Memory as State Abstraction over Trajectories

Aaron Kirtland[*]
Brown University
aaron_kirtland@brown.edu

Alexander Ivanov[*]
Brown University
alexander_ivanov@brown.edu

Cameron Allen
University of California, Berkeley
camallen@berkeley.edu

Michael Littman
Brown University
mlittman@cs.brown.edu

George Konidaris
Brown University
gdk@cs.brown.edu

Abstract

Reinforcement learning is provably difficult in non-Markovian environments, which motivates identifying tractable environment subclasses. We propose a systematic structuring of POMDP subclasses that naturally arise from considering agents with memory, which we view as a temporally-extended abstraction over the agent's observation-action history. We proceed by classifying memory functions as "optimal", "improving", or "neither" with respect to the same targets as in state abstractions, namely model, optimal state-action values Q^*, and optimal policy π^*, with each type of abstraction contained by the previous one. Additionally, we extend traditional state abstraction to "soft" (stochastic) abstractions and show that the abstraction hierarchy also holds for stochastic memory functions. Then, we define classes of POMDPs by whether they admit a specific kind of memory function. Concretely, we classify POMDPs in terms of memory functions with the following attributes: size (number of memory states), stochasticity (deterministic, stochastic), target (model, Q^*, π^*), and quality (improving, optimal, neither). We prove that non-trivial relationships between these POMDP classes do not exist, with two notable exceptions: 1) with an unbounded memory capacity, deterministic memory can approach the expected return of finite-size stochastic memory, when rewards are bounded; 2) with a finite memory capacity, there exist POMDPs where stochastic memory is strictly more powerful than deterministic memory. Lastly, we show how these classes both systematize several previously considered types of POMDPs and, using approximate abstractions, generalize them to approximate variants.

1 Introduction

Much of reinforcement learning makes the Markov assumption, which is unrealistic and restrictive for many applications. Hence, it is useful to generalize MDPs to partially observable POMDPs. The class of POMDPs is too broad, however; worst-case performance is provably difficult. Previous literature has thus studied tractable subclasses of POMDPs that admit practical algorithms (see Section 5).

At the same time, recent memory-based approaches with recurrent neural networks (RNNs) and learned memory controllers have met success. Yet, there has been no systematic exploration of the space of POMDPs with respect to memory specifically.

We propose to structure the class of POMDPs by identifying structure in the memory functions which they support. Memory functions map the agent's history to a memory output; they are therefore

[*]These authors contributed equally.

Preprint. Under review.

temporally-extended observation abstractions. This observation allows us to apply frameworks for reasoning about state abstractions. In particular, we are interested in memory functions that preserve or improve the agent's ability to represent a Markov model, optimal value estimation, and optimal policy. We also consider stochastic memory functions, which require extending the state abstraction literature.

For example, consider Bakker's T-maze in Figure 1 [Bakker, 2001]. The agent begins at the left end of one of two mazes (upper/lower diagonal), with rewards of +1 or -1 at terminal states as clued by the initial observation. The 2-state deterministic memory depicted in Figure 2 is a π^*-optimal memory function for this environment because it perfectly recalls the initial state color, and knowledge of the initial color and current observation is all that is necessary for the optimal policy. However, it is not a model-optimal memory function because the agent cannot correctly predict which hallway state they are in using only the initial observation.

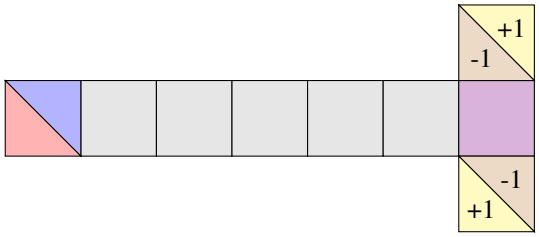

Figure 1: Bakker's T-maze environment

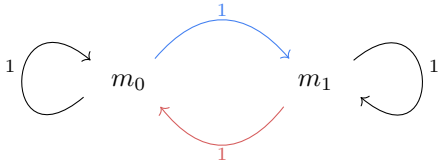

Figure 2: A π^*-optimal memory function for the T-maze.

Concretely, we use memory functions to augment the agent's observations and allow it to recall features of the past to improve certain targets such as the optimal expected return, value estimation error, and model error. In Section 2, we give a background for POMDPs, memory functions, and state abstractions. Our contributions begin in Section 3, where we show how memory functions induce state abstractions over a trajectory MDP defined over any POMDP, and so we can therefore consider memory functions that preserve the traditional state abstraction targets: π^*, Q^*, or model. Given any of these targets, we define three collections of memory functions: those that achieve the target optimally, "optimal"; those that improve on the target compared to having no memory whatsoever, "improving"; or those that do nothing with respect to the target, "nonimproving". Each of these targets defines classes of POMDPs based on whether such a memory function exists, and in Section 4, we prove which relationships hold between these POMDP classes. Lastly, in Section 5, we show how these classes of POMDPs systematize previously considered types of POMDPs and, using approximate abstractions, generalize them to approximate classes.

2 Background

POMDPs: We formalize the agent's decision problem as a partially observable Markov decision process, a generalization of Markov decision processes where the agent cannot observe the state directly. A POMDP is defined as a 7-tuple $(S, A, P, R, \Omega, \Phi, \gamma)$ where S is the set of states, A is the set of actions, $P : S \times A \to \Delta S$ is the transition function, $R : S \times A \to \mathbb{R}$ is the reward function, Ω is the set of observations, $\Phi : S \times A \to \Omega$ is the observation function, and γ is the discount factor. On each timestep, $\omega_t \sim \Phi(s_t)$, $a_t \sim \Phi(\omega_t)$, and $s_{t+1} \sim P(s_t, a_t)$.

Memory Functions: We define a k-state **memory function** μ as a k-state finite state machine (FSM) mapping a memory state m from the set M, an observation ω, and an action a, to a distribution over new memory states m'. Formally, $\mu : M \times \Omega \times A \to \Delta M$. m_0 is sampled from the initial state distribution, and on each timestep, $a_t \sim \pi(m_t, \omega_t)$ and $m_{t+1} \sim \mu(m_t, \omega_t, a_t)$. This flow is also described in Figure 3.

We use FSM-based memory in this work as it produces a good model of systems like RNNs. Furthermore, its action on the POMDP can be cleanly described as augmenting the state and observation space [Allen et al., 2024]. We allow the memory function μ to be stochastic as stochastic memory functions are provably more powerful for goals such as improving expected return (see Subsection 4.2).

State Abstraction: Given an MDP with states S, a state abstraction φ is a mapping $s \mapsto x$ for x in some abstract set of states X.[2] In this work, we consider three types of state abstractions, initially defined by [Li et al., 2006]: model, which preserves the one-step model, Q^*, which preserves the state-action value function for the optimal policy, and π^*, which preserves optimal actions. These state abstractions can be generalized to approximate forms and parameterized by values of ε [Abel et al., 2016, Jiang, 2018]. These are defined using the lifting operator, which for a function $f : \varphi(S) \to X$, is defined as $[f]_\varphi(s) \coloneqq f(\varphi(s))$. Additionally, these definitions require the introduction of two norms. The ∞-norm is used for scalar outputs and requires that the norm argument satisfies the epsilon constraint for all inputs, namely states s, and, if relevant, actions a. In other words, for $f : A \to B$, $\|f\|_\infty = \max_{a \in A} |f(a)|$. If the output is a distribution, we use a 1-norm over the output while taking a max over all inputs. In other words, if $g : A \to \Delta B$, $\|f\|_1 = \max_{a \in A} \|f(a)\|_1$ with $f(a)$ viewed as a vector. We list the definitions for exact and approximate state abstractions in Table 1.

Target	Exact State Abstractions	Approximate State Abstractions
model	$\forall s^{(1)}, s^{(2)}.\forall \bar{s}.\forall a.\varphi(s^{(1)}) = \varphi(s^{(2)}) \Rightarrow$ $R(s^{(1)}, a) = R(s^{(2)}, a)$ $\sum_{s' \in \varphi^{-1}(\bar{s})} P(s'\|s^{(1)}, a) = \sum_{s' \in \varphi^{-1}(\bar{s})} P(s'\|s^{(2)}, a)$	$\exists f_P : \varphi(T) \times A \to \Delta\Omega$ $\|[f_P]_\varphi - P_o\|_1 < \varepsilon_P$ $\exists f_R : \varphi(T) \times A \to \mathbb{R}$ $\|[f_R]_\varphi - R\|_\infty < \varepsilon_R$
Q^*	$\forall s, s'.\forall a.\varphi(s) = \varphi(s') \Rightarrow$ $Q^*(s, a) = Q^*(s', a)$	$\exists f : \varphi(S) \times A \to \mathbb{R}$ $\|[f]_\varphi - Q^*_M\|_\infty \leq \varepsilon_{Q^*}$
π^*	$\forall s, s'.\exists a^*.\varphi(s) = \varphi(s') \Rightarrow$ $Q^*(s, a^*) = \max_a Q^*(s, a)$ $\max_a Q^*(s', a) = Q^*(s', a^*)$	$\exists \pi : \varphi(S) \to \Delta A$ $\left\|V_M^{[\pi]_\varphi} - V_M^*\right\|_\infty \leq \varepsilon_{\pi^*}$

Table 1: Exact and Approximate State Abstractions

The types of abstractions form a hierarchy, as shown in Theorem 2.1. A model-preserving abstraction is necessarily a Q^*-preserving abstraction, and a Q^*-preserving abstraction is necessarily a π^*-preserving abstraction.

Theorem 2.1 (Abstraction hierarchy)**.** *Let (S, A, P, R, γ) be an MDP.*

1. *An $(\varepsilon_P, \varepsilon_R)$-approximate* model*-preserving abstraction is also a Q^*-preserving abstraction with $\varepsilon_{Q^*} = \frac{\varepsilon_R}{1-\gamma} + \frac{\gamma \varepsilon_P R_{max}}{2(1-\gamma)^2}$.*
2. *A Q^*-preserving abstraction with ε_{Q^*} is also a π^*-preserving abstraction with $\varepsilon_{\pi^*} = 2\varepsilon_{Q^*}/(1-\gamma)$.*

3 Memory functions and state abstraction

3.1 Trajectory MDP and Abstraction Definitions

We would like to define classes of memory functions in terms of state abstractions. To do this, we need to have a base MDP on which to define the abstractions. Given a POMDP, we must construct an MDP from it. We can do this via the trajectory MDP, the MDP given by taking as states the agent's entire history of observations and actions. The observation-only construction was previously considered by Timmer and Riedmiller [2009] and Hong et al. [2023].

Definition 3.1 (Trajectory MDP)**.** Given a POMDP $(S, A, P, \gamma, \Omega, \Phi, R)$ with initial state distribution s_0, we define the trajectory MDP to be (T, A, P', R', γ), where $T \coloneqq \{\tau \in (\Omega \times A)^* \times \Omega\}$ is the space of observation-action partial trajectories, $P : \tau \times a_t \mapsto \tau \oplus a_t \oplus \omega_{t+1}$ with $\omega_{t+1} \sim \mathbb{P}(\cdot|\tau, a_t)$ and $\oplus$ denoting concatenation, and $R'(\tau = (\omega_t, a_0, \ldots, \omega_t), a_t) \coloneqq \mathbb{E}_{s_t|\tau}[R(s_t, a_t)]$. This decision process is Markov by definition as a trajectory τ_t up to time t being a prefix of τ_{t+j} implies that $\mathbb{P}(\tau_{t+k}|\tau_t, \tau_{t-1}, \ldots) = \mathbb{P}(\tau_{t+k}|\tau_t)$.

[2] Some authors call general maps "aggregation" and use "abstraction" to refer specifically to maps that preserve model, Q^*, or π^*.

Next, in Table 2, we adapt the approximate state abstraction definitions (Table 1) for the trajectory setting (i.e., abstracting the trajectory MDP T). For now, we focus on the first column, "Optimal"; we will discuss the second column later. For Q^* and π^*, the resulting definition is the same, but for model, we make an adjustment so that a good abstraction represents Markov predictions with respect to the underlying MDP. We place an additional restriction that abstractions must preserve the present observation. In other words, we consider $\varphi : (\Omega \times A)^* \times \Omega \to M \times \Omega$; i.e. φ can be written as the product of a function $(\Omega \times A)^* \to M$ and the identity function on the observation space Ω. This resolves definitional issues that arise with considering more general memory functions and matches the intuition that the agent should always be able to perceive the present observation.

Additionally, applying these definitions to stochastic memory functions requires an extension from deterministic abstractions $\varphi : S \to X$ to soft abstractions $\varphi : S \to \Delta X$. To accommodate this, we extend the definition of lifting to $[f]_\varphi(s) \coloneqq \mathbb{E}_{x\sim\varphi(s)}\, f(x) = \sum_x \varphi(x|s) f(x)$. Likewise, we define the lift $[f]_\varphi$ of a function on the domain $\varphi(S) \times A$ to be $s, a \mapsto \mathbb{E}_{x\sim\varphi(s)}\, f(x, a)$. The definitions in Table 2 use this modification. Soft abstractions were previously considered by Singh et al. [1994] and Sorg and Singh [2009]. Stochastic memory functions require this extension because the abstraction map is defined given a memory function, and if the abstraction map were purely deterministic, then given some trajectory τ, the (m_t, ω_t) pair it corresponds to would be fixed. The definitions of Q^* and φ^* require only minimal changes to support this generalization. The changes are hidden in the notation here and are discussed in detail in Appendix C.

With well-defined abstractions over trajectories, we next define how memory functions induce abstractions. This is useful so that we can define "good" memory functions in terms of known types of abstractions. Recall that memory functions are $\mu : M \times A \times \Omega \to \Delta M$. Given a μ, an abstraction φ is well-defined when there exists a map φ such that the diagram in Figure 3 commutes. We can define such a φ as:

$$\varphi\big((\omega_0)\big) \coloneqq (m_0, \omega_0); \quad \varphi\big((\dots, \omega_t, a_t, \omega_{t+1})\big) \coloneqq \big(\mu(\mu(\dots \mu(m_0, a_0, \omega_1), \dots), a_t, \omega_{t+1}), \omega_{t+1}\big)$$

i.e., defining the abstraction to follow single steps forward in the memory function.

Type	Optimal	Improving
model	$\exists f_P : \varphi(T) \times A \to \Delta\Omega. \lVert[f_P]_\varphi - P_o\rVert_1 \le \varepsilon_P$	$\forall f_P : \varphi(T) \times A \to \Delta\Omega. \left\lVert[f_P]_\varphi - \hat{P}_o\right\rVert_1 > \varepsilon_P$
	$\exists f_R : \varphi(T) \times A \to \mathbb{R}. \lVert[f_R]_\varphi - R\rVert_\infty \le \varepsilon_R$	$\forall f_R : \varphi(T) \times A \to \mathbb{R}. \left\lVert[f_R]_\varphi - \hat{R}\right\rVert_\infty > \varepsilon_R$
Q^*	$\exists f : \varphi(T) \times A \to \mathbb{R}. \lVert[f]_\varphi - Q^*_M\rVert_\infty \le \varepsilon_{Q^*}$	$\forall f : \varphi(T) \times A \to \mathbb{R}. \left\lVert[f]_\varphi - Q^*_{\hat{M}}\right\rVert_\infty > \varepsilon_{Q^*}$
π^*	$\exists \pi : \varphi(T) \to \Delta A. \left\lVert V_M^{[\pi]_\varphi} - V_M^*\right\rVert_\infty \le \varepsilon_{\pi^*}$	$\forall \pi : \varphi(T) \to \Delta A. \left\lVert V_{\hat{M}}^{[\pi]_\varphi} - V_{\hat{M}}^*\right\rVert_\infty > \varepsilon_{\pi^*}$

Table 2: Left: State abstractions of optimal targets for memory functions. Here, $\varphi : T \to \Delta M \times \Delta\Omega$, P_o is the distribution over next observations $\mathbb{P}(\omega_{t+1}|\varphi(\tau_t), a_t)$, and R is as defined in Definition 3.1. Right: The definitions of ε-improving memory functions, where $\hat{M}$ is the effective MDP with components $\hat{P}$ and $\hat{R}$. Here, $\hat{P}_o$ maps $(\varphi(\tau_t), a_t)$ to the distribution $\hat{P}(\omega_{t+1}|\omega_t, a_t)$.

3.2 Optimal and improving memory functions

Now that we know how memory functions induce abstractions, we want to quantify how good memory functions are using these abstractions. Figure 4 shows the general relationships between abstractions and memory functions with respect to some target metric, but there are two useful kinds of memory to draw attention to: **improving** memory functions that improve over having no memory at all, and **optimal** memory functions improve. A memory function is ε-optimal if the abstraction it induces has targets ε-close to optimal. A memory function is ε-improving if it is *not* ε-nonimproving, where a memory function is ε-nonimproving if the abstraction it induces has targets ε-close to the so-called effective MDP, which models a memoryless agent [Allen et al., 2024] (See the right column of Table 2 for the formal definition). Just as the trajectory MDP plays a role in measuring optimality, the effective MDP plays a parallel role in measuring a lack of improvement.

[2] In Jiang [2018], $\pi : \varphi(S) \to A$ was assumed to be deterministic, but the results extend to stochastic π as well.

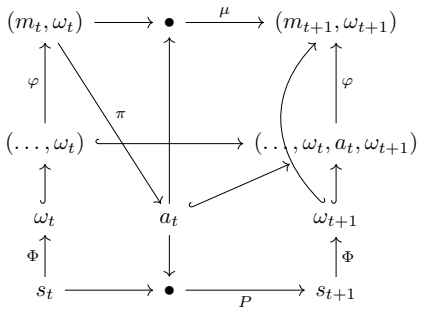

Figure 3: The relationship between memory functions and abstractions (see text). Hooks denote inclusion into a tuple, and reward is omitted for clarity.

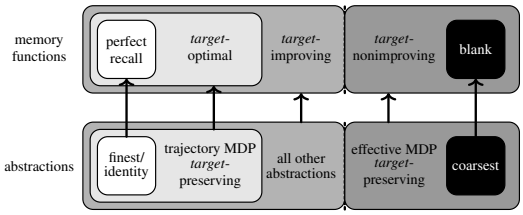

Figure 4: The relationships between types of memory functions and types of trajectory abstractions. We elaborate on the types of abstractions in Appendix D.

Formally, $\hat{P}(\omega'|\omega, a) := \sum_{s,s' \in S} \Phi(\omega'|s') P(s'|s, a)\, \mathbb{P}(s|\omega)$ and $\hat{R}(\omega, a) := \sum_{s \in S} R(s, a)\, \mathbb{P}(s|\omega)$, where $\mathbb{P}(s|\omega)$ is policy-dependent and describes how each hidden state $s_i \in \mathcal{S}$ contributes to the overall environment behavior when we see observation ω.

From defined classes of memory functions, we define classes of POMDPs, given four attributes: *stochasticity* (stochastic or deterministic), *quality* (optimal, improving, nonimproving), *target* (model-preserving, Q^*-preserving, or π^*-preserving), and *number* (k). Here, "improving" means 0-improving ($\varepsilon = 0$), and "nonimproving" means 0-nonimproving. This is the setting in which we present our later results.

Definition 3.2. We say that a POMDP is (stochastic/deterministic) k-memory *target*-(optimal/improvable/nonimprovable) if it admits a k-state (stochastic/deterministic) *target*-(optimal/improving/nonimproving) memory function.

4 Relationships between classes of memory functions

Next, let's consider what relationships exist *between* these classes. We want to know, in particular, when a POMDP admitting a memory function in one class implies it admits a memory function in another class. We will henceforth ignore the "nonimproving" memory functions as they are not useful, and every POMDP admits a trivial 1-state nonimproving blank memory function, and we will also focus on the cases of 2 or an arbitrary finite number k memory states. This still, however, leads to $(2 \cdot 2 \cdot 3 \cdot 2)^2 = 24^2$ possibilities! Conveniently, all of these cases can be grouped into three families that are straightforward to consider:

1. With *stochasticity*, *number*, and *quality* constant, we consider if $\textit{target}_1 \Rightarrow \textit{target}_2$ in Subsection 4.1.
2. With *target* and *quality* constant, we consider if $\textit{stochasticity}_1, \textit{number}_1 \Rightarrow \textit{stochasticity}_2, \textit{number}_2$ in Subsection 4.2.
3. With *stochasticity*, *number*, and *target* constant, we consider if $\textit{quality}_1 \Rightarrow \textit{quality}_2$ in Subsection 4.3.

We conjecture that these are the *only* families of cases that we must consider because for any other cases, the implication does not follow.

4.1 Target: state abstraction hierarchy

The first family of cases, $\textit{target}_1 \Rightarrow \textit{target}_2$ given that *stochasticity*, *number*, and *quality* are held constant, are covered by the abstraction hierarchy. Namely, an ε-model-optimal memory function implies the existence of a $\frac{\varepsilon_R}{1-\gamma} + \frac{\gamma \varepsilon_P R_{\max}}{2(1-\gamma)^2}$-$Q^*$-optimal memory function, and an ε-Q^*-optimal memory function implies the existence of a $2\varepsilon_{Q^*}/(1-\gamma)$-$\pi^*$-optimal memory function. Non-trivial relationships do not appear to hold for improving memory functions.

These two bounds were presented earlier in the background Section 2. Though the hierarchy is a well-known result proven by previous authors, our work requires slight extensions because we consider soft approximate abstractions. The Q^*-preservation implies π^*-preservation proof follows directly (see Appendix C for details), and for model-preservation implies Q^*-preservation (see Appendix C.1) we require a modified lemma because we use a different model error definition.

4.2 Stochasticity and number

Next, let's consider when admitting a deterministic/stochastic memory function of a certain size implies admitting a deterministic/stochastic memory function of another size. Through a set of proofs and counterexamples filling out tables of fixed *target* and *quality*, we prove that non-trivial relationships between these POMDP classes do not exist, with two notable exceptions:

Theorem 4.1. ***With an unbounded memory capacity, deterministic memory can approach the expected return of finite-size stochastic memory, when rewards are bounded****: Let μ_k^* be a k-state stochastic memory function. For any POMDP with bounded reward and all ε, there exists a k'-DFA which achieves an expected return that is only ε less than μ_k^*. Furthermore, it is sufficient to choose $k' \geq k \ln(\varepsilon(1-\gamma)/R_{max})/\ln(\gamma)$ where R_{max} is the bound on reward and γ is the discount factor. See Appendix G for the proof.*

Example 4.1. ***With a finite memory capacity, there exist POMDPs where stochastic memory is strictly more powerful than deterministic memory****: We show this via counterexample. Consider the POMDP depicted in Figure 5. The agent spawns in one of two corridors and observes a sequence of binary observations, with all actions in the corridor moving to the right. At the junction (red), the agent receives a positive reward if they choose the action "up", indicating they were in the top corridor, while they receive a positive reward in the bottom corridor if they choose "down".*

There exists no two-state DFA capable of distinguishing the strings 1000 and 0010, and thus the agent's memory will be identical at the junction regardless of which of the 16 two-state DFAs they have. This problem is, however, resolvable with 3 states of memory (such as with the automata that count the number of 0's mod 3 since the most recent 1) or with stochastic memory, which is shown in Figure 6. Given the policy that the agent goes up given m_0 and down given m_1, the agent will receive an expected reward of $11/16 = 0.6875$, which is higher than the 0.5 expected reward given by a deterministic memory or memoryless policy.

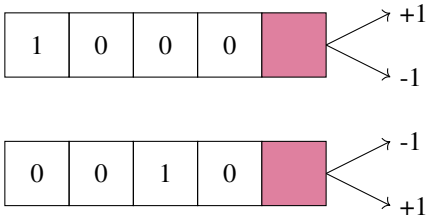

Figure 5: Two corridors, the sequence of observations of which cannot be recalled by any deterministic 2-state DFA.

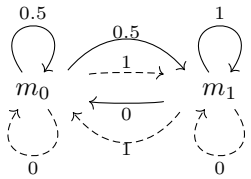

Figure 6: m_0 is the initial state, the solid line gives transitions upon observations of 0, and the dashed line gives transitions upon observations of 1.

These two results highlight important entries in the following tables, the first of which, Table 3, shows which implications hold for π^*. We give a reference after each implication result to a counterexample or proof in the appendix, and relationships that hold trivially by set inclusion are marked with [SI].

	Expected π^*-optimal				Expected π^*-improving			
if $\exists \downarrow$ then $\exists \rightarrow$	2 det	k' det	2 sto	k' sto	2 det	k' det	2 sto	k' sto
2 det	✓ [SI]	✓ [SI]	✓ [SI]	✓ [SI]	✓ [SI]	✓ [SI]	✓ [SI]	✓ [SI]
k det	× H.1	✓ [SI]	× H.1	✓ [SI]	× H.1	✓ [SI]	× H.1	✓ [SI]
2 sto	× E.1	*	✓ [SI]	✓ [SI]	× E.1	*	✓ [SI]	✓ [SI]
k sto	× H.1	*	× H.1	✓ [SI]	× H.1	*	× H.1	✓ [SI]

Table 3: POMDP class implications for the π^* target. Here, * denotes that the result depends on whether rewards are bounded, i.e., an $R_{\max}$ exists. If rewards are bounded, then the result holds by Theorem 4.1, while if they are unbounded, we have counterexample E.1.

Example 4.1 corresponds to the third row, first column entry, while Theorem 4.1 corresponds to the third row, second column entries. However, to minimize the number of counterexamples we need to cite in this table and present in the appendix, we leave Example 4.1 for intuition and instead reference other counterexamples that refute the same claims in the tables. The relationships given in the table above can be visualized in Figure 7, which continues to hold for later tables, albeit without the converse marked by $*$.

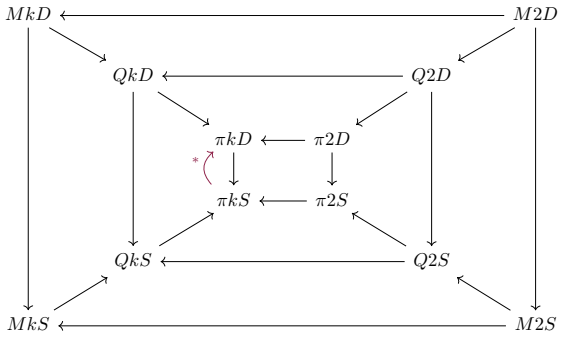

Figure 7: The general relationships between our defined POMDP classes. Here, "M" denotes model, "Q", Q^*, and π, π^*; 2 denotes 2-state and k denotes k-state (with each entry potentially a different k), and D denotes deterministic, while S denotes stochastic. The asterisk and purple arrow mark the result from Example 4.1, when a converse holds.

Here, we also note that Table 3 is for the expected case, meaning it is defined with an expectation over initial trajectories τ rather than an ∞-norm, i.e., max over all initial trajectories. With this expected case, π^*-optimality is equivalent to expected return preservation, and, indeed, this is the way we present the proofs/counterexamples. Similarly, expected Q^*-optimality is equivalent to the preservation of the expected value error of π^*. However, expected model-optimality is less significant than ∞-norm model-optimality, which is equivalent to the memory-augmented POMDP being Markov in transitions and rewards. Hence, it is apparent that both options are useful in some contexts; this is a choice we have to make in our results. There is a relationship between the two norms; any target ∞-norm preservation implies expected case preservation. Because the results involving the abstraction hierarchy are in terms of the ∞-norm, we use the ∞-norm to present our later tables for Q^* and π^*. See Appendix F for more details on the ∞-norm and expected options.

It turns out that Table 4 for Q^* and Table 5 for model are simple; their entries can be proven with just two distinct counterexamples in addition to set inclusion. We include them for completeness below.

	Q^*-optimal				Q^*-improving			
if $\exists \downarrow$ then $\exists \rightarrow$	2 det	k' det	2 sto	k' sto	2 det	k' det	2 sto	k' sto
2 det	✓ [SI]	✓ [SI]	✓ [SI]	✓ [SI]	✓ [SI]	✓ [SI]	✓ [SI]	✓ [SI]
k det	× E.3	✓ [SI]	× E.3	✓ [SI]	× E.3	✓ [SI]	× E.3	✓ [SI]
2 sto	× E.5	× E.5	✓ [SI]	✓ [SI]	× E.5	× E.5	✓ [SI]	✓ [SI]
k sto	× E.3	× E.5	× E.3	✓ [SI]	× E.3	× E.5	× E.3	✓ [SI]

Table 4: POMDP class implications for the Q^* target

	model-optimal				model-improving			
if $\exists \downarrow$ then $\exists \rightarrow$	2 det	k' det	2 sto	k' sto	2 det	k' det	2 sto	k' sto
2 det	✓ [SI]	✓ [SI]	✓ [SI]	✓ [SI]	✓ [SI]	✓ [SI]	✓ [SI]	✓ [SI]
k det	× E.3	✓ [SI]	× E.3	✓ [SI]	× E.3	✓ [SI]	× E.3	✓ [SI]
2 sto	× E.5	× E.5	✓ [SI]	✓ [SI]	× E.5	× E.5	✓ [SI]	✓ [SI]
k sto	× E.3	× E.5	× E.3	✓ [SI]	× E.3	× E.5	× E.3	✓ [SI]

Table 5: POMDP class implications for the model target

4.3 Quality: Improving/optimal

Lastly, we must consider when, holding *stochasticity*, *number*, and *target* constant, does *improving* imply *optimal*? The answer is no, and to show this, we can modify Figure 5 to admit a memory function that is improving but suboptimal for each target. Consider a modified Figure 5 in which there are four possible corridors the agent spawns in, with a fifth observation appended independently to the end of each corridor, either a 0 or 1. The agent has a choice of two tasks: recalling the most recent observation or recalling the entire sequence of observations. The harder task yields higher rewards. A 2-state DFA can recall the most recent observation, and therefore receive a small reward. However, just as before, the agent is unable to recall the sequence of the 4 initial observations. A larger DFA can, however, recall the full sequence and receive the large reward. This example also works for Q^* following similar reasoning. For model, it works because recalling the first observation is sufficient for predicting the full observation sequence, and recalling the full observation sequence is required for predicting reward.

5 Related Work

We can use the POMDP classes defined in this paper to generalize previously considered classes of restricted POMDPs, or otherwise to situate them within a natural hierarchical framework. The POMDP framework was first developed for control theory in Åström [1965] and later applied to AI problems in works such as Kaelbling et al. [1998]. Schmidhuber [1990], Lin and Mitchell [1992], and Meeden et al. [1993] developed the use of history features, and Lin and Mitchell [1992] and Ring [1994] developed variable-length history windows. Some approaches to memory were previously surveyed in Kaelbling et al. [1996]. McCallum [1996] implies (without explicitly mentioning "abstraction") that memory can be viewed as an abstraction over histories. However, he did not have access to the state abstraction hierarchy that came later due to Li et al. [2006].

Model-abstractions: The regular decision processes (RDPs) defined by Brafman and De Giacomo [2024] are essentially the class of POMDPs that admit finite transition-model and reward-model optimal memory functions. This follows from the FSM specification of RDPs. By formalizing this in the framework of approximate abstractions, however, our work can further accommodate approximate ε-optimal classes in addition to exact target-preserving classes. This fact generally holds for the other classes of POMDPs mentioned below; our framework enables approximate variants.

As Brafman and De Giacomo [2024] point out, k-order Markov POMDPs [Ching and Ng, 2006], in which the future observations are independent of the past history given the last k steps of observations, are a special case where the memory functions act only on the most recent k observations. If we decouple transition-model-optimality and reward-model-optimality, then we get the generalization

of k-order Markovianity pursued by Ni et al. [2023]. Those authors allowing the k parameter to vary between rewards and transitions via the "reward memory length" (k such that $\mathbb{E}[r_t|h_{1:t}, a_t] = \mathbb{E}[r_t|h_{t-k+1:t}, a_t]$) and "transition memory length". Thus, we can recover the class of POMDPs with a reward (resp. transition) memory length of k as the class of POMDPs that admit reward-model-optimal (resp. transition-model-optimal) memory functions that act only on the most recent k observations.

It is often useful to compress the history in addition to memorizing it, and it can be cheaper to maintain a long-term small FSM of memory than a short-term high-fidelity memory. Metrics like k-order Markov and memory length only care about memorizing past sequences, while our approach accommodates either by restricting the class of memory functions we consider. The memorization-based approach is somewhat specialized for context-based architectures such as transformers. Additionally, we mention that Efroni et al. [2022] defines the class of k-step decodable POMDPs, which are stronger than k-order Markov POMDPs, in which the last k observations can predict not only a function (the observation function) of the next state, but the state itself.

Policy and value abstractions: Just as k-order Markov POMDPs can be described as a type of model-optimal memory functions, the classic group of POMDPs defined by finitely-transient policies of size k [Sondik, 1978] are equivalent to POMDPs that admit a π^*-optimal memory function with k states. Furthermore, just as Ni et al. [2023]'s "reward" and "transition" memory lengths slightly generalized k-Markovianity, Ni et al. [2023] also define a "policy memory length" that generalizes finite transience in the same way. They correspond to π^*-optimal memory functions of a certain size in our framework. Ni et al. [2023] also define "value memory length" that corresponds to Q^*-optimal memory functions in our framework. They also discuss a credit assignment length that does not appear to easily fit within our system.

Belief space methods: We could define classes of POMDPs based on the size of the reachable belief states, dependent on a specific policy, such as the optimal policy or a set of policies. Zhang and Zhang [2001] defines "informative POMDPs", where each new observation yields information that helps partition the belief space. Roy et al. [2005] proposes a method of compressing the belief space of POMDPs with exponential family principal component analysis. Lee et al. [2007] proposes a different method that utilizes the covering number instead of PCA. Doing so, the authors achieve guarantees on the difficulty of finding approximately optimal solutions; the time required to find such a solution is polynomial in the covering number. It may be possible to connect our abstraction-based approach to belief space methods, but that is outside the scope of this paper.

Learnability: Another class of POMDPs is defined with restrictions to the observation or transition space such that efficient learnability is guaranteed. Jin et al. [2020] defines "undercomplete" POMDPs where there are more observations than latent states. Azizzadenesheli et al. [2016] and Guo et al. [2016] both place restrictions on the allowed observation functions (full column rank), transition function (full rank), with Azizzadenesheli et al. [2016] having an additional assumption of ergodicity and Guo et al. [2016] of full reward column rank. Given these assumptions, they find efficient learning techniques. These classes of POMDPs are different from the ones we define, which are focused on expressability rather than learnability, and we place no assumptions on the observation or transition functions. In the future, we hope to tie the two notions together.

Problem-specific: There are also specialized classes of POMDPs that are efficiently solvable by methods such as SLAM and Kalman filters, where constraints are placed on the transition dynamics and/or observation function, such as assuming Gaussian noise in it. Overall, while many of the traditional classes of POMDPs fit into our framework, this seems to be the first time they fit into a known hierarchy, and we are the first authors to consider relationships between the classes.

6 Future Work

We are excited about how the theory in this paper can apply to further settings, such as considering exploration in addition to exploitation. While the memory functions in this paper concern the latter, some environments, such as mazes, may require a large amount of memory to explore, even though the resulting policy does not require much memory to express. Similarly, we hope to later relate the complexity classes in this paper to useful metrics such as the speed of learning. Lastly, we note two ways that results in this paper could be sharpened: by considering variable ε in our tables instead of just $\varepsilon = 0$, and by considering quantitative and asymptotic estimates on how parameters such as the

number of states k required to achieve a certain performance level vary, similar to the estimate in Lemma 4.1.

7 Conclusion

We prove that memory functions induce state abstractions over the trajectory MDP, and therefore we can judge them based on typical state abstraction targets of π^*-preservation, Q^*-preservation, and model-preservation. We define classes of POMDPs based on whether, for any of these targets, a deterministic or stochastic memory function of a certain size exists that is optimal or improves the target. We prove which inclusions hold between these POMDP classes, and we show that these classes both systematize previously considered types of POMDPs (regular decision processes, k-order Markov models, memory lengths, finitely transient POMDPs) and generalize them with ε-approximate variants.

References

D. Abel, D.E. Hershkowitz, and M.L. Littman. Near optimal behavior via approximate state abstraction. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 2915–2923, 2016.

Cameron Allen, Aaron T Kirtland, Ruo Yu Tao, Sam Lobel, Daniel Scott, Nicholas Petrocelli, Omer Gottesman, Ronald Parr, Michael Littman, and George Konidaris. Mitigating partial observability in decision processes via the lambda discrepancy. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Karl Johan Åström. Optimal control of markov processes with incomplete state information i. *Journal of mathematical analysis and applications*, 10:174–205, 1965.

Kamyar Azizzadenesheli, Alessandro Lazaric, and Animashree Anandkumar. Reinforcement learning of pomdps using spectral methods. In *Conference on Learning Theory*, pages 193–256. PMLR, 2016.

Bram Bakker. Reinforcement learning with long short-term memory. *Advances in Neural Information Processing Systems*, 14, 2001.

Ronen I Brafman and Giuseppe De Giacomo. Regular decision processes. *Artificial Intelligence*, 331:104113, 2024.

Wai-Ki Ching and Michael K Ng. Markov chains. *Models, algorithms and applications*, 650:111–139, 2006.

Yonathan Efroni, Chi Jin, Akshay Krishnamurthy, and Sobhan Miryoosefi. Provable reinforcement learning with a short-term memory. In *International Conference on Machine Learning*, pages 5832–5850. PMLR, 2022.

Zhaohan Daniel Guo, Shayan Doroudi, and Emma Brunskill. A pac rl algorithm for episodic pomdps. In *Artificial Intelligence and Statistics*, pages 510–518. PMLR, 2016.

Joey Hong, Anca Dragan, and Sergey Levine. Offline rl with observation histories: Analyzing and improving sample complexity, 2023. URL https://arxiv.org/abs/2310.20663.

Nan Jiang. Notes on state abstractions, 2018. URL http://nanjiang.cs.illinois.edu/files/cs598/note4.pdf.

Chi Jin, Sham Kakade, Akshay Krishnamurthy, and Qinghua Liu. Sample-efficient reinforcement learning of undercomplete pomdps. *Advances in Neural Information Processing Systems*, 33: 18530–18539, 2020.

Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

L.P. Kaelbling, M.L. Littman, and A.R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, 1998.

Wee Lee, Nan Rong, and David Hsu. What makes some pomdp problems easy to approximate? *Advances in neural information processing systems*, 20, 2007.

L. Li, T.J. Walsh, and M.L. Littman. Towards a unified theory of state abstraction for MDPs. In *Proceedings of the Ninth International Symposium on Artificial Intelligence and Mathematics*, 2006.

Long-Ji Lin and Tom M Mitchell. *Memory approaches to reinforcement learning in non-Markovian domains*. Citeseer, 1992.

Andrew Kachites McCallum. *Reinforcement learning with selective perception and hidden state*. University of Rochester, 1996.

Lisa Meeden, Gary McGraw, and Douglas Blank. Emergent control and planning in an autonomous vehicle. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 15, 1993.

Tianwei Ni, Michel Ma, Benjamin Eysenbach, and Pierre-Luc Bacon. When do transformers shine in rl? decoupling memory from credit assignment. *Advances in Neural Information Processing Systems*, 36:50429–50452, 2023.

Mark Bishop Ring. *Continual learning in reinforcement environments*. The University of Texas at Austin, 1994.

Nicholas Roy, Geoffrey Gordon, and Sebastian Thrun. Finding approximate pomdp solutions through belief compression. *Journal of artificial intelligence research*, 23:1–40, 2005.

Jürgen Schmidhuber. Reinforcement learning in markovian and non-markovian environments. *Advances in neural information processing systems*, 3, 1990.

Satinder Singh, Tommi Jaakkola, and Michael Jordan. Reinforcement learning with soft state aggregation. *Advances in neural information processing systems*, 7, 1994.

Satinder P Singh and Richard C Yee. An upper bound on the loss from approximate optimal-value functions. *Machine Learning*, 16:227–233, 1994.

Edward J Sondik. The optimal control of partially observable markov processes over the infinite horizon: Discounted costs. *Operations research*, 26(2):282–304, 1978.

Jonathan Sorg and Satinder Singh. Transfer via soft homomorphisms. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 741–748, 2009.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018.

Stephan Timmer and Martin Riedmiller. *Reinforcement learning with history lists.* PhD thesis, University of Osnabrück, Germany, 2009.

Weihong Zhang and Nevin L Zhang. Solving informative partially observable markov decision processes. In *Proceedings of the 6th European Conference on Planning (ECP)*, 2001.

A Limitations

As discussed in Subsection 4.2, we consider only the expected case of π^*-optimality and the ∞-norm case of Q^*-optimality and model-optimality. We anticipate other asymptotic or numerical results may hold about approximations, but we do not show those here. Additionally, our results are for the $\varepsilon = 0$ case of strict optimality or improvement, while potentially more sophisticated results could follow from considering variable ε.

B Broader Impacts

The theory presented in this paper could potentially be applied to any areas of sequential decision making or reinforcement learning, such as robotics. However, it is foundational research not tied to particular applications or deployments, and thus has no societal impacts beyond what any work towards reinforcement learning theory entails. Impact control is the same as for any of these works.

C Abstraction Hierarchy

In order to accommodate soft abstractions, we alter the definition of lifting from

$$[f]_\varphi(s) \coloneqq f(\varphi(s))$$

to

$$[f]_\varphi(s) \coloneqq \mathop{\mathbb{E}}_{x\sim\varphi(s)} f(x)$$

With this change, the definitions of approximate π^*-preservation and Q^*-preservation remain the same as in Jiang [2018], where we take the space of trajectories as the state space.

However, the definition of model requires a slight modification from Jiang's definition. Jiang defines a model-preserving abstraction to satisfy an approximate bisimulation condition: for all $s, s' \in S$ with $\varphi(s) = \varphi(s')$ and for all $a \in A$,

$$|R(s,a) - R(s',a)| \leq \varepsilon_R$$
$$\|\Phi P(s,a) - \Phi P(s',a)\|_1 \leq \varepsilon_P$$

However, this definition does not necessarily generalize well to the case that $\varphi(s)$ is a distribution. We could instead require closeness between $\varphi(s)$ and $\varphi(s')$, but it seems easier to instead take a slightly different definition, namely a result that Jiang proves as a corollary of the approximate bisimulation condition.

Jiang proves in his Lemma 3 that when the approximate bisimulation condition holds, there exists an MDP $M_\varphi = (\varphi(S), A, P_\varphi, R_\varphi, \gamma)$ such that for all $s \in S$ and $a \in A$,

$$|R_\varphi(\varphi(s),a) - R(s,a)| \leq \varepsilon_R$$
$$\|P_\varphi(x,a) - \Phi P(s,a)\|_1 \leq \varepsilon_P$$

The actual definitions of R_φ and P_φ do not matter for latter proofs in the hierarchy (for model-preservation to imply Q^*-preservation); only their existence. Importantly, this definition works naturally for soft abstractions, which will allow us to work with stochastic memory functions. This is therefore the basis of how we define model-preservation in our paper.

Here's how we derive our definitions: The conclusion of Lemma 3 is precisely that if the bisimulation condition is met, then there exists a $P_\varphi : X \times A \to \Delta X$, $R_\varphi : X \times A \to \mathbb{R}$ such that for all $s \in S$ and $a \in A$,

1. $\|P_\varphi(x,a) - \Phi P(s,a)\|_1 \leq \varepsilon_P$
2. $|R_\varphi(\varphi(s),a) - R(s,a)| \leq \varepsilon_R$

Renaming these components, we get

1. $\exists f_P : X \times A \to \Delta X. \|[f_P]_\varphi - \Phi \circ P\| < \varepsilon_P$
2. $\exists f_R : X \times A \to \mathbb{R}. \|[f_R]_\varphi - R\|_\infty < \varepsilon_R$

and additional substitutions of $\varphi(S)$ for X and P_o for $\Phi \circ P$ yields

1. $\exists f_P : \varphi(S) \times A \to \Delta\varphi(S). \|[f_P]_\varphi - P_o\| < \varepsilon_P$
2. $\exists f_R : \varphi(S) \times A \to \mathbb{R}. \|[f_R]_\varphi - R\|_\infty < \varepsilon_R$

This is very close to our definition of model-preservation, which we recall below:

1. $\exists f_P : \varphi(T) \times A \to \Delta\Omega. \|[f_P]_\varphi - P_o\| < \varepsilon_P$
2. $\exists f_R : \varphi(T) \times A \to \mathbb{R}. \|[f_R]_\varphi - R\|_\infty < \varepsilon_R$

The differences are that we need to substitute T, the space of trajectories, in for S, and we need to replace Ω with $\varphi(T)$. Additionally, the output of f_P would be $\varphi(T) = \Delta M \times \Delta\Omega$, not $\Delta\Omega$, with this substitution. We change the output space of f_P because there is no need to predict next memory states; predicting only next observations should be considered the definition.

In the following subsections, we prove that the abstraction hierarchy continues to hold.

C.1 model-preservation implies Q^*-preservation

Consider an MDP with states $X = M \times \Omega$, actions A, transitions $(m', \omega') = x' \sim P_M(x, a) = (\mu(x, a), f_P(x, a))$ defined by the standard memory update but with averaging over observation transitions following f_P, and rewards given by f_R. Note that the transitions are stochastic and their probabilities can be written as $P_M(x'|x, a)$.

Because this is an MDP, we can define a Q-value function $f^* : \varphi(T) \times A \to \mathbb{R}$ for it that satisfies the optimal Bellman equation:

$$f^*(x, a) = f_R(x, a) + \gamma \max_{a' \in A} \mathop{\mathbb{E}}_{x' \sim P_M(x,a)} f^*(x', a')$$

Given that this holds for all x, we can take an expectation of both sides over $x \sim \varphi(\tau)$ to get

$$\mathop{\mathbb{E}}_{x \sim \varphi(\tau)} f^*(x, a) = \mathop{\mathbb{E}}_{x \sim \varphi(\tau)} f_R(x, a) + \gamma \max_{a' \in A} \mathop{\mathbb{E}}_{x \sim \varphi(\tau)} \mathop{\mathbb{E}}_{x' \sim P_M(x,a)} f^*(x', a')$$

We now rewrite the expectation over $f^*(x', a')$ in terms of τ' for the given τ and a'. For this we will define a new function $P_T : T \times A \to \Delta T$, such that $P_T(\tau, a) = \tau \oplus (a, \mathbb{E}_{x \sim \varphi(\tau)} f_P(x, a))$.

$$\begin{aligned}
&\mathop{\mathbb{E}}_{x \sim \varphi(\tau)} \mathop{\mathbb{E}}_{x' \sim P_M(x,a)} f^*(x', a') \\
&= \mathop{\mathbb{E}}_{x \sim \varphi(\tau)} \sum_{x'} P_M(x'|x, a) f^*(x', a') \\
&= \mathop{\mathbb{E}}_{x \sim \varphi(\tau)} \sum_{\omega'} \sum_{m'} P_M((m', \omega')|x, a) f^*((m', \omega'), a') \\
&= \mathop{\mathbb{E}}_{x \sim \varphi(\tau)} \sum_{\omega'} \sum_{m'} f_P(\omega'|x, a) \mu(m'|x, a) f^*((m', \omega'), a') \\
&= \mathop{\mathbb{E}}_{x \sim \varphi(\tau)} \sum_{\omega'} f_P(\omega'|x, a) \sum_{m'} \mu(m'|x, a) f^*((m', \omega'), a')
\end{aligned}$$

Note that for a given τ and a' the sum over ω' is equivalent to a sum over τ' with ω' being the last observation. Specifically, we can rewrite $\sum_{\omega'} f_P(\omega'|x, a)$ as an expectation over τ' as follows:

$$= \mathop{\mathbb{E}}_{x \sim \varphi(\tau)} \sum_{\tau'} f_P(\tau'_{\text{last } \omega}|x, a) \sum_{m'} \mu(m'|x, a) f^*((m', \tau'_{\text{last } \omega}), a')$$

Now we notice that for a given τ' and x, $(m', \tau'_{\text{last }\omega})$ is simply $\varphi(\tau')$ and $\mu(m'|x,a)$ is equivalent to $\mathbb{P}[m'|\varphi(\tau')]$ by the definition of φ.

$$
\begin{aligned}
&= \mathop{\mathbb{E}}_{x\sim\varphi(\tau)} \sum_{\tau'} f_P(\tau'_{\text{last }\omega}|x,a) \sum_{m'} \varphi(m'|\tau') f^*((m', \tau'_{\text{last }\omega}), a') \\
&= \sum_{\tau'} \mathop{\mathbb{E}}_{x\sim\varphi(\tau)} f_P(\tau'_{\text{last }\omega}|x,a) \sum_{m'} \varphi(m'|\tau') f^*((m', \tau'_{\text{last }\omega}), a') \\
&= \mathop{\mathbb{E}}_{\tau'\sim P_T(\tau,a)} \sum_{m'} \varphi(m'|\tau') f^*((m', \tau'_{\text{last }\omega}), a') \\
&= \mathop{\mathbb{E}}_{\tau'\sim P_T(\tau,a)} \mathop{\mathbb{E}}_{(m',\omega')\sim\varphi(\tau')} f^*((m', \omega'), a') \\
&= \mathop{\mathbb{E}}_{\tau'\sim P_T(\tau,a)} \mathop{\mathbb{E}}_{x'\sim\varphi(\tau')} f^*(x', a')
\end{aligned}
$$

Let $f^*(\tau, a) = \mathbb{E}_{x\sim\varphi(\tau)} f^*(x, a)$. Assuming that rewards are bounded, i.e. there exists an $R_{\max}$ such that $|R| < R_{\max}$, we can compute $\|f^*(\tau, a) - Q^*(\tau, a)\|$ as:

$$
\begin{aligned}
&\|f^*(\tau,a) - Q^*(\tau,a)\| \\
&= \left\| E_{x\sim\varphi(\tau)} f_R(x,a) - R(\tau,a) + \gamma \max_{a'\in A} \mathop{\mathbb{E}}_{\tau'\sim P_T(\tau,a)} [f^*(\tau', a)] - \max_{a'\in A} \mathop{\mathbb{E}}_{\tau'\sim P(\tau,a)} [Q^*(\tau', a')] \right\| \\
&\leq \left\| \mathop{\mathbb{E}}_{x\sim\varphi(\tau)} f_R(x,a) - R(\tau,a) \right\| + \gamma \left\| \max_{a'\in A} \mathop{\mathbb{E}}_{\tau'\sim P_T(\tau,a)} [f^*(\tau', a)] - \max_{a'\in A} \mathop{\mathbb{E}}_{\tau'\sim P(\tau,a)} [Q^*(\tau', a')] \right\| \\
&\leq \varepsilon_R + \gamma \max_{a'\in A} \left\| \mathop{\mathbb{E}}_{\tau'\sim P_T(\tau,a)} [f^*(\tau', a)] - \mathop{\mathbb{E}}_{\tau'\sim P(\tau,a)} [Q^*(\tau', a')] \right\| \\
&\leq \varepsilon_R + \gamma \max_{a'\in A} \left\| \sum_{\tau'\in T} [P_T(\tau'|\tau,a) f^*(\tau', a)] - \sum_{\tau'\in T} [P(\tau'|\tau,a) Q^*(\tau', a')] \right\| \\
&= \varepsilon_R + \gamma \max_{a'\in A} \left\| \sum_{\tau'\in T} P_T(\tau'|\tau,a) f^*(\tau', a) - P(\tau'|\tau,a) Q^*(\tau', a') \right\| \\
&= \varepsilon_R + \gamma \max_{a'\in A} \left\| \sum_{\tau'\in T} P_T(\tau'|\tau,a) f^*(\tau', a) - P(\tau'|\tau,a) f^*(\tau', a) + P(\tau'|\tau,a) f^*(\tau', a) - P(\tau'|\tau,a) Q^*(\tau', a') \right\| \\
&= \varepsilon_R + \gamma \max_{a'\in A} \left\| \sum_{\tau'\in T} \left(P_T(\tau'|\tau,a) - P(\tau'|\tau,a)\right) f^*(\tau', a) + P(\tau'|\tau,a) \left(f^*(\tau', a) - Q^*(\tau', a')\right) \right\| \\
&< \varepsilon_R + \gamma \max_{a'\in A} \left\| \sum_{\tau'\in T} \left(P_T(\tau'|\tau,a) - P(\tau'|\tau,a)\right) f^*(\tau', a) \right\| + \max_{a'\in A} \left\| \sum_{\tau'\in T} P(\tau'|\tau,a) \left(f^*(\tau', a) - Q^*(\tau', a')\right) \right\|
\end{aligned}
$$

which follows from the triangle inequality. The next inequality follows from $\|\max \cdot\| \leq \max \|\cdot\|$.

$$
\begin{aligned}
&\leq \varepsilon_R + \gamma \max_{a'\in A} \left\| \sum_{\tau'\in T} \left(P_T(\tau'|\tau,a) - P(\tau'|\tau,a)\right) f^*(\tau', a) \right\| + \max_{a'\in A, \tau'\in T} \|f^*(\tau', a) - Q^*(\tau', a')\| \\
&\leq \varepsilon_R + \gamma \max_{a'\in A} \left(\left(\sum_{\tau'\in T} \|P_T(\tau'|\tau,a) - P(\tau'|\tau,a)\| \right) \max_{\tau'\in T} \|f^*(\tau', a)\| \right) + \max_{a'\in A, \tau'\in T} \|f^*(\tau', a) - Q^*(\tau', a')\|
\end{aligned}
$$

From the definition of model preserving we have that $\varepsilon_P > \sum_{\tau'\in T} \mathbb{E}_{x\sim\varphi(\tau')} f_P(x,a) - P(\tau', a)_o = \sum_{\tau'\in T} P_T(\tau'|\tau,a) - P(\tau'|\tau,a)$. We also have that the function f^* is bounded by $R_{\max}/(1-\gamma)$ which is obtained by considering the maximum reward $R_{\max}$ in the Bellman equation for f^*.

$$\leq \varepsilon_R + \gamma \max_{a' \in A} \varepsilon_P \frac{R_{\max}}{(1-\gamma)} + \max_{a' \in A, \tau' \in T} \|f^*(\tau', a) - Q^*(\tau', a')\|$$
$$\leq \varepsilon_R + \frac{\gamma \varepsilon_P R_{\max}}{(1-\gamma)} + \max_{a' \in A, \tau' \in T} \|f^*(\tau', a) - Q^*(\tau', a')\|$$

Letting $E(\tau, a) = \|f^*(\varphi(\tau, a) - Q^*(\tau, a)\|$ we can rewrite the inequality as,

$$E(\tau, a) \leq \varepsilon_R + \gamma \varepsilon_P R_{\max} + \gamma \max_{a' \in A, \tau' \in T} E(\tau', a')$$
$$\max_{a \in A, \tau \in T} E(\tau, a) \leq \max_{a \in A, \tau \in T} \left(\varepsilon_R + \frac{\gamma \varepsilon_P R_{\max}}{(1-\gamma)} + \gamma \max_{a' \in A, \tau' \in T} E(\tau', a') \right)$$
$$\max_{a \in A, \tau \in T} E(\tau, a) \leq \varepsilon_R + \frac{\gamma \varepsilon_P R_{\max}}{(1-\gamma)} + \gamma \max_{a \in A, \tau \in T} \max_{a' \in A, \tau' \in T} E(\tau', a')$$
$$\max_{a \in A, \tau \in T} E(\tau, a) \leq \varepsilon_R + \frac{\gamma \varepsilon_P R_{\max}}{(1-\gamma)} + \gamma \max_{a \in A, \tau \in T} E(\tau, a)$$
$$\max_{a \in A, \tau \in T} E(\tau, a) \leq \frac{\varepsilon_R}{(1-\gamma)} + \frac{\gamma \varepsilon_P R_{\max}}{(1-\gamma)^2}$$

so for all $\tau \in T, a \in A$

$$\left\| \mathop{\mathbb{E}}_{x \sim \varphi(\tau)} f^*(\varphi(x, a) - Q^*(\tau, a) \right\| \leq \frac{\varepsilon_R}{(1-\gamma)} + \frac{\gamma \varepsilon_P R_{\max}}{(1-\gamma)^2}$$

So the Q^* error is bounded by a function of the two sources of model error.

C.2 Q^*-preservation implies π^*-preservation

To prove the other implication in the hierarchy, that Q^*-preservation implies π^*-preservation, we can utilize the same approach is in Jiang [2018]. Namely, the lemma

Lemma C.1.

$$\|V^* - V^{\pi_f}\|_\infty \leq \frac{2\|f - Q^*\|_\infty}{1-\gamma}$$

continues to hold in our generalized setting because the lifted function $[\pi]_\varphi$ has the same signature $S \to \Delta A$ in both of the exact abstraction and soft abstraction cases.

Proof. See Singh and Yee [1994]. □

D Abstractions to Memory Functions

The finest/identity abstraction maps each trajectory to its own memory state. For standard environments with unbounded trajectories, this yields an infinite-state automata memory function capable of perfect recall. The opposite of this would be an abstraction that maps all trajectories to a single memory state. This yields a trivial memory function with a single state, which is effectively always blank.

Two intermediate classes of memory functions that are useful to define are those that are ε-close to *target*-preserving abstractions over either the trajectory MDP or the effective MDP. The trajectory MDP models having perfect information, so being ε-close to *target*-preserving over it implies almost having enough information to recreate the *target*. The effective MDP models the opposite situation in which no information is available, so ε-close *target* preservation means having almost no advantage over a blank memory.

Anything that is not ε-close to *target*-preserving over the effective MDP is called "improving".

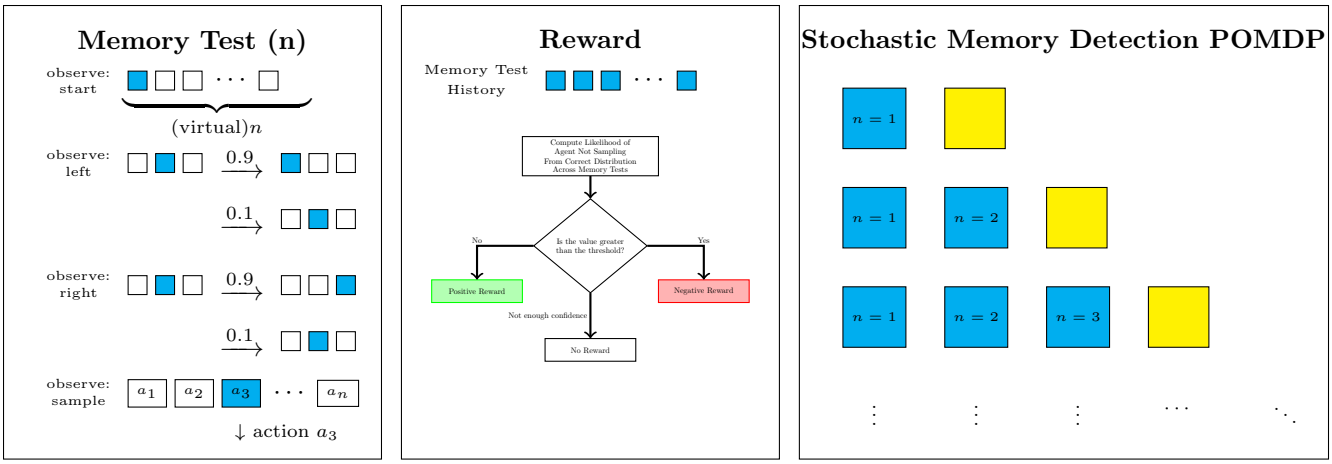

Figure 8: Example E.1 POMDP.

E Counterexamples

Example E.1. The following counterexample is for the following results:

- The existence of 2-stochastic expected return optimal memory doesn't imply the existence of k-deterministic expected return optimal memory if rewards are unbounded.
- The existence of 2-stochastic expected return improving memory doesn't imply the existence of k-deterministic expected return improving memory if rewards are unbounded.
- The existence of 2-stochastic or k-stochastic expected return optimal memory doesn't imply the existence of k-deterministic expected return optimal memory if rewards are unbounded.
- The existence of 2-stochastic or k-stochastic expected return improving memory doesn't imply the existence of k-deterministic expected return improving memory if rewards are unbounded.

To show both the optimal conditions and the improvable conditions it is sufficient to show that for a given POMDP where there exists a k-stochastic memory function which is expected return optimal there doesn't necessarily exist a k-deterministic memory function which is improvable. This is derived from the fact that an optimal memory function is improving, along with its contrapositive: if there doesn't exist an improving memory function there cannot exist an optimal one. For this purpose, we construct the following counter example and depict its structure in Figure 8.

Consider an environment where the agent's task is to simulate trajectories in a stochastic virtual environment and sample from the state distribution after some number of steps. The agent's actual environment is constructed adversarially such that the agent is rewarded if it produces different samples for the same simulated trajectory over multiple trials. The environment is composed of a detection mechanism and a rewarding mechanism. The detection mechanism detects whether an agent has deterministic or stochastic memory irrespective of the stochasticity of the policy. The rewarding mechanism produces different reward based on if the agent memory is deterministic or stochastic and induces non-improving reward for deterministic memory. Finally, there is also an opt-out action that can be taken at the first time step giving the agent 0 reward.

The detection mechanism is composed of a series of tests where the agent is asked to simulate a particular stochastic environment. The virtual environment is a board with $n = 10$ spaces in a line and a token in the first space. The value of n corresponds to the number of memory states of the stochastic memory function and we choose it to be 10 for the purposes of this example. For an agent with 2 stochastic memory states we would similarly have $n = 2$. Each test starts with ω_{reset} which indicates that the token should be virtually placed on the first space. Then, an arbitrary long sequence of observations from the set $\{\omega_{\text{left}}, \omega_{\text{right}}\}$ is provided to the agent. When receiving ω_{left} or ω_{right} the agent is expected to simulate the token moving left or right respectively with probability 0.9 or

otherwise staying in place (the choice of probability here is arbitrary as long as it isn't uniform). Finally, the agent gets the observation ω_{sample} for which the agent is expected to take an action from $a_1, \ldots, a_{10}$ corresponding to where the simulated token ended up. For all other observations, the agent is expected to provide the action a_0.

After a single test the likelihood that the sampled action was in fact from the expected virtual distribution is computed and by repeating the test the statistical confidence can be increased. To run infinitely many tests infinitely many times a list of current tests is constructed and run sequentially. Upon completion a new longer test is added and the full list of tests is repeated. This ensures that in the limit as a finite time step t goes to infinity, infinitely many tests, are run infinitely many times, and the length of the tests also approaches infinity.

A single test cannot be executed perfectly by an agent with deterministic memory while it is trivially handled by a stochastic memory agent with 10 memory states. For any choice of finite deterministic memory size there will eventually be a test that requires remembering more possible distributions than there are memory states. In this case, the best that the agent would be able to do is sample from a distribution that is ε close to the true distribution. As that particular test is repeated infinitely many times, the discrepancy between the agents sampling distribution and the true distribution will always become statistically significant and detectable.

For the rewarding mechanism of the environment, we simply give the agent reward depending on if it is believed to have stochastic or deterministic memory. A reward that is exponential in the time step, $|R(t)| = O(1/\gamma^t)$, would be sufficient to overshadow the discount factor γ. For this particular counter example, we will provide a positive reward for agents believed to have stochastic memory and a negative reward for agents believed to have deterministic memory (according to the detection mechanism). Positive and negative rewards are equal in magnitude. If the current likelihood estimate doesn't have sufficient confidence a reward of 0 is given. In the limit, the probability that the detecting mechanism is wrong about the nature of the agents memory goes to zero and so in the limit, all agents with receive positive/negative reward if they have stochastic/deterministic memory respectively.

The opt-out action is available for the agent at the first time step and if taken gives the agent 0 reward and ends the episode. This reward is arbitrary as long as it is better than the reward achieved by an agent with deterministic memory. This ensures that the best option for a deterministic agent is to opt out and so achieve the same performance as no memory.

We now combine the detecting mechanism with the rewarding mechanism. Importantly, the detecting piece is never certain that a given agent has deterministic or stochastic stochastic memory for any finite time step t. This means we cannot switch to the rewarding piece indefinitely. Instead, after each test in the detecting piece we allow the rewarding piece to take over for a single time step to provide reward based on the current likelihood of the agent having deterministic memory. Because of randomness an agent with stochastic memory may be believed to have deterministic memory but in the limit this will resolve and the expected return will be positive. A deterministic memory agent with finitely many states will be detected after finitely many time steps and so will have a negative expected return even if it receives zero or positive reward for some finite number of time steps initially. Because of this, any non-stochastic agent or agent with insufficient memory will take the opt-out action when maximizing expected return and so achieve the same reward as a no memory function behavior.

Note that for this counter example we require non-finite trajectories and we only detect deterministic memory in the limit as the time step goes to infinity. If trajectories are finite, then proof 4.1 gives a deterministic memory that is better or equal to any given stochastic memory.

Example E.2. The following counter example is for the following results:

- The existence of k-stochastic expected Q^* optimal memory doesn't imply the existence of 2-deterministic Q^* optimal memory
- The existence of 2-stochastic expected Q^* optimal memory doesn't imply the existence of 2-deterministic Q^* optimal memory
- The existence of k-stochastic expected Q^* improving memory doesn't imply the existence of 2-deterministic Q^* improving memory

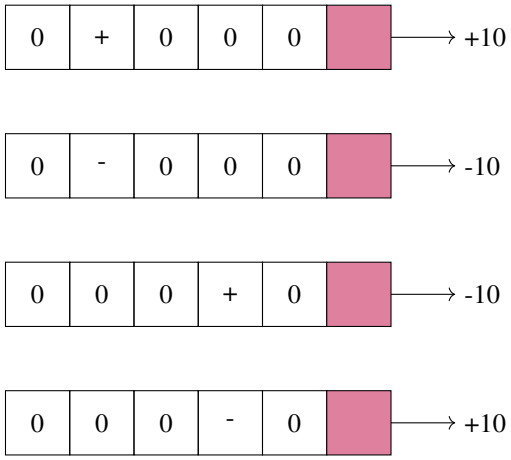

Figure 9: Four corridors.

- The existence of 2-stochastic expected Q^* improving memory doesn't imply the existence of 2-deterministic Q^* improving memory

This environment is depicted in Figure 9.

Consider an environment in which the agent is in one of four corridors with a single action and receives observations for 5 time steps before receiving a final reward. All observations are blank (0) with the except of one, on either the 2nd or 4th time step. That observation is either a $+$ or a $-$. If the observations seen at time step 2 are $+$ or $-$, the final reward is 10 or -10 respectively. If the unique observation is instead seen on the 4th time step, then the final rewards are flipped. Because the agent only has a single action at each time step, there is only one possible policy.

In this environment, an agent with k memory states can track both the time step and the specific observation, $+$ or $-$, to know the exact reward that will be received at the final time step.

For an agent with 2 deterministic memory states, m_1 and m_2, there are at most 2^6 possible memory functions. We can think of these in terms of the possible memory transitions for the three observations, 0, $+$, and $-$, independently.

First consider the possible memory transitions for the blank observation 0. Two options are collapsing the memory to either m_1 or m_2 in which case there is no information at the final timestep and the best possible error is 10. So for the 0 observation we can either transpose the memory states or keep them the same (identity). For the $+$ observation we similarly cannot collapse the memory to either m_1 or m_2 or else reach the final state with a fixed memory state in corridors 2 and 4 and the best possible error is 10. The same logic follows for the $-$ memory transitions.

So we have shown that the memory function for all observations either swaps m_1 to m_2 and m_2 to m_1 or is the identity. If the transitions for the 0 observation are the identity then we notice that for either the $-$ or $+$ observations we reach the final state with the same observation in corridors 2 and 4 or corridors 1 and 3 and so the possible error is 10. If the transitions for the 0 observation transpose the memory state we have the same result because there is always an odd number of 0 observations before and after any $+$ or $-$ observation.

So we have shown that for all possible 2-state deterministic memory functions, the agents memory at the final state is the same for both a corridor giving 10 reward and -10 reward meaning the best possible Q^* error is 10, the same as no memory.

Example E.3. Consider an environment where the agent needs to keep track of the multiplicity of the time step. First, the agent receives a sequence of 0, 1, 2, or 3 ω_{null} observations followed by a single ω_{end} observation. At each time step, the agent can only take the action a. After each observation, the agent receives a reward of 0 unless the observation is ω_{end} and the time step is a multiple of 3. More specifically, here are the rewards for the following observation sequences:

1. $R(\omega_{\text{null}}) = 1$
2. $R(\omega_{\text{null}}, \omega_{\text{end}}) = 0$
3. $R(\omega_{\text{null}}, \omega_{\text{null}}, \omega_{\text{end}}) = 0$
4. $R(\omega_{\text{null}}, \omega_{\text{null}}, \omega_{\text{null}}, \omega_{\text{end}}) = 1$

Each of the possible trajectory sequences is equally likely.

A 3-state deterministic memory function is sufficient to achieve 0 reward error in this environment. Consider three states m_1, m_2, and m_3 that transition in a cycle with 100% probability. Whenever the memory is in state m_1, the initial memory state, the agent can predict a reward of 1 and otherwise predict a reward of 0. This gives a candidate f which satisfies $\|[f]_\varphi - Q^*_M\|_\infty \leq \varepsilon_{Q^*}$.

An agent with no memory could achieve a maximum error of $1/2$ by predicting a reward of $1/2$ in all cases.

We now consider the constraints that a 2-state stochastic memory function would need to satisfy in order to perform better than an agent with no memory. Note that because the agent only has one available action, we can think of f as a function only of memory. We will also reduce our view to only the ω_{end} observation because if no f exists which outperforms a memoryless agent over just one of the observations it also cannot exist over both. Because we are considering only a single action and a single observation, f becomes a function of only the memory state. This lets us succinctly express $[f]_\varphi$ as $\mathbb{E}_{(m,\omega)\sim\varphi(\tau)}[f(m)] = \mathbb{P}(m_1|\tau)f(m_1) + \mathbb{P}(m_2|\tau)f(m_2)$. For convenience, we write $f = (f(m_1), f(m_2))$.

We now need to determine what $Pr(m_1)$ and $Pr(m_2)$ would be for a given trajectory. Because the observation is always ω_{null} for all time steps before the ω_{end} observation, and because the agents action is always a, the memory function update reduces to a function of only the previous memory state. This also lets us express it as a two by two matrix $A = \begin{pmatrix} p & 1-p \\ 1-q & q \end{pmatrix}$, where p is the probability of transitioning to m_1 when in m_1 and q is the probability of transitioning to m_2 when in m_2. Now, given a vector representing the probabilities of each memory state, we can find the corresponding probabilities at the next time step by multiplying this vector by A. Finally, we say the initial memory state distribution is $y = (y_1, y_2)$ where y_1 is the probability of starting in state m_1 and y_2 is the probability of starting in state m_2. We can now express the final memory state distribution of a trajectory of length n as yA^n. We can then get $[f]_\varphi$, by computing $\mathbb{E}_{(m)\sim\varphi(\tau)}[f(m)] = yA^n f^T$.

Using y, A, and f, we can express $[f]_\varphi$, which is equivalently the final predicted reward, for the terminal states of the four possible trajectories of this environment. If we assume that this 2-stochastic memory agent is improving, we know that these predictions must be greater or less than $1/2$ based on the true Q^* value.

1. $yf^T > 1/2$
2. $yAf^T < 1/2$
3. $yA^2f^T < 1/2$
4. $yA^3f^T > 1/2$

Note that these inequalities are strict because predicting $1/2$ would mean that the error of the agent is at least $|1/2 - 1|$ or $|1/2 - 0|$ which is not better than a no-memory agent.

From the first two conditions, we get that $yAf < yf$, and subtracting $yf = yIf$, where I is the identity matrix, we get $y(A - I)f < 0$. From the second two conditions we get that $yA^2f < yA^3f$ and subtracting yA^2f gives $0 < y(A^3 - A^2)f$. We can calculate $A^3 - A^2$ to be $(a + b - 1)^2(A - I)$ so we get the final condition of $0 < (a + b - 1)^2 y(A - I)f$.

Because $(a + b - 1)^2$ is positive we have a contradiction. Both $0 < (a + b - 1)^2 y(A - I)f$ and $y(A - I)f < 0$ cannot be true. This implies that a 2-stochastic memory agent cannot perform any better than a no memory agent on this environment.

Lemma E.4. *If there exists a terminal trajectory, τ, such that $|[f]_\varphi(\tau) - R(\tau)| \geq \varepsilon$ for all $f : \varphi(T) \times A \to \mathbb{R}$, then:*

1. $\varepsilon_{Q^*} \geq \varepsilon$
   Because τ is a terminal trajectory $Q^*_M(\tau) = R(\tau)$ and by the definition of infinity norm $\|\cdot\|_\infty$, we must have that ε_{Q^*} is at least ε

2. $\varepsilon_R \geq \varepsilon$
   By the definition of infinity norm $\|\cdot\|_\infty$, ε_R must at least be ε

Example E.5. The following counter example is for the following results:

- The existence of 2-stochastic Q* improving memory doesn't imply the existence of k-deterministic Q* improving memory.
- The existence of 2-stochastic Model improving memory doesn't imply the existence of k-deterministic Model improving memory.
- The existence of 2-stochastic Q* optimal memory doesn't imply the existence of k-deterministic Q* optimal memory.
- The existence of 2-stochastic Model optimal memory doesn't imply the existence of k-deterministic Model optimal memory.

First we define a virtual MDP with two states s_1 and s_2 and a parameterized set of actions $A = \{a_x | x \in [-1, 1]\}$. Actions a_x with $x >= 0$ result in the the following two transitions $P(s_2|s_1, a_x) = x$, $P(s_1|s_1, a_x) = 1 - x$, and $P(s_2|s_2, a_x) = 1$. Actions a_x with $x < 0$ result in the the following two transitions $P(s_1|s_2, a_x) = x$, $P(s_2|s_2, a_x) = 1 - x$, and $P(s_1|s_1, a_x) = 1$. Actions are selected uniformly at random at each time step.

We now wrap this MDP with a POMDP to produce the desired counter example. The POMDP tracks the running probability of state s_1 and at each time step communicates the action taken in the MDP, a_x, to the agent as observation o_x. At each time step the POMDP has a .1 probability of terminating and presenting the agent with the o_{end} observation. For this observation the reward is equal to the probability of the MDP being in state s_1. The reward for all other observations is 0. The trajectory terminates after the o_{end} observation. The action space for the agent is $A = \{a\}$, a single action for all time steps.

There exists a 2-stochastic optimal memory which is sufficient to predict the reward at each time step. Specifically, we take the memory function which for observations transitions its memory states m_1 and m_2 in the same way as the virtual MDP transitions its states s_1 and s_2 at each time step. This is Q^* optimal. f can be chosen such that $f((m_1, o_{\text{end}}), a) = 1$ and $f((m_2, o_{\text{end}}), a) = 0$ which gives an Q^* error of 0 for terminal trajectories. For non-terminal partial trajectories we note that the true future return is independent of the actual time step because there is no time dependence for transitions nor termination. This lets us define $\hat{R}(s_1)$ to be the future discounted rewards if the current MDP state is s_1 and $\hat{R}(s_2)$ to be the future discounted rewards if the current MDP state is s_2. We can then choose $f((m_1, o_x \neq o_{\text{end}}), a) = \hat{R}(s_1)$ and $f((m_2, o_x \neq o_{\text{end}}), a) = \hat{R}(s_2)$ which also gives $\varepsilon_{Q^*} = 0$. This is because $P(m_1|\tau) = P(s_1|\tau)$ and $P(m_2|\tau) = P(s_2|\tau)$ so when lifting for a given trajectory τ we get $P(m_1) * f((m_1, o \neq o_{\text{end}}), a) + P(m_2) * f((m_2, o \neq o_{\text{end}}), a) = P(s_1) * \hat{R}(s_1) + P(s_2) * \hat{R}(s_2)$ which is exactly the true future discounted reward.

Following similar reasoning, this memory function is also Model optimal. For terminal trajectories f_R can match f and for non-terminal trajectories $f_R((\cdot, o \neq o_{\text{end}}), a) = 0$ which gives $\varepsilon_R = 0$. For transitions, the probability of o_{end} is always .1 and the probability of the observations o_x follows $U(-1, 1) * .9$ which gives a natural choice of f_P with $\varepsilon_P = 0$.

No memory can at best achieve $\varepsilon_{Q^*} = 1/2$ and $\varepsilon_R = 1/2$ because the true reward at the final observation can be either 0 or 1 and $f(o_{\text{end}}, a)$ can at best be assigned to the middle of this range to minimize error.

We now consider a k-deterministic memory function μ, with corresponding . Lets assume that exists function $f : \varphi(T) \times A \to \mathbb{R}$ such that $\|f(\varphi(\tau)) - R(\tau)\|_\infty \leq \varepsilon < 1/2$. Note that this is identical to the terminal trajectory requirement for f in Lemma E.4. For simplicity we can exclude the observation and action, which are always o_{end} and a respectively, to get an identical $f : M \to \mathbb{R}$. We now prove by contradiction that such f cannot exist.

For each memory state m_i we define $S_i = \{\tau | \varphi(\tau) = m_i, \tau \text{ is terminal}\}$ and $R(S) = \{R(\tau) | \tau \in S\}$. Let S, generated by memory state m, be the set for which $\sup R(S) - \inf R(S) \geq \sup R(S_i) - \inf R(S_i)$ for all S_i. The best choice of $f(m)$ is $(\sup R(S) + \inf R(S))/2$ because for all $\tau \in S$, $|f(m) - R(\tau)| \leq (\sup R(S) - \inf R(S))/2 \leq \varepsilon < 1/2$. This implies that $\sup R(S) - \inf R(S) \leq 2\varepsilon < 1$. Either $\inf S > 0$ or $\sup S < 1$. Without loss of generality, assume that $\inf S > 0$.

We now consider an arbitrary trajectory τ and define the operation $\tau \oplus o_x$ for observation o_x which generates a new terminal trajectory by inserting the observation o_x before o_{end} in the trajectory. Notice that for any $\tau_1, \tau_2 \in S$ and o_x, we have that $\varphi(\tau_1 \oplus o_x) = \varphi(\tau_2 \oplus o_x) = \mu(m, a, o_x)$. We also have that for positive x, $R(\tau \oplus o_x) = (1 - x)R(\tau)$ as defined by the probability of transitioning from s_1 to s_1 in the virtual MDP.

For any choice $0 < \varepsilon' < \frac{1}{4} \sup R$ we can choose $\tau_1, \tau_2 \in S$ and o_x such that $\sup R(S) - R(\tau_2 \oplus o_x) = \varepsilon'$ and $R(\tau_1 \oplus o_x) < \inf R(S)$. First we pick $\tau_2 \in S$ such that $\sup R(S) - R(\tau_2) = \delta < \varepsilon'$, for $\delta \in \mathbb{R}$, which gives $R(\tau_2) = \sup R(S) - \delta$. This then means we can choose $x = 1 - (\sup R(S) - \varepsilon')/(\sup R(S) - \delta)$ which gives the desired $\sup R(S) - R(\tau_2 \oplus o_x) = \varepsilon'$. The condition $0 < \varepsilon' < \frac{1}{4} \sup R$ ensures $x \in (0, 1]$. We can now choose $\tau_1 \in S$ such that $R(\tau_1) - \inf R(S) = \delta' < \frac{x}{1-x} \inf R(S)$ which reduces to the desired $R(\tau_1)(1 - x) < \inf R(S)$ which is equivalent to $R(\tau_1 \oplus o_x) < \inf R(S)$.

We now consider a sequence of choices of $\varepsilon', \varepsilon_1, \varepsilon_2, \ldots$, such that $\varepsilon_i = \varepsilon_{i-1}/2$. For each choice of epsilon ε_i we have $\tau_{1,i}, \tau_{2,i} \in S$ and o_x such that $\sup R(S) - R(\tau_{2,i} \oplus o_x) = \varepsilon'$ and $R(\tau_{1,i} \oplus o_x) < \inf R(S)$. Let $m_i = \varphi(\tau_{1,i} \oplus o_x) = \varphi(\tau_{2,i} \oplus o_x)$ for each ε_i. For the infinite sequence of m_i, there must be some particular $\hat{m}$ that repeats infinitely many times. Let $\hat{m}$ generate $\hat{S}$ and I be the set of $\{i | m_i = \hat{m}\}$. For the pairs $\tau_{1,i} \oplus o_x, \tau_{2,i} \oplus o_x \in \hat{S}$, we have that $\inf R(\hat{S}) \leq \inf_{i \in I} R(\tau_{1,i} \oplus o_x) < \inf R(S)$ and $\sup_{i \in I} R(\tau_{2,i} \oplus o_x) = \sup R(S) \leq \sup R(\hat{S})$. This implies that $\sup R(\hat{S}) - \inf R(\hat{S}) > \sup R(S) - \inf R(S)$ which contradicts the definition of S. So we have that $\|f(\varphi(\tau)) - R(\tau)\|_\infty \geq 1/2$ and by Lemma E.4 we have that $\varepsilon_{Q^*} \geq 1/2$ and $\varepsilon_M \geq 1/2$.

Example E.6. We can consider a simpler but related counter example to E.5 to prove only the optimal cases:

- The existence of 2-stochastic Q* optimal memory doesn't imply the existence of k-deterministic Q* optimal memory.
- The existence of 2-stochastic Model optimal memory doesn't imply the existence of k-deterministic Model optimal memory.

We first define a virtual state machine with two states s_1 and s_2. The initial state is s_1 and at each time step there is a 10% chance of s_1 transitioning to state s_2. The state s_2 always transitions to itself.

We now consider a POMDP with two observations o and o_{end} and one action a which wraps the virtual state machine. At each time step the agent receives observation o and when the agent takes its action a the virtual state machine is updated. At each time step there is a 10% chance of the trajectory terminating in which case the agent receives the o_{end} observation and the following reward is equal to the probability of the virtual state machine being in state s_1. All other rewards are 0.

There exists a 2-stochastic optimal memory which is sufficient to predict the reward at each time step. Specifically, we take the memory function which for observations transitions its memory states m_1 and m_2 in the same way as the virtual MDP transitions its states s_1 and s_2 at each time step. This is Q^* optimal. f can be chosen such that $f((m_1, o_{\text{end}}), a) = 1$ and $f((m_2, o_{\text{end}}), a) = 0$ which gives an Q^* error of 0 for terminal trajectories. For non-terminal partial trajectories the value of f for each memory state can be adjusted based on the probability distribution of time steps before the environment terminates. This memory function is similarly Model optimal. For terminal trajectories f_R can match f and for non-terminal trajectories $f_R((\cdot, o \neq o_{\text{end}}), a) = 0$ which gives a model reward error of 0. For transitions, the probability of receiving the next observation is always fixed so the choice of f_P is trivial.

Note that an agent with no memory, upon getting the terminal observation o_{end} can at best guess the center of the range of possible rewards $[1, 0)$ and so has $\varepsilon_{Q^*} \geq .5$ and $\varepsilon_R \geq .5$ by Lemma E.4.

We now consider a k-deterministic memory agent. At the final observations there are an infinite number of possible rewards that the agent may receive in the range $[1, 0)$. However, because there

are only k memory states, $f((m_i, o_{\text{end}}), a)$ can only take on k possible values, one for each m_i. By pigeon hole, there must exist at least one terminal trajectory, τ for which $|f(\tau) - R(\tau)| > 0$. This then implies that $\varepsilon_{Q^*} > 0$ and $\varepsilon_R > 0$ by Lemma E.4. So, no k-deterministic memory function exists.

F Infinity Norm to Expected Case

Here we explain the relationships between the infinity norm and expected cases.

First, we show model error being zero implies the agent's memory makes the gives a Markov representation of the MDP. Suppose that the the model error definition

$$\exists f_P : \varphi(T) \times A \to \Delta\Omega. \|[f_P]_\varphi - P_o\|_1 < \varepsilon_P$$

is satisfied for $\varepsilon_P = 0$. Then,

$$\exists f_P : \varphi(T) \times A \to \Delta\Omega. [f_P]_\varphi = P_o$$

where $P_o = \mathbb{P}(\cdot|\tau, a_t)$. This says that given $\varphi(\tau_t) = (m_t, \omega_t)$ and a_t, the agent can predict the distribution over next observations $\omega_{t+1} \sim \mathbb{P}(\omega_{t+1}|\tau_t, a_t)$ perfectly. Thus, $\mathbb{P}(\omega_{t+1}|\tau_t) = \mathbb{P}(\omega_{t+1}|m_t, \omega_t, a_t)$, which is the definition of memory yielding a Markov representation. Model error being zero implying that rewards are Markov follows similarly.

Second, we show the connection between π^*-preservation and expected return. Observe

$$\left|\mathop{\mathbb{E}}_{\tau}[V_M^{[\pi]_\varphi} - V_M^*]\right| \leq \mathop{\mathbb{E}}_{\tau}\left|V_M^{[\pi]_\varphi} - V_M^*\right| \leq \left\|V_M^{[\pi]_\varphi} - V_M^*\right\|_\infty \leq \varepsilon_{\pi^*}$$

where the first inequality follows by Jensen's inequality. Thus, if there exists a $\pi : \varphi(T) \to \Delta A$ such that the final inequality follows (which is, by definition, π^*-preservation), then for this π, the expected return is also constrained.

Third, the relationship between Q^*-preservation and expected value error of π^* in the text from Sutton and Barto [2018] is similar. Recall the expected value error definition, defined in the context of function approximation, where μ is some distribution over states, $\hat{v}$ is the function approximator value function, w is the approximated state, and $v_\pi(s)$ is the true value of state s under the policy π:

$$\overline{VE}(\omega) \coloneqq \mu(s)\left[v_\pi(s) - \hat{v}(s, w)\right]^2$$

and recall the Q^*-preservation definition:

$$\exists f : \varphi(T) \times A \to \mathbb{R}. \|[f]_\varphi - Q_M^*\|_\infty \leq \varepsilon_{Q^*}$$

To get from Q^*-preservation to expected value error for π^*, we must: First, define Q^*-preservation with a 2-norm over outputs rather than ∞-norm (notated as $\|\cdot\|_{\infty,2}$ to show that a max is still taken over inputs) to get

$$\exists f : \varphi(T) \times A \to \mathbb{R}. \|[f]_\varphi - Q_M^*\|_{\infty,2} \leq \varepsilon_{Q^*}$$

Second, consider only state-values instead of state-action values:

$$\exists f : \varphi(T) \to \mathbb{R}. \|[f]_\varphi - V_M^*\|_{\infty,2} \leq \varepsilon_{V^*}$$

Third, take an expectation over τ rather than a maximum over τ.

G Expected Case Proofs

Here we restate Theorem 4.1 as given in the main text.

Lemma G.1. *Let μ_k^* be a k-state stochastic finite automata that will serve as a memory function in a POMDP. For any POMDP with bounded reward and for all ε, there exists a k'-DFA which achieves an expected return that is only ε less than μ_k^*. Furthermore, it is sufficient to choose $k' \geq k \ln(\varepsilon(1-\gamma)/R_{max})/\ln(\gamma)$ where R_{max} is the bound on reward and γ is the discount factor.*

This result is used for the following individual results:

- The existence of 2-stochastic Expected Return improving memory doesn't imply the existence of k-deterministic expected return improving memory.

- The existence of 2-stochastic Expected Return optimal memory doesn't imply the existence of k-deterministic expected return improving memory.

Proof. Let μ_k^* be the given k-SFA memory function with the corresponding policy π^*. Let $\hat{\mu}_{k'}$ be the k'-DFA memory function with corresponding policy $\hat{\pi}$.

For a given POMDP, let τ_t be a trajectory in the environment of states, observations, memory states, actions and rewards up to time step t where memory state m_t is being chosen. Let the observation, memory states, and rewards for a time step t be ω_t, m_t, and r_t, respectively.

For a given τ_t, we define $G_{\pi,\mu}(\tau_t)$ as the expected sum of discounted rewards for trajectories that start with τ_t and then proceed according to the policy π and memory function μ.

$$G_{\pi,\mu}(\tau_t) = \mathop{\mathbb{E}}_{\tau|\tau_t}\left[\sum_{i=t}^{\infty}\gamma^{i-t}r_i\right]$$

We then define $G_{\pi,\mu}(m_t,\tau_t)$ as the expected sum of discounted rewards for trajectories that start with τ_t, transition to memory state m_t at time step t, and then proceed according to the policy π and memory function μ.

$$G_{\pi,\mu}(m_t,\tau_t) = \sum_{\tau_{t+1}}\mathbb{P}(\tau_{t+1}|\tau_t,m_t)G_{\pi,\mu}(\tau_{t+1})$$

where $\mathbb{P}(\tau_{t+1}|\tau_t,m_t)$ is the probability of a trajectory of length $t+1$ given that it starts with trajectory τ_t of length t and that the memory state at time step t is m_t given the policy π.

Let $P(m'|m,a,\omega)$ be the probability distribution for the transitions of the memory function μ. This gives $P^*(m'|m,a,\omega)$, the probability distribution of the stochastic memory function μ_k^*, and $\hat{P}(m'|m,a,\omega)$, the probability distribution of the deterministic memory function $\hat{\mu}_{k'}$.

For any time step t we can write the expected return of the stochastic policy as:

$$G_{\pi^*,\mu_k^*} = \mathop{\mathbb{E}}_{\tau_t}\left[\sum_{m_t\in M}P^*(m_t|\omega_t,a_{t-1},m_{t-1})G_{\pi^*,\mu_k^*}(m_t,\tau_t)\right]$$

Because M is finite, there must exist a $\hat{m}_t$ such that for all possible $m_t \in M$

$$G_{\pi^*,\mu_k^*}(\hat{m}_t,\tau_t) \geq G_{\pi^*,\mu_k^*}(m_t,\tau_t)$$

We then let $\hat{P}(\hat{m}_t|\omega_t,a_{t-1},m_{t-1}) = 1$ and have $\hat{p}$ be 0 for all other m_t. This guarantees that

$$\mathop{\mathbb{E}}_{\tau_t}\left[\sum_{m_t\in M}P^*(m_t|\omega_t,a_{t-1},m_{t-1})G_{\pi^*,\mu_k^*}(m_t,\tau_t)\right] \leq \mathop{\mathbb{E}}_{\tau_t}\left[\sum_{m_t\in M}\hat{P}(m_t|\omega_t,a_{t-1},m_{t-1})G_{\pi^*,\mu_k^*}(m_t,\tau_t)\right]$$

Note that such assignment of $\hat{P}$ is equivalent to a deterministic memory function. So we have that for a particular time step t the memory state can be chosen in a deterministic way to achieve the same or better expected return when compared to choosing the memory state according to μ_k^*

The same argument can be made inductively, conditioning on a finite initial trajectory τ_{start}. We can consider longer and longer starting trajectories and in each case we can deterministically assign $\hat{P}$ to achieve the same or better expected return when compared to μ_k^*.

$$\begin{aligned}&\mathop{\mathbb{E}}_{\tau_t|\tau_{start}}\left[\sum_{m_t\in M}p^*(m_t|\omega_t,m_{t-1})G_{\pi^*,\mu_k^*}(m_t,\tau_t)\right] \leq \\ &\mathop{\mathbb{E}}_{\tau_t|\tau_{start}}\left[\sum_{m_t\in M}\hat{p}(m_t|\omega_t,m_{t-1})G_{\pi^*,\mu_k^*}(m_t,\tau_t)\right]\end{aligned} \tag{1}$$

where $E_{\tau_t|\tau_{start}}$ is the expectation over trajectories τ_t that start with τ_{start}. Importantly, this holds only for finite trajectories τ_{start}. Consider picking the memory states deterministically as described for trajectories τ_{start} of increasing length. Equation 1 will continue to hold and at some finite point the $G_{\pi^*,\mu_k^*}(m_t, \tau_t)$ term will become epsilon small due to the bounded reward. This means that a deterministic memory can achieve the same or better expected return compared to the stochastic memory function for a finite number of time steps t and after is ε close. To achieve this however, we need to distinguish identical observation, action, memory state pairs that might occur when considering trajectories of different lengths. To remedy possible conflicts that would prevent always selecting the optimal memory transitions, we can augment the memory with the current time step t.

We construct $\hat{\mu}_{k'}$ by making t copies of each memory state in μ_k^*, one for each of the first t time steps. So for a given memory state m from μ_k^* we now have m_t for each time step t. The policy $\hat{\pi}$ can be defined to return the same action as π^* for each of the t duplicates of a given memory state, ignoring the time step. We then construct $\hat{\mu}$ as described above by taking the best choice of memory state transition at each time step ensuring that $G_{\pi^*,\mu_k^*} \leq G_{\hat{\pi},\hat{\mu}_{k'}}$.

This gives us a k'-deterministic memory function with $k' = k * t$. To guarantee $G_{\pi^*,\mu_k^*} - G_{\hat{\pi},\hat{\mu}_{k'}} < \varepsilon$ for a given ε we can consider the worst case which would be a difference of R_{max} right after the first t time steps. This gives the expression $R_{\text{max}}\gamma^t(1 + \gamma + \gamma^2 + \dots) \leq \varepsilon$ which means it is sufficient to take t greater than $\ln(\varepsilon(1-\gamma)/R_{\text{max}})/\ln(\gamma)$. This works because once the deterministic memory function matches the performance of the stochastic memory function for all trajectories of a sufficiently large finite length, all further rewards are negligibly small due to the discount factor γ.

□

H Expected Case Counterexamples

Example H.1. The following counter example is for the following results:

- The existence of k-deterministic expected π^* optimal memory doesn't imply the existence of 2-stochastic π^* optimal memory
- The existence of k-deterministic expected π^* improving memory doesn't imply the existence of 2-stochastic π^* improving memory

Consider an environment where the agent is first shown an integer observation ω_i between 1 and k, and then a recall observation, ω_{recall}. At the recall observation, the agent can either take the a_{exit} action to receive a reward of 0, or an action $a_1, a_2, a_3, \dots a_k$ corresponding to one of the possible observations it received. If the agent selects the correct action it receives a reward of 1 and otherwise $-k$.

An agent with k memory states can update the memory state based on the first observation ω_i, $\mu(\cdot, \omega_i, \cdot) = m_i$. We then have the policy $\pi(m_i, \omega_{\text{recall}}) = a_i$ which achieves an expected return of 1.

Now, consider an agent with 2 stochastic memory states, $M = \{m_1, m_2\}$, which doesn't take the exit action. We have two steps that occur probabilistically, the selection of the memory state and the selection of the action. We can write the probability of a particular action in terms of the initial observation as $P(a_i|o_j)$. When $i = j$ the agent took the correct action and gets a reward of 1 and otherwise, it gets a reward of $-k$. This means we can write the expected return as

$$\begin{aligned}\text{Expected Return} &= \frac{1}{k}\sum_{i=1}^{k} 1 * P(a_i|\omega_i) - k(k-1)(1 - P(a_i|\omega_i)) = \\ &= \frac{1}{k}\sum_{i=1}^{k} P(a_i|\omega_i)(1 + k(k-1)) - k(k-1) \\ &= -k(k-1) + \frac{1 + k(k-1)}{k}\sum_{i=1}^{k} P(a_i|\omega_i)\end{aligned}$$

We have that $P(a_i|\omega_i) = P(a_i|m_1)P(m_1|\omega_i) + P(a_i|m_2)P(m_2|\omega_i)$ and because $\sum_{i=1}^{k} P(a_i|m) = 1$ for all m we can bound $\sum_{i=1}^{k} P(a_i|\omega_i) = \sum_{i=1}^{k} P(a_i|m_1)P(m_1|\omega_i) + P(a_i|m_2)P(m_2|\omega_i) \leq max_i P(m_1|\omega_i) + P(m_2|\omega_i) \leq 2$.

For $k \geq 3$ this gives:

$$\begin{aligned}\text{Expected Return} &\leq -k(k-1) + \frac{1 + k(k-1)}{k} 2 \\ &\leq -k(k-1) + \frac{1}{k} + 2(k-1) \\ &\leq \frac{1}{k} + (2-k)(k-1) \leq 0\end{aligned}$$

This means that taking the exit action a_{exit} is always optimal for the 2 stochastic memory function and this matches the expected return of no-memory.

This counter example shows that the existence of k-deterministic memory doesn't imply the existence of 2 stochastic. By set inclusion this also gives us that k-stochastic memory doesn't imply 2-stochastic memory, k-deterministic memory doesn't imply 2-stochastic memory, and k-stochastic memory doesn't imply 2-deterministic memory. Because we have that the k-deterministic memory is optimal and the 2-stochastic memory is not improving we have that this example extends to both the optimal and improving tables.