{2d}H)$-admissible, therefore Lemma 5.4.7 immediately provides $\theta'_h$ satisfying the required conditions.

Therefore, the MDP is shown to be linear with misspecification $\eta + \sqrt{2d}H\eta_0$, and parameter bound $B(4Hd_0\sqrt{2d}/\alpha + 1)$. ■

***Sketch of the $q^\pi$-to-linear MDP conversion argument.*** We elaborate on the conversion to linear MDP mechanism presented in Section 5.3. As the basis of this argument is that an idealistic range-determining oracle is present, we note that this argument only serves as intuition and is otherwise tangential to our proof. Instead of a direct approach of learning this oracle, our proof argues that learning about this oracle happens whenever there is a need (performance shortfall) for it. A formal reduction to linear MDPs given this oracle however is fairly straight-forward but cumbersome, with the caveat that the linear MDP will end up with $dH$ (instead of $d$) dimensional features. One would proceed by copying the features of each state $s$ in stage $h$ into the $h^{\text{th}}$ chunk of size $d$ of this vector of size $dH$ (the rest of the vector remains zero). A similar transformation is applied to all $\theta_h(\pi)$. Then, $H$ copies are made of each high-enough-range state, with all possible stages (but keeping the feature vectors). These will be the states of the new MDP we construct. When a transition from state $s$ leads to skipped states, the linear MDP returns with the copy of the first non-

skipped state that has a stage counter of stage$(s)+1$, so that in this linear MDP the stage numbers are consecutive (as required by our definitions). $q^\pi$-realizability of this modified MDP is easy to show, and (as it has no low-range states) Proposition 5.3.4 can be used to show that the modified MDP is linear. To account for the fact that this new MDP may finish an episode in fewer than $H$ steps due to the skips, we add a special, zero-reward, self-transitioning state called "episode-over". To ensure that the MDP stays linear, we extend the feature vectors of each state by a scalar 1, and a scalar indicator of being in this state, with all original features of the "episode-over" state defined to be zero. It is easy to see that this construction leads to a linear MDP with the desired action-value functions.

## 5.D. Intuition behind our method and proof strategy from the perspective of ELEANOR (Zanette et al., 2020b)

The starting point of our method is the ELEANOR algorithm, which is designed for linear MDPs. Similarly to SKIPPYELEANOR, ELEANOR solves an optimistic optimization problem inside a loop. The optimization problem computes optimistic estimates $\bar{\theta}_t$ of the parameters of the MDP simultaneously for all $t \in [H]$, and in each iteration of the loop, more data is collected according to the policy that is optimal for the MDP defined by the estimated parameters. Initial estimates $\hat{\theta}_t$ are computed via solving least-squares problems whose covariates are the features corresponding to state-action pairs $(S_t, A_t)$ from all the data collected so far, while the corresponding least-squares targets are computed as the sum of the immediate reward $R_t$ and the estimated value for $S_{t+1}$, computed from $\bar{\theta}_{t+1}$. $\bar{\theta}_t$ is then optimistically chosen as the solution of the optimization problem, in the neighborhood (confidence ellipsoid) of $\hat{\theta}_t$, the solution to this least-squares problem. It is shown that this optimistic choice of estimates results in an optimistic estimate of the value of $v^\star$ of the initial state, and the regret is upper bounded in terms of the sum of elliptic potentials of the covariates.

This argument appears in our analysis too, with minor modifications due to our PAC-like setting (instead of aiming to bound the regret), leading to our final-iteration condition of Line 18 in Algorithm 7. Our Optimization Problem 5.4.10 is similar to ELEANOR's, and the parameters $\bar{\theta}_t$ and $\hat{\theta}_t$ have the same meaning. A key difference between the optimization problems of ELEANOR and SKIPPYELEANOR are how the least-squares targets are determined. For ELEANOR, it is the sum of the immediate reward $R_t$ and its estimated value for $S_{t+1}$); with this target, only one on-policy rollout is required for each episode in order to get the least-squares parameter estimate for all $H$ stages. In contrast, our least-squares targets are formed as the sum of $R_t + \ldots + R_{t+i}$ and the estimated value

for $S_{t+i+1}$, where $i$, the number of stages "skipped", depends on the guess $\hat{G}$. The guess $\hat{G}$ is selected only in Optimization Problem 5.4.10, and we do not know its value at the time of data collection, so we cannot know which stages will have to be skipped for each rollout. Therefore, (i) we need access to the rewards of the current policy at any stage (similarly to ELEANOR), and hence we run the current policy to any stage (including the last one); and (ii) perform rollouts with the fixed policy $\pi^0$ (from any stage) to be able to estimate the reward $R_t + \ldots + R_{t+i}$ collected while skipping over $i$ stages (for any $i$). To ensure this happens for every stage, we start Phase II from every stage $k$, resulting in the additional for loop in Line 6 of Algorithm 7 compared to ELEANOR. Finally, the randomization in Phase I is applied to make the optimization problem smooth, as described in Section 5.4.

One could analyze this algorithm similarly to the analysis of ELEANOR if it were not for the fact that the least-squares targets we just introduced are not realizable in general. We can, however, prove the realizability of certain components of the matrix-valued version of these targets, $F$ (Lemma 5.4.9 and Corollary 5.4.11). This enables us to detect when the realizability of our least-squares targets fail, measure the direction (component) of the largest error, and learn from that. This is the job of Optimization Problem 5.4.12: $\hat{F}^{ki}_{\hat{G}\bar{\theta}}$ corresponds to the matrix-valued empirical measurements of $F$, while the $y^{ki}_{\hat{G}\bar{\theta}}$ are the average predictions of the same quantities. If the targets are realizable, which happens if we manage to skip the right number of stages), these matrices are very close; if not, the direction of their largest discrepancy tells us something about $\perp(Q, i)$, and allows us to learn.

Optimism ties all this together: either there is no shortfall between predicted and measured $q$-values (and we are done) or we grow the elliptical potential of $X$ (the two cases present in the analysis of ELEANOR, Zanette et al. (2020b)), or we grow the elliptical potential of $\mathcal{Q}$ (the new case due to the lack of realizability guarantees).

## 5.E. Proof of Theorem 5.4.1

In this section we present the proof of Theorem 5.4.1. Recall that some quantities are defined in Section 5.B.

### 5.E.1. Checking consistency

We introduce some lemmas to establish the required guarantees of the consistency checker. Their proofs, which rely on the usual least squares analysis techniques and covering arguments, are presented in Section 5.H.

**Lemma 5.E.1.** *Let $(k,i,v)$ be the outcome of Optimization Problem 5.4.12 any time during the execution of* SkippyEleanor*, and let $w = \mathrm{Proj}_{Z(\mathcal{Q},i)} v$ as in the algorithm. Then,*

$$w^\top \left(y^{ki}_{\hat{G}\bar{\theta}} - \hat{F}^{ki}_{\hat{G}\bar{\theta}}\right) w \geq v^\top \left(y^{ki}_{\hat{G}\bar{\theta}} - \hat{F}^{ki}_{\hat{G}\bar{\theta}}\right) v - \frac{\varepsilon}{dH^2\omega}$$

**Lemma 5.E.2.** *There is an event $\mathcal{E}_1$ that happens with probability at least $1-\zeta$, such that under $\mathcal{E}_1$, during the execution of* SkippyEleanor*, when the beginning of any iteration (Line 5) is executed, for any $t \in [H-1]$, $i \in [t+1:H]$, for any $\hat{G} \in \mathbf{G}$, $\bar{\theta} \in \bar{\Theta}$, and $v, w \in \mathcal{B}(1)$, for all $(s,a) \in \mathcal{S}_t \times [\mathcal{A}]$,*

$$|v_\parallel^\top \left(\varphi(s,a)^\top \hat{\theta}^{ti}_{\hat{G}\bar{\theta}} - \mathbb{E}_{\pi^0,s,a} \bar{F}_{\hat{G}\bar{\theta}}(S_i)\right) w| \leq \|\varphi(s,a)\|_{X_{mt}^{-1}} \beta + \frac{\varepsilon}{dH^2\omega},$$

*where $\bullet_\parallel$ denotes $\bullet_{\parallel(\mathcal{Q},i)}$.*

The next lemma uses the average least-squares predictions' (capped) uncertainty term $\bar{\sigma}^m_k$ (defined in Eq. (105)), where the average is taken over predictions from the state-action pair where Phase I of SkippyPolicy$(\cdot,\cdot,k)$ ends.

**Lemma 5.E.3.** *There is an event $\mathcal{E}_2$ with probability at least $1-\zeta$, such that under $\mathcal{E}_1 \cap \mathcal{E}_2$, during the execution of* SkippyEleanor*, when Optimization Problem 5.4.12 is solved (Line 11), for $(\hat{G},\bar{\theta})$ as recorded in Line 5 for all $k \in [H-1]$, $i \in [k+1:H]$, and $v, w \in \mathcal{B}(1)$,*

$$|v_\parallel^\top \left(y^{ki}_{\hat{G}\bar{\theta}} - \hat{F}^{ki}_{\hat{G}\bar{\theta}}\right) w| \leq \bar{\sigma}^m_k \beta + 3\frac{\varepsilon}{dH^2\omega}$$

*where $\bullet_\parallel$ denotes $\bullet_{\parallel(\mathcal{Q},i)}$.*

Together, these lemmas can be used to show that the vector $w$ derived from Line 11 in SkippyEleanor is sufficiently aligned with both $Z(\mathcal{Q},\cdot)$ and the subspace $\mathrm{Proj}_{\perp(\mathcal{Q},\cdot)}$ projects to, which leads to the following important result:

**Lemma 5.E.4.** *Under the $\mathcal{E}_1 \cap \mathcal{E}_2$, if Line 14 is executed any time during the execution of* SkippyEleanor *(i.e., when the consistency check fails), then the resulting $\mathcal{Q}$ continues to be a valid preconditioning.*

From now on, our lemmas assume the high-probability events of Lemmas 5.E.2 and 5.E.3 hold, and therefore $\mathcal{Q}$ is a valid preconditioning at any time during the execution by Lemma 5.E.4.

### 5.E.2. Query complexity bounds

We bound the number of iterations of $m$ that SkippyEleanor can execute. The proof of the following lemma is presented in Section 5.I:

**Lemma 5.E.5.** *Throughout the execution of* SkippyEleanor, $m \le m_{max}$.

Note that throughout the execution of SkippyEleanor, $m' \le m'_{\max}$. As $m' - m$ equals the number of times Line 14 is executed, i.e., the sum of sequence lengths corresponding to $\mathcal{Q}$, by Lemma 5.4.4,

**Corollary 5.E.6.** *Under $\mathcal{E}_1 \cap \mathcal{E}_2$,* SkippyEleanor *returns with a policy before exiting the while loop of Line 3, and as each iteration executes $Hn$ trajectories in Line 8, the number of interactions of* SkippyEleanor *with the MDP is bounded by $\tilde{\mathcal{O}}\left(H^{11} d^7 / \varepsilon^2\right)$.*

### 5.E.3. Performance guarantee

We next consider the $m^{\text{th}}$ iteration of SkippyEleanor under the assumption that the consistency check passes, that is, Line 18 is executed. We intend to guarantee the performance of $\pi^{mH}$ in terms of $\sum_{t=1}^{H} \bar{\sigma}_k^m$, given that the optimization value $x$ satisfies $x \le \bar{\sigma}_k^m \beta\omega + 3\frac{\varepsilon}{dH^2}$ (which follows from the execution reaching Line 18). Next we introduce variants of $c_{ki}^j$ and $\tilde{c}_{ki}^j$ (Eq. (107)) which act, instead of the data collected during the execution of the algorithm, on a trajectory $(S_h, A_h, R_h)_{h\in[H]}$ and corresponding stage mapping $p$ obtained by an independent run of SkippyPolicy, which will be clear from the context: $\tilde{c}_{ki} = \mathbb{1}\{p(k) < i\}$, and

$$c_{ki} = \mathbb{1}\left\{ p(k) < i \text{ and } \left\|\varphi(S_{p(k)}, A_{p(k)})\right\|_{X_{m,p(k)}^{-1}} < 2(\beta\omega dH)^{-1} \text{ and } \left\langle \varphi(S_{p(k)}, A_{p(k)}), \bar{\theta}_{p(k)} \right\rangle \ge 0 \right\}.$$

**Remark 5.E.7.** *In our analysis we rely on the obvious fact that the laws of the trajectories of* SkippyPolicy$(\hat{G}, \bar{\theta}, k)$ *and* SkippyPolicy$(\hat{G}, \bar{\theta}, k+1)$ *are the same until stage $p(k+1)$ (as the policies are the same until then), for any parameters $\hat{G}$ and $\bar{\theta}$. This includes $S_{p(k+1)}$ but not $A_{p(k+1)}$ if $p(k+1) \le H$, and includes the whole trajectory ending with $R_H$ otherwise.*

We prove the following using induction on $k = H, \ldots, 1$ in Section 5.J:

**Lemma 5.E.8.** *There is an event $\mathcal{E}_3$ with probability at least $1 - 3\zeta$, such that under $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3$, during the execution of* SkippyEleanor, *whenever Line 18 is executed, for $(\hat{G}, \bar{\theta})$ as recorded in Line 5 of the current iteration, for $k \in [H]$,*

$$\bar{C}^k := \mathbb{E}_{\pi^{mk}, s_1} \tilde{c}_{k,H+1} C_{\hat{G}\bar{\theta}}(S_{p(k)}) \le \mathbb{E}_{\pi^{mH}, s_1} \sum_{u=p(k)}^{H} R_u + 2\sum_{i=k}^{H} \bar{\sigma}_k^m \beta\omega dH + 4(H-k+1)\frac{\varepsilon}{H}. \quad (109)$$

As $S_1 = s_1$ is fixed and $\tau(s_1) = 1$, we get the following corollary, which shows that the value $C_{\hat{G}\bar{\theta}}$ of the solution $(\hat{G}, \bar{\theta})$ of Optimization Problem 5.4.10 can be used as a lower bound on the value of the policy $\pi^{mH}$ up to the uncertainty and some $\varepsilon$ terms:

**Corollary 5.E.9.** *Under $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3$, the value of Optimization Problem 5.4.10 with the solution $(\hat{G}, \bar{\theta})$ satisfies*

$$C_{\hat{G}\bar{\theta}}(s_1) = \bar{C}^1 \le \mathbb{E}_{\pi^{mH}, s_1}\sum_{u=1}^{H} R_u + 2\sum_{i=1}^{H} \bar{\sigma}_k^m \beta\omega d H^2 + 4\varepsilon = v^{\pi^{mH}}(s_1) + 2\sum_{i=1}^{H} \bar{\sigma}_k^m \beta\omega d H^2 + 4\varepsilon\,.$$

## 5.E.4. Optimism of Optimization Problem 5.4.10

The following establishes the optimistic property, that is, that the value of Optimization Problem 5.4.10 competes with $v^\star(s_1)$. The proof relies on the fact that the correct guess $\hat{G}$ and a good choice of $\bar{\theta}$ are feasible for the optimization problem, combined with the fact that this $\bar{\theta}$ induces a policy $\pi = \textsc{SkippyPolicy}(\hat{G}, \bar{\theta}, H)$ that takes action-value maximizing actions according to a very accurate approximation of action-values almost everywhere. In fact, it only skips states whose range is at most $\varepsilon/H$. The proof is presented in Section 5.K.

**Lemma 5.E.10.** *There is an event $\mathcal{E}_4$ with probability at least $1 - \zeta$, such that under $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_4$, throughout the execution of* SkippyEleanor*, the value of Optimization Problem 5.4.10 is at least $v^\star(s_1) - 2\varepsilon$.*

*Proof of Theorem 5.4.1.* We combine Lemma 5.E.10 with Corollary 5.E.9, Corollary 5.E.6, and the fact that the condition of Line 18 is satisfied when SkippyEleanor returns with a policy, to get that under $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3 \cap \mathcal{E}_4$, that is, with probability at least $1 - 6\zeta$, SkippyEleanor interacts with the MDP for at most $\tilde{\mathcal{O}}\left(H^{11} d^7/\varepsilon^2\right)$ many steps, before returning with the policy $\pi^{mH}$ that satisfies

$$v^\star(s_1) \le C_{\hat{G}\bar{\theta}}(s_1) + 2\varepsilon \le v^{\pi^{mH}}(s_1) + 2\sum_{i=1}^{H} \bar{\sigma}_k^m \beta\omega d H^2 + 6\varepsilon \le v^{\pi^{mH}}(s_1) + 8\varepsilon\,,$$

where the final inequality follows from the fact that when SkippyEleanor returns in Line 19, $\sum_{k=1}^{H} \bar{\sigma}_k^m \le \varepsilon/(\beta\omega d H^2)$. By scaling the parameters, this finishes the proof of Theorem 5.4.1. ■

## 5.F. Deferred definitions and proofs for Section 5.4.1

***Proof of Lemma 5.4.3.*** For any $\theta \in \Theta_h^{\mathcal{Q}}$, it holds that $\theta = \mathcal{Q}_h^{-1}\hat{\theta}$ for some $\hat{\theta} \in \Theta_h$. Since $\left\|\hat{\theta}\right\|_2 \le B$, and writing $\mathcal{Q}_h$ as in Definition 5.4.2,

$$\|\theta\|_2^2 = \hat{\theta}^\top \left(B^{-2}I + \textstyle\sum_{v \in C_h} vv^\top\right)\hat{\theta} \le B^{-2}B^2 + |C_h| \le 1 + d_1\,,$$

where we used Definition 5.4.2 and Lemma 5.4.4. Finally, we conclude that $\|\theta\|_2 \le \sqrt{d_1+1}$. ■

**Definition 5.F.1.** *$(G_h^{\mathcal{Q}}, \rho_h^{\mathcal{Q}})$ is a near-optimal design for $\Theta_h^{\mathcal{Q}}$, if for any $\theta \in \Theta_h^{\mathcal{Q}}$,*

$$\langle v, \theta\rangle = 0 \quad \textit{for all } v \in \mathrm{Ker}(V(G_h^{\mathcal{Q}}, \rho_h^{\mathcal{Q}})), \textit{ and} \tag{110}$$

$$\|\theta\|^2_{V(G_h^{\mathcal{Q}}, \rho_h^{\mathcal{Q}})^\dagger} \le 2d, \tag{111}$$

$$\textit{where } V(G_h^{\mathcal{Q}}, \rho_h^{\mathcal{Q}}) = \sum_{\pi \in G_h^{\mathcal{Q}}} \rho_h^{\mathcal{Q}}(\pi)(\theta_h^{\mathcal{Q}}(\pi))(\theta_h^{\mathcal{Q}}(\pi))^\top. \tag{112}$$

An important corollary of the above definition is that if $M = \mathrm{Proj}_{\mathrm{Im}(V(G_h^{\mathcal{Q}}, \rho_h^{\mathcal{Q}}))}$, then $V(G_h^{\mathcal{Q}}, \rho_h^{\mathcal{Q}})^{\dagger\frac{1}{2}} V(G_h^{\mathcal{Q}}, \rho_h^{\mathcal{Q}})^{\frac{1}{2}} Mv = Mv$, and $\langle \theta, Mv\rangle = \langle \theta, v\rangle$ due to Eq. (110), and so

$$\theta^\top v = \theta^\top V(G_h^{\mathcal{Q}}, \rho_h^{\mathcal{Q}})^{\dagger\frac{1}{2}} V(G_h^{\mathcal{Q}}, \rho_h^{\mathcal{Q}})^{\frac{1}{2}} v \quad \text{for all } \theta \in \Theta_h^{\mathcal{Q}} \text{ and } v \in \mathbb{R}^d. \tag{113}$$

***Proof of Proposition 5.4.5.*** Take any $h \in [H]$, $s \in \mathcal{S}_h$, and $\mathcal{Q} \in \mathrm{PD}^H$. Take $i, j \in [\mathcal{A}]$ such that $\mathrm{range}(s) = \sup_{\theta \in \Theta_h} \langle \varphi(s,i,j), \theta\rangle$. Then,

$$\begin{aligned}
\mathrm{range}(s)^2 &= \sup_{\theta \in \Theta_h} \langle \varphi(s,i,j), \theta\rangle^2 = \sup_{\theta \in \Theta_h^{\mathcal{Q}}} \langle \varphi_{\mathcal{Q}}(s,i,j), \theta\rangle^2 \\
&\le \sup_{\theta \in \Theta_h^{\mathcal{Q}}} \|\theta\|^2_{V(G_h^{\mathcal{Q}}, \rho_h^{\mathcal{Q}})^\dagger} \|\varphi_{\mathcal{Q}}(s,i,j)\|^2_{V(G_h^{\mathcal{Q}}, \rho_h^{\mathcal{Q}})} \\
&\le 2d\varphi_{\mathcal{Q}}(s,i,j)^\top \left(\sum_{\pi \in G_h^{\mathcal{Q}}} \rho_h^{\mathcal{Q}}(\pi)(\theta_h^{\mathcal{Q}}(\pi))(\theta_h^{\mathcal{Q}}(\pi))^\top\right) \varphi_{\mathcal{Q}}(s,i,j) \\
&= 2d\varphi(s,i,j)^\top \left(\sum_{\pi \in G_h^{\mathcal{Q}}} \rho_h^{\mathcal{Q}}(\pi)(\theta_h(\pi))(\theta_h(\pi))^\top\right) \varphi(s,i,j) \\
&\le 2d \max_{\pi \in G_h^{\mathcal{Q}}} \left\langle \varphi(s,i,j)^\top, \theta_h(\pi)\right\rangle^2 \le 2d\,\mathrm{range}_{\mathcal{Q}}(s)^2,
\end{aligned}$$

where the first inequality uses Eq. (113) and the Cauchy-Schwarz inequality, and the second inequality follows by substituting the definition of $V(G_h^{\mathcal{Q}}, \rho_h^{\mathcal{Q}})$ and using Eq. (111). Finally, the first inequality in the last line holds as we replace the weighted sum from the previous line with the maximum operator. We therefore get that $\mathrm{range}(s) \le \sqrt{2d}\,\mathrm{range}_{\mathcal{Q}}(s)$, finishing the proof. ■

***Proof of Lemma 5.4.4.*** Take any $h \in [H]$ and the sequence $C_h$ corresponding to $\mathcal{Q}$. Assume that this sequence is of length $l$, and let $\Sigma_{h,i} = B^{-2}I + \sum_{j=1}^{i} v_j v_j^\top$ for $i \in [l]$. By the second part of Eq. (92),

$$l = \sum_{i=1}^{l} \min\left\{1, 2\left\|\left(B^{-2}I + \sum_{j=1}^{i-1} v_j v_j^\top\right)^{-\frac{1}{2}} v_i\right\|_2^2\right\} \le 2\sum_{i=1}^{l} \min\left\{1, \|v_i\|^2_{\Sigma_{h,i-1}^{-1}}\right\}.$$

Applying the elliptical potential lemma (Lemma 5.L.1),

$$l \le 2\sum_{i=1}^{l} \min\left\{1, \|v_i\|^2_{\Sigma_{h,i-1}^{-1}}\right\} \le 4d \log\left(\frac{\mathrm{Tr}(\Sigma_{h,0}) + lL_3^2}{d\det(\Sigma_{h,0})^{1/d}}\right) = 4d\log\left(1 + \frac{lL_3^2}{B^{-2}d}\right).$$

where $\Sigma_{h,0} = B^{-2}I$ by definition. Using that $\log(1+x) \le \sqrt{x}$ for $x \ge 0$, we have $l \le 4d\sqrt{lL_3^2B^2/d}$, which implies $l \le 16dL_3^2B^2$. Substituting this into the previous bound yields

$$l \le 4d\log\left(1 + \frac{(16dL_3^2B^2)L_3^2}{B^{-2}d}\right) = 4d\log\left(1 + 16L_3^4B^4\right) = d_1.$$ ■

## 5.G. Deferred proofs for Section 5.4.2

For any vector $v \in \mathbb{R}^d$, we define $\overline{v} = v/\|v\|_2$ as the unit vector in the direction of $v$ if $v \neq \mathbf{0}$ and $\overline{\mathbf{0}} = 0$ otherwise. For any $h \in [2:H]$, $s \in \mathcal{S}_h$, the normalized version of the largest preconditioned feature difference is denoted by

$$\bar{\varphi}_{\mathcal{Q}}(s) = \overline{\varphi_{\mathcal{Q}}(s,i,j)} \quad \text{where } (i,j) = \operatorname*{arg\,max}_{i',j' \in [\mathcal{A}]} \|\varphi_{\mathcal{Q}}(s,i',j')\|_2. \tag{114}$$

***Proof of Lemma 5.4.7.*** Fix $h \in [H]$, $\alpha$-admissible $f : \mathcal{S}_h \to \mathbb{R}$, $t \in [h-1]$, and $\pi \in \Pi$. Our aim is to construct policies $\pi_k^+, \pi_k^- \in \Pi$ for $k \in [d_0]$, such that for all $(s,a) \in \mathcal{S}_t \times [\mathcal{A}]$, $\sum_{k\in[d_0]}(q^{\pi_k^+}(s,a) - q^{\pi_k^-}(s,a))$ is approximately proportional to the desired $\mathbb{E}_{\pi,s,a} f(S_h)$. Let $G_{h,1}^{\mathcal{Q}}, G_{h,2}^{\mathcal{Q}}, \ldots$ denote the policies in $G_h^{\mathcal{Q}}$ underlying the near-optimal design of $\Theta_h^{\mathcal{Q}}$, and for any $s \in \mathcal{S}_h$, denote by $\mathrm{ord}(s) \in [d_0]$ the index of the policy maximizing the range of the action-value function in state $s$, that is,

$G^{\mathcal{Q}}_{h,\mathrm{ord}(s)} = \arg\max_{\pi \in G^{\mathcal{Q}}_h} \max_{i,j\in[\mathcal{A}]}(q^\pi(s,i) - q^\pi(s,j))$; to simplify notation, we define $\tilde{G}(s) = G^{\mathcal{Q}}_{h,\mathrm{ord}(s)}$. For $s \in \mathcal{S}_h$ let

$$(a^+(s), a^-(s)) = \begin{cases} \arg\max_{i,j\in[\mathcal{A}]} \hat{q}^{\tilde{G}(s)}(s,i) - \hat{q}^{\tilde{G}(s)}(s,j) & \text{if } f(s) \geq 0 \\ \arg\min_{i,j\in[\mathcal{A}]} \hat{q}^{\tilde{G}(s)}(s,i) - \hat{q}^{\tilde{G}(s)}(s,j) & \text{otherwise.} \end{cases}$$

By Eq. (94) and Definition 5.4.6 have that

$$|\hat{q}^{\tilde{G}(s)}(s,a^+(s)) - \hat{q}^{\tilde{G}(s)}(s,a^-(s))| = \mathrm{range}_{\mathcal{Q}}(s) \geq \alpha|f(s)| \geq 0\,.$$

Since $q^{\tilde{G}(s)}(s,a^+(s)) - q^{\tilde{G}(s)}(s,a^-(s)) \approx_{2\eta} \hat{q}^{\tilde{G}(s)}(s,a^+(s)) - \hat{q}^{\tilde{G}(s)}(s,a^-(s))$, if $\alpha|f(s)| \geq 4\eta$, we have

$$\begin{aligned} q^{\tilde{G}(s)}(s,a^+(s)) - q^{\tilde{G}(s)}(s,a^-(s)) &\geq \alpha f(s) - 2\eta \geq \frac{\alpha}{2} f(s) > 0 && \text{if } f(s) \geq 0 \\ q^{\tilde{G}(s)}(s,a^+(s)) - q^{\tilde{G}(s)}(s,a^-(s)) &\leq \alpha f(s) + 2\eta \leq \frac{\alpha}{2} f(s) < 0 && \text{otherwise.} \end{aligned} \tag{115}$$

Let us define $f' : \mathcal{S}_h \to \mathbb{R}$ as

$$f'(s) = \begin{cases} \frac{\alpha f(s)/2}{q^{\tilde{G}(s)}(s,a^+(s)) - q^{\tilde{G}(s)}(s,a^-(s))} & \text{if } \alpha|f(s)| \geq 4\eta \\ 0 & \text{otherwise.} \end{cases}$$

By Eq. (115), there can be no division by zero in the above definition, and $0 \leq f'(s) \leq 1$.

Now we are ready to define $\pi^+_k$ and $\pi^-_k$. Both policies follow $\pi$ up to stage $h-1$, when they switch to $G^{\mathcal{Q}}_{h,k}$, except if at stage $h$ a state $s \in \mathcal{S}_h$ is such that $G^{\mathcal{Q}}_{h,k}$ has the maximal action-value function range. In this case $\pi^+_k$ selects $a^+(s)$ with probability $f'(s)$ and $a^-(s)$ with probability $1 - f'(s)$, while $\pi^-_k$ always selects $a^-(s)$. Formally, for $k \in [d_0]$, we define for $s \in \mathcal{S}$

$$\pi^+_k(s) = \begin{cases} \pi(s) & \text{if } \mathrm{stage}(s) < h; \\ a^+(s) \text{ w.p. } f'(s), \text{ and } a^-(s) \text{ w.p. } 1 - f'(s) & \text{if } \mathrm{stage}(s) = h \text{ and } \mathrm{ord}(s) = k; \\ G^{\mathcal{Q}}_{h,k}(s), & \text{otherwise,} \end{cases}$$

where w.p. stands for *with probability*. Similarly,

$$\pi_k^-(s) = \begin{cases} \pi(s) & \text{if } \text{stage}(s) < h; \\ a^-(s) \text{ w.p. } 1 & \text{if } \text{stage}(s) = h \text{ and } \text{ord}(s) = k; \\ G^{\mathcal{Q}}_{h,k}(s) & \text{otherwise.} \end{cases}$$

Note that $\pi_k^+ \in \Pi$ and $\pi_k^- \in \Pi$, as desired. Since for all $k \in [d_0]$, the policies follow $G_{h,k}$ for $s \in \mathcal{S}_{t'}$ for $t' > h$, therefore for all $k \in [d_0]$,

$$v^{\pi_k^-}(s) = v^{\pi_k^+}(s) = v^{G^{\mathcal{Q}}_{h,k}}(s) \quad \text{for all } s \in \mathcal{S}_{h+1}, \text{ and} \tag{116}$$

$$q^{\pi_k^-}(s,a) = q^{\pi_k^+}(s,a) = q^{G^{\mathcal{Q}}_{h,k}}(s,a) \quad \text{for all } (s,a) \in \mathcal{S}_h \times [\mathcal{A}]. \tag{117}$$

Also, for any $s \in \mathcal{S}$ with $\text{stage}(s) < h$ and any $a \in [\mathcal{A}]$,

$$\begin{aligned}
\sum_{k\in[d_0]} \left(q^{\pi_k^+}(s,a) - q^{\pi_k^-}(s,a)\right) &= \mathbb{E}_{\pi,s,a} \sum_{k\in[d_0]} \left(v^{\pi_k^+}(S_h) - v^{\pi_k^-}(S_h)\right) \\
&= \mathbb{E}_{\pi,s,a} \left(v^{\pi^+_{\text{ord}(S_h)}}(S_h) - v^{\pi^-_{\text{ord}(S_h)}}(S_h)\right) \\
&= \mathbb{E}_{\pi,s,a} \Big(q^{\tilde{G}(S_h)}(S_h, a^+(S_h)) f'(S_h) + q^{\tilde{G}(S_h)}(S_h, a^-(S_h))(1 - f'(S_h)) \\
&\qquad - q^{\tilde{G}(S_h)}(S_h, a^-(S_h))\Big) \\
&= \mathbb{E}_{\pi,s,a} \left(f'(S_h) \left(q^{\tilde{G}(S_h)}(S_h, a^+(S_h)) - q^{\tilde{G}(S_h)}(S_h, a^-(S_h))\right)\right) \\
&= \mathbb{E}_{\pi,s,a} \mathbb{I}\{\alpha |f(S_h)| \geq 4\eta\} \frac{\alpha}{2} f(S_h) \approx_{2\eta} \frac{\alpha}{2} \mathbb{E}_{\pi,s,a} f(S_h),
\end{aligned}$$

where the first line is due to both $q^{\pi_k^+}$ and $q^{\pi_k^-}$ following $\pi$ on states with stage less than $h$, the second line follows from the fact that for any $s \in \mathcal{S}_h$, $\pi_k^+(s) = \pi_k^-(s)$ for any $k \neq \text{ord}(s)$; combining this with Eq. (116) leads to all $k \neq \text{ord}(s)$ terms of the sum to cancel. The third line follows from expanding the definition of the policies and Eq. (117).

Let $\tilde{\theta} = \frac{2}{\alpha} \sum_{k\in[d_0]} \left(\theta_t(\pi_k^+) - \theta_t(\pi_k^-)\right)$. Since $\|\theta_t(\cdot)\|_2 \leq B$ by definition, we have $\left\|\tilde{\theta}\right\|_2 \leq 4d_0 B/\alpha$. By Definition 5.3.2, for all $(s,a) \in \mathcal{S}_t \times [\mathcal{A}]$,

$$\left\langle \varphi(s,a), \frac{\alpha}{2}\tilde{\theta} \right\rangle \approx_{2d_0\eta} \sum_{k\in[d_0]} q^{\pi_k^+}(s,a) - q^{\pi_k^-}(s,a) \approx_{2\eta} \frac{\alpha}{2} \mathbb{E}_{\pi,s,a} f(S_h),$$

and hence

$$\left\langle \varphi(s,a), \tilde{\theta} \right\rangle \approx_{4(d_0+1)\eta/\alpha} \mathbb{E}_{\pi,s,a} f(S_h)\,.$$

Since $4(d_0+1)\eta/\alpha \le \eta_0 = 5d_0\eta/\alpha$ as $d_0 \ge 4$ by definition, this completes the proof. ■

***Proof of Lemma 5.4.9.*** Take any $s \in \mathcal{S}_h$. For the correct guess, $\mathrm{range}_{\mathcal{Q}}^{\hat{G}}(s) = \mathrm{range}_{\mathcal{Q}}(s)$. Then, using that $\|\bar{\varphi}_{\mathcal{Q}}(\cdot)\|_2 \le 1$, $\mathrm{Tr}(\mathbf{f}(s)) \le \mathrm{range}_{\mathcal{Q}}(s)\frac{\sqrt{2d}H^2}{\varepsilon}$, proving the second claim of the lemma (as $\gamma \le 1$).

For the first claim, take any $\hat{G} = (\hat{\vartheta}_h^i)_{h\in[2:H],i\in[d_0]} \in \mathbf{G}$. Let $\varphi'$ be the unnormalized version of $\bar{\varphi}_{\mathcal{Q}}(s)$ of Eq. (114), that is, $\varphi' = \varphi_{\mathcal{Q}}(s,i,j)$ for the same $i,j$ as in Eq. (114) (i.e., with the largest $\ell_2$-norm). Then, using that $\hat{G} \in \mathbf{G}$,

$$\mathrm{range}_{\mathcal{Q}}^{\hat{G}}(s) = \max_{k\in[d_0]} \max_{i,j} \left\langle \varphi_{\mathcal{Q}}(s,i,j), \hat{\vartheta}_h^k \right\rangle \le \|\varphi'\|_2 \max_{k\in[d_0]} \left\|\hat{\vartheta}_h^k\right\|_2 \le \|\varphi'\|_2 \sqrt{d_1+1}.$$

Using that above in combination with $|f(s)| \le H$, $v,w \in \mathcal{B}(1)$, $\|\bar{\varphi}_{\mathcal{Q}}(s)\|_2 \le 1$, we obtain

$$\begin{aligned} |v_{\|}^{\top}\mathbf{f}(s)w| &\le |\left\langle \bar{\varphi}_{\mathcal{Q}}(s), v_{\|} \right\rangle \left\langle \bar{\varphi}_{\mathcal{Q}}(s), w \right\rangle| \,\mathrm{range}_{\mathcal{Q}}^{\hat{G}}(s) \frac{\sqrt{2d}H^2}{\varepsilon} \\ &\le \left\|\bar{\varphi}_{\mathcal{Q}}(s)_{\|}\right\|_2 \|\varphi'\|_2 \sqrt{d_1+1} \frac{\sqrt{2d}H^2}{\varepsilon} = \left\|\varphi'_{\|}\right\|_2 \sqrt{d_1+1} \frac{\sqrt{2d}H^2}{\varepsilon}\,. \end{aligned}$$

As the eigenvalues of $V(G_h^{\mathcal{Q}}, \rho_h^{\mathcal{Q}}) = \sum_{\pi\in G_h^{\mathcal{Q}}} \rho_h^{\mathcal{Q}}(\pi)(\theta_h^{\mathcal{Q}}(\pi))(\theta_h^{\mathcal{Q}}(\pi))^{\top}$ corresponding to the subspace in which $\varphi'_{\|}$ lies are by definition at least $\gamma$, we can write

$$(\mathrm{range}_{\mathcal{Q}}(s))^2 \ge \max_{\pi\in G_h^{\mathcal{Q}}} \left\langle \varphi', \theta_h^{\mathcal{Q}}(\pi) \right\rangle^2 \ge \varphi'^{\top} V(G_h^{\mathcal{Q}}, \rho_h^{\mathcal{Q}}) \varphi' \ge \varphi'^{\top}_{\|} V(G_h^{\mathcal{Q}}, \rho_h^{\mathcal{Q}}) \varphi'_{\|} \ge \left\|\varphi'_{\|}\right\|_2^2 \gamma\,.$$

Combining with the previous result, we get that

$$\mathrm{range}_{\mathcal{Q}}(s) \ge \sqrt{\gamma}\left\|\varphi'_{\|}\right\|_2 \ge \frac{\sqrt{\gamma}\varepsilon}{\sqrt{2d}\sqrt{d_1+1}H^2} |v_{\|}^{\top}\mathbf{f}(s)w| = \alpha |v_{\|}^{\top}\mathbf{f}(s)w|,$$

finishing the proof. ■

## 5.H. Deferred proofs for Section 5.E.1

The definitions (Eqs. (96) and (97)) immediately give rise to the following facts:

$$\begin{aligned} &D(s_i\to) \in \left[-\sum_{u=i}^{H} r_u, H\right] \subseteq [-H,H] \text{ and } \tau(\cdot) \in [0,1], \text{ implying} \\ &E^{\to}(s_i\to) \in \left[-\sum_{u=i}^{H} r_u, H\right] \subseteq [-H,H], \text{ implying} \\ &E(s_i\to) \in [-2\tau(s_i)H, 2\tau(s_i)H] \subseteq [-2H,2H]\,. \end{aligned} \tag{118}$$

Furthermore, since either $\tau(s_i) = 0$ or $\|\bar{\varphi}_{\mathcal{Q}}(s_i)\|_2 = 1$ (as $\|\bar{\varphi}_{\mathcal{Q}}(s_i)\| = 0$ implies that $\mathrm{range}_{\mathcal{Q}}(s_i)$ and hence $\tau(s_i)$ are both zero), we have

$$\mathrm{Tr}(F(s_i\to)) = E(s_i\to)\,, \tag{119}$$

which was used to establish the last part of Corollary 5.4.11.

***Proof of Lemma 5.E.1.*** We drop the subscripts $(\hat{G},\bar{\theta})$. Let $(\hat{\vartheta}_h^i)_{h\in[2:H],i\in[d_0]} = \hat{G} \in \mathbf{G}$. Let $z = v - w$ be the projection of $v$ to the subspace orthogonal to $Z(\mathcal{Q},i)$, denoted by $Z(\mathcal{Q},i)^\perp$. In other words, $z = \mathrm{Proj}_{Z(\mathcal{Q},i)^\perp} v$. Let $\mathbf{M} = y^{ki} - \hat{F}^{ki}$. By the symmetry of $\mathbf{M}$,

$$v^\top \mathbf{M} v = z^\top \mathbf{M}(v+w) + w^\top \mathbf{M} w\,.$$

It is enough to prove therefore that

$$\frac{\varepsilon}{dH^2\omega} \geq z^\top \mathbf{M}(v+w)\,.$$

As $\|v\|_2 \leq 1$ and $\|v+w\|_2 \leq 2$, and using the definitions and Eq. (118), for any input $(s_i\to)$,

$$\begin{aligned} |z^\top F(s_i\to)(v+w)| &= |\langle z, \bar{\varphi}_{\mathcal{Q}}(s_i)\rangle \langle v+w, \bar{\varphi}_{\mathcal{Q}}(s_i)\rangle E(s_i\to)| \\ &\leq 4H\tau(s_i)|\langle z, \bar{\varphi}_{\mathcal{Q}}(s_i)\rangle| \leq 4\,\mathrm{range}_{\mathcal{Q}}^{\hat{G}}(s_i)\frac{\sqrt{2d}H^2}{\varepsilon}|\langle z, \bar{\varphi}_{\mathcal{Q}}(s_i)\rangle| \\ &\leq 4|\langle z, \bar{\varphi}_{\mathcal{Q}}(s_i)\rangle| \max_{a,b,k\in[d_0]} \left\langle \varphi_{\mathcal{Q}}(s,a,b), \hat{\vartheta}_h^k \right\rangle \frac{\sqrt{2d}H^2}{\varepsilon} \\ &\leq 4\left\|\mathrm{Proj}_{Z(\mathcal{Q},i)^\perp} \bar{\varphi}_{\mathcal{Q}}(s)\right\|_2 \|\varphi'\|_2 \max_{k\in[d_0]} \left\|\hat{\vartheta}_h^k\right\|_2 \frac{\sqrt{2d}H^2}{\varepsilon} \\ &\leq 4\left\|\mathrm{Proj}_{Z(\mathcal{Q},i)^\perp} \varphi'\right\|_2 \sqrt{d_1+1}\frac{\sqrt{2d}H^2}{\varepsilon}\,, \end{aligned}$$

where $\varphi'$ is the unnormalized version of $\bar{\varphi}_{\mathcal{Q}}(s_i)$ of Eq. (114), that is, $\varphi' = \varphi_{\mathcal{Q}}(s_i, a, b)$ for the same $a, b$ as in Eq. (114) (i.e., with the largest $\ell_2$-norm).

As $\mathrm{Proj}_{Z(\mathcal{Q},i)^\perp} \varphi' = \mathrm{Proj}_{Z(\mathcal{Q},i)^\perp}(\varphi_{\mathcal{Q}}(s,a) - \varphi_{\mathcal{Q}}(s,b)) = \mathrm{Proj}_{Z(\mathcal{Q},i)^\perp} \mathcal{Q}_i(\varphi(s,a) - \varphi(s,b))$ for some $s \in \mathcal{S}_i, a, b \in [\mathcal{A}]$, and by definition $\mathrm{Proj}_{Z(\mathcal{Q},i)^\perp} \mathcal{Q}_i \preceq L_3^{-2} I$, $\left\|\mathrm{Proj}_{Z(\mathcal{Q},i)^\perp} \varphi'\right\|_2 \leq L_3^{-2} \|\varphi(s,a) - \varphi(s,b)\|_2 \leq 2L_3^{-2} L$, so

$$|z^\top F(s_i \rightarrow)(v+w)| \leq 8L_3^{-2} L \sqrt{d_1 + 1} \frac{\sqrt{2d} H^2}{\varepsilon}, \tag{120}$$

and hence

$$|z^\top \hat{F}^{ki}(v+w)| \leq 8L_3^{-2} L \sqrt{d_1 + 1} \frac{\sqrt{2d} H^2}{\varepsilon}. \tag{121}$$

To bound $|z^\top y^{ki}(v+w)|$, note that by the definition $y^{ki}$,

$$z^\top y^{ki}(v+w) = \frac{1}{n} \sum_{j \in [n]} c_{ki}^j \left\langle \varphi^{mkj}_{\mathrm{p}(k)}, \check{\theta}^{\mathrm{p}(mkj),i} \right\rangle$$

$$\text{where} \qquad \check{\theta}^{ti} = X_{mt}^{-1} \sum_{lkj \in \mathbf{I}^m(t)} \varphi_t^{lkj} \left( z^\top F(S_i^{lkj}, \ldots, R_H^{lkj})(v+w) \right) \qquad \text{for } t \in [i-1]$$

Therefore

$$|z^\top y^{ki}(v+w)| \leq \max_{t \in [i-1], s \in \mathcal{S}_t, a \in [\mathcal{A}]} \left\langle \varphi(s,a), \check{\theta}^{ti} \right\rangle.$$

Fix any $t \in [i-1], s \in \mathcal{S}_t, a \in [\mathcal{A}]$. By repeated application of the Cauchy-Schwarz inequality, the fact that $X_{mt} \succeq \lambda I$, the triangle inequality, and using Eq. (120),

$$\begin{aligned}
\left| \left\langle \varphi(s,a), \check{\theta}^{ti} \right\rangle \right| &\leq \|\varphi(s,a)\|_{X_{mt}^{-1}} \left\| \sum_{lkj \in \mathbf{I}^m(t)} \varphi_t^{lkj} \left( z^\top F(S_i^{lkj}, \ldots, R_H^{lkj})(v+w) \right) \right\|_{X_{mt}^{-1}} \\
&\leq \|\varphi(s,a)\|_2 \lambda^{-1/2} \cdot 8L_3^{-2} L \sqrt{d_1 + 1} \frac{\sqrt{2d} H^2}{\varepsilon} \sum_{lkj \in \mathbf{I}^m(t)} \left\| \varphi_t^{lkj} \right\|_{X_{mt}^{-1}} \\
&\leq 8L_3^{-2} L^2 \lambda^{-1/2} \sqrt{d_1 + 1} \frac{\sqrt{2d} H^2}{\varepsilon} \sqrt{|\mathbf{I}^m(t)|} \sqrt{\sum_{lkj \in \mathbf{I}^m(t)} \left\| \varphi_t^{lkj} \right\|_{X_{mt}^{-1}}^2} \\
&\leq 8L_3^{-2} L^2 \lambda^{-1/2} \sqrt{d_1 + 1} \frac{\sqrt{2d} H^2}{\varepsilon} \sqrt{m_{\max} n H d},
\end{aligned}$$

where we use that $|\mathbf{I}^m(t)| \le mnH$, $m \le m_{\max}$ by Lemma 5.E.5, and that

$$\sqrt{\sum_{lkj \in \mathbf{I}^m(t)} \left\|\varphi_t^{lkj}\right\|_{X_{mt}^{-1}}^2} = \sqrt{\sum_{lkj \in \mathbf{I}^m(t)} \mathrm{Tr}(X_{mt}^{-1} \varphi_t^{lkj} \varphi_t^{lkj\top})} \le \sqrt{\mathrm{Tr}\, X_{mt}^{-1} X_{mt}} = \sqrt{d}\,.$$

Combining with Eq. (121), with an appropriate choice of $L_3$, we obtain

$$|z^\top \mathbf{M}(v+w)| \le 8L_3^{-2} L \sqrt{d_1+1} \frac{\sqrt{2d}H^2}{\varepsilon} \left(1 + L\lambda^{-1/2}\sqrt{m_{\max} nHd}\right) \le \frac{\varepsilon}{dH^2\omega} \tag{122}$$

as desired. ∎

***Proof of Lemma 5.E.2.*** Choose $\beta$

$$\beta \le 2 + 2H\sqrt{2dH(d_0+1)\log\frac{12d_0HB}{\alpha\xi} + 2\log\frac{m'_{\max}H^2}{\zeta} + d\log\left(\lambda + m'_{\max}nHL^2/d\right)}, \tag{123}$$

satisfying $\beta = \tilde{\mathcal{O}}(H^{3/2}d)$ as given in Eq. (102), and define

$$\xi = \frac{\varepsilon}{5\sqrt{2d}(H+1)^3 L}\left(\min\left\{\varepsilon/(dH^2\omega), 1/\sqrt{m'_{\max}nH}\right\} - \eta_0\right).$$

Note that subtracting $\eta_0$ keeps $\xi$ positive, and of the same order, by our assumption that $\eta$ is small enough: $\eta_0 \le \frac{1}{2}\min\left\{\varepsilon/(dH^2\omega), 1/\sqrt{m'_{\max}nH}\right\}$, which follows from Eq. (108).

We start with a covering argument for the set of functions of the form $v_\parallel^\top \bar{F}_{\hat{G}\bar{\theta}} w$, for different choices of $\hat{G}$, $\bar{\theta}$, $v$, and $w$. By (Vershynin, 2018, Corollary 4.2.13), there is a set $C_\xi \subset \mathcal{B}(1)$ with $|C_\xi| \le (3/\xi)^d$ such that for all $x \in \mathcal{B}(1)$ there exists a $y \in C_\xi$ with $\|x-y\|_2 \le \xi$. Therefore, there is a set $C_\xi^\times \subset \left(\bigtimes_{h\in[2:H],k\in[d_0]} \mathcal{B}(\sqrt{d_1+1})\right) \times \left(\bigtimes_{h\in[2:H]} \mathcal{B}(4d_0HB/\alpha)\right) \times \mathcal{B}(1) \times \mathcal{B}(1)$ with $|C_\xi^\times| \le (12d_0HB/(\alpha\xi))^{dH(d_0+1)}$ such that for any $\hat{G} = (\hat{\vartheta}_h^i)_{h\in[2:H],i\in[d_0]} \in \mathbf{G}$, $\bar{\theta} \in \bar{\Theta}$, and $v, w \in \mathcal{B}(1)$, there exists a $y \in C_\xi^\times$, such that if we let $\tilde{G} = (\tilde{\vartheta}^i)_{h\in[2:H],i\in[d_0]} = (y_{(h-1)d_0+i})_{h\in[2:H],i\in[d_0]}$, $\tilde{\theta} = (\tilde{\theta}_h)_{h\in[2:H]} = (y_{(H-1)d_0+h})_{h\in[2:H]}$, and $a = y_{(H-1)(d_0+1)+1}$, $b = y_{(H-1)(d_0+1)+2}$, then $\tilde{G} \in \mathbf{G}$, $\tilde{\theta} \in \bar{\Theta}$, $a, b \in \mathcal{B}(1)$, and

$$\|a - v\|_2 \le \xi \quad \text{and} \quad \|b - w\|_2 \le \xi, \quad \text{and}$$

$$\left\|\hat{\vartheta}_h^i - \tilde{\vartheta}^i\right\|_2 \le \xi \text{ and } \left\|\bar{\theta}_h - \tilde{\theta}_h\right\|_2 \le \xi \text{ for all } h \in [2:H], i \in [d_0]\,.$$

As a result, for all $s \in \mathcal{S} \setminus \mathcal{S}_1$, $|\mathrm{range}_Q^{\hat{G}}(s) - \mathrm{range}_Q^{\tilde{G}}(s)| \le 2L\xi$, and therefore $|\tau_{\hat{G}\bar{\theta}}(s) - \tau_{\tilde{G}\bar{\theta}}(s)| \le 2\sqrt{2d}HL\xi/\varepsilon$. Furthermore, $|D_{\hat{G}\bar{\theta}}(s,\dots,r_H) - D_{\tilde{G}\bar{\theta}}(s,\dots,r_H)| \le L\xi$. Combining these with the facts that in either case, $\tau(\cdot) \in [0,1]$, $D(\cdot) \in [-H,H]$, and $E^{\rightarrow}(\cdot) \in [-H,H]$ (Eq. (118)), and using

the definition of $E$ and $E^{\rightarrow}$,we have that for any $i \in [H+1]$ and inputs,

$$\begin{aligned}|E_{\hat{G}\tilde{\theta}}(s_i\rightarrow) - E_{\tilde{G}\tilde{\theta}}(s_i\rightarrow)| &\leq 4\sqrt{2d}H^2L\xi/\varepsilon + L\xi + |E^{\rightarrow}_{\hat{G}\tilde{\theta}}(s_{i+1}\rightarrow) - E^{\rightarrow}_{\tilde{G}\tilde{\theta}}(s_{i+1}\rightarrow)| \\ &= 4\sqrt{2d}H^2L\xi/\varepsilon + L\xi + \sum_{j=i+1}^{H} |E_{\hat{G}\tilde{\theta}}(s_j\rightarrow) - E_{\tilde{G}\tilde{\theta}}(s_j\rightarrow)| \\ &\leq (H+1)5\sqrt{2d}H^2L\xi/\varepsilon,\end{aligned}$$

where the first inequality sums over the contributions of $\tau$, $D$, and $E^{\rightarrow}$, and the second applies induction. By combining this bound with the bounds on $\|v-a\|_2$ and $\|w-b\|_2$, and that $E(\cdot) \in [-2H, 2H]$ (Eq. (118)) implying that $\bar{F}(\cdot) \in [-2H, 2H]$, for all $s \in \mathcal{S} \setminus \mathcal{S}_1$, we have that

$$\begin{aligned}|v_{\|}^{\top}\bar{F}_{\hat{G}\tilde{\theta}}(s)w - a_{\|}^{\top}\bar{F}_{\tilde{G}\tilde{\theta}}b|(s) &\leq 6H\xi + (H+1)5\sqrt{2d}H^2L\xi/\varepsilon \\ &\leq 5\sqrt{2d}(H+1)^3L\xi/\varepsilon = \min\{\varepsilon/(dH^2\omega), 1/\sqrt{m'_{\max}nH}\} - \eta_0\end{aligned} \tag{124}$$

Take any $m' \in [m'_{\max}]$ (this includes the entire execution of SKIPPYELEANOR). and let the quantities of Section 5.4.3 (such as $F$) be calculated with the value of $\mathcal{Q}$ at the beginning iteration $m'$ (Line 5). Take any $t \in [H-1]$, $i \in [t+1:H]$. Take any $y \in C_{\xi}^{\times}$ and assign values to $a, b, \tilde{G}$, and $\tilde{\theta}$ based on $y$ as above. For any $lkj \in \mathbf{I}^m(t)$, observe that given all the history of SKIPPYELEANOR interacting with the MDP up to (and including) $S_t^{lkj}, A_t^{lkj}$, the trajectory $S_{t+1}^{lkj}, A_{t+1}^{lkj}, \ldots, R_H^{lkj}$ is an independent rollout with policy $\pi^0$, with its law given by $\mathcal{P}_{\pi^0, S_t^{lkj}, A_t^{lkj}}$. The random variable $a_{\|}^{\top}F_{\tilde{G}\tilde{\theta}}(S_i^{lkj}\ldots, R_H^{lkj})b$ has range $[-2H, 2H]$ and expectation (conditioned on this history) $\mathbb{E}_{\pi^0, S_t^{lkj}, A_t^{lkj}} a_{\|}^{\top}\bar{F}_{\tilde{G}\tilde{\theta}}(S_i)b$. Let $\check{\theta}_{ti}$ be $\tilde{\theta}_{ti}$ from Corollary 5.4.11, satisfying $\left\|\check{\theta}_{ti}\right\|_2 \leq 1/\sqrt{\lambda}$ and Eq. (98) for $a_{\|}$, $b$, $\tilde{G}$, and $\tilde{\theta}$ instead of $v_{\|}$, $w$, $\hat{G}$, and $\bar{\theta}$:

$$\mathbb{E}_{\pi^0, s, a} a_{\|}^{\top}\bar{F}_{\tilde{G}\tilde{\theta}}(S_i)b \approx_{\eta_0} \left\langle \varphi(s,a), \check{\theta}_{ti} \right\rangle. \tag{125}$$

Take the sequence $A$ formed of $\varphi_t^{lkj}$ (for $lkj \in \mathbf{I}^m(t)$, in the order that these random variables are observed), and the sequence $X$ formed of $v_{\|}F_{\hat{G}\tilde{\theta}}(S_i^{lkj}, \ldots, R_H^{lkj})w$ (for $lkj \in \mathbf{I}^m(t)$, in the same order), and the sequence $\Delta$ formed of $\mathbb{E}_{\pi^0, S_t^{lkj}, A_t^{lkj}} v_{\|}\bar{F}_{\hat{G}\tilde{\theta}}(S_i)w - \left\langle \varphi_t^{lkj}, \check{\theta}_{ti} \right\rangle$ (for $lkj \in \mathbf{I}^m(t)$, in the same order, for any $v, w, \hat{G}$, and $\bar{\theta}$ as in the statement of this lemma). Then the sequences $A$, $X$, and $\Delta$ satisfy the conditions of Lemma 5.M.4 with a subgaussianity parameter $\sigma = 2H$. Due to this

lemma, with probability at least $1-\zeta/(m'_{\max}H^2|C^\times_\xi|)$, for any choice of $v, w, \hat{G}$, and $\bar{\theta}$ (as above),

$$\left\|\tilde{\theta}_{ti}-\check{\theta}_{ti}\right\|_{X_{mt}} < \sqrt{\lambda}\left\|\check{\theta}_{ti}\right\|_2 + \|\Delta\|_\infty\sqrt{|\mathbf{I}^m(t)|} + 2H\sqrt{2\log\left(\frac{m'_{\max}H^2|C^\times_\xi|}{\zeta}\right)+\log\left(\frac{\det X_{mt}}{\lambda^d}\right)} \tag{126}$$

$$\text{where}\qquad \tilde{\theta}_{ti} = X_{mt}^{-1}\sum_{lkj\in\mathbf{I}^m(t)}\varphi_t^{lkj}v_\parallel^\top F_{\hat{G}\bar{\theta}}(S_i^{lkj},\dots,R_H^{lkj})w$$

A union bound over all $m' \in [m'_{\max}]$, $t, i$, and $y \in C^\times_\xi$ guarantees with probability at least $1-\zeta$, the above holds for all choice of these variables, any time beginning of any iteration (Line 5) is executed. Note that we need the union bound over $m$ because the value of $\mathcal{Q}$ underlying the targets of least-squares estimations can potentially change between iterations.

To finish the proof, under this high-probability event, take any $m, t, i, \hat{G}$, and $\bar{\theta}$ as in the statement of this lemma, and choose $y \in C^\times_\xi$ as before, to satisfy Eq. (124). Combined with Eq. (125), this immediately implies that the sequence $\Delta$ formed of quantities with absolute value

$$\begin{aligned}
&|\mathbb{E}_{\pi^0,S_t^{lkj},A_t^{lkj}}v_\parallel\bar{F}_{\hat{G}\bar{\theta}}(S_i)w - \left\langle\varphi_t^{lkj},\check{\theta}_{ti}\right\rangle| \\
&\quad\le |\mathbb{E}_{\pi^0,S_t^{lkj},A_t^{lkj}}v_\parallel\bar{F}_{\hat{G}\bar{\theta}}(S_i)w - a_\parallel\bar{F}_{\tilde{G}\tilde{\theta}}(S_i)b| + |a_\parallel\bar{F}_{\tilde{G}\tilde{\theta}}(S_i)b - \left\langle\varphi_t^{lkj},\check{\theta}_{ti}\right\rangle| \\
&\quad\le \min\{\varepsilon/(dH^2\omega), 1/\sqrt{m'_{\max}nH}\} - \eta_0 + \eta_0
\end{aligned} \tag{127}$$

satisfies $\|\Delta\|_\infty \le \min\{\varepsilon/(dH^2\omega), 1/\sqrt{m'_{\max}nH}\}$. Take any $(s,a) \in \mathcal{S}_t \times [\mathcal{A}]$, and let $\tilde{\theta}_{ti}$ and $\check{\theta}_{ti}$ be as above (in Eq. (126)) for $v_\parallel$, $w$, $\hat{G}$, and $\bar{\theta}$. Note that

$$v_\parallel^\top\varphi(s,a)^\top\hat{\theta}^{ti}_{\hat{G}\bar{\theta}}w = \left\langle\varphi(s,a),\tilde{\theta}_{ti}\right\rangle,$$

By the triangle inequality, using Cauchy-Schwarz, and Eqs. (126) and (127),

$$
\begin{aligned}
&|v_{\|}^{\top}\left(\varphi(s,a)^{\top}\hat{\theta}_{\hat{G}\bar{\theta}}^{ti}-\mathbb{E}_{\pi^0,s,a}\bar{F}_{\hat{G}\bar{\theta}}(S_i)\right)w| \\
&\quad\le|\left\langle\varphi(s,a),\tilde{\theta}_{ti}-\check{\theta}_{ti}\right\rangle|+|\mathbb{E}_{\pi^0,s,a}v_{\|}^{\top}\bar{F}_{\hat{G}\bar{\theta}}(S_i)w-\left\langle\left\langle\varphi(s,a),\check{\theta}_{ti}\right\rangle\right\rangle| \\
&\quad\le\|\varphi(s,a)\|_{X_{mt}^{-1}}\left(\sqrt{\lambda}\left\|\check{\theta}_{ti}\right\|_2+\frac{\sqrt{|\mathbf{I}^m(t)|}}{\sqrt{m'_{\max}nH}}+2H\sqrt{2\log\left(\frac{m'_{\max}H^2|C_{\xi}^{\times}|}{\zeta}\right)+\log\left(\frac{\det X_{mt}}{\lambda^d}\right)}\right)+\frac{\varepsilon}{dH^2\omega} \\
&\quad\le\|\varphi(s,a)\|_{X_{mt}^{-1}}\left(2+2H\sqrt{2dH(d_0+1)\log\frac{12d_0HB}{\alpha\xi}+2\log\frac{m'_{\max}H^2}{\zeta}+d\log\left(\lambda+m'_{\max}nHL^2/d\right)}\right)+\frac{\varepsilon}{dH^2\omega} \\
&\quad\le\|\varphi(s,a)\|_{X_{mt}^{-1}}\beta+\frac{\varepsilon}{dH^2\omega},
\end{aligned}
\tag{128}
$$

where in the fourth line we used that $|\mathbf{I}^m(t)|\le m'_{\max}nH$, $|C_{\xi}^{\times}|\le(12d_0HB/(\alpha\xi))^{dH(d_0+1)}$, and we used the inequality of arithmetic and geometric means to bound $\det X_{mt}\le\left(\frac{1}{d}\operatorname{Tr}X_{mt}\right)^d\le\left(\frac{\operatorname{Tr}\lambda I+|\mathbf{I}^m(t)|L^2}{d}\right)^d$. ∎

***Proof of Lemma 5.E.3.*** Choose $n$ to satisfy

$$
n=\left\lceil 64\frac{(dH^2\omega)^2}{\varepsilon^2}H^2\left(2d\log\frac{18dH^3}{\varepsilon}+\log\frac{2m'_{\max}H^2}{\zeta}\right)\right\rceil. \tag{129}
$$

This leads to $n=\tilde{\mathcal{O}}(d^5H^6/\varepsilon^2)$.

Similarly to the proof of Lemma 5.E.2, we start with a covering argument. This time, as $\hat{G}$ and $\bar{\theta}$ are fixed, we only consider $v$ and $w$, to cover $v_{\|}^{\top}\bar{F}_{t'}^{(j)}w$ and $v_{\|}\hat{F}_{t'}^{(j)}w$. Let $\xi'=\frac{\varepsilon}{12dH^3}$. There is a set $C_{\xi'}^{+}\subset\mathcal{B}(1)\times\mathcal{B}(1)$ with $|C_{\xi'}|\le(3/\xi')^{2d}$ such that for all $v,w\in\mathcal{B}(1)$, there exists an $(a,b)\in C_{\xi'}^{+}$ with $\|v-a\|_2\le\xi'$ (and therefore $\left\|v_{\|}-a_{\|}\right\|_2\le\xi'$), and $\|w-b\|_2\le\xi'$. Take such a choice of $(a,b)$ for any $(v,w)$. As $E(\cdot)\in[-2H,2H]$ by Eq. (118), and $\|\bar{\varphi}_{\mathcal{Q}}(\cdot)\|_2\le1$, For $i\in[2:H]$ and any input,

$$
|v_{\|}^{\top}F(s_i\to)w-a_{\|}^{\top}F(s_i\to)b|\le6H\xi'=\frac{\varepsilon}{2dH^2},
$$

and therefore for any $s\in\mathcal{S}\setminus\mathcal{S}_1$, $|v_{\|}^{\top}\bar{F}(s)w-a_{\|}^{\top}\bar{F}(s)b|\le\varepsilon/(2dH^2)$. For $j\in[n]$ let

$$
\tilde{F}_j^{ki}=\mathbb{E}_{\pi^0,S_{\mathrm{p}(k)}^{mkj},A_{\mathrm{p}(k)}^{mkj}}F_{\hat{G}\bar{\theta}}(S_i^{mkj},\ldots,R_H^{mkj})=\mathbb{E}_{\pi^0,S_{\mathrm{p}(k)}^{mkj},A_{\mathrm{p}(k)}^{mkj}}\bar{F}_{\hat{G}\bar{\theta}}(S_i^{mkj})
$$

By the triangle inequality, for any $k \in [H-1]$, $i \in [k+1:H]$,

$$
\begin{aligned}
&|v_\|^\top \left(y^{ki}_{\hat{G}\bar{\theta}} - \hat{F}^{ki}_{\hat{G}\bar{\theta}}\right) w| \\
&\quad \le |\frac{1}{n}\sum_{j\in[n]} c^j_{ki} v^\top \left(\varphi^{mkj}_{\mathrm{p}(k)}{}^\top \hat{\theta}^{\mathrm{p}(mkj),i}_{\hat{G}\bar{\theta}} - \tilde{F}^{ki}_j\right) w| + |\frac{1}{n}\sum_{j\in[n]} c^j_{ki} v^\top \left(\tilde{F}^{ki}_j - F_{\hat{G}\bar{\theta}}(S^{mkj}_i, \dots, R^{mkj}_H)\right) w| \\
&\quad \le \frac{1}{n}\sum_{j\in[n]} c^j_{ki} \left\|\varphi^{mkj}_{\mathrm{p}(k)}\right\|_{X^{-1}_{m,\mathrm{p}(mkj)}} \beta + \frac{\varepsilon}{dH^2\omega} + \frac{\varepsilon}{dH^2\omega} + |\frac{1}{n}\sum_{j\in[n]} c^j_{ki} a^\top \left(\tilde{F}^{ki}_j - F_{\hat{G}\bar{\theta}}(S^{mkj}_i, \dots, R^{mkj}_H)\right) b|,
\end{aligned}
\tag{130}
$$

where the second inequality uses Lemma 5.E.2 and applies the triangle inequality twice again. Observe that for all $j \in [n]$, given all the history of SkippyEleanor interacting with the MDP up to (and including) $S^{mkj}_{\mathrm{p}(k)}, A^{mkj}_{\mathrm{p}(k)}$ (which also includes the value of $c^j_{ki}$ for $i \in [H+1]$), the trajectory $S^{mkj}_{\mathrm{p}(k)+1}, A^{mkj}_{\mathrm{p}(k)+1}, \dots, R^{mkj}_H$ is an independent rollout with policy $\pi^0$, with its law given by $\mathcal{P}_{\pi^0, S^{mkj}_{\mathrm{p}(k)}, A^{mkj}_{\mathrm{p}(k)}}$. Therefore, for any fixed $(a,b) \in C^+_{\xi'}$, $c^j_{ki} a^\top \left(\tilde{F}^{ki}_j - F_{\hat{G}\bar{\theta}}(S^{mkj}_i, \dots, R^{mkj}_H)\right) b$ are independent zero-mean random variables with range $[-4H, 4H]$. Applying Hoeffding's inequality with a union bound over $m', k, i, a$, and $b$, with probability at least $1-\zeta$, for any of the $m' \in [m'_{\max}]$ times the beginning of the iteration (Line 5) is executed (this includes the entire execution of SkippyEleanor),

$$
\begin{aligned}
|\frac{1}{n}\sum_{j\in[n]} c^j_{ki} a^\top \left(\tilde{F}^{ki}_j - F_{\hat{G}\bar{\theta}}(S^{mkj}_i, \dots, R^{mkj}_H)\right) b| &\le \frac{8H}{\sqrt{n}} \sqrt{\log \frac{2m'_{\max} H^2 |C^+_{\xi'}|}{\zeta}} \\
&= \frac{8H}{\sqrt{n}} \sqrt{2d \log \frac{18dH^3}{\varepsilon} + \log \frac{2m'_{\max} H^2}{\zeta}} \le \frac{\varepsilon}{dH^2\omega},
\end{aligned}
$$

where we used Eq. (129). To finish, note that unless $c^j_{ki} = 0$, $\left\|\varphi^{mkj}_{\mathrm{p}(k)}\right\|_{X^{-1}_{m,\mathrm{p}(mkj)}} < 2(\beta\omega dH)^{-1}$, so we can continue from Eq. (130) by bounding the average feature-norm by $\bar{\sigma}^m_k$ as

$$
|v_\|^\top \left(y^{ki}_{\hat{G}\bar{\theta}} - \hat{F}^{ki}_{\hat{G}\bar{\theta}}\right) w| \le \bar{\sigma}^m_k \beta + 3\frac{\varepsilon}{dH^2\omega}. \qquad \blacksquare
$$

***Proof of Lemma 5.E.4.*** Recall that $(k,i,v)$ are the arguments and $x$ the value of Optimization Problem 5.4.12. Throughout the proof we write $\mathcal{Q}$ to refer to its value *just before* Line 14 is executed. We write $\bullet_\|$ for $\bullet_{\|(\mathcal{Q},i)}$, and $\bullet_\perp$ for $\bullet_{\perp(\mathcal{Q},i)}$. Let $\mathbf{M} = y^{ki}_{\hat{G}\bar{\theta}} - \hat{F}^{ki}_{\hat{G}\bar{\theta}}$. Therefore, $v^\top \mathbf{M} v = x > \bar{\sigma}^m_k \beta\omega + 3\frac{\varepsilon}{dH^2}$, and by Lemma 5.E.1, $w^\top \mathbf{M} w > \bar{\sigma}^m_k \beta\omega + 2\frac{\varepsilon}{dH^2}$.

Line 14 changes $\mathcal{Q}_i$ by appending $\mathcal{Q}_i^{-1} w$ to the sequence $C_i$ of vectors from which $\mathcal{Q}$ is calculated according to Eq. (91). Eq. (92) lists the conditions on the new sequence $C_i$ that need to be

satisfied for $\mathcal{Q}$ to stay a valid preconditioning. Consider the third condition, i.e., $\left\|\mathcal{Q}_i^{-1}w\right\|_2 \le L_3$. Observe that $\mathcal{Q}_i^{-1}\operatorname{Proj}_{Z(\mathcal{Q},i)} \le L_3^2 I$ and $\|v\|_2 = 1$, therefore $\left\|\mathcal{Q}_i^{-1}w\right\|_2 = \left\|\mathcal{Q}_i^{-1}\operatorname{Proj}_{Z(\mathcal{Q},i)} v\right\|_2 \le L_3$.

Now consider the second condition. To prove that it holds, we need to show that $\left\|\mathcal{Q}_i\mathcal{Q}_i^{-1}w\right\|_2 = \|w\|_2 \ge \frac{1}{2}$. Let $x = \|w\|_2^{-1}$. Since $v$ was the argument of the optimization problem, and using Lemma 5.E.1,

$$x^2 w^\top \mathbf{M} w \le v^\top \mathbf{M} v \le w^\top \mathbf{M} w + \frac{\varepsilon}{dH^2\omega} \le w^\top \mathbf{M} w(1+1/2)$$

Therefore, $\|w\|_2^2 \ge \frac{2}{3}$. We immediately get that

$$\left\|\mathcal{Q}_i\mathcal{Q}_i^{-1}w\right\|_2^2 \ge \frac{2}{3},$$

satisfying the second condition.

It remains to prove that the first condition also holds. First, noting that $\mathbf{M}$ is symmetric, we can decompose $w^\top \mathbf{M} w$ as

$$w^\top \mathbf{M} w = w_\parallel^\top \mathbf{M} w + w_\parallel^\top \mathbf{M} w_\perp + w_\perp^\top \mathbf{M} w_\perp .$$

Applying Lemma 5.E.3 on the first two terms,

$$w^\top \mathbf{M} w \le 2\bar{\sigma}_k^m \beta + 6\frac{\varepsilon}{dH^2\omega} + w_\perp^\top \mathbf{M} w_\perp .$$

Due to $\omega > 3$ and $w^\top \mathbf{M} w > \bar{\sigma}_k^m \beta\omega + 2\frac{\varepsilon}{dH^2}$ and the above, $w_\perp \ne \mathbf{0}$. Let $w' = w_\perp/\|w_\perp\|_2$. Since $v$ was the argument of the optimization problem, have that $v^\top \mathbf{M} v \ge w'^\top \mathbf{M} w'$. Putting this together,

$$\|w_\perp\|_2^{-2} w_\perp^\top \mathbf{M} w_\perp = w'^\top \mathbf{M} w' \le v^\top \mathbf{M} v \le w^\top \mathbf{M} w + \frac{\varepsilon}{dH^2\omega} \le 2\bar{\sigma}_k^m \beta + 7\frac{\varepsilon}{dH^2\omega} + w_\perp^\top \mathbf{M} w_\perp ,$$

Since $v^\top \mathbf{M} v > \bar{\sigma}_k^m \beta\omega + 3\frac{\varepsilon}{dH^2}$, $w_\perp^\top \mathbf{M} w_\perp \ge (\omega - 7/3)\left(\bar{\sigma}_k^m \beta + 3\frac{\varepsilon}{dH^2\omega}\right) > 0$ and therefore dividing the above by $w_\perp^\top \mathbf{M} w_\perp$,

$$\begin{aligned}
\|w_\perp\|_2^{-2} &\le \frac{7/3}{\omega - 7/3} + 1 \\
\|w_\perp\|_2^2 &\ge \frac{1}{1+c} \qquad \text{for } c = \frac{7/3}{\omega - 7/3} \\
\left\|w_\parallel\right\|_2^2 &\le 1 - \frac{1}{1+c} \qquad \text{as } \|w\|_2 \le 1 .
\end{aligned}$$

Now to prove that the first condition also holds,

$$\begin{aligned}
\sup_{\theta\in\Theta_i} |\langle \theta, \mathcal{Q}_i^{-1} w\rangle| &= \sup_{\theta\in\Theta_i^{\mathcal{Q}}} |\langle \theta, w\rangle| \le \sup_{\theta\in\Theta_i^{\mathcal{Q}}} \|\theta\|_2 \left\|w_{\|}\right\|_2 + \sup_{\theta\in\Theta_i^{\mathcal{Q}}} |\langle \theta, w_\perp\rangle| \\
&\le \sqrt{d_1+1}\sqrt{1-\frac{1}{1+c}} + \sup_{\theta\in\Theta_i^{\mathcal{Q}}} \|\theta\|_{V(G_h^{\mathcal{Q}},\rho_h^{\mathcal{Q}})^\dagger} \|w_\perp\|_{V(G_h^{\mathcal{Q}},\rho_h^{\mathcal{Q}})} \\
&\le \sqrt{d_1+1}\sqrt{1-\frac{1}{1+c}} + \sqrt{2d w_\perp^\top(\gamma I) w_\perp} \\
&\le \sqrt{d_1+1}\sqrt{1-\frac{1}{1+c}} + \sqrt{2d\gamma} = \sqrt{d_1+1}\sqrt{1-\frac{1}{1+c}} + \frac{1}{2},
\end{aligned}$$

where in the second line we used Lemma 5.4.3 to bound $\sup_{\theta\in\Theta_i^{\mathcal{Q}}} \|\theta\|_2$, and for the second term we used Eq. (113) with Cauchy-Schwarz. In the third line we used Eq. (111), and the definition of $\mathrm{Proj}_\perp$. Finally in the last line we use that $w_\perp$ is perpendicular to $a_i$ for $i \le d'$ (by definition) and that $\lambda_i \le \gamma$ for $i > d'$. It is left to prove that $\sqrt{d_1+1}\sqrt{1-\frac{1}{1+c}} \le \frac{1}{2}$. This holds if $c \ge 1/(4(d_1+1)-1)$, which is satisfied as $c = 1/(3(d_1+1))$, due to $\omega = 7(d_1+1)+7/3$ (Eq. (100)). ■

## 5.I. Deferred proofs for Section 5.E.2

***Proof of Lemma 5.E.5.*** The features $\varphi^{lkj}_{\mathrm{p}(k)}$ are observed by SkippyEleanor in the order of increasing $l$, within that increasing $k$, and within that, increasing $j$. Each time the next $\varphi^{lkj}_{\mathrm{p}(k)}$ is observed, we sum the elliptic potential as follows.

For $i \in [m], r \in [H], u \in [n], t \in [H]$, let the set of indices observed before $\varphi^{iru}_{\mathrm{p}(r)}$ whose Phase II (rollout phase) starts at some stage $t$ be:

$$\mathbf{I}^{iru}(t) = \{l \in [i], k \in [H], j \in [n] : lHn + kn + j < iHn + rn + u \text{ and } \mathrm{p}(lkj) = t\}$$

Let a version of this where only the whole iteration $i$'s data is included be

$$\mathbf{J}^{i}(t) = \{l = i, k \in [H], j \in [n] : \mathrm{p}(lkj) = t\}$$

Let

$$X_{iru}(t) = \lambda I + \sum_{lkj \in \mathbf{I}^{iru}(t)} \varphi^{lkj}_{\mathrm{p}(k)} \varphi^{lkj\ \top}_{\mathrm{p}(k)}$$

Observe that $X_{it}$, defined in Optimization Problem 5.4.10, is the version of this that only updates at the start of each iteration $i$, that is,

$$X_{it} = X_{i11}(t)\,.$$

The total elliptic potential, observed by the end of iteration $m$ is, writing $k = \mathrm{p}(iru)$ on the left hand side:

$$\sum_{i\in[m],r\in[H],u\in[n]} \mathbb{1}\{k < H+1\} \min\left\{1, \left\|\varphi_k^{iru}\right\|^2_{X_{iru}(k)^{-1}}\right\} = \sum_{i\in[m],t\in[H]} \sum_{lkj\in\mathbf{J}^i(t)} \min\left\{1, \left\|\varphi_t^{lkj}\right\|^2_{X_{lkj}(t)^{-1}}\right\}.$$

Applying the elliptical potential lemma (Lemma 5.L.1) $H$ times for $t \in [H]$, this can be bounded as

$$\sum_{t\in[H],i\in[m]} \sum_{lkj\in\mathbf{J}^i(t)} \min\left\{1, \left\|\varphi_t^{lkj}\right\|^2_{X_{lkj}(t)^{-1}}\right\} \le 2dH\log\left(1+\frac{HmnL^2}{d\lambda}\right)$$

On the other hand, by Lemma 5.L.2, then switching to an $\ell_1$-bound, then observing that by definition, $\sum \bar{\sigma}_k^i$ sums the same quantities but caps them by some threshold,

$$\begin{aligned}
\sum_{t\in[H],i\in[m]} \sum_{lkj\in\mathbf{J}^i(t)} \min\left\{1, \left\|\varphi_t^{lkj}\right\|^2_{X_{lkj}(t)^{-1}}\right\} &\ge \sum_{t\in[H],i\in[m]} \min\left\{1, \frac{1}{2}\sum_{lkj\in\mathbf{J}^i(t)} \left\|\varphi_t^{lkj}\right\|^2_{X_{it}^{-1}}\right\} \\
&\ge \sum_{i\in[m]} \min\left\{1, \frac{1}{2}\sum_{t\in[H]} \sum_{lkj\in\mathbf{J}^i(t)} \left\|\varphi_t^{lkj}\right\|^2_{X_{it}^{-1}}\right\} \\
&\ge \sum_{i\in[m]} \min\left\{1, \frac{1}{2Hn}\left(\sum_{t\in[H]} \sum_{lkj\in\mathbf{J}^i(t)} \left\|\varphi_t^{lkj}\right\|_{X_{it}^{-1}}\right)^2\right\} \\
&\ge \sum_{i\in[m]} \min\left\{1, \frac{1}{2Hn}\left(n\sum_{k\in[H]} \bar{\sigma}_k^i\right)^2\right\}
\end{aligned}$$

Whenever an iteration finishes without returning in Line 19, $\sum_{k\in[H]} \bar{\sigma}_k^m > \varepsilon/(dH^2\beta\omega)$. Therefore,

$$\begin{aligned}
2dH\log\left(1+\frac{HmnL^2}{d\lambda}\right) &\ge \sum_{i\in[m]} \min\left\{1, \frac{1}{2Hn}\left(n\sum_{k\in[H]} \bar{\sigma}_k^i\right)^2\right\} \\
&\ge \sum_{i\in[m]} \min\left\{1, \frac{1}{2H}n\left(\frac{\varepsilon}{dH^2\beta\omega}\right)^2\right\} \\
&\ge \sum_{i\in[m]} \min\left\{1, Hd/\beta^2\right\} = mHd/\beta^2,
\end{aligned}$$

Therefore, even for the iteration that returns in Line 19,

$$m \le \beta^2 \log\left(1 + \frac{HmnL^2}{d\lambda}\right) + 1 = m_{\max}\,. \qquad \blacksquare$$

## 5.J. Deferred proofs for Section 5.E.3

***Proof of Lemma 5.E.8.*** For notational simplicity we drop the subscripts $(\hat{G}, \bar{\theta})$. We first use the usual high-probability bounds on the least squares predictor and Hoeffding's inequality on the empirical mean quantities, to prove that with probability at least $1 - 3\zeta$, during the execution of SKIPPYELEANOR whenever Line 18 is executed, for all $k \in [H]$,

$$\mathbb{E}_{\pi^{mk}, s_1} \tilde{c}_{k,H+1} C(S_{p(k)}) \le \mathbb{E}_{\pi^{mk}, s_1} \sum_{u=p(k)}^{H} R_u + \tilde{c}_{k,H+1} E^{\rightarrow}(S_{p(k)+1}, \ldots, R_H) + 2\bar{\sigma}_k^m \beta \omega d H + 4\frac{\varepsilon}{H}\,. \tag{131}$$

The proof of this is presented as Lemma 5.J.1.

Next, to prove the statement for $k \in [H]$, assume by induction that Eq. (109) holds for $i \in [k+1 : H]$.

Observe that SKIPPYPOLICY performs a rollout with policy $\pi^0$ for the rest of the episode starting from stage $p(k)+1$, that is, $1 = A_{p(k)+1} = \cdots = A_H$. Therefore, the law of the random variables $S_{p(k)+1}, \ldots, R_H$, given $(S_{p(k)}, A_{p(k)})$ is fully determined by the dynamics of the MDP, and is independent of the values of $p(k+1), \ldots, p(H)$. Therefore,

$$\begin{aligned} \mathbb{E}_{\pi^{mk}, s_1} \tilde{c}_{k+1,H+1} C(S_{p(k+1)}) &= \mathbb{E}_{\pi^{mk}, s_1} \tilde{c}_{k+1,H+1} D(S_{p(k+1)}, \ldots, R_H) + \sum_{u=p(k+1)}^{H} R_u \\ &= \mathbb{E}_{\pi^{mk}, s_1} \tilde{c}_{k,H+1} E^{\rightarrow}(S_{p(k)+1}, \ldots, R_H) + \sum_{u=p(k+1)}^{H} R_u\,, \end{aligned} \tag{132}$$

where we use Eq. (96), and that $\pi^{mk}$ (SKIPPYPOLICY) is in phase II after stage $p(k)$, but defines the the mapping $p(\cdot)$ independently of whether the policy is in phase I or phase II, in such a way that for any $H \ge j > p(k)$,

$$\mathcal{P}_{\pi^{mk}, s_1}\left[p(k+1) = j \,\middle|\, p(k), S_{p(k)}, A_{p(k)}\right] = \mathcal{P}_{\pi^{mk}, s_1}\left[\tau(S_j) \prod_{j'=p(k)+1}^{j-1} (1 - \tau(S_{j'})) \,\middle|\, p(k), S_{p(k)}, A_{p(k)}\right].$$

Combining Eq. (132) with Eq. (131),

$$\mathbb{E}_{\pi^{mk},s_1}\tilde{c}_{k,H+1}C(S_{p(k)}) \le \mathbb{E}_{\pi^{mk},s_1}\sum_{u=p(k)}^{p(k+1)-1} R_u + \tilde{c}_{k+1,H+1}C(S_{p(k+1)}) + 2\bar{\sigma}_k^m\beta\omega dH + 4\frac{\varepsilon}{H}.$$

By Remark 5.E.7, $\mathbb{E}_{\pi^{mk},s_1}\tilde{c}_{k+1,H+1}C(S_{p(k+1)}) = \mathbb{E}_{\pi^{m,k+1},s_1}\tilde{c}_{k+1,H+1}C(S_{p(k+1)}) = \bar{C}^{k+1}$. Therefore, combining with the inductive hypothesis,

$$\begin{aligned}
\mathbb{E}_{\pi^{mk},s_1}\tilde{c}_{k,H+1}C(S_{p(k)}) &\le \mathbb{E}_{\pi^{mk},s_1}\sum_{u=p(k)}^{p(k+1)-1} R_u + \bar{C}^{k+1} + 2\bar{\sigma}_k^m\beta\omega dH + 4\frac{\varepsilon}{H} \\
&\le \mathbb{E}_{\pi^{mk},s_1}\sum_{u=p(k)}^{p(k+1)-1} R_u + \mathbb{E}_{\pi^{mH},s_1}\sum_{u=p(k+1)}^{H} R_u + 2\sum_{i=k}^{H}\bar{\sigma}_k^m\beta\omega dH + 4(H-k+1)\frac{\varepsilon}{H} \\
&= \mathbb{E}_{\pi^{mH},s_1}\sum_{u=p(k)}^{H} R_u + 2\sum_{i=k}^{H}\bar{\sigma}_k^m\beta\omega dH + 4(H-k+1)\frac{\varepsilon}{H}
\end{aligned}$$

where the last equation uses Remark 5.E.7 again, finishing the induction. ■

**Lemma 5.J.1.** *Adopt the notation of Lemma 5.E.8. With probability at least* $1-3\zeta$*, during the execution of* SkippyEleanor*, whenever Line 18 is executed, for all* $k \in [H]$*,*

$$\mathbb{E}_{\pi^{mk},s_1}\tilde{c}_{k,H+1}C(S_{p(k)}) \le \mathbb{E}_{\pi^{mk},s_1}\sum_{u=p(k)}^{H} R_u + \tilde{c}_{k,H+1}E^{\rightarrow}(S_{p(k)+1},\dots,R_H) + 2\bar{\sigma}_k^m\beta\omega dH + 4\frac{\varepsilon}{H}.$$

*Proof.* We refer as $\hat{\theta}$ to the value of the argument of Optimization Problem 5.4.10 recorded in Line 5. For $k \in [H]$, recall the definition of $\bar{\sigma}_k^m$ (Eq. (105)), along with the fact that unless $c^j_{k,H+1} = 0$, $\left\|\varphi^{mkj}_{\mathrm{p}(k)}\right\|_{X^{-1}_{m,\mathrm{p}(mkj)}} < 2(\beta\omega dH)^{-1}$, we get a useful bound on the average norm of the features under consideration:

$$\frac{1}{n}\sum_{j\in[n]} c^j_{k,H+1}\left\|\varphi^{mkj}_{\mathrm{p}(k)}\right\|_{X^{-1}_{m,\mathrm{p}(mkj)}} \le \bar{\sigma}_k^m. \tag{133}$$

If Line 18 is executed, the consistency check passed, and therefore for all $k \in [H-1], i \in [k+1:H]$,

$$\mathrm{Tr}\left(y^{ki} - \hat{F}^{ki}\right) \le \bar{\sigma}_k^m\beta\omega d + 3\frac{\varepsilon}{H^2} \tag{134}$$

For $t \in [H]$ let the least-squares predictor of rewards sums under the policy $\pi^0$ be

$$\breve{\theta}^{t,H+1} = X_{mt}^{-1} \sum_{lkj \in \mathbf{I}^m(t)} \varphi_t^{lkj} \sum_{u=t}^{H} R_u^{lkj} .$$

For $k \in [H]$ and $j \in [n]$ let us introduce the shorthand

$$R_{k\to}^{mkj} = \sum_{u=\mathrm{p}(mkj)}^{H} R_u^{mkj} ,$$

and similarly when the trajectory is clear from context: $R_{k\to} = \sum_{u=p(k)}^{H} R_u$. For $k \in [H]$ let

$$\begin{aligned}
\hat{E}^k &= \frac{1}{n} \sum_{j\in[n]} c_{k,H+1}^j \left( E^{\to}(S_{\mathrm{p}(k)+1}^{mkj}, \dots, R_H^{mkj}) + R_{k\to}^{mkj} \right) \\
\hat{C}^k &= \frac{1}{n} \sum_{j\in[n]} c_{k,H+1}^j C(S_{\mathrm{p}(k)}^{mkj}) \\
y^{k,H+1} &= \frac{1}{n} \sum_{j\in[n]} c_{k,H+1}^j \left\langle \varphi_{\mathrm{p}(k)}^{mkj}, \breve{\theta}^{\mathrm{p}(mkj),H+1} \right\rangle \\
z^{k,H+1} &= \frac{1}{n} \sum_{j\in[n]} c_{k,H+1}^j R_{k\to}^{mkj}
\end{aligned}$$

For $t \in [H-1]$, $i \in [t+1:H]$, along with $\breve{\theta}^{t,H+1}$, let

$$\breve{\theta}^{ti} = X_{mt}^{-1} \sum_{lkj \in \mathbf{I}^m(t)} \varphi_t^{lkj} \mathrm{Tr}(F(S_i^{lkj}, \dots, R_H^{lkj})) = X_{mt}^{-1} \sum_{lkj \in \mathbf{I}^m(t)} \varphi_t^{lkj} E(S_i^{lkj}, \dots, R_H^{lkj}) ,$$

where the second equality is by Eq. (119). Observe that for any $v \in \mathbb{R}^d$, $\mathrm{Tr}(v^\top \hat{\theta}^{ti}) = \left\langle v, \breve{\theta}^{ti} \right\rangle$. Therefore, for $k \in [H]$,

$$y^{k,H+1} + \sum_{i=k+1}^{H} \mathrm{Tr}(y^{ki}) = \frac{1}{n} \sum_{j\in[n]} \sum_{i=k+1}^{H+1} c_{ki}^j \left\langle \varphi_{\mathrm{p}(k)}^{mkj}, \breve{\theta}^{\mathrm{p}(mkj),i} \right\rangle = \frac{1}{n} \sum_{j\in[n]} c_{k,H+1}^j \left\langle \varphi_{\mathrm{p}(k)}^{mkj}, \sum_{i=\mathrm{p}(mkj)+1}^{H+1} \breve{\theta}^{\mathrm{p}(mkj),i} \right\rangle$$

For any $t \in [H]$, by the definitions,

$$\begin{aligned}
\sum_{i=t+1}^{H+1} \breve{\theta}^{ti} &= X_{mt}^{-1} \sum_{lkj \in \mathbf{I}^m(t)} \varphi_t^{lkj} \left( \sum_{i=t+1}^{H} E(S_i^{lkj}, \dots, R_H^{lkj}) + \sum_{u=t}^{H} R_u^{mkj} \right) \\
&= X_{mt}^{-1} \sum_{lkj \in \mathbf{I}^m(t)} \varphi_t^{lkj} \left( E^{\to}(S_{t+1}^{lkj}, \dots, R_H^{lkj}) + \sum_{u=t}^{H} R_u^{mkj} \right) = \hat{\theta}_t
\end{aligned}$$

Plugging this into the previous calculation,

$$\begin{aligned} y^{k,H+1} + \sum_{i=k+1}^{H} \mathrm{Tr}(y^{ki}) &= \frac{1}{n} \sum_{j\in[n]} c^j_{k,H+1} \left\langle \varphi^{mkj}_{\mathrm{p}(k)}, \hat{\theta}_{\mathrm{p}(mkj)} \right\rangle \\ &\geq \frac{1}{n} \sum_{j\in[n]} c^j_{k,H+1} \left\langle \varphi^{mkj}_{\mathrm{p}(k)}, \bar{\theta}_{\mathrm{p}(mkj)} \right\rangle \\ &\quad - \frac{1}{n} \sum_{j\in[n]} c^j_{k,H+1} \left\| \varphi^{mkj}_{\mathrm{p}(k)} \right\|_{X^{-1}_{m,\mathrm{p}(mkj)}} \left\| \bar{\theta}_{\mathrm{p}(mkj)} - \hat{\theta}_{\mathrm{p}(mkj)} \right\|_{X_{m,\mathrm{p}(mkj)}} \\ &\geq \frac{1}{n} \sum_{j\in[n]} c^j_{k,H+1} \left\langle \varphi^{mkj}_{\mathrm{p}(k)}, \bar{\theta}_{\mathrm{p}(mkj)} \right\rangle - \bar{\sigma}^m_k \beta H \\ &\geq \frac{1}{n} \sum_{j\in[n]} c^j_{k,H+1} \mathrm{clip}_{[0,H]} \left\langle \varphi^{mkj}_{\mathrm{p}(k)}, \bar{\theta}_{\mathrm{p}(mkj)} \right\rangle - \bar{\sigma}^m_k \beta H \\ &= \frac{1}{n} \sum_{j\in[n]} c^j_{k,H+1} C(S^{mkj}_{\mathrm{p}(k)}) - \bar{\sigma}^m_k \beta H = \hat{C}^k - \bar{\sigma}^m_k \beta H \,, \end{aligned} \tag{135}$$

where the first inequality uses Cauchy-Schwarz. The second inequality bounds the average of the first norm by Eq. (133), and the bound on the second norm (for any $j$) is by definition of Optimization Problem 5.4.10. The third inequality relies on the fact that $c^j_{k,H+1} = 0$ if the clipped inner product is negative, and the final equality is due to the definition of $C$ along with the fact that $A^{mkj}_{\mathrm{p}(k)} = \pi^+(S^{mkj}_{\mathrm{p}(k)})$, as this is the last state in the trajectory where SkippyPolicy takes the inner-product maximizing action ($\pi^+$) before rolling out with $\pi^0$.

By Eqs. (99) and (119), we have that

$$\begin{aligned} z^{k,H+1} + \sum_{i\in[k+1:H]} \mathrm{Tr}(\hat{F}^{ki}) &= \frac{1}{n} \sum_{j\in[n]} c^j_{k,H+1} \left( \sum_{i=\mathrm{p}(mkj)+1}^{H+1} E(S^{mkj}_i, \ldots, R^{mkj}_H) + R^{mkj}_{k\rightarrow} \right) \\ &= \frac{1}{n} \sum_{j\in[n]} c^j_{k,H+1} \left( E^{\rightarrow}(S^{mkj}_{\mathrm{p}(k)+1}, \ldots, R^{mkj}_H) + R^{mkj}_{k\rightarrow} \right) = \hat{E}^k \end{aligned} \tag{136}$$

Combining Eqs. (135) and (136),

$$\begin{aligned} \hat{C}^k - \hat{E}^k &\leq \bar{\sigma}^m_k \beta H + \left( y^{k,H+1} - z^{k,H+1} \right) + \sum_{i\in[k+1:H]} \mathrm{Tr}(y^{ki} - \hat{F}^{ki}) \\ &\leq \bar{\sigma}^m_k \beta H + \left( |\mathbb{E}_{\pi^{mk},s_1} c_{k,H+1} R_{k\rightarrow} - z^{k,H+1}| + |y^{k,H+1} - \mathbb{E}_{\pi^{mk},s_1} c_{k,H+1} R_{k\rightarrow}| \right) + \bar{\sigma}^m_k \beta \omega d H + 3\frac{\varepsilon}{H} \end{aligned} \tag{137}$$

where the sum (last term) is bounded by Eq. (134), and we apply a triangle inequality on the second term. To continue bounding this term, we apply Hoeffding's inequality on the independent random

variables $c^j_{k,H+1} R_{k\to}$ (for $j \in [n]$) that have range $[0,H]$, along with a union bound over the iteration $m' \in [m'_{\max}]$ and $k \in [H]$, to get that with probability at least $1-\zeta$,

$$|\mathbb{E}_{\pi^{mk},s_1} c_{k,H+1} R_{k\to} - z^{k,H+1}| \le \frac{H}{\sqrt{n}} \sqrt{\log \frac{2m'_{\max} H}{\zeta}} \le \frac{\varepsilon}{dH^2\omega}. \tag{138}$$

The remaining term $|y^{k,H+1} - \mathbb{E}_{\pi^{mk},s_1} c_{k,H+1} R_{k\to}|$ is bounded using the realizability of $q^{\pi^0}$ (Definition 5.3.2) as follows. Take any $t \in [H]$. By definition there exists $\theta^\star_t \in \Theta^{\mathcal{Q}}_t \subseteq \mathcal{B}(B)$, such that for all $s \in \mathcal{S}_t$ and $a \in [\mathcal{A}]$, $q^{\pi^0}(s,a) \approx_\eta \langle \varphi(s,a), \theta^\star_t \rangle$. Take the sequence $A$ formed of $\varphi^{lkj}_t$ (for $lkj \in \mathbf{I}^m(t)$, in the order that these random variables are observed), and the sequence $X$ formed of $R^{mkj}_{k\to}$ (for $lkj \in \mathbf{I}^m(t)$, in the same order), and the sequence $\Delta$ formed of $q^{\pi^0}(S^{lkj}_t, A^{lkj}_t) - \langle \varphi^{lkj}_t, \theta^\star_t \rangle$ (for $lkj \in \mathbf{I}^m(t)$, in the same order). Then the sequences $A$, $X$, and $\Delta$ satisfy the conditions of Lemma 5.M.4 with a subgaussianity parameter $\sigma = H$. Due to this lemma, applied with a union bound over $m' \in [m'_{\max}]$ and $t \in [H]$, with probability at least $1-\zeta$,

$$\begin{aligned}\left\|\check{\theta}^{t,H+1} - \theta^\star_t\right\|_{X_{mt}} &< \sqrt{\lambda}\left\|\theta^\star_t\right\|_2 + \|\Delta\|_\infty \sqrt{|\mathbf{I}^m(t)|} + H\sqrt{2\log\left(\frac{m'_{\max}H}{\zeta}\right) + \log\left(\frac{\det X_{mt}}{\lambda^d}\right)} \\ &\le 2 + H\sqrt{2\log\frac{m'_{\max}H}{\zeta} + \log\left(\frac{\det X_{mt}}{\lambda^d}\right)} \le \beta,\end{aligned}$$

by Eq. (128). Therefore by Cauchy-Schwarz and Eq. (133),

$$\begin{aligned}|y^{k,H+1} - \mathbb{E}_{\pi^{mk},s_1} c_{k,H+1} R_{k\to}| &\le \frac{1}{n}\sum_{j\in[n]} c^j_{k,H+1}\left(\left\|\varphi^{mkj}_{\mathrm{p}(k)}\right\|_{X^{-1}_{m,\mathrm{p}(mkj)}} \left\|\check{\theta}^{\mathrm{p}(mkj),H+1} - \theta^\star_{\mathrm{p}(mkj)}\right\|_{X_{mt}} + \eta\right) \\ &\le \bar{\sigma}^m_k \beta + \eta \le \bar{\sigma}^m_k \beta + \frac{\varepsilon}{dH^2\omega}.\end{aligned}$$

Combining this with Eqs. (137) and (138),

$$\hat{C}^k - \hat{E}^k \le 1.5\bar{\sigma}^m_k \beta\omega dH + 3\frac{\varepsilon}{H} + 2\frac{\varepsilon}{dH^2\omega}. \tag{139}$$

We introduce the following notation for $j \in [n]$, $k \in [H+1]$, $i \in [H+1]$:

$$\begin{aligned}\bar{c}^j_{ki} &= \mathbb{1}\left\{\mathrm{p}(mkj) < i \text{ and } \left\|\varphi^{mkj}_{\mathrm{p}(k)}\right\|_{X^{-1}_{m,\mathrm{p}(mkj)}} \ge 2(\beta\omega dH)^{-1} \text{ and } \left\langle \varphi^{mkj}_{\mathrm{p}(k)}, \bar{\theta}_{\mathrm{p}(mkj)} \right\rangle \ge 0\right\} \\ \hat{c}^j_{ki} &= \mathbb{1}\left\{\mathrm{p}(mkj) < i \text{ and } \left\langle \varphi^{mkj}_{\mathrm{p}(k)}, \bar{\theta}_{\mathrm{p}(mkj)} \right\rangle < 0\right\},\end{aligned}$$

such that for all $j$,

$$\tilde{c}^j_{ki} = c^j_{ki} + \bar{c}^j_{ki} + \hat{c}^j_{ki}\,. \tag{140}$$

Continuing from Eq. (139), as $E^{\rightarrow}(s_i \rightarrow) + \sum_{u=i}^{H} r_u \geq 0$ by Eq. (118), and if $\hat{c}^j_{k,H+1} = 1$ then $C(S^{mkj}_{\mathrm{p}(k)}) = 0$, we have that

$$\frac{1}{n}\sum_{j\in[n]} (c^j_{k,H+1} + \hat{c}^j_{k,H+1}) \left(C(S^{mkj}_{\mathrm{p}(k)}) - \left(E^{\rightarrow}(S^{mkj}_{\mathrm{p}(k)+1}, \ldots, R^{mkj}_H) + R^{mkj}_{k\rightarrow}\right)\right) \leq 1.5\bar{\sigma}^m_k \beta\omega dH + 3\frac{\varepsilon}{H} + 2\frac{\varepsilon}{dH^2\omega}\,.$$

As (even if $\bar{c}^j_{k,H+1} = 1$) $C(S^{mkj}_{\mathrm{p}(k)}) \leq H$,

$$\frac{1}{n}\sum_{j\in[n]} \bar{c}^j_{k,H+1} C(S_{p(k)}) \leq H\bar{\sigma}^m_k / (2(\beta\omega dH)^{-1}) = \frac{1}{2}\bar{\sigma}^m_k \beta\omega dH\,,$$

which combined with the previous inequality and Eq. (140) yields

$$\frac{1}{n}\sum_{j\in[n]} \tilde{c}^j_{k,H+1} \left(C(S^{mkj}_{\mathrm{p}(k)}) - \left(E^{\rightarrow}(S^{mkj}_{\mathrm{p}(k)+1}, \ldots, R^{mkj}_H) + R^{mkj}_{k\rightarrow}\right)\right) \leq 2\bar{\sigma}^m_k \beta\omega dH + 3\frac{\varepsilon}{H} + 2\frac{\varepsilon}{dH^2\omega}\,.$$

Observe that the random variables $\tilde{c}^j_{k,H+1} \left(C(S^{mkj}_{\mathrm{p}(k)}) - \left(E^{\rightarrow}(S^{mkj}_{\mathrm{p}(k)+1}, \ldots, R^{mkj}_H) + R^{mkj}_{k\rightarrow}\right)\right)$ are independent (for $j \in [n]$) with range $[-2H, H]$ (Eq. (118)). By Hoeffding's inequality, with probability at least $1-\zeta$, for all iteration $m' \in [m'_{\max}]$ (this includes the entire execution of SKIPPYELEANOR) and $k \in [H]$,

$$\begin{aligned}
&\Bigg| \mathbb{E}_{\pi^{mk}, s_1} \tilde{c}_{k,H+1} \left(C(S_{p(k)}) - \left(E^{\rightarrow}(S_{p(k)+1}, \ldots, R_H) + R_{k\rightarrow}\right)\right) \\
&\qquad - \frac{1}{n}\sum_{j\in[n]} \tilde{c}^j_{k,H+1} \left(C(S^{mkj}_{\mathrm{p}(k)}) - \left(E^{\rightarrow}(S^{mkj}_{\mathrm{p}(k)+1}, \ldots, R^{mkj}_H) + R^{mkj}_{k\rightarrow}\right)\right) \Bigg| \\
&\qquad \leq \frac{4H}{\sqrt{n}} \sqrt{\log \frac{2m'_{\max} H}{\zeta}} \leq \frac{\varepsilon}{dH^2\omega}\,.
\end{aligned}$$

Combining with the previous bound, under the intersection of the high-probability events referred to above, which by a union bound has a probability of at least $1-3\zeta$, we have that for all $k \in [H]$,

$$\mathbb{E}_{\pi^{mk}, s_1} \tilde{c}_{k,H+1} C(S_{p(k)}) \leq \mathbb{E}_{\pi^{mk}, s_1} \sum_{u=p(k)}^{H} R_u + \tilde{c}_{k,H+1} E^{\rightarrow}(S_{p(k)+1}, \ldots, R_H) + 2\bar{\sigma}^m_k \beta\omega dH + 4\frac{\varepsilon}{H}\,. \quad \blacksquare$$

## 5.K. Deferred proofs for Section 5.E.4

***Proof of Lemma 5.E.10.*** Let $m$ be the current iteration. Unlike in previous lemmas, here we introduce $(\hat{G}, \bar{\theta})$ that does *not* refer to the outcome of Optimization Problem 5.4.10. Instead, let $\hat{G} = (\vartheta_h^i)_{h\in[2:H],i\in[d_0]} \in \mathbf{G}$ be the correct guess. For $h = H, \ldots, 1$, $\bar{\theta}_h$ is defined in sequence along with the behavior of a policy $\pi$ on stage $h$.

For $h = H, \ldots, 1$, assuming that this process already defined $\bar{\theta}_{h+1}, \ldots, \bar{\theta}_H$ (in Eq. (142)), let $\pi$ be the policy that, for any $t > h$ and $s \in \mathcal{S}_t$, takes action on $s$ as $\pi^+_{\hat{G}\bar{\theta}}(s)$ with probability $\tau_{\hat{G}\bar{\theta}}(s)$, and action 1 with probability $1 - \tau_{\hat{G}\bar{\theta}}(s)$ ($\tau$ is defined in Eq. (95)). Simultaneously, using the second part of Corollary 5.4.11, define $\tilde{\theta}_{hi} \in \mathcal{B}(4d_0B/\alpha)$ for $i \in [h+1:H]$ to satisfy for all $s \in \mathcal{S}_h$, $a \in [\mathcal{A}]$:

$$\mathbb{E}_{\pi^0,s,a} \operatorname{Tr}(\bar{F}_{\hat{G}\bar{\theta}}(S_i)) \approx_{\eta_0} \left\langle \varphi(s,a), \tilde{\theta}_{hi} \right\rangle .$$

We also define $\tilde{\theta}_{h,H+1} \in \mathcal{B}(B)$ to satisfy for all $s \in \mathcal{S}_h$, $a \in [\mathcal{A}]$:

$$\mathbb{E}_{\pi^0,s,a} \sum_{u=h}^{H} R_u \approx_{\eta} \left\langle \varphi(s,a), \tilde{\theta}_{h,H+1} \right\rangle .$$

By Eq. (119),

$$\mathbb{E}_{\pi^0,s,a} \sum_{i\in[h+1:H]} \operatorname{Tr}(\bar{F}_{\hat{G}\bar{\theta}}(S_i)) + \sum_{u=h}^{H} R_u = \mathbb{E}_{\pi^0,s,a} E^{\rightarrow}_{\hat{G}\bar{\theta}}(S_{h+1}, \ldots, R_H) + \sum_{u=h}^{H} R_u \approx_{H\eta_0} \left\langle \varphi(s,a), \bar{\theta}_h \right\rangle , \tag{141}$$

where we define

$$\bar{\theta}_h = \sum_{i\in[h+1:H+1]} \tilde{\theta}_{hi} . \tag{142}$$

We first show that $(\hat{G}, \bar{\theta})$ is feasible for Optimization Problem 5.4.10. Clearly, $\left\|\bar{\theta}_h\right\|_2 \le 4d_0HB/\alpha$. For any $i \in [h+1:H]$, let

$$\hat{\theta}_{hi} = X^{-1}_{mh} \sum_{lkj\in\mathbf{I}^m(h)} \varphi_h^{lkj} \operatorname{Tr}(F_{\hat{G}\bar{\theta}}(S^{lkj}_{h+1}, \ldots, R^{lkj}_H)) ,$$

and let

$$\hat{\theta}_{h,H+1} = X^{-1}_{mh} \sum_{lkj\in\mathbf{I}^m(h)} \varphi_h^{lkj} \sum_{u=h}^{H} R_u^{lkj} .$$

Then, $\hat{\theta}$ of Optimization Problem 5.4.10 satisfies for all $h \in [H]$, by Eq. (119),

$$\hat{\theta}_h = \sum_{i \in [h+1:H+1]} \hat{\theta}_{hi} \,.$$

To show that $(\hat{G}, \bar{\theta})$ is feasible, it thus suffices to show for all $h \in [H]$, $i \in [h+1 : H+1]$, that $\left\|\tilde{\theta}_{hi} - \hat{\theta}_{hi}\right\|_{X_{mh}} \leq \beta$.

Fix any $h \in [H]$ and $i \in [h+1 : H+1]$. Take the sequence $A$ formed of $\varphi_t^{lkj}$ (for $lkj \in \mathbf{I}^m(h)$, in the order that these random variables are observed). For $i < H+1$ take the sequence $X$ formed of $\mathrm{Tr}(F_{\hat{G}\bar{\theta}}(S_i^{lkj}, \ldots, R_H^{lkj}))$ (for $lkj \in \mathbf{I}^m(h)$, in the same order), and the sequence $\Delta$ formed of $\mathbb{E}_{\pi^0, S_h^{lkj}, A_h^{lkj}} \mathrm{Tr}(\bar{F}_{\hat{G}\bar{\theta}}(S_i)) - \left\langle \varphi_h^{lkj}, \tilde{\theta}_{hi} \right\rangle$ (for $lkj \in \mathbf{I}^m(h)$, in the same order). For $i = H+1$, the sequence $X$ is formed of $\sum_{u=h}^{H} R_u^{lkj}$, and $\Delta$ is formed of $q^{\pi^0}(S_h^{lkj}, A_h^{lkj}) - \left\langle \varphi_h^{lkj}, \tilde{\theta}_{hi} \right\rangle$. Then the sequences $A$, $X$, and $\Delta$ satisfy the conditions of Lemma 5.M.4 with a subgaussianity parameter $\sigma = H$. Due to this lemma, applied with a union bound over $m' \in [m'_{\max}]$, $t$, and $i$, with probability at least $1 - \zeta$,

$$\begin{aligned}
\left\|\hat{\theta}_{hi} - \tilde{\theta}_{hi}\right\|_{X_{mh}} &< \sqrt{\lambda} \left\|\tilde{\theta}_{hi}\right\|_2 + \|\Delta\|_\infty \sqrt{|\mathbf{I}^m(t)|} + H \sqrt{2 \log\left(\frac{m'_{\max} H^2}{\zeta}\right) + \log\left(\frac{\det X_{mt}}{\lambda^d}\right)} \\
&\leq 2 + H \sqrt{2 \log \frac{m'_{\max} H^2}{\zeta} + \log\left(\frac{\det X_{mt}}{\lambda^d}\right)} \leq \beta \,,
\end{aligned}$$

by Eq. (128).

Next, we show that the resulting policy $\pi$ is near-optimal. Assume by induction on $h = H, \ldots, 1$, that for all $t \in [h+1 : H]$, all $s \in \mathcal{S}_t$ and $a \in [\mathcal{A}]$,

$$v^\pi(s) \geq v^\star(s) - (H - t + 1)(\varepsilon/H + 2H^2\eta_0) \quad \text{and} \tag{143}$$

$$\left\langle \varphi(s,a), \bar{\theta}_t \right\rangle \approx_{(H-t+1)H\eta_0} q^\pi(s,a) \,. \tag{144}$$

To prove the above for $t = h$ as well, take any $s \in \mathcal{S}_h, a \in [\mathcal{A}]$. Introduce the random variable $P$ that, for a trajectory following $\mathcal{P}_{\pi^0, s, a}$, takes as its value the index of the first Bernoulli draw of 1 (starting from index $h+1$), when the Bernoullis have means $\tau_{\hat{G}\bar{\theta}}(S_j)$ for $j \in [h+1 : H]$, and takes the value $H+1$ if all of these Bernoullis have outcome 0. Write $\mathbb{E}_{\pi^0, s, a, P}[\cdot]$ for $\mathbb{E}_{\pi^0, s, a} \mathbb{E}_P[\cdot \,|\, S_{h+1}, \ldots, R_H]$.

Then,

$$\mathbb{E}_{\pi^0,s,a}\overrightarrow{E}_{\hat{G}\bar{\theta}}(S_{h+1},\dots,R_H)+\sum_{u=h}^{H}R_u=\mathbb{E}_{\pi^0,s,a,P}D_{\hat{G}\bar{\theta}}(S_P,\dots,R_H)+\sum_{u=h}^{H}R_u$$
$$=\mathbb{E}_{\pi^0,s,a,P}\sum_{u=h}^{P-1}R_u+\mathbb{1}\{P<H+1\}C_{\hat{G}\bar{\theta}}(S_P)$$

where we use Eq. (96). Combining with Eq. (141),

$$\left\langle\varphi(s,a),\bar{\theta}_h\right\rangle\approx_{H\eta_0}\mathbb{E}_{\pi^0,s,a,P}\sum_{u=h}^{P-1}R_u+\mathbb{1}\{P<H+1\}C_{\hat{G}\bar{\theta}}(S_P)$$
$$=\mathbb{E}_{\pi^0,s,a,P}\sum_{u=h}^{P-1}R_u+\mathbb{1}\{P<H+1\}\operatorname{clip}_{[0,H]}\left\langle\varphi(S_P,\pi^+_{\hat{G}\bar{\theta}}(S_P)),\bar{\theta}_P\right\rangle$$
$$\approx_{(H-h)H\eta_0}\mathbb{E}_{\pi^0,s,a,P}\sum_{u=h}^{P-1}R_u+\mathbb{1}\{P<H+1\}q^\pi(S_P,\pi^+_{\hat{G}\bar{\theta}}(S_P)),$$

where we used the inductive assumption along with the fact that action-values are bounded in $[0,H]$. Observe also that

$$q^\pi(s,a)=\mathbb{E}_{\pi^0,s,a,P}\sum_{u=h}^{P-1}R_u+\mathbb{1}\{P<H+1\}q^\pi(S_P,\pi^+_{\hat{G}\bar{\theta}}(S_P)),$$

and therefore

$$\left\langle\varphi(s,a),\bar{\theta}_h\right\rangle\approx_{(H-h+1)H\eta_0}q^\pi(s,a),$$

proving Eq. (144) of the inductive assumption for $t=h$.

To show Eq. (143) for $t=h$, by Eq. (144) for $t=h$ and the inductive assumption for $t>h$,

$$\left\langle\varphi(s,a),\bar{\theta}_h\right\rangle\approx_{H^2\eta_0}q^\pi(s,a)\geq q^\star(s,a)-(H-h)(\varepsilon/H+2H^2\eta_0).$$

Either $\pi$ chooses the action $a'$ maximizing the inner product above, for which

$$q^\pi(s,a')\geq\max_{a\in[\mathcal{A}]}q^\star(s,a)-(H-h)(\varepsilon/H+2H^2\eta_0)-2H^2\eta_0\geq v^\star(s)-(H-h+1)(\varepsilon/H+2H^2\eta_0),$$

or it chooses action 1. This can only happen with non-zero probability if $\tau_{\hat{G}\bar{\theta}}(s)<1$, in which case we have by definition that $\operatorname{range}^{\hat{G}}_{\mathcal{Q}}(s)=\operatorname{range}_{\mathcal{Q}}(s)\leq\frac{\varepsilon}{\sqrt{2dH}}$. Combining with Eq. (90) and

Proposition 5.4.5, range$(s) \le \frac{\varepsilon}{H}$, and therefore, using Eq. (143) for $t = h+1$, in this case

$$\begin{aligned} q^{\pi}(s,1) &\ge q^{\star}(s,1) - (H-h)(\varepsilon/H + 2H^2\eta_0) \\ &\ge v^{\star}(s) - \frac{\varepsilon}{H} - 2\eta - (H-h)(\varepsilon/H + 2H^2\eta_0) \ge v^{\star}(s) - (H-h+1)(\varepsilon/H + 2H^2\eta_0)\,. \end{aligned}$$

Therefore for any choice of action $a'$ of policy $\pi$ in state $s$, $q^{\pi}(s,a') \ge v^{\star}(s) - (H-h+1)(\varepsilon/H + 2H^2\eta_0)$. Therefore

$$v^{\pi}(s) \ge v^{\star}(s) - (H-h+1)(\varepsilon/H + 2H^2\eta_0)\,,$$

finishing the induction.

We thus conclude that

$$v^{\pi}(s_1) \ge v^{\star}(s_1) - \varepsilon - 2H^3\eta_0\,.$$

Combined with Eq. (144) of the inductive assumption, the value of Optimization Problem 5.4.10 can be bounded as

$$C_{\hat{G}\bar{\theta}}(s_1) = \mathrm{clip}_{[0,H]}\left\langle \varphi(s_1, \pi(s_1)), \bar{\theta}_1 \right\rangle \ge H^2\eta_0 + v^{\pi}(s_1) \ge v^{\star}(s_1) - 2\varepsilon\,,$$

by assumption on $\eta$ being relatively small (Eq. (108)). ■

## 5.L. Deferred lemmas

**Lemma 5.L.1** (Elliptical potential, Lemma 19.4 from Lattimore and Szepesvári (2020))**.** *Let $V_0 \in \mathbb{R}^{d\times d}$ be positive definite and $a_1 \ldots, a_n \in \mathbb{R}^d$ be a sequence of vectors with $\|a_t\|_2 \le L < \infty$ for all $t \in [n]$, $V_t = V_0 + \sum_{s\le t} a_s a_s^{\top}$. Then,*

$$\sum_{t=1}^{n} \min\left\{1, \|a_t\|^2_{V_{t-1}^{-1}}\right\} \le 2\log\left(\frac{\det V_n}{\det V_0}\right) \le 2d\log\left(\frac{\mathrm{Tr}\, V_0 + nL^2}{d\det(V_0)^{1/d}}\right).$$

**Lemma 5.L.2.** *Let $V \in \mathbb{R}^{d\times d}$ be a symmetric positive definite matrix and $(a_i)_{i\in[n]}$ be a sequence of $n$ $d$-dimensional real vectors. Let $V_i = V + \sum_{j\in[i]} a_j a_j^{\top}$. Then,*

$$\sum_{i\in[n]} \|a_i\|^2_{V_i^{-1}} \ge \min\left\{1, \frac{1}{2}\sum_{i\in[n]} \|a_i\|^2_{V^{-1}}\right\}$$

*Proof.* If $\sum_{i\in[n]} a_i a_i^\top \preceq V$, then $V_i \preceq 2V$, and therefore

$$\sum_{i\in[n]} \|a_i\|^2_{V_i^{-1}} \geq \sum_{i\in[n]} \|a_i\|^2_{2V^{-1}} = \frac{1}{2}\|a_i\|_{V^{-1}} .$$

Otherwise, $\sum_{i\in[n]} a_i a_i^\top V^{-1}$ has an eigenvalue that is at least 1. As all the other eigenvalues are non-negative (as $V$ is symmetric positive definite), we have that

$$\sum_{i\in[n]} \|a_i\|^2_{V^{-1}} = \mathrm{Tr}\left(\sum_{i\in[n]} a_i a_i^\top V^{-1}\right) \geq 1 . \qquad \blacksquare$$

## 5.M. Estimation error blow-up guarantees

We borrow Assumption 5.M.1 and Theorem 5.M.2 from Lattimore and Szepesvári (2020) and refer the reader to the book for the corresponding proof.

**Assumption 5.M.1** (Prerequisites for Theorem 5.M.2)**.** *Let $\lambda > 0$. For $k \in \mathbb{N}^+$, let $A_k$ be random variables taking values in $\mathbb{R}^d$. For some $\theta_\star \in \mathbb{R}^d$, let $X_k = \langle A_k, \theta_\star\rangle + \eta_k$ for all $k \in \mathbb{N}^+$. Here, $\eta_k$ is a conditionally 1-subgaussian random variable ("noise"), ie. it satisfies:*

$$\text{for all } \alpha \in \mathbb{R} \text{ and } t \geq 1, \qquad \mathbb{E}[\exp(\alpha\eta_k) \,|\, \mathcal{F}_{k-1}] \leq \exp\left(\frac{\alpha^2}{2}\right) \quad a.s.,$$

*where $\mathcal{F}_{k-1}$ is such that $A_1, X_1, \ldots, A_{k-1}, X_{k-1}, A_k$ are $\mathcal{F}_{k-1}$-measurable.*

**Theorem 5.M.2** (Lattimore and Szepesvári (2020), Theorem 20.5)**.** *Let $\zeta \in (0,1)$. Under Assumption 5.M.1, with probability at least $1-\zeta$, it holds that for all $k \in \mathbb{N}$,*

$$\left\|\hat{\theta}_k - \theta_\star\right\|_{V_k(\lambda)} < \sqrt{\lambda}\,\|\theta_\star\|_2 + \sqrt{2\log\left(\frac{1}{\zeta}\right) + \log\left(\frac{\det V_k(\lambda)}{\lambda^d}\right)},$$

*where for $k \in \mathbb{N}$,*

$$V_k(\lambda) = \lambda I + \sum_{s=1}^{k} A_s A_s^\top$$
$$\hat{\theta}_k = V_k(\lambda)^{-1} \sum_{s=1}^{k} X_s A_s$$

We generalize this theorem to handle non-zero-mean noise with parametrized subgaussianity. To handle non-zero-mean noise, we use (Zanette et al., 2020b, Lemma 8). We state the lemma here and refer the reader to Zanette et al. (2020b) for the proof:

**Lemma 5.M.3** (Zanette et al. (2020b), Lemma 8)**.** *For $n \in N^+$, let $\{A_i\}_{i=1,\dots,n}$ be any sequence of vectors in $\mathbb{R}^d$ and $\{\Delta_i\}_{i=1,\dots,n}$ be any sequence of scalars such that $|\Delta_i| \leq \xi \in \mathbb{R}$ with $\xi \geq 0$. For any $\lambda \geq 0$ and $V(\lambda) = \sum_{i=1}^n A_i A_i^\top + \lambda I$ we have:*

$$\left\| \sum_{i=1}^n A_i \Delta_i \right\|^2_{V(\lambda)^{-1}} \leq n\xi^2$$

**Lemma 5.M.4.** *Let $\zeta \in (0,1)$, $\lambda > 0$, $\sigma > 0$, and $\xi \geq 0$. For $k \in \mathbb{N}^+$, let $A_k$ be random variables taking values in $\mathbb{R}^d$. For some $\theta_\star \in \mathbb{R}^d$, let $\tilde{X}_k = \langle A_k, \theta_\star \rangle + \eta_k$ for all $k \in \mathbb{N}^+$. Here, $\eta_k$ is a conditionally $\sigma$-subgaussian random variable, ie. it satisfies:*

$$\text{for all } \alpha \in \mathbb{R} \text{ and } t \geq 1, \qquad \mathbb{E}[\exp(\alpha \eta_k) \,|\, \mathcal{F}_{k-1}] \leq \exp\left(\frac{\alpha^2 \sigma^2}{2}\right) \quad a.s.,$$

*where $\mathcal{F}_{k-1}$ is such that $A_1, \tilde{X}_1, \dots, A_{k-1}, \tilde{X}_{k-1}, A_k$ are $\mathcal{F}_{k-1}$-measurable. With probability at least $1-\zeta$, it holds that for any sequence $\{\Delta_i\}_{i=1,\dots}$ such that $|\Delta_i| \leq \xi$, for all $k \in \mathbb{N}$,*

$$\left\| \hat{\theta}_k - \theta_\star \right\|_{V_k(\lambda)} < \sqrt{\lambda}\, \|\theta_\star\|_2 + \xi\sqrt{k} + \sigma \sqrt{2 \log\left(\frac{1}{\zeta}\right) + \log\left(\frac{\det V_k(\lambda)}{\lambda^d}\right)}.$$

*where for $k \in \mathbb{N}$,*

$$X_k = \tilde{X}_k + \Delta_k$$
$$V_k(\lambda) = \lambda I + \sum_{s=1}^k A_s A_s^\top$$
$$\hat{\theta}_k = V_k(\lambda)^{-1} \sum_{s=1}^k X_s A_s$$

*Proof.* Let $X'_k = (X_k - \Delta_k)/\sigma_k$, $A'_k = A_k/\sigma_k$, $\lambda' = \lambda/\sigma_k^2$, and $\theta'_\star = \theta_\star$, $V'_k(\lambda') = \lambda' I + \sum_{s=1}^k A'_s A'^\top_s$, and $\hat{\theta}'_k = V'_k(\lambda')^{-1} \sum_{s=1}^k X'_s A'_s$. By assumption, $X'_k$, $A'_k$, $\lambda'$ and $\theta'_\star$ then satisfy Assumption 5.M.1. Therefore by applying Theorem 5.M.2, with probability at least $1-\zeta$, it holds that for all $k \in \mathbb{N}$,

$$\left\| \hat{\theta}'_k - \theta_\star \right\|_{V'_k(\lambda')} < \sqrt{\lambda'}\, \|\theta_\star\|_2 + \sqrt{2 \log\left(\frac{1}{\zeta}\right) + \log\left(\frac{\det V'_k(\lambda')}{\lambda'^d}\right)}.$$

Under this high-probability event, since $V'_k(\lambda') = V_k(\lambda)/\sigma^2$, substituting into the previous display yields

$$\left\|\hat{\theta}'_k - \theta_\star\right\|_{V_k(\lambda)} < \sqrt{\lambda}\,\|\theta_\star\|_2 + \sigma\sqrt{2\log\left(\frac{1}{\zeta}\right) + \log\left(\frac{\det V_k(\lambda)}{\lambda^d}\right)}. \tag{145}$$

Take any sequence $\{\Delta_i\}_{i=1,\dots}$ such that $|\Delta_i| \le \xi$ and apply the triangle inequality:

$$\left\|\hat{\theta}_k - \theta_\star\right\|_{V_k(\lambda)} \le \left\|\hat{\theta}'_k - \theta_\star\right\|_{V_k(\lambda)} + \left\|\hat{\theta}'_k - \hat{\theta}_k\right\|_{V_k(\lambda)}, \tag{146}$$

so it remains to bound $\left\|\hat{\theta}'_k - \hat{\theta}_k\right\|_{V_k(\lambda)}$.

$$\begin{aligned}
\left\|\hat{\theta}'_k - \hat{\theta}_k\right\|_{V_k(\lambda)} &= \left\| V'_k(\lambda')^{-1} \sum_{s=1}^{k} X'_s A'_s - V_k(\lambda)^{-1} \sum_{s=1}^{k} X_s A_s \right\|_{V_k(\lambda)} \\
&= \left\| V_k(\lambda)^{-1} \sum_{s=1}^{k} (X_s - \Delta_s) A_s - V_k(\lambda)^{-1} \sum_{s=1}^{k} X_s A_s \right\|_{V_k(\lambda)} \\
&= \left\| V_k(\lambda)^{-1} \sum_{s=1}^{k} \Delta_s A_s \right\|_{V_k(\lambda)} = \left\| \sum_{s=1}^{k} \Delta_s A_s \right\|_{V_k(\lambda)^{-1}} \\
&\le \sqrt{k}\xi,
\end{aligned} \tag{147}$$

where the final inequality uses Lemma 5.M.3. The proof is finished by plugging in the bounds of Eqs. (145) and (147) into the triangle inequality of Eq. (146). ■

## Chapter 6

# Summary

In this thesis we presented contributions to the field of reinforcement learning with linear function approximation. Most notable are the query complexity results under optimal (action-)value function approximation for various access settings, and results under $q^\pi$-realizability (all-policy realizability) for both local access planning and online RL. For $q^\star$ and $v^\star$-realizability, this includes the lower bounds of Theorems 1.4.1 and 1.4.2 (Theorem 9 of Weisz et al., 2021b, and Theorem 1.1 of Weisz et al., 2022b) that hold in the most permissive global access setting, and the algorithm and corresponding upper bound of Theorem 1.4.4 (Theorem 1.2 of Weisz et al., 2022b), that holds in the less permissive local access setting. For $q^\pi$-realizability, we present an algorithm and corresponding upper bound of Theorem 1.5.1 (consequence of Theorem 1.2 and Theorem 1.3 of Weisz et al., 2022a) for local access planning, and another algorithm and corresponding upper bound of Theorem 1.5.2 (consequence Theorem 4.1 of Weisz et al., 2023) for online RL. These results are summarized in Sections 1.4 and 1.5. Along the way, we present various open problems and promising directions for future work, many of which are summarized in Chapter 1.

# Bibliography


Alekh Agarwal, Nan Jiang, Sham M Kakade, and Wen Sun. Reinforcement learning: Theory and algorithms. *CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep*, pages 10–4, 2019.

Alex Ayoub, Zeyu Jia, Csaba Szepesvari, Mengdi Wang, and Lin Yang. Model-based reinforcement learning with value-targeted regression. In *International Conference on Machine Learning*, pages 463–474. PMLR, 2020.

R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, New Jersey, 1957.

R. Bellman, R. Kalaba, and B. Kotkin. Polynomial approximation – a new computational technique in dynamic programming: Allocation processes. *Mathematics of Computation*, 17(8):155–161, 1963.

Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control: Approximate dynamic programming*, volume II. 4 edition, 2012.

Dimitri P. Bertsekas and John N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

C. S. Chow and J. N. Tsitsiklis. The complexity of dynamic programming. *Journal of Complexity*, 5:466–488, 1989.

Christoph Dann, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. On oracle-efficient PAC RL with rich observations. *arXiv preprint arXiv:1803.00606*, 2018.

Holger Dell, Thore Husfeldt, Dániel Marx, Nina Taslaman, and Martin Wahlén. Exponential time complexity of the permanent and the tutte polynomial. *ACM Transactions on Algorithms (TALG)*, 10(4):1–32, 2014.

Simon S Du, Sham M Kakade, Ruosong Wang, and Lin F Yang. Is a good representation sufficient for sample efficient reinforcement learning? In *International Conference on Learning Representations*, 2019a.

Simon S Du, Yuping Luo, Ruosong Wang, and Hanrui Zhang. Provably efficient $Q$-learning with function approximation via distribution shift error checking oracle. In *Advances in Neural Information Processing Systems*, pages 8060–8070, 2019b.

Simon S Du, Sham M Kakade, Jason D Lee, Shachar Lovett, Gaurav Mahajan, Wen Sun, and Ruosong Wang. Bilinear classes: A structural framework for provable generalization in RL. In *ICML*, volume 139, pages 2826–2836, 2021.

Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. Contextual decision processes with low Bellman rank are PAC-learnable. In *International Conference on Machine Learning*, pages 1704–1713. PMLR, 2017.

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143, 2020a.

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143. PMLR, 2020b.

Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline rl? In *International Conference on Machine Learning*, pages 5084–5096. PMLR, 2021.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *In Proc. 19th International Conference on Machine Learning*. Citeseer, 2002.

Lodewijk Kallenberg. Finite state and action mdps. In *Handbook of Markov decision processes: methods and applications*, pages 21–87. Springer, 2002.

Michael Kearns, Yishay Mansour, and Andrew Y Ng. A sparse sampling algorithm for near-optimal planning in large Markov decision processes. *Machine Learning*, 49:193–208, 2002.

Chandrashekar Lakshminarayanan, Shalabh Bhatnagar, and Csaba Szepesvári. A linearly relaxed approximate linear program for Markov decision processes. *IEEE Transactions on Automatic control*, 63(4):1185–1191, 2017.

T. Lattimore and Cs. Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

Tor Lattimore, Csaba Szepesvári, and Gellért Weisz. Learning with good feature representations in bandits and in RL with a generative model. In *ICML*, pages 9464–9472, 2020.

Gen Li, Yuxin Chen, Yuejie Chi, Yuantao Gu, and Yuting Wei. Sample-efficient reinforcement learning is feasible for linearly realizable MDPs with limited revisiting. *arXiv preprint arXiv:2105.08024*, 2021.

Michael L. Littman, Thomas L. Dean, and Leslie Pack Kaelbling. On the complexity of solving markov decision problems. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, pages 394–402, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1558603859.

Sihan Liu, Gaurav Mahajan, Daniel Kane, Shachar Lovett, Gellért Weisz, and Csaba Szepesvári. Exponential hardness of reinforcement learning with linear function approximation. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 1588–1617. PMLR, 2023.

Mausam and Andrey Kolobov. Planning with Markov decision processes: An ai perspective. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–210, 2012.

A Woodbury Max. Inverting modified matrices. In *Memorandum Rept. 42, Statistical Research Group*, page 4. Princeton Univ., 1950.

Sean Meyn. *Control systems and reinforcement learning*. Cambridge University Press, 2022.

Aditya Modi, Nan Jiang, Ambuj Tewari, and Satinder Singh. Sample complexity of reinforcement learning using linearly combined model ensembles. In *International Conference on Artificial Intelligence and Statistics*, pages 2010–2020. PMLR, 2020.

Remi Munos. Error bounds for approximate policy iteration. In *ICML*, pages 560–567, 2003.

Remi Munos. Error bounds for approximate value iteration. In *AAAI*, pages 1006–1011, 2005.

Rémi Munos. From bandits to Monte-Carlo tree search: The optimistic principle applied to optimization and planning. *Foundations and Trends® in Machine Learning*, 7(1):1–129, 2014.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley-Interscience, 1994.

Walter Rudin et al. *Principles of mathematical analysis*, volume 3. McGraw-hill New York, 1976.

Daniel Russo. Approximation benefits of policy gradient methods with aggregated states. *arXiv preprint arXiv:2007.11684*, 2020.

Daniel Russo and Benjamin Van Roy. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243, 2014.

John Rust. Using randomization to break the curse of dimensionality. *Econometrica: Journal of the Econometric Society*, pages 487–516, 1997.

Bruno Scherrer. Approximate policy iteration schemes: a comparison. In *International Conference on Machine Learning*, pages 1314–1322. PMLR, 2014.

Bruno Scherrer. Improved and generalized upper bounds on the complexity of policy iteration. *Mathematics of Operations Research*, 41(3):758–774, August 2016.

Bruno Scherrer and Boris Lesner. On the use of non-stationary policies for stationary infinite-horizon Markov decision processes. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25, 2012.

Paul J. Schweitzer and Abraham Seidmann. Generalized polynomial approximations in Markovian decision processes. *Journal of Mathematical Analysis and Applications*, 110(2):568–582, September 1985.

Roshan Shariff and Csaba Szepesvári. Efficient planning in large MDPs with weak linear function approximation. *arXiv preprint arXiv:2007.06184*, 2020.

Aaron Sidford, Mengdi Wang, Xian Wu, and Yinyu Ye. Variance reduced value iteration and faster algorithms for solving markov decision processes. *Naval Research Logistics (NRL)*, 70(5): 423–442, 2023.

Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44, 1988.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Cs. Szepesvári and R. Munos. Finite time bounds for sampling based fitted value iteration. In *ICML*, pages 881–886, 2005.

Michael J Todd. *Minimum-volume ellipsoids: Theory and algorithms*. SIAM, 2016.

John N Tsitsiklis and Benjamin Van Roy. Feature-based methods for large scale dynamic programming. *Machine Learning*, 22(1):59–94, 1996.

Benjamin Van Roy and Shi Dong. Comments on the Du-Kakade-Wang-Yang lower bounds. *arXiv preprint arXiv:1911.07910*, 2019.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Andrew Wagenmaker, Yifang Chen, Max Simchowitz, Simon S Du, and Kevin Jamieson. Reward-free RL is no harder than reward-aware RL in linear Markov decision processes. *arXiv preprint arXiv:2201.11206*, 2022.

Ruosong Wang, Dean P. Foster, and Sham M. Kakade. What are the statistical limits of offline RL with linear function approximation?, 2020a.

Ruosong Wang, Ruslan Salakhutdinov, and Lin F Yang. Provably efficient reinforcement learning with general value function approximation. *arXiv preprint arXiv:2005.10804*, 2020b.

Yining Wang, Ruosong Wang, Simon S Du, and Akshay Krishnamurthy. Optimism in reinforcement learning with generalized linear function approximation. *arXiv preprint arXiv:1912.04136*, 2019.

Yuanhao Wang, Ruosong Wang, and Sham Kakade. An exponential lower bound for linearly realizable MDP with constant suboptimality gap. *Advances in Neural Information Processing Systems*, 34, 2021.

Gellért Weisz, Philip Amortila, Barnabás Janzer, Yasin Abbasi-Yadkori, Nan Jiang, and Csaba Szepesvári. On query-efficient planning in MDPs under linear realizability of the optimal state-value function. In *COLT*, volume 134 of *Proceedings of Machine Learning Research*, pages 4355–4385, 2021a.

Gellért Weisz, Philip Amortila, and Csaba Szepesvári. Exponential lower bounds for planning in MDPs with linearly-realizable optimal action-value functions. In *ALT*, volume 132 of *Proceedings of Machine Learning Research*, pages 1237–1264, 2021b.

Gellért Weisz, András György, Tadashi Kozuno, and Csaba Szepesvari. Confident approximate policy iteration for efficient local planning in $q^\pi$-realizable MDPs. In *Advances in Neural Information Processing Systems*, 2022a.

Gellért Weisz, Csaba Szepesvári, and András György. Tensorplan and the few actions lower bound for planning in MDPs under linear realizability of optimal value functions. In *International Conference on Algorithmic Learning Theory*, pages 1097–1137. PMLR, 2022b.

Gellért Weisz, András György, Tadashi Kozuno, and Csaba Szepesvari. Online rl in linearly $q^\pi$-realizable mdps is as easy as in linear mdps if you learn what to ignore. *Advances in Neural Information Processing Systems*, 36, 2023.

Zheng Wen and Benjamin Van Roy. Efficient exploration and value function generalization in deterministic systems. In *Advances in Neural Information Processing Systems*, pages 3021–3029, 2013.

Lin F Yang and Mengdi Wang. Sample-optimal parametric $q$-learning using linearly additive features. *arXiv preprint arXiv:1902.04779*, 2019.

Yinyu Ye. The simplex and policy-iteration methods are strongly polynomial for the Markov decision problem with a fixed discount rate. *Mathematics of Operations Research*, 36(4):593–603, 2011.

Dong Yin, Botao Hao, Yasin Abbasi-Yadkori, Nevena Lazić, and Csaba Szepesvári. Efficient local planning with linear function approximation. In *International Conference on Algorithmic Learning Theory*, pages 1165–1192. PMLR, 2022.

Andrea Zanette, Alessandro Lazaric, Mykel J Kochenderfer, and Emma Brunskill. Limiting extrapolation in linear approximate value iteration. In *Advances in Neural Information Processing Systems*, pages 5615–5624, 2019.

Andrea Zanette, Alessandro Lazaric, Mykel Kochenderfer, and Emma Brunskill. Learning near optimal policies with low inherent Bellman error. *arXiv:2003.00153 [cs]*, Mar 2020a. arXiv: 2003.00153.

Andrea Zanette, Alessandro Lazaric, Mykel Kochenderfer, and Emma Brunskill. Learning near optimal policies with low inherent Bellman error. In *International Conference on Machine Learning*, pages 10978–10989. PMLR, 2020b.