Timing as an Action: Learning When to Observe and Act

Helen Zhou* CMU
Audrey Huang* UIUC
Kamyar Azizzadenesheli NVIDIA
David Childers CMU
Zachary C. Lipton CMU

Abstract

In standard reinforcement learning setups, the agent receives observations and performs actions at evenly spaced intervals. However, in many real-world settings, observations are expensive, forcing agents to commit to courses of action for designated periods of time. Consider that doctors, after each visit, typically set not only a treatment plan but also a follow-up date at which that plan might be revised. In this work, we formalize the setup of *timing-as-an-action*. Through theoretical analysis in the tabular setting, we show that while the choice of delay intervals could be naively folded in as part of a composite action, these actions have a special structure and handling them intelligently yields statistical advantages. Taking a model-based perspective, these gains owe to the fact that delay actions do not add any parameters to the underlying model. For model estimation, we provide provable sample-efficiency improvements, and our experiments demonstrate empirical improvements in both healthcare simulators and classical reinforcement learning environments.

1 INTRODUCTION

In the real-world, decisions are often spread across irregular intervals of time. After each visit, doctors must commit to not only a course of treatment but also to a follow up plan. Each office visit offers an opportunity to gain fresh information and course correct if the current treatment regime is unsuccessful. On the other hand, excessive visits are expensive, consuming hospital resources and consuming time that could be spent on patients in greater need. Thus doctors must trade off the value of information gained the cost of more frequent opportunities to observe and intervene. Similarly, research advisors must decide not only how to advise students in each meeting, but also how frequently to schedule these touchpoints. Economists have considered related scenarios where firms incur a cost for observing market state and must set pricing policies that will hold in between observations (Mankiw and Reis, 2002; Stokey, 2008). When the state is fully unobserved between actions, agents must anticipate both the state and the speed at which their information of the state will go stale in order to choose both action and time to next observation.

Several works in disease progression modeling have applied multi-state models, which assign probabilities or intensities to transitions between different discrete states, to capture state transitions across periods of non-observation (Jackson, 2011; Young et al., 2020; Lorenzi et al., 2019; Cheung et al., 2022). For cardiovascular disease, Lindbohm et al. (2019) utilized multi-state Markov models to estimate rates of progression for different risk groups, demonstrating how different screening intervals can lead to different trade-offs between cost and quality-adjusted life years. Breast cancer screening has been the subject of substantial controversy (Esserman, 2017; Marmot et al., 2013), with different organizations recommending different screening policies (Ren et al., 2022). However, the prior literature leaves open the question of how a reinforcement learner ought to go about learning a joint policy over actions and observation intervals.

In this work, we explore how one might *learn* policies in an action space augmented by the choice of when to observe and take the next action. We show that this setting is amenable to standard model-free and model-based reinforcement learning algorithms in this augmented action space, but also propose a new *timing-aware* model-based approach which can leverage the temporal nature of the timing action. We prove theoretically that the timing-aware algorithm has improved sample efficiency compared to the aforementioned standard approaches, which arises from more efficient model estimation, and empirically characterize the estimation

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

error rates of timing-aware, timing-naive, and model-free strategies under various quantities of samples and exploration policies, showing that the timing-aware strategy is able to consistently achieve the lowest estimation error with fewer samples. In the disease progression, windy gridworld, and glucose reinforcement learning environments, we demonstrate empirically that timing-aware learning consistently achieves the lowest estimation error the quickest, and is also able to achieve the highest average cumulative reward. At the same time, we empirically find that low estimation error is not always necessary for good performance as measured by average cumulative reward. Finally, we release our timing-as-an-action simulators to encourage further model and algorithmic development in this setting.

2 TIMING-AS-AN-ACTION

Consider the motivating setting where a patient with a chronic illness visits a doctor, who prescribes them a daily medication and schedules a follow-up appointment. We design the timing-as-an-action problem setting to mimic this interaction dynamic, where importantly, (1) the doctor must choose not only which treatment (*action*) to recommend but also how long (*delay*) to recommend it for, (2) the doctor *does not observe* the patient's intermediate state or the benefit of the medication until the next appointment (no observations of state or reward until after the delay), (3) there is some *cost* to each appointment (observation and action cost). With these characteristics in mind, we define the timing-as-an-action Markov decision process (MDP) and reinforcement learning (RL) setup.

Timing-as-an-action MDP The *timing-as-an-action Markov decision process* is an infinite-horizon MDP defined by the tuple $M = (\mathcal{S}, \mathcal{A}, \mathcal{K}, P, R, \gamma, C, s_0)$, with state space $\mathcal{S}$, action space $\mathcal{A}$, and *delay space* $\mathcal{K} = \{1, 2, ..., K\}$, where $K \in \mathbb{N}$. The delay space $\mathcal{K}$ represents the set of numbers of timesteps for which an action can be repeated, and a policy in the timing-as-an-action MDP must make decisions over both actions and how long to take them for (the delay), i.e., $\pi : \mathcal{S} \to \Delta(\mathcal{A} \times \mathcal{K})$, where Δ indicates the probability simplex. Note that with different choices of k, the resulting sequence of observations will be unevenly spaced in terms of the underlying timestep, which we will refer to as the *primitive* timestep. The underlying one-step reward function, or *primitive reward function* $R : \mathcal{S} \times \mathcal{A} \to [0, 1]$, is assumed to be deterministic and in a bounded nonnegative interval. Additionally, there is a discount factor $\gamma \in [0, 1)$; a fixed, known interaction cost $C \in \mathbb{R}_{\geq 0}$; and a deterministic starting state s_0. The k-step transition probabilities are given by $P : \mathcal{S} \times \mathcal{A} \times \mathcal{K} \to \Delta(\mathcal{S})$, where Δ is the probability simplex, and $P(s'|s, a, k)$ denotes the probability of transitioning to a next state s' after taking action a for k timesteps from a state s.

Importantly, the true transition probabilities P have the structure that the k-step transitions are induced by the 1-step transitions. Before formalizing this property we introduce some additional notation. For any valid transition P', let $P'_{a,k}(s'|s) := P'(s'|s, a, k)$ for short, and let the bolded version $\mathbf{P'}$ denote the corresponding $A \times K \times S \times S$ tensor, where indexing into the tensor is denoted as $\mathbf{P'}[a, k, s, s'] := P'(s'|s, a, k)$, and we also denote $\mathbf{P'}_{a,k} := \mathbf{P'}[a, k, :, :]$ and $\mathbf{P'}_{a,k}(s'|s) := \mathbf{P'}_{a,k}[s, s']$. In the timing-as-an-action MDP, we have $\mathbf{P}_{a,k} = \mathbf{P}^k_{a,1}$ for all $(a, k) \in \mathcal{A} \times \mathcal{K}$, which refers to the one-step transition probability matrix multiplied by itself k times.

Timing-as-an-action RL setup In the *timing-as-an-action RL setup*, the agent alternately observes a state s, chooses an action a to commit to, as well as a delay k, that corresponds to the number of timesteps the action a is played for. The agent then observes state s' as well as the aggregated k-step reward g,

$$g = -C + \textstyle\sum_{j=0}^{k-1} \gamma^j r_j, \tag{1}$$

that is the discounted sum of the (unobserved) one-step rewards encountered along the k steps of taking action a, from which C, the cost of interaction, is subtracted. For clarity, one step of an agent's interaction with the environment `env` $= M$, i.e. calling `s', g = env.step(a,k)`, is summarized below (ignoring termination conditions and `done`'s for simplicity):

> **Timing-as-an-action** `env.step(a,k)`
>
> **Given**: `env` $= M$, current state s.
>
> Initialize $s_0 = s$. For $j = 0, \ldots, k-1$:
>
> 1. Sample $r_j \sim R(s_j, a)$
> 2. Transition to $s_{j+1} \sim P(\cdot|s_j, a, 1)$
>
> **Out:** aggregate reward $g = -C + \sum_{j=0}^{k-1} \gamma^j r_j$; next state $s' = s_k$

Crucially, the intermediate states $(s_1, \ldots, s_{k-1})$ and intermediate one-step rewards $(r_1, \ldots, r_{k-1})$ *are not* observed—only s_k and the aggregate rewards g (defined above) are. This captures the challenges of learning when to observe, core to healthcare applications as discussed previously, and distinguishes our problem setting from the "standard" RL setup.

Value functions We call $G : \mathcal{S} \times \mathcal{A} \times \mathcal{K} \to \Delta([-C, \frac{1}{1-\gamma} - C])$ the distribution over aggregated rewards induced by R and P, i.e., $g \sim G(s, a, k)$, with expected value

$$\mathbb{E}[G(s, a, k)] = -C + \textstyle\sum_{j=0}^{k-1} \gamma^j \mathbb{E}[R(s_j, a)|s, a].$$

When a policy π interacts continuously with M, it observes a trajectory $(s_0, a_0, g_0, s_1, a_1, g_1, \ldots)$, and its state-action value function of policy π is the expectation of its total discounted return over the infinite horizon of interaction:

$$Q^\pi(s,a,k) = \mathbb{E}\left[\sum_{\tau=0}^{\infty} \gamma^{(\sum_{\tau'=0}^{\tau-1} k_{\tau'})} g_\tau | \pi, s_0 = s, a_0 = a, k_0 = k\right]$$

and its state value function is $V^\pi(s) = \mathbb{E}_{a,k\sim\pi(\cdot|s)}[Q^\pi(s,a)]$. Lastly, the goal of the *timing-as-an-action reinforcement learning problem* is to find the optimal policy $\pi^* = \text{argmax}_{\pi:\mathcal{S}\to\Delta(\mathcal{A}\times\mathcal{K})} V^\pi(s_0)$ that learns which actions *and* delays to take in order to maximize its total discounted return. We define the optimal value function $Q^* := Q^{\pi^*}$, and same for V^*.

3 RELATED WORK

The timing-as-an-action framework is most closely related to the options and hierarchical RL literatures, but there are key differences, described shortly, that make them very different learning problems. Broadly, an option is a pre-defined, temporally extended sequence of actions. An MDP endowed with a set of options is called a semi-MDP, and the agent's policy chooses options to take. The options framework has been commonly used for reasoning at different levels of temporal abstraction (Sutton et al., 1998, 1999; Bacon et al., 2017; Machado et al., 2023). Options belong to a class of reinforcement learning (RL) approaches called hierarchical RL, which involves decomposing a task into subtasks at varying levels of granularity (Barto and Mahadevan, 2003; Dietterich, 2000; Vezhnevets et al., 2017; Co-Reyes et al., 2018; Eysenbach et al., 2019; Hafner et al., 2022).

The key difference between the timing-as-an-action framework and semi-MDPs or hierarchical RL is that the latter frameworks generally assume that per-step observations and rewards are available to the learner, and subtasks are often accomplished with a combination of different granular actions. In contrast, our work utilizes repeated actions (to reflect, e.g., a patient following a treatment plan), and does not assume access to per-step observations or rewards, but rather the aggregated reward and final observation after the chosen duration for the action has passed (e.g. when the patient comes back for a follow-up visit). Semi-MDP methods which rely on per-step rewards and observations are thus not applicable.

Model-based RL has offered a sample-efficient approach for settings where interactions may be expensive to collect (Kaelbling et al., 1996; Deisenroth et al., 2011; Sutton and Barto, 2018). Motivated by human cognition, Ha and Schmidhuber (2018) proposed a model-based framework that learns an autoencoder vision network and recurrent neural memory network to represent the environment dynamics. In a "dream" world simulated by these learned networks, a small controller network is trained. For continuous-time domains with irregularly observed data, Du et al. (2020) use neural ordinary differential equations for model-based reinforcement learning in semi-Markov decision processes. However, as far as we are aware, none of these setups assign a cost to observing and acting, and proactively jointly optimize for the next choice of delay and action.

Repetition of actions has been found to be useful for improving exploration in simple classical RL environments (Dabney et al., 2020) as well as gaming environments such as Atari (Braylan et al., 2015) and VizDoom (Khan et al., 2019), where skipping frames can lead to improvements in learning speed and final performance. Prior work on learning action repetitions has used a Q-network with multiple output heads per action for different repetition lengths (Lakshminarayanan et al., 2017), a framework that jointly learns an action policy and a second policy that decides how often to repeat (Sharma et al., 2017), and using all pairs of intermediate observations to learn the values of multi-step actions (Biedenkapp et al., 2021). However, these works assume access to intermediate observations and rewards.

4 METHODS

4.1 Timing-as-an-action Bellman Backup

To facilitate planning in the timing-as-an-action MDP, we begin with defining the following timing-as-an-action Bellman optimality equation, that recursively relates the value function to itself. For any (s,a,k),

$$Q(s,a,k) = \mathbb{E}[G(s,a,k)] + \gamma^k \mathbb{E}_{s'\sim P(\cdot|s,a,k)}[\max_{a',k'} Q(s',a',k')] \quad (2)$$

Such recursive equations are the backbone of value-based RL methods (Agarwal et al., 2019), that optimize policies from learned value functions. In particular, finding a value function that satisfies (2) for all (s,a,k) implies that we have obtained the optimal value function Q^* (see Appendix A.1 for proof).

Lemma 1. *The timing-as-an-action Bellman optimality equation (2) has a unique fixed point for Q^*.*

As Lemma 1 is analogous to well-established results for value-based learning in standard MDPs, the immediate implication is that one could solve the timing-as-an-action RL problem by applying standard value-based RL algorithms (e.g., Q-learning), with an expanded

action space equal to the cross product of actions and delays $\mathcal{A}' = \mathcal{A} \times \mathcal{K}$. Indeed, we will show that this is the case in Section 4.2.1.

However, it should also be immediately clear that such methods will be sample-inefficient because they do not leverage the structure of the timing-as-an-action MDP, namely, that observing k-step transitions also provides information about the dynamics for $k' \neq k$. In general, the sample complexity of RL algorithms depends on the size of the action space (Azar et al., 2012; Agarwal et al., 2019), here $|\mathcal{A}'| = |\mathcal{A}||\mathcal{K}|$. This can grow rapidly depending on the choice of delay space $\mathcal{K}$, which, in general, we expect to be relatively large as it represents discretized time. For example, if one chose delays up to one day with one-minute intervals between them, the action space would be 1,440 times as large as the single-timestep action space. Ideally, leveraging the temporal nature of the timing action should result in more efficient learning.

4.2 Learning Algorithms

We define three value-based RL algorithms based on the timing-as-an-action Bellman backup (2). Two approaches, one model-free and one model-based, give a naive treatment of the delay by treating it as any other action, and can be viewed as standard RL algorithms translated directly to the timing-as-an-action setup. The third approach is also model-based, but leverages the temporal nature of the delay, i.e, that $\mathbf{P}_{a,k} = \mathbf{P}_{a,1}^k$, to share information between different values of delays. As an extreme example, obtaining perfect 1-step transitions automatically translates to perfect estimation of $P_{a,k}$ for all $k \in K$; more generally, observations of any delay allows for reasoning about the transitions for other delays. Because the delay structure is embedded in the transitions, model-based methods are a natural choice for leveraging this structure (which is not encoded in the Q-values).

4.2.1 Model-free

After taking action a for k steps from state s, the environment returns an aggregated reward g and the next state s'. The *model-free* approach updates the action-values using the standard Q-learning update (Watkins and Dayan, 1992):

$$\widehat{Q}(s,a,k) \leftarrow g + \gamma^k \max_{a',k'} \widehat{Q}(s',a',k'). \tag{3}$$

Here the action space is simply the cross product of all actions and delays $(a,k) \in \mathcal{A} \times \mathcal{K}$, and a sample of experience with delay k does not inform the values associated with $k-1$, $k+1$, etc. To help improve sample efficiency, we add experience replay, iterating through tuples (s,a,k,g,s') and updating using (3) (details in Algorithm 1). To further improve sample efficiency, we consider model-based methods.

Algorithm 1 Model-free learning procedure

Input: MDP M, environment env to interact with M, policy transformation $\pi_Q : Q \to \Delta(\mathcal{A} \times \mathcal{K})^S$
Initialize replay buffer $B = \emptyset$ and Q-values $\widehat{Q}(s,a,k) = 0 \; \forall s,a,k \in \mathcal{S} \times \mathcal{A} \times \mathcal{K}$.
for each episode **do**
  s, done = env.reset(), False
  while not done **do**
    $a, k \sim \pi_{\widehat{Q}}(\cdot,\cdot|s)$
    s', g, done = env.step(a,k)
    Append $B \leftarrow B \cup \{(s,a,k,s',g)\}$
    for $(s,a,k,g,s') \in B$ **do**
      Update $\widehat{Q}$ via (3):

$$\widehat{Q}(s,a,k) \leftarrow g + \gamma^k \max_{a',k'} \widehat{Q}(s',a',k').$$

    end for
  end while
end for

4.2.2 Model-based

In the model-based approaches, we learn models of the transition probabilities $\widehat{P}$ and one-step rewards $\widehat{R}$ using a dataset of the form $\{(s_i,a_i,k_i,g_i,s_i')\}_{i=1}^N$, which are then used to obtain the Q-value estimates $\widehat{Q}$.

For the *timing-naive model-based* approach, the transitions are learned through maximum likelihood estimation from the function class $\mathcal{P}$:

$$\widehat{P} = \operatorname*{argmax}_{p \in \mathcal{P}} \sum_{i=1}^{N} \log p_{a_i,k_i}(s_i'|s_i), \tag{4}$$

where $\mathcal{P} = \{P : \mathcal{S} \times \mathcal{A} \times \mathcal{K} \to \Delta(\mathcal{S})\}$ is the set of all valid transitions (involving actions and delays). For the *timing-aware model-based* approach,

$$\widehat{P} = \operatorname*{argmax}_{p \in \mathcal{P}_1} \sum_{i=1}^{n} \log \left(p_{a_i,1}^{k_i}(s_i'|s_i) \right). \tag{5}$$

where $\mathcal{P}_1 = \{p : p_{a,k}(\cdot|s) = [\mathbf{p}_{a,1}]^k(\cdot|s), \forall (s,a,k) \in \mathcal{S} \times \mathcal{A} \times \mathcal{K}\}$, recalling that $\mathbf{p}$ is the tensor version of p thus $[\mathbf{p}_{a,1}]^k$ is the $S \times S$ one-step transition probability matrix multiplied by itself k times. Note that $\mathcal{P}_1 \subseteq \mathcal{P}$ from (4), and the true transitions $P \in \mathcal{P}_1$ given the structure the environment.

Then, given an estimate of the transition probabilities $\widehat{P}$ (from either (4) or (5)), estimates of the one-step reward function $\widehat{R}$ are obtained as follows:

$$\widehat{R} = \operatorname*{argmin}_{R' \in \mathcal{R}} \frac{1}{N} \sum_{i=1}^{N} (\mathcal{G}_{R',\widehat{P}}(s_i,a_i,k_i) - g_i)^2, \tag{6}$$

Algorithm 2 Model-based learning procedure

Given: MDP M, policy transformation $\pi_Q : Q \to \Delta(\mathcal{A} \times \mathcal{K})^S$, $\mathcal{P}' = \mathcal{P}$ in the timing-naive approach and $\mathcal{P}' = \mathcal{P}_1\}$ in the timing-naive approach.
Initialize replay buffer $B = \emptyset$, Q-values $\widehat{Q} = 0$, transitions $\widehat{P}$, and one-step rewards $\widehat{r}$.
for each episode **do**
  $s, \text{done} = \text{env.reset}(), \text{False}$
  while not done **do**
    $a, k \sim \pi_Q(\cdot, \cdot|s)$
    $s', g, \text{done} = \text{env.step}(a, k)$
    Append $B \leftarrow B \cup \{(s, a, k, s', g)\}$
    for $(s, a, k, g, s') \in B$ **do**
      Update $\widehat{P}$ using (4) (timing-naive):

$$\widehat{P} = \operatorname*{argmax}_{p \in \mathcal{P}} \sum_{i=1}^{n} \log p_{a_i,k_i}(s_i'|s_i)$$

      or (5) (timing-aware):

$$\widehat{P} = \operatorname*{argmax}_{p \in \mathcal{P}} \sum_{i=1}^{n} \log \left(p^{k_i}_{a_i,1}(s_i'|s_i) \right).$$

      Update $\widehat{R}$ using (6). $\mathcal{G}_{R',P'}$ is defined in (7):

$$\widehat{R} = \operatorname*{argmin}_{R' \in \mathcal{R}} \frac{1}{N} \sum_{i=1}^{N} (\mathcal{G}_{R',\widehat{P}}(s_i, a_i, k_i) - g_i)^2.$$

      Update $\widehat{Q} \leftarrow \text{value\_iteration}(\widehat{P}, \mathcal{G}_{\widehat{R},\widehat{P}})$.
    end for
  end while
end for

where $\mathcal{R}$ is a one-step reward function class and $\mathcal{G}$ is a deterministic mapping from a one-step reward function R' and transition P' to the corresponding expected aggregated multi-step reward,

$$\mathcal{G}_{R',P'}(s, a, k) = -C + \sum_{\tau=0}^{k-1} \gamma^\tau \mathbb{E}_{s' \sim P'_{a,k}(\cdot|s)}[R'(s', a)]. \quad (7)$$

Note that by plugging in the true R and P, we have $\mathcal{G}_{R,P}(s, a, k) = \mathbb{E}[G(s, a, k)]$. The $\widehat{P}$ and corresponding $\widehat{R}$ are then used to learn the Q-value functions via value iteration, i.e., by finding $\widehat{Q}$ that is the fixed point of the Bellman equation involving the estimated $\mathcal{G}_{\widehat{R},\widehat{P}}$ and $\widehat{P}$ below:

$$\widehat{Q}(s, a, k) = \mathcal{G}_{\widehat{R},\widehat{P}}(s, a, k) + \gamma^k \mathbb{E}_{s' \sim \widehat{P}_{a,k}(\cdot|s)}[\max_{a',k'} \widehat{Q}(s', a', k')]. \quad (8)$$

4.3 Analysis

To highlight the sample complexity improvements in model learning achieved by (5), we provide our guarantees in the generative setting, which isolates the estimation problem from the challenges of exploration in RL (Azar et al., 2012; Agarwal et al., 2019):

Definition 1 (Generative Setting)**.** *A generative model takes as input (s, a, k) and outputs $s' \sim P(\cdot|s, a, k)$. In the generative setting, we obtain n samples of s' from each $(s, a, k) \in \mathcal{S} \times \mathcal{A} \times K$ using the generative model, i.e. $\mathcal{D}_{s,a,k} = \{(s, a, k, s'_i)\}_{i=1}^n$.*

Proofs for the below are provided in Appendix A.2, in addition to the more general versions for the non-generative setting:

Lemma 2. *In the generative setting (Definition 1), for $\widehat{P}$ from (4), with probability $\geq 1 - \delta$,*

$$\max_{s,a,k} \|\widehat{P}_{a,k}(\cdot|s) - P_{a,k}(\cdot|s)\|_1 \lesssim S\sqrt{\frac{AK \log(1/\delta)}{n}}.$$

Lemma 3. *In the generative setting (Definition 1), for $\widehat{P}$ from (5), with probability $\geq 1 - \delta$,*

$$\max_{s,a,k} \|\widehat{P}_{a,k}(\cdot|s) - P_{a,k}(\cdot|s)\|_1 \lesssim S\sqrt{\frac{A \log(K/\delta)}{n}}.$$

Comparison of the above transition estimation results reveals the sample complexity gains from (5) are obtained by leveraging the delay structure, as evidenced by their respective dependencies on K. While the timing-naive estimate has $\sqrt{K}$ in its upper bound (Lemma 2), the timing-smart estimate obtains $\log K$ (Lemma 3), and this is because learning just the 1-step transitions from all samples is more efficient, while still being sufficient to express all k-step transitions.

Lemma 4. *Fix $\widehat{P}$ and let $\varepsilon_{\widehat{P}} = \max_{s,a,k} \left\| \widehat{P}(\cdot|s, a, k) - P(\cdot|s, a, k) \right\|_1$. Then in the generative setting (Definition 1), with probability $\geq 1 - \delta$ we have*

$$\left\| \mathcal{G}_{\widehat{R},\widehat{P}} - \mathcal{G}_{R,P} \right\|_\infty \lesssim \frac{1}{(1-\gamma)} (SAK)^{1/2} \varepsilon_{\widehat{P}} + \left(\frac{G_{\max}^2 S^2 A^2 K}{n} \right)^{1/2} + \left(\frac{1}{(1-\gamma)^2} \frac{G_{\max}^2 S^2 A^2 K}{n} \varepsilon_{\widehat{P}}^2 \right)^{1/4}$$

where $G_{\max} = \max\{C, |\frac{1}{1-\gamma} - C|\}$.

Lemma 4 demonstrates that the error of aggregated reward estimation is directly related to the error of transition estimation through $\varepsilon_{\widehat{P}}$; better transition estimation (smaller $\varepsilon_{\widehat{P}}$) translates to faster reward convergence. For $\widehat{P}$ used in the timing-aware or timing-naive model updates, $\varepsilon_{\widehat{P}}$ is given by the bounds in

Lemma 3 and Lemma 2, respectively, both of which are $n^{-1/2}$ thus endowing the RHS of Lemma 4 with a fast $n^{-1/2}$ rate of estimation, with $\mathcal{G}_{\widehat{R},\widehat{P}} \to \mathcal{G}_{R,P}$ as $n \to \infty$. For example, Lemma 4 is $\tilde{O}(SA^{3/4}K^{1/2}n^{-1/2})$ for the timing-aware model-based approach in (5) since $\varepsilon_{\widehat{P}} = \tilde{O}(SA^{1/2}\log K n^{-1/2})$ from Lemma 3.[1] The sample complexity gains in timing-aware model estimation thus translate to reward estimation as well.

As $\widehat{Q}$ is formed directly from $\widehat{P}$ and $\mathcal{G}_{\widehat{R},\widehat{P}}$ in the model-based update (8), the Q-value estimate directly inherits the quality of the $\widehat{P}$ and $\mathcal{G}_{\widehat{R},\widehat{P}}$ estimates. In other words, the better the transition estimate, the better $\widehat{Q}$ approximates Q^*, and the better the downstream learned policy is. This is formalized below.

Proposition 1. *For any $\widehat{P}$ and $\mathcal{G}_{\widehat{R},\widehat{P}}$, let $\widehat{Q}$ satisfy (8). Let $\pi_{\widehat{Q}}$ be the greedy policy with respect to $\widehat{Q}$, i.e., $\pi_{\widehat{Q}}(a,k|s) = \mathbf{1}[a = \text{argmax}_{a',k'} \widehat{Q}(s,a',k')]$. Then*

$$\|Q^* - Q^{\pi_{\widehat{Q}}}\|_\infty \le \frac{2}{1-\gamma}\left\|\mathcal{G}_{R,P} - \mathcal{G}_{\widehat{R},\widehat{P}}\right\|_\infty + \frac{2\gamma}{(1-\gamma)^2}\max_{s,a,k}\left\|P(\cdot|s,a,k) - \widehat{P}(\cdot|s,a,k)\right\|_1.$$

5 EXPERIMENTS

Our experiments investigate the model estimation problem (Section 5.1) separately from the policy learning problem (Section 5.2).

Implementation Details All models are implemented using PyTorch, with transition probabilities $\widehat{P}$ initialized uniformly, and single-step reward estimates $\widehat{R}$ initialized to -1. For the timing-aware model, the estimate of the one-step transition matrix for action a is $\widehat{P}_{a,1} = \text{softmax}(\mathbf{T}[a,:,:])$, where $\mathbf{T}$ is an unconstrained $A \times S \times S$ parameter tensor initialized with ones, and the softmax is over the last dimension. For the timing-naive model, the estimate of the k-step transition matrix for action a is $\widehat{P}_{a,k} = \text{softmax}(\mathbf{T}'[a,k,:,:])$, where $\mathbf{T}'$ is an unconstrained $A \times K \times S \times S$ parameter tensor initialized with ones, and the softmax is over the last dimension. The single-step rewards $\widehat{R}$ are initialized as an $A \times S$ tensor, and estimates of the expected aggregate reward are computed using $\widehat{g} = \mathcal{G}_{\widehat{R},\widehat{P}}(s,a,k)$, defined in (7). Additional optimization details are in Appendix B.

[1] While we obtain the desired $\log K$ dependence in our timing-aware transition estimation bound, this becomes a looser $K^{1/2}$ factor in the reward estimation result, which we believe is an artifact of the analysis that arises from translating between norms. However, this inflation applies to any plug-in transition estimation so our argument for sample efficiency gains still applies.

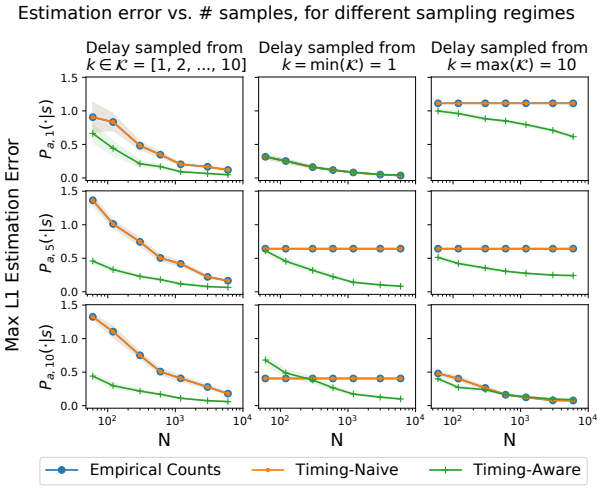

Figure 1: Maximum L1 estimation error $\max_{a,s} \|P_{a,k}(\cdot|s) - \widehat{P}_{a,k}(\cdot|s)\|_1$ (with 95% CI) for $\widehat{P}_{a,1}$, $\widehat{P}_{a,5}$, and $\widehat{P}_{a,10}$ vs. N, the # of samples generated from three sampling regimes: (a) the generative setting of Definition 1, (b) only sampling $k = \min(\mathcal{K})$, and (c) only sampling $k = \max(\mathcal{K})$.

5.1 Transition Model Estimation

As the likelihood objective in the timing-aware model-based approach (5) may be non-convex in the one-step parameters $p_{a,1}$, we first verify empirically that standard gradient-based optimization methods can learn $\widehat{P} \approx P$ in Figure 3. To better compare the rates of learning the transition probabilities in the timing-aware and timing-naive approaches, we examine the L1 error curves for three sampling regimes: (a) drawing samples in the generative setting (Definition 1), (b) drawing an equivalent number of samples selecting actions uniformly with just the minimum delay, and (c) drawing an equivalent number of samples selecting actions uniformly with just the maximum delay. For (a), we draw $n = [1, 2, 5, 10, 20, 50, 100]$ per-(s,a,k) samples $\forall (s,a,k) \in \mathcal{S} \times \mathcal{A} \times \mathcal{K}$, giving $N = nSAK = [60, 120, ..., 6000]$ samples to estimate $\widehat{P}$. For (b) and (c) which sample just one value of k, we draw ten times as many per-(s,a,k) samples to obtain the same number of samples N. True transition probabilities P come from the disease progression environment, where $S = 3, A = 2$, and $K = 10$. We also sanity-check against the estimate of P from empirically counting transitions to each state given each state and action. Results are averaged over 30 trials.

When all delays are sampled, the timing-aware model-based approach achieves lower estimation error faster than the timing-naive model-based approach and empirical counts. In the generative setting (Definition 1), for example, it only takes the timing-aware approach $n = 20$ draws of all $(s,a,k) \in \mathcal{S} \times \mathcal{A} \times \mathcal{K}$ to achieve

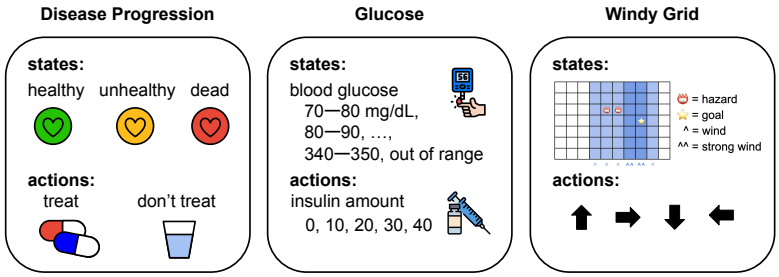

Figure 2: Summary of disease, glucose, and windy grid environments. Additional details in Appendix C.

Table 1: Final average cumulative reward after 200 episodes. Values are written in the hundreds.

	Disease Progression	Windy Grid	Glucose
Timing-Aware	**4.26 (4.00–4.53)**	**81.5 (80.3–82.6)**	**0.420 (0.287–0.554)**
Timing-Naive	**4.24 (4.00–4.47)**	76.9 (75.0–78.8)	0.334 (0.224–0.443)
Model-Free	3.47 (3.28–3.65)	3.96 (1.83–6.10)	0.270 (0.183–0.356)

a maximum L1 error less than 0.1 in the estimate of the one-step transition probabilities, whereas it takes timing-naive approach more than $n = 100$ per-(s, a, k) draws to achieve the same. While in the timing-naive approach each sample only contributes information towards the corresponding entry with the same delay, in the timing-aware approach each sample contributes to the estimates of transition probabilities of all other delays. This is further demonstrated in the second and third columns of Figure 3, where when only one delay is sampled, the timing-naive approach only improves its estimate with samples of the same delay, and the estimates for the other delays remain unchanged.

5.2 Reinforcement Learning Setting

For $\pi_{\widehat{Q}}$, we use an ϵ-greedy policy with $\epsilon = 0.1$, where with probability $1 - \epsilon$ the delay and action are $\arg\max_{a',k'} \widehat{Q}(s, a', k')$, and otherwise the delay is $k = 1$ and the action is drawn uniformly from $\mathcal{A}$. Each experiment has 200 episodes (limited to mimic real-world situations with limited data), and 50 trials of each experiment are run. We explore three environments of increasing size of state and action space: (1) a three-state disease progression simulator, (2) a glucose simulator, and (3) a windygrid environment (Figure 2). We implement an augmented version of all three simulators which accepts both the action a and delay k, and returns the aggregated rewards g and state s' after having taken k steps with action a (see box in Section 2). Code for these simulators is in the supplement.

Progression Environment This simulator is an environment with three states (healthy, unhealthy, dead) and two actions (treat and do not treat). We consider delays of up to ten timesteps, $k \in \mathcal{K} = [1, 2, ..., 10]$. The simulator is based on models commonly used in disease progression modeling, such as multi-state Markov models used in breast cancer progression modeling (Yen et al., 2003; Olsen et al., 2006; Chen et al., 1996; Duffy et al., 1995). The true transition probabilities P are included in Appendix C. The healthy state has a reward of 25, the unhealthy state has a reward of 5, and the dead state terminates the episode and has as reward of 0. The action cost is $C = 5$, and $\gamma = 0.99$.

Glucose Environment We implement an augmented version of the SimGlucose simulator (Xie, 2018), with 29 states corresponding to ranges of blood glucose measurements, and five actions corresponding to different insulin amounts. We consider delays $\mathcal{K} = [1, 2, 3, 4]$. The reward and transition probabilities are not defined explicitly, but rather according to dynamics in Clarke and Kovatchev (2009); Xie (2018); Man et al. (2014). The action cost is $C = 0.5$, and $\gamma = 0.99$.

Windy Grid Environment The windy grid simulator is a classic RL environment (Sutton and Barto, 2018), consisting of a 7×10 grid with 70 states, and four actions (up, down, left, right). We consider a delay space $\mathcal{K} = [1, 2, ..., 10]$. The agent starts at (3, 0), and the goal state is at (3, 7). With probability 0.5, wind (columns 4–6 and 9) pushes the agent up one space, and strong wind (columns 7 and 8) pushes the agent up two spaces. Except for states with wind, the actions produce the expected transition to adjacent states with probability 1. In row 3, columns 5 and 6 have hazards which have a negative reward, -5. The goal state has a reward of 10,000, and the remaining states have a reward of -1. Upon reaching the goal state, the episode terminates. The action cost is $C = 1$, and $\gamma = 0.99$.

RL Results Across episodes, the timing-aware ap-

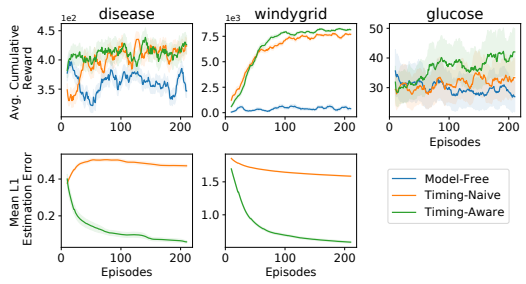

Figure 3: Average cumulative reward and mean L1 error ($\|\widehat{P}_{a,k}(\cdot|s) - P_{a,k}(\cdot|s)\|_1$ averaged over all s, a, k) across 50 trials, smoothed with a running average over 20 episodes. Shaded regions are the standard errors.

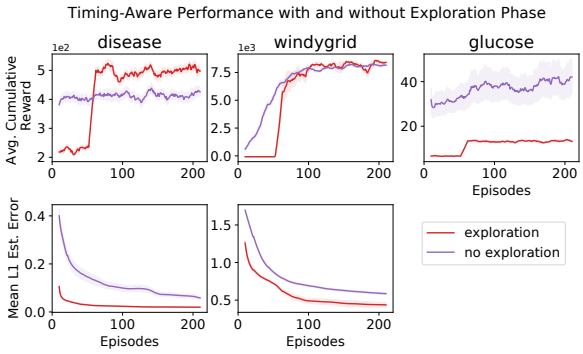

Figure 4: Avg. cumulative reward and mean L1 error for timing-aware with/without exploration phase.

proach achieves the highest cumulative reward in all environments (Table 1 and Figure 2, top). In the disease progression environment it only achieves slightly higher cumulative reward than the timing-naive approach, but achieves it quicker with fewer episodes. In the windy grid and glucose environments, the timing-aware approach significantly outperforms the timing-naive approach. In all cases, the model-based approaches outperform the model-free approach. In the disease progression and windy grid environments (where we have access to the true P for evaluation purposes), the timing-aware approach is able to obtain significantly lower estimation error $\max_{s,a,k} \|P_{a,k}(\cdot|s) - \widehat{P}_{a,k}(\cdot|s)\|_1$ than the timing-naive approach (Figure 2, bottom).

We also experiment with adding an exploration phase of 50 episodes, where actions are taken uniformly at random in the exploration phase before reverting to the ϵ-greedy policy. This approach does decrease the estimation error more quickly (Figure 4 second row), however depending on the environment, it has an inconsistent effect on the resulting cumulative reward. In the windy grid environment, we observe that while the exploration phase improves transition estimation, this does not necessarily translate to a higher return. It can even worse performance when purely random exploration causes early termination, as in the glucose environment. On the other hand, it can also result in improved transition estimation and higher reward, as seen in the disease progression environment.

In the disease simulator, all methods execute the "don't treat" action more frequently (assigns higher probabilities to staying in the same state) rather than the "treat" action (encourages switching between states) (Appendix Figure 6). The learned policies align with clinical intuition, learning not to treat in the healthy state, and to treat in the unhealthy state. Once the agent is in a healthy state, it is incentivized to remain there as long as possible. The timing-aware method often takes the largest delay (10 timesteps), whereas the timing-naive method executes intermediate delays more frequently, and the model-free methods execute much shorter delays.

In the glucose simulator, timing-aware most frequently administers the second lowest amount of insulin for the largest delay (4 timesteps) (Appendix Figure 6). By contrast, the timing-naive method administers a greater variety of quantities of insulin, and does so for an intermediate number of timesteps, most frequently administering for two timesteps. The model-free method utilizes all actions and delays more uniformly. These policies align with the clinical intuition that one should avoid administering large doses for extended periods of time.

In the windy grid simulator, all methods tend to utilize shorter delays closer to the goal state, where wind pushes the agent up one or two squares with probability 0.5 (Appendix Figure 7). Along the first and last rows (rows 0 and 9) of the grid, the timing-aware policy learns to repeat the move right action just long enough to get in the vicinity of the goal. Since there is wind pushing the agent upward in columns 7 and 8, along the top half of the grid the agent learns to go to column 9 first (where there is no wind), and then walk downward to the same row as the goal state before walking left. The optimal policy more closely resembles the timing-aware approach than the timing-naive approach.

6 DISCUSSION

The *timing-as-an-action* problem setting poses interesting theoretical and practical challenges for bringing reinforcement learning into real-world settings where opportunities to observe and act can be costly. We demonstrate that the timing-aware model-based method leverages the structure of the timing-as-an-action environment to obtain sample complexity advantages over either model-free (corresponding to standard value-based RL methods translated to our setting) or the

timing-naive model-based method. This aligns with intuition, as the timing-naive method must learn SAK parameters for $\widehat{P}$ whereas the timing-aware method only needs to learn SA parameters for $\widehat{P}$.

We demonstrate empirically that estimation using the timing-aware approach is more sample-efficient than the timing-naive approach (Figure 3). Additionally, the timing-aware approach updates its estimates for transition probabilities of delay actions other than those which were sampled (middle and right columns of Figure 3), whereas the timing-naive approach does not.

In all RL experiments, the timing-aware approach achieves the highest or ties for the highest average cumulative reward (Table 1). We note that these results are after 200 episodes, and it is likely that with more episodes the other methods would eventually do as well as the timing-aware approach. In the disease progression and windy grid RL settings, timing-naive has substantially higher estimation error than timing-aware, however the resulting policy is still able substantially outperform the model-free approach and obtain performance comparable or almost comparable to timing-aware, indicating that the learned policy can still perform well even if $\widehat{P}$ is inaccurate (Figure 2). Similarly, although an exploration phase helps improve the estimation error (Figure 4), the cumulative reward may not necessarily improve. Although our results use the same exploration strategy across all settings, future works may find it beneficial in some settings to have an exploration phase for the first few episodes in order to quickly learn $\widehat{P}$.

Acknowledgements

We gratefully acknowledge the NSF (FAI 2040929 and IIS2211955), UPMC, Highmark Health, Abridge, Ford Research, Mozilla, the PwC Center, Amazon AI, JP Morgan Chase, the Block Center, the Center for Machine Learning and Health, and the CMU Software Engineering Institute (SEI) via Department of Defense contract FA8702-15-D-0002, for their generous support of ACMI Lab's research.

References

Agarwal, A., Jiang, N., Kakade, S. M., and Sun, W. (2019). Reinforcement learning: Theory and algorithms. *CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep*, 32.

Agarwal, A., Kakade, S., Krishnamurthy, A., and Sun, W. (2020). Flambe: Structural complexity and representation learning of low rank mdps. *Advances in neural information processing systems*, 33:20095–20107.

Azar, M. G., Munos, R., and Kappen, B. (2012). On the sample complexity of reinforcement learning with a generative model. *arXiv preprint arXiv:1206.6461*.

Bacon, P.-L., Harb, J., and Precup, D. (2017). The option-critic architecture. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Barto, A. G. and Mahadevan, S. (2003). Recent advances in hierarchical reinforcement learning. *Discrete event dynamic systems*, 13(1-2):41–77.

Biedenkapp, A., Rajan, R., Hutter, F., and Lindauer, M. (2021). Temporl: Learning when to act. In *International Conference on Machine Learning*, pages 914–924. PMLR.

Braylan, A., Hollenbeck, M., Meyerson, E., and Miikkulainen, R. (2015). Frame skip is a powerful parameter for learning to play atari. In *Workshops at the twenty-ninth AAAI conference on artificial intelligence*.

Chen, H.-H., Duffy, S. W., and Tabar, L. (1996). A markov chain method to estimate the tumour progression rate from preclinical to clinical phase, sensitivity and positive predictive value for mammography in breast cancer screening. *Journal of the Royal Statistical Society Series D: The Statistician*, 45(3):307–317.

Cheung, L. C., Albert, P. S., Das, S., and Cook, R. J. (2022). Multistate models for the natural history of cancer progression. *British Journal of Cancer*, 127(7):1279–1288.

Clarke, W. and Kovatchev, B. (2009). Statistical tools to analyze continuous glucose monitor data. *Diabetes technology & therapeutics*, 11(S1):S–45.

Co-Reyes, J., Liu, Y., Gupta, A., Eysenbach, B., Abbeel, P., and Levine, S. (2018). Self-consistent trajectory autoencoder: Hierarchical reinforcement learning with trajectory embeddings. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1009–1018. PMLR.

Dabney, W., Ostrovski, G., and Barreto, A. (2020). Temporally-extended {\epsilon}-greedy exploration. *arXiv preprint arXiv:2006.01782*.

Deisenroth, M. P., Rasmussen, C. E., and Fox, D. (2011). Learning to control a low-cost manipulator using data-efficient reinforcement learning. *Robotics: Science and Systems VII*, 7:57–64.

Dietterich, T. G. (2000). Hierarchical reinforcement learning with the maxq value function decomposition. *Journal of artificial intelligence research*, 13:227–303.

Du, J., Futoma, J., and Doshi-Velez, F. (2020). Model-based reinforcement learning for semi-markov decision processes with neural odes. *Advances in Neural Information Processing Systems*, 33:19805–19816.

Duffy, S. W., Chen, H.-H., Tabar, L., and Day, N. E. (1995). Estimation of mean sojourn time in breast cancer screening using a markov chain model of both entry to and exit from the preclinical detectable phase. *Statistics in medicine*, 14(14):1531–1543.

Esserman, L. J. (2017). The wisdom study: breaking the deadlock in the breast cancer screening debate. *NPJ breast cancer*, 3(1):34.

Eysenbach, B., Salakhutdinov, R. R., and Levine, S. (2019). Search on the replay buffer: Bridging planning and reinforcement learning. *Advances in Neural Information Processing Systems*, 32.

Ha, D. and Schmidhuber, J. (2018). World models. *arXiv preprint arXiv:1803.10122*.

Hafner, D., Lee, K.-H., Fischer, I., and Abbeel, P. (2022). Deep hierarchical planning from pixels. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 26091–26104. Curran Associates, Inc.

Huang, A., Chen, J., and Jiang, N. (2023). Reinforcement learning in low-rank mdps with density features. *arXiv preprint arXiv:2302.02252*.

Jackson, C. (2011). Multi-state models for panel data: the msm package for r. *Journal of statistical software*, 38:1–28.

Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285.

Khan, A., Feng, J., Liu, S., Asghar, M. Z., et al. (2019). Optimal skipping rates: training agents with fine-grained control using deep reinforcement learning. *Journal of Robotics*, 2019.

Lakshminarayanan, A., Sharma, S., and Ravindran, B. (2017). Dynamic action repetition for deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Lindbohm, J. V., Sipilä, P. N., Mars, N. J., Pentti, J., Ahmadi-Abhari, S., Brunner, E. J., Shipley, M. J., Singh-Manoux, A., Tabak, A. G., and Kivimäki, M. (2019). 5-year versus risk-category-specific screening intervals for cardiovascular disease prevention: a cohort study. *The Lancet Public Health*, 4(4):e189–e199.

Liu, Q., Chung, A., Szepesvári, C., and Jin, C. (2022). When is partially observable reinforcement learning not scary? In *Conference on Learning Theory*, pages 5175–5220. PMLR.

Liu, Q., Netrapalli, P., Szepesvari, C., and Jin, C. (2023). Optimistic mle: A generic model-based algorithm for partially observable sequential decision making. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 363–376.

Lorenzi, M., Filippone, M., Frisoni, G. B., Alexander, D. C., and Ourselin, S. (2019). Probabilistic disease progression modeling to characterize diagnostic uncertainty: Application to staging and prediction in alzheimer's disease. *NeuroImage*, 190:56–68. Mapping diseased brains.

Machado, M. C., Barreto, A., Precup, D., and Bowling, M. (2023). Temporal abstraction in reinforcement learning with the successor representation. *Journal of Machine Learning Research*, 24(80):1–69.

Man, C. D., Micheletto, F., Lv, D., Breton, M., Kovatchev, B., and Cobelli, C. (2014). The uva/padova type 1 diabetes simulator: new features. *Journal of diabetes science and technology*, 8(1):26–34.

Mankiw, N. G. and Reis, R. (2002). Sticky information versus sticky prices: a proposal to replace the new keynesian phillips curve. *The Quarterly Journal of Economics*, 117(4):1295–1328.

Marmot, M. G., Altman, D., Cameron, D., Dewar, J., Thompson, S., and Wilcox, M. (2013). The benefits and harms of breast cancer screening: an independent review. *British journal of cancer*, 108(11):2205–2240.

Olsen, A. H., Agbaje, O. F., Myles, J. P., Lynge, E., and Duffy, S. W. (2006). Overdiagnosis, sojourn time, and sensitivity in the copenhagen mammography screening program. *The Breast Journal*, 12(4):338–342.

Ren, W., Chen, M., Qiao, Y., and Zhao, F. (2022). Global guidelines for breast cancer screening: a systematic review. *The Breast*, 64:85–99.

Sharma, S., Srinivas, A., and Ravindran, B. (2017). Learning to repeat: Fine grained action repetition for deep reinforcement learning. *arXiv preprint arXiv:1702.06054*.

Stokey, N. L. (2008). *The Economics of Inaction: Stochastic Control models with fixed costs*. Princeton University Press.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Sutton, R. S., Precup, D., and Singh, S. (1998). Intra-option learning about temporally abstract actions. In *ICML*, volume 98, pages 556–564.

Sutton, R. S., Precup, D., and Singh, S. (1999). Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211.

Vezhnevets, A. S., Osindero, S., Schaul, T., Heess, N., Jaderberg, M., Silver, D., and Kavukcuoglu, K.

(2017). Feudal networks for hierarchical reinforcement learning. In *International Conference on Machine Learning*, pages 3540–3549. PMLR.

Watkins, C. J. and Dayan, P. (1992). Q-learning. *Machine learning*, 8:279–292.

Xie, J. (2018). Simglucose v0.2.1.

Yen, M.-F., Tabar, L., Vitak, B., Smith, R., Chen, H.-H., and Duffy, S. (2003). Quantifying the potential problem of overdiagnosis of ductal carcinoma in situ in breast cancer screening. *European journal of cancer*, 39(12):1746–1754.

Young, A. L., Bragman, F. J., Rangelov, B., Han, M. K., Galbán, C. J., Lynch, D. A., Hawkes, D. J., Alexander, D. C., and Hurst, J. R. (2020). Disease progression modeling in chronic obstructive pulmonary disease. *American journal of respiratory and critical care medicine*, 201(3):294–302.

A Proofs for Section 4

A.1 Proof of Lemma 1

Proof. Let us show that the Bellman backup operator H described in (2) is a contraction mapping in the finite space $(\mathbb{R}, L_\infty)$.

$$
\begin{aligned}
&(Hq)(s,a,k) = \sum_{s' \in \mathcal{S}} P_{a,k}(s'|s) \left[\mathbb{E}[G(s,a,k)] + \gamma^k \max_{a',k'} q(s',a',k') \right] \\
&||Hq_1 - Hq_2||_\infty \\
&= \max_{s,a,k} \left| \sum_{s' \in \mathcal{S}} P_{a,k}(s'|s) \left[\mathbb{E}[G(s,a,k)] + \gamma^k \max_{a',k'} q_1(s',a',k') - \mathbb{E}[G(s,a,k)] - \gamma^k \max_{a',k'} q_2(s',a',k') \right] \right| \\
&= \max_{s,a,k} \left| \sum_{s' \in \mathcal{S}} P_{a,k}(s'|s) \left[\gamma^k (\max_{a',k'} q_1(s',a',k') - \max_{a',k'} q_2(s',a',k')) \right] \right| \\
&\leq \max_{s,a,k} \gamma^k \sum_{s' \in \mathcal{S}} P_{a,k}(s'|s) \left| \max_{a',k'} q_1(s',a',k') - \max_{a',k'} q_2(s',a',k') \right| \\
&\leq \max_{s,a,k} \gamma^k \sum_{s' \in \mathcal{S}} P_{a,k}(s'|s) \max_{a',k'} |q_1(s',a',k') - q_2(s',a',k')| \\
&= \max_{s,a,k} \gamma^k \sum_{s' \in \mathcal{S}} P_{a,k}(s'|s) ||q_1 - q_2||_\infty \\
&= \gamma^k ||q_1 - q_2||_\infty \\
&\leq \gamma ||q_1 - q_2||_\infty
\end{aligned}
$$

where the last line follows from the fact that $k \geq 1$. Thus, the operator H is a contraction. Hence, by the Banach fixed point theorem, there exists a unique optimal Q^*. □

A.2 Proofs for Lemma 2 and Lemma 3

First we prove Lemma 3. Applying Lemma 5 to $\{\mathcal{D}_i\}_{i=1}^N$ drawn as in the generative setting (Definition 1), the LHS becomes

$$
\sum_{i=1}^{N} \mathbb{E}_{(s,a,k) \sim \mathcal{D}_i} \left\| \widehat{P}(s'|s,a,k) - P(s'|s,a,k) \right\|_1^2 = n \sum_{s,a,k} \left\| \widehat{P}(s'|s,a,k) - P(s'|s,a,k) \right\|_1^2
$$

and we have the bound that with probability at least $1-\delta$,

$$
\sum_{s,a,k} \left\| \widehat{P}(s'|s,a,k) - P(s'|s,a,k) \right\|_1^2 \leq \frac{12 \log(|\overline{\mathcal{P}}_1|/\delta)}{n} + 6\epsilon SAK + \epsilon^2 SAK
$$

Then choosing $\epsilon = 1/nSAK$, Lemma 8 states that $|\overline{\mathcal{P}}_1|$ has cardinality $\leq (nS^2AK^2)^{S^2A}$, thus

$$
\sum_{s,a,k} \left\| \widehat{P}(s'|s,a,k) - P(s'|s,a,k) \right\|_1^2 \leq \frac{12 S^2 A \log(S^2AK^2n/\delta)}{n} + \frac{6}{n} + \frac{1}{SAKn^2},
$$

which implies that (suppressing log factors)

$$
\max_{s,a,k} \left\| \widehat{P}(s'|s,a,k) - P(s'|s,a,k) \right\|_1 \lesssim S \sqrt{\frac{A \log(K/\delta)}{n}}
$$

The proof of Lemma 2 proceeds similarly but with $|\overline{\mathcal{P}}| \leq (nS^2AK)^{S^2AK}$ from Lemma 8.

Maximum likelihood estimation We state and prove a general MLE guarantee for conditional probability estimation. This section is takes inspiration from the results in Agarwal et al. (2020); Liu et al. (2022, 2023); Huang et al. (2023). We consider the problem of estimating the conditional density $f^*(y|x)$, for all $x \in \mathcal{X}$ (the input space) and $y \in \mathcal{Y}$ (the target space). We are given a function class $\mathcal{F} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, and suppose $f^* \in \mathcal{F}$. In addition, we have an adaptive dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ where $x_i \sim \mathcal{D}_i(x_{<i}, y_{<i})$, and $y_i \sim f^*(\cdot|x_i)$. We output $\widehat{f} = \operatorname{argmax}_{f \in \mathcal{F}} \sum_{i=1}^n \log f(y_i|x_i)$. Note that this is analogous to the model-based estimation in (5) and (4) with function classes $\mathcal{P}_1$ and $\mathcal{P}$, respectively, and $\mathcal{X} = \mathcal{S} \times \mathcal{A} \times \mathcal{K}$ and $\mathcal{Y} = \mathcal{S}$.

We allow $\mathcal{F}$ to be an infinite function class, and the MLE bound will depend on the statistical complexity of the function class $\mathcal{F}$, which is quantified using the ℓ_1 optimistic cover, defined below:

Definition 2 (ℓ_1 optimistic cover)**.** *For a function class $\mathcal{F} \subseteq (\mathcal{X} \to \mathbb{R})$, we call function class $\overline{\mathcal{F}}$ an ℓ_∞ optimistic cover of $\mathcal{F}$ with scale ϵ, if for any $f \in \mathcal{F}$ there exists $\overline{f} \in \overline{\mathcal{F}}$, such that $\max_{x \in \mathcal{X}} \|f(\cdot|x) - \overline{f}(\cdot|x)\|_1 \le \epsilon$ and $f(y|x) \le \overline{f}(y|x)$, $\forall x \in \mathcal{X}, y \in \mathcal{Y}$.*

The formal bound for MLE estimation is stated below:

Lemma 5 (MLE guarantee)**.** *Suppose $\mathcal{F}$ satisfies: (i) $f^* \in \mathcal{F}$, (ii) each function $f \in \mathcal{F}$ is a valid probability distribution over $\mathcal{Y}$ given x (i.e., $f(\cdot|x) \in \Delta(\mathcal{Y})$ for all $x \in \mathcal{X}$), and (iii) $\mathcal{F}$ has a finite ℓ_1 optimistic cover (Definition 2) $\overline{\mathcal{F}}$ with scale ϵ and $\overline{\mathcal{F}} \subseteq (\mathcal{X} \to \mathbb{R}_{\ge 0})$. Then with probability at least $1 - \delta$, the MLE solution $\widehat{f}$ has an ℓ_1 error guarantee*

$$\sum_{i=1}^N \mathbb{E}_{x_i \sim \mathcal{D}_i} \left\| \widehat{f}(\cdot|x_i) - f^*(\cdot|x_i) \right\|_1^2 \le 12 \log(|\overline{\mathcal{F}}|/\delta) + 6\epsilon N + \epsilon^2 N$$

Proof of Lemma 5. First, define

$$\mathcal{L}(f, \mathcal{D}) = \frac{1}{2} \sum_{i=1}^N \log \left(\frac{f(y_i|x_i)}{f^*(y_i|x_i)} \right).$$

Next, we decouple the dependencies between samples, and state the following result from Agarwal et al. (2020) without proof:

Lemma 6 (Lemma 24 of Agarwal et al. (2020))**.** *Let $\mathcal{D}$ be a dataset of N examples, and let $\mathcal{D}'$ be a tangent sequence. A tangent sequence $\{(x_i', y_i')\}_{i=1}^N$ is sampled as $x_i' \sim \mathcal{D}_i(x_{1:i-1}, y_{1:i-1})$ and $y_i' \sim f^*(\cdot|x_i')$, which is independent conditioned on $\mathcal{D}$. Let $\mathcal{L}(f, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^N l(f, (x_i, y_i))$ be any function that decomposes additively across examples, where l is any function, and let $f(\mathcal{D})$ be any estimator taking as input the random variable $\mathcal{D}$ and with range $\mathcal{F}$. Then*

$$\mathbb{E}_{\mathcal{D}} \left[\exp \left(\mathcal{L}(f(\mathcal{D}), \mathcal{D}) - \log \mathbb{E}_{\mathcal{D}'} \exp(\mathcal{L}(f(\mathcal{D}), \mathcal{D}')) - \log |\mathcal{F}| \right) \right] \le 1.$$

Using Chernoff's method with union bound over $\overline{\mathcal{F}}$, with probability at least $1 - \delta$ we have for any $f \in \overline{\mathcal{F}}$ that

$$- \log \mathbb{E}_{\mathcal{D}'} \exp \left(\mathcal{L}(f(\mathcal{D}), \mathcal{D}') \right) \le -\mathcal{L}(f(\mathcal{D}), \mathcal{D}) + \log(|\overline{\mathcal{F}}|/\delta)$$

Now let $\overline{f} \in \overline{\mathcal{F}}$ be the ϵ-close ℓ_1 optimistic approximator of the MLE solution $\widehat{f} \in \mathcal{F}$. Applying the above to $\overline{f}$, in the RHS we have

$$\begin{aligned}
-\mathcal{L}(\overline{f}(\mathcal{D}), \mathcal{D}) &= \frac{1}{2} \sum_{i=1}^N \log \frac{f^*(y_i|x_i)}{\overline{f}(y_i|x_i)} \\
&\le \frac{1}{2} \sum_{i=1}^N \log \frac{f^*(y_i|x_i)}{\widehat{f}(y_i|x_i)} && (\overline{f} \text{ is optimistic cover}) \\
&= \frac{1}{2} \left(\sum_{i=1}^N \log f^*(y_i|x_i) - \sum_{i=1}^N \log \widehat{f}(y_i|x_i) \right) \\
&\le 0 && (\widehat{f} \text{ optimal})
\end{aligned}$$

Combining inequalities, we have

$$\begin{aligned}\log(|\overline{\mathcal{F}}|/\delta) \geq & -\log \mathbb{E}_{\mathcal{D}'} \exp\left(\mathcal{L}(\overline{f}(\mathcal{D}), \mathcal{D}')\right) \\ = & -\log \mathbb{E}_{\mathcal{D}'}\left[\exp\left(\frac{1}{2}\sum_{i=1}^{N}\log\left(\frac{\overline{f}(y_i'|x_i')}{f^*(y_i'|x_i')}\right)\right)|\mathcal{D}\right] \\ = & -\sum_{i=1}^{N}\log \mathbb{E}_{x\sim\mathcal{D}_i, y\sim f^*(\cdot|x_i)}\left[\sqrt{\frac{\overline{f}(y|x)}{f^*(y|x)}}\right] \end{aligned} \tag{9}$$

Next, we show that for any $\mathcal{D}$, we have

$$\mathbb{E}_{x\sim\mathcal{D}}\left\|\overline{f}(\cdot|x) - f^*(\cdot|x)\right\|_1^2 \leq -12\log \mathbb{E}_{x\sim\mathcal{D}, y\sim f^*(\cdot|x)}\left[\sqrt{\frac{\overline{f}(y|x)}{f^*(y|x)}}\right] + 6\epsilon \tag{10}$$

Combining (9) and (10), we have

$$\sum_{i=1}^{N}\mathbb{E}_{x\sim\mathcal{D}_i}\left\|\overline{f}(\cdot|x) - f^*(\cdot|x)\right\|_1^2 \leq 12\log(|\overline{\mathcal{F}}|/\delta) + 6\epsilon N \tag{11}$$

Finally, using the triangle inequality, we have

$$\begin{aligned}\sum_{i=1}^{N}\mathbb{E}_{x\sim\mathcal{D}_i}\left\|\widehat{f}(\cdot|x) - f^*(\cdot|x)\right\|_1^2 \leq & \sum_{i=1}^{N}\mathbb{E}_{x\sim\mathcal{D}_i}\left\|\widehat{f}(\cdot|x) - \overline{f}(\cdot|x)\right\|_1^2 + \sum_{i=1}^{N}\mathbb{E}_{x\sim\mathcal{D}_i}\left\|\overline{f}(\cdot|x) - f^*(\cdot|x)\right\|_1^2 \\ \leq & \epsilon^2 N + 12\log(|\overline{\mathcal{F}}|/\delta) + 6\epsilon N\end{aligned}$$

□

Lemma 7 (Optimistic cover for $\mathcal{P}$)**.** *For the function class $\mathcal{P}$ (from (4)) there exists an ℓ_1 optimistic cover (Definition 2) with scale ϵ of size $\left(\lceil\frac{S}{\epsilon}\rceil\right)^{S^2AK}$.*

Lemma 8 (Optimistic cover for $\mathcal{P}_1$)**.** *For the function class $\mathcal{P}_1$ (from (5)) there exists an ℓ_1 optimistic cover (Definition 2) with scale ϵ of size $\left(\lceil\frac{KS}{\epsilon}\rceil\right)^{S^2A}$.*

Proof of Lemma 7. Denote $\mathcal{P} = \{\mathcal{P}_k\}_{k\in[K]}$, where $\mathcal{P}_k$ denotes the model class for the k-step transitions, and we will construct $\overline{\mathcal{P}} = \{\overline{\mathcal{P}}_k\}_{k\in[K]}$ its optimistic covering set. For any $P \in \mathcal{P}_k$, set its optimistic covering function to be $\overline{P}(s'|s,a,k) = \epsilon'\lceil\frac{P(s'|s,a,k)}{\epsilon'}\rceil$ and include this $\overline{P}$ in $\overline{\mathcal{P}}_k$. Clearly for any (s,a,k,s') we have $\overline{P}(s'|s,a,k) \geq P(s'|s,a,k)$, and $\|P(\cdot|s,a,k) - \overline{P}(\cdot|s,a,k)\|_1 \leq \epsilon'|S|$ so we need to set $\epsilon' = \epsilon/|S|$. Then $|\overline{\mathcal{P}}| \leq \left(\lceil\frac{|S|}{\epsilon}\rceil\right)^{S^2AK}$. □

Proof of Lemma 8. For any $P, P' \in \mathcal{P}_1$ and $k \in [K]$,

$$\left\|P_{a,k}(\cdot|s) - P'_{a,k}(\cdot|s)\right\|_1 = \left\|P^k_{a,1}(\cdot|s) - (P'_{a,1})^k(\cdot|s)\right\|_1 \leq |S|k\sup_s\left\|P_{a,1}(\cdot|s) - P'_{a,1}(\cdot|s)\right\|_1$$

Let $\mathcal{P}'_1 = \{P : \mathcal{S}\times\mathcal{A}\to\Delta(\mathcal{S})\}$ to be the set of all valid one-step transitions. Then it suffices to first find a cover $\overline{\mathcal{P}}'_1$ of $\mathcal{P}'_1$ using a grid of size $\frac{\epsilon}{|S|K}$, then set the cover of $\mathcal{P}_1$ to be $\overline{\mathcal{P}} = \{[(\overline{P}_{a,1})^k]_{a\in\mathcal{A},k\in\mathcal{K}} : \overline{P} \in \overline{\mathcal{P}}'_1\}$. Then $|\overline{\mathcal{P}}| \leq \left(\lceil\frac{K|S|}{\epsilon}\rceil\right)^{S^2A}$. □

A.3 Proof of Lemma 4

We treat $\widehat{P}$ as independent of $\widehat{G}$ in this section, which can be accomplished by splitting the samples N into two folds, one for $\widehat{P}$ estimation, and one for $\widehat{G}$ estimation, which dilutes the final bound by only a small constant factor without changing the dependencies, and we discuss this at the end of this section.

Let $(s_i, a_i, k_i) \sim \mu$ independently. For a fixed $\widehat{P}$, rewrite the regression as

$$\widehat{G} = \operatorname*{argmin}_{f \in \mathcal{F}_{\widehat{P}}} \frac{1}{N} \sum_{i=1}^{N} \left(f(s_i, a_i, k_i) - g_i\right)^2$$

where $\mathcal{F}_{\widehat{P}} = \{\mathcal{G}_{R',\widehat{P}} : R' \in [0,1]^{SA}\}$ is the set of aggregated rewards induced by $\widehat{P}$ and any valid one-step reward function. We will bound the error $\left\|\widehat{G} - \mathcal{G}_{R,P}\right\|_{2,\mu}^2$, starting with the decomposition

$$\begin{aligned}
\left\|\widehat{G} - \mathcal{G}_{R,P}\right\|_{2,\mu}^2 &= \left\|\widehat{G} - g\right\|_{2,\mu\times G}^2 - \left\|\mathcal{G}_{R,P} - g\right\|_{2,\mu\times G}^2 \\
&= \left\|\widehat{G} - g\right\|_{2,\mu\times G}^2 - \left\|\mathcal{G}_{R,\widehat{P}} - g\right\|_{2,\mu\times G}^2 + \left\|\mathcal{G}_{R,\widehat{P}} - g\right\|_{2,\mu\times G}^2 - \left\|\mathcal{G}_{R,P} - g\right\|_{2,\mu\times G}^2 \\
&= \underbrace{\left\|\widehat{G} - g\right\|_{2,\mu\times G}^2 - \left\|\mathcal{G}_{R,\widehat{P}} - g\right\|_{2,\mu\times G}^2}_{\text{T1}} + \underbrace{\left\|\mathcal{G}_{R,\widehat{P}} - \mathcal{G}_{R,P}\right\|_{2,\mu}^2}_{\text{T2}}
\end{aligned}$$

We can bound T2 as follows:

$$\left\|\mathcal{G}_{R,\widehat{P}} - \mathcal{G}_{R,P}\right\|_{2,\mu}^2 \le \frac{1}{(1-\gamma)^2} \max_{s,a,k} \left\|\widehat{P}(\cdot|s,a,k) - P(\cdot|s,a,k)\right\|_1^2 := \frac{1}{(1-\gamma)^2}\varepsilon_P \tag{12}$$

We have the following bound for T1 from Lemma 9:

$$\left\|\widehat{G} - g\right\|_{2,\mu\times G}^2 - \left\|\mathcal{G}_{R,\widehat{P}} - g\right\|_{2,\mu\times G}^2 \lesssim \sqrt{\frac{SAG_{\max}^2 \log(1/\delta')}{N} \left\|\mathcal{G}_{R,\widehat{P}} - \mathcal{G}_{R,P}\right\|_{2,\mu}^2} + \frac{SAG_{\max}^2 \log(1/\delta')}{N}$$

Then combining (12) and Lemma 9, we have

$$\begin{aligned}
\left\|\widehat{G} - \mathcal{G}_{R,P}\right\|_{2,\mu}^2 &\lesssim \sqrt{\frac{SAG_{\max}^2 \log(1/\delta')}{N} \left\|\mathcal{G}_{R,\widehat{P}} - \mathcal{G}_{R,P}\right\|_{2,\mu}^2} + \frac{SAG_{\max}^2 \log(1/\delta')}{N} + \left\|\mathcal{G}_{R,\widehat{P}} - \mathcal{G}_{R,P}\right\|_{2,\mu}^2 \\
&\le \frac{1}{1-\gamma}\sqrt{\frac{SAG_{\max}^2 \log(1/\delta')}{N}\varepsilon_P} + \frac{SAG_{\max}^2 \log(1/\delta')}{N} + \frac{1}{(1-\gamma)^2}\varepsilon_P
\end{aligned}$$

Finally, to translate the above inequality to the ℓ_∞ guarantee of Lemma 4 in the generative setting (Definition 1),

$$\left\|\widehat{G} - \mathcal{G}_{R,P}\right\|_{2,\mu}^2 = \frac{1}{SAK} \sum_{s,a,k} \left(\widehat{G}(s,a,k) - \mathcal{G}_{R,P}(s,a,k)\right)^2 \ge \frac{1}{SAK} \max_{s,a,k} \left(\widehat{G}(s,a,k) - \mathcal{G}_{R,P}(s,a,k)\right)^2$$

Combining the above two inequalities and rearranging gives the result.

Bounds for timing-aware model-based and timing-naive model-based methods We briefly discuss the bound for timing-aware model-based, and the bound for timing-naive model-based follows the same argument. Due to sample splitting, we call the results in Lemma 3 and Lemma 4 with $\frac{1}{2}N$ samples and $\delta' = \frac{1}{2}\delta$, then union bound over the two results. For timing-smart, we have $\varepsilon_P \lesssim \frac{S^2 A \log(K/\delta)}{n}$.

Lemma 9. *Let $\widehat{R}$ be the output of (6) with transition $\widehat{P}$, and define $\widehat{G} = \mathcal{G}_{\widehat{R},\widehat{P}}$. With probability at least $1-\delta'$ we have*

$$\left\|\mathcal{G}_{\widehat{R},\widehat{P}} - g\right\|_{2,\mu\times G}^2 - \left\|\mathcal{G}_{R,\widehat{P}} - g\right\|_{2,\mu\times G}^2 \lesssim \sqrt{\frac{SAG_{\max}^2 \log(1/\delta')}{N} \left\|\mathcal{G}_{R,\widehat{P}} - \mathcal{G}_{R,P}\right\|_{2,\mu}^2} + \frac{SAG_{\max}^2 \log(1/\delta')}{N}$$

Proof of Lemma 9. For any f, define the empirical loss $\mathcal{L}_{\mathcal{D}}(f)$ and its expectation $\mathcal{L}_\mu(f)$, respectively, as

$$\begin{aligned}
\mathcal{L}_{\mathcal{D}}(f) &:= \frac{1}{N}\sum_{i=1}^{N}(f(s_i,a_i,k_i) - g_i)^2 \\
\mathcal{L}_\mu(f) &:= \mathbb{E}_{(s,a,k)\sim\mu, g\sim G(s,a,k)}\left[(f(s,a,k) - g)^2\right].
\end{aligned}$$

Let $\overline{\mathcal{F}}_{\widehat{P}}$ be an ℓ_∞ covering of $\mathcal{F}_{\widehat{P}}$ with scale ϵ, in other words, for any $f \in \mathcal{F}_{\widehat{P}}$ there exists $\overline{f} \in \overline{\mathcal{F}}_{\widehat{P}}$ such that $|\overline{f}(s,a,k) - f(s,a,k)| \leq \epsilon$ for any (s,a,k). Lemma 10 shows that such a covering exists and has cardinality $|\overline{\mathcal{F}}_{\widehat{P}}| = (\lceil 1/(1-\gamma)\epsilon \rceil)^{SA}$. For any $f \in \overline{\mathcal{F}}_{\widehat{P}}$, for a random $(s,a,k,g) \sim \mu \times G$, define

$$Z(f) := (f(s,a,k) - g)^2 - \left(\mathcal{G}_{R,\widehat{P}}(s,a,k) - g\right)^2$$

and let $Z_i(f)$ be the corresponding variable for each $(s_i, a_i, k_i, g_i) \in \mathcal{D}_i$. Observe that $\mathcal{L}_{\mathcal{D}}(f) - \mathcal{L}_{\mathcal{D}}(\mathcal{G}_{R,\widehat{P}}) = \frac{1}{N}\sum_{i=1}^N Z_i(f)$. Applying Bernstein's inequality with union bound over $\overline{\mathcal{F}}_{\widehat{P}}$, with probability $\geq 1-\delta$ we have that for any $f \in \overline{\mathcal{F}}_{\widehat{P}}$,

$$\mathbb{E}[Z(f)] - \frac{1}{N}\sum_{i=1}^N Z_i(f) \leq \sqrt{\frac{2\mathbb{V}[Z(f)] \log \frac{|\mathcal{F}_{\widehat{P}}|}{\delta}}{N}} + \frac{8G_{\max}^2 \log \frac{|\mathcal{F}_{\widehat{P}}|}{\delta}}{3N} \tag{13}$$

For any $f \in \overline{\mathcal{F}}_{\widehat{P}}$, we can upper bound $\mathbb{V}[Z(f)]$ as follows (with the constant $G_{\max}$ such that $g \in [-G_{\max}, G_{\max}]$):

$$\begin{aligned}
\mathbb{V}_{\mu\times P}[Z(f)] &\leq \mathbb{E}_{\mu\times G}[Z(f)^2] \\
&= \mathbb{E}_{\mu\times G}\left[\left((f(s,a,k) - g)^2 - \left(\mathcal{G}_{R,\widehat{P}}(s,a,k) - g\right)^2\right)^2\right] \\
&= \mathbb{E}_{\mu\times G}\left[\left(f(s,a,k) - \mathcal{G}_{R,\widehat{P}}(s,a,k)\right)^2 \left(f(s,a,k) + \mathcal{G}_{R,\widehat{P}}(s,a,k) - 2g\right)^2\right] \\
&\leq 16G_{\max}^2 \mathbb{E}_{\mu}\left[\left(f(s,a,k) - \mathcal{G}_{R,\widehat{P}}(s,a,k)\right)^2\right] \\
&= 16G_{\max}^2 \left\|f - \mathcal{G}_{R,\widehat{P}}\right\|_{2,\mu}^2
\end{aligned}$$

Further,

$$\begin{aligned}
\left\|f - \mathcal{G}_{R,\widehat{P}}\right\|_{2,\mu}^2 &\leq 2\left(\|f - \mathcal{G}_{R,P}\|_{2,\mu}^2 + \left\|\mathcal{G}_{R,P} - \mathcal{G}_{R,\widehat{P}}\right\|_{2,\mu}^2\right) \\
&= 2\left(\|f - \mathcal{G}_{R,P}\|_{2,\mu}^2 - \left\|\mathcal{G}_{R,\widehat{P}} - \mathcal{G}_{R,P}\right\|_{2,\mu}^2 + 2\left\|\mathcal{G}_{R,P} - \mathcal{G}_{R,\widehat{P}}\right\|_{2,\mu}^2\right) \\
&= 2\left(\mathcal{L}_\mu(f) - \mathcal{L}_\mu(\mathcal{G}_{R,P}) - \left(\mathcal{L}_\mu(\mathcal{G}_{R,\widehat{P}}) - \mathcal{L}_\mu(\mathcal{G}_{R,P})\right) + 2\left\|\mathcal{G}_{R,P} - \mathcal{G}_{R,\widehat{P}}\right\|_{2,\mu}^2\right) \\
&= 2\left(\mathcal{L}_\mu(f) - \mathcal{L}_\mu(\mathcal{G}_{R,\widehat{P}}) + 2\left\|\mathcal{G}_{R,P} - \mathcal{G}_{R,\widehat{P}}\right\|_{2,\mu}^2\right) \\
&= 2\left(\mathbb{E}[Z(f)] + 2\left\|\mathcal{G}_{R,P} - \mathcal{G}_{R,\widehat{P}}\right\|_{2,\mu}^2\right)
\end{aligned}$$

To summarize the above series of inequalities, we have upper bounded the variance as:

$$\mathbb{V}_{\mu\times P}[Z(f)] \leq 32G_{\max}^2\left(\mathbb{E}[Z(f)] + 2\left\|\mathcal{G}_{R,P} - \mathcal{G}_{R,\widehat{P}}\right\|_{2,\mu}^2\right) \tag{14}$$

Plugging this back into (13), with probability $\geq 1-\delta$ for any $f \in \overline{\mathcal{F}}_{\widehat{P}}$ we have

$$\mathbb{E}[Z(f)] - \frac{1}{N}\sum_{i=1}^N Z_i(f) \leq \sqrt{\frac{64G_{\max}^2\left(\mathbb{E}[Z(f)] + 2\left\|\mathcal{G}_{R,P} - \mathcal{G}_{R,\widehat{P}}\right\|_{2,\mu}^2\right)\log\frac{|\overline{\mathcal{F}}_{\widehat{P}}|}{\delta}}{N}} + \frac{8G_{\max}^2 \log\frac{|\overline{\mathcal{F}}_{\widehat{P}}|}{\delta}}{3N} \tag{15}$$

Now let $\widehat{f} = \mathcal{G}_{\widehat{R},\widehat{P}} \in \mathcal{F}_{\widehat{P}}$, and let $\overline{f}$ be its covering function. Then for any (s,a,k,g),

$$\begin{aligned}\left|Z(\overline{f}) - Z(\widehat{f})\right| &= \left|\left(\overline{f}(s,a,k) - g\right)^2 - \left(\widehat{f}(s,a,k) - g\right)^2\right| \\ &= \left|\left(\overline{f}(s,a,k) - \widehat{f}(s,a,k)\right)\left(\overline{f}(s,a,k) + \widehat{f}(s,a,k) - 2g\right)\right| \\ &\leq 4G_{\max}\left\|\overline{f} - \widehat{f}\right\|_\infty \\ &\leq 4G_{\max}\epsilon\end{aligned}$$

since $\overline{\mathcal{F}}_{\widehat{P}}$ is an ℓ_∞ cover of scale ϵ. By extension, $\left|\mathbb{E}[Z(\overline{f})] - \mathbb{E}[Z(\widehat{f})]\right| \leq \epsilon$, and same for its empirical approximation. Then

$$\begin{aligned}\mathbb{E}[Z(\widehat{f})] - \frac{1}{N}\sum_{i=1}^N Z_i(\widehat{f}) &= \mathbb{E}[Z(\overline{f})] - \frac{1}{N}\sum_{i=1}^N Z_i(\overline{f}) + \left(\mathbb{E}[Z(\widehat{f})] - \mathbb{E}[Z(\overline{f})]\right) + \left(\frac{1}{N}\sum_{i=1}^N Z_i(\overline{f}) - \frac{1}{N}\sum_{i=1}^N Z_i(\widehat{f})\right) \\ &\leq \mathbb{E}[Z(\overline{f})] - \frac{1}{N}\sum_{i=1}^N Z_i(\overline{f}) + 2\epsilon \\ &\leq \sqrt{\frac{64G_{\max}^2\left(\mathbb{E}[Z(\overline{f})] + 2\left\|\mathcal{G}_{R,P} - \mathcal{G}_{R,\widehat{P}}\right\|_{2,\mu}^2\right)\log\frac{|\overline{\mathcal{F}}_{\widehat{P}}|}{\delta}}{N}} + \frac{8G_{\max}^2\log\frac{|\overline{\mathcal{F}}_{\widehat{P}}|}{\delta}}{3N} + 2\epsilon \\ &\leq \sqrt{\frac{64G_{\max}^2\left(\mathbb{E}[Z(\widehat{f})] + \epsilon + 2\left\|\mathcal{G}_{R,P} - \mathcal{G}_{R,\widehat{P}}\right\|_{2,\mu}^2\right)\log\frac{|\overline{\mathcal{F}}_{\widehat{P}}|}{\delta}}{N}} + \frac{8G_{\max}^2\log\frac{|\overline{\mathcal{F}}_{\widehat{P}}|}{\delta}}{3N} + 2\epsilon\end{aligned}$$

Since $\widehat{f}$ is the regression loss minimizer, $\frac{1}{N}\sum_{i=1}^N Z_i(\widehat{f}) \leq \frac{1}{N}\sum_{i=1}^N Z_i(\mathcal{G}_{R,\widehat{P}}) = 0$, and we have

$$\mathbb{E}[Z(\widehat{f})] \leq \sqrt{\frac{64G_{\max}^2\left(\mathbb{E}[Z(\widehat{f})] + \epsilon + 2\left\|\mathcal{G}_{R,P} - \mathcal{G}_{R,\widehat{P}}\right\|_{2,\mu}^2\right)\log\frac{|\overline{\mathcal{F}}_{\widehat{P}}|}{\delta}}{N}} + \frac{8G_{\max}^2\log\frac{|\overline{\mathcal{F}}_{\widehat{P}}|}{\delta}}{3N} + 2\epsilon.$$

Completing the square gives

$$\mathbb{E}[Z(\widehat{f})] \leq \sqrt{\frac{128G_{\max}^2\log\frac{|\overline{\mathcal{F}}_{\widehat{P}}|}{\delta}}{N}\left(\epsilon + \left\|\mathcal{G}_{R,P} - \mathcal{G}_{R,\widehat{P}}\right\|_{2,\mu}^2\right)} + \frac{112G_{\max}^2\log\frac{|\overline{\mathcal{F}}_{\widehat{P}}|}{\delta}}{3N} + 28\epsilon$$

and $\epsilon = \frac{1}{N}$, along with the identity of $\mathbb{E}[Z(\widehat{f})]$ and $|\overline{\mathcal{F}}_{\widehat{P}}|$ from Lemma 10, gives the final bound. □

Lemma 10 (ℓ_∞ cover of $\mathcal{F}_{P'}$)**.** *Let P' be a valid transition matrix in the timing − as − an − action MDP, and let $\mathcal{F}_{P'} = \{\mathcal{G}_{R',\widehat{P}} : R' \in [0,1]^{SA}\}$ be the induced function class of aggregate rewards. There exists an ℓ_∞ cover $\overline{\mathcal{F}}_{P'}$, meaning that for any $f \in \mathcal{F}_{P'}$ there exists $\overline{f} \in \overline{\mathcal{F}}_{P'}$ with $\|f - \overline{f}\|_\infty \leq \epsilon$, of cardinality $\left(\lceil\frac{1}{(1-\gamma)\epsilon}\rceil\right)^{SA}$.*

Proof of Lemma 10. $\overline{\mathcal{F}}_{P'}$ is induced by an ℓ_∞ covering of the one-step reward functions. Let $\mathcal{R} = \{R' : R' \in [0,1]^{SA}\}$, and let $\overline{\mathcal{R}}$ be its ℓ_∞ covering of scale ϵ', such that any $R' \in \mathcal{R}$ has $\overline{R} \in \overline{\mathcal{R}}$ with $\|R' - \overline{R}\|_\infty \leq \epsilon'$. It is easy to verify that the cardinality of $\overline{\mathcal{R}}$ is $\lceil\frac{1}{\epsilon'}\rceil^{SA}$, by discretizing the interval $[0,1]$ at a scale of ϵ' for each (s,a). Then we define $\overline{\mathcal{F}}_{P'} = \{\mathcal{G}_{\overline{R},\widehat{P}} : \overline{R} \in \overline{\mathcal{R}}\}$.

For any $f = \mathcal{G}_{R',P'} \in \mathcal{F}_{P'}$, consider $\overline{f} = \mathcal{G}_{\overline{R},P'} \in \overline{\mathcal{F}}_{P'}$, where $\|\overline{R} - R'\|_\infty \leq \epsilon'$. Then using the definition of $\mathcal{G}$ in

(7), for any (s, a, k) we have

$$\begin{aligned}
\left|f(s,a,k) - \overline{f}(s,a,k)\right| &= \left|\sum_{\tau=0}^{k-1} \gamma^\tau \sum_{s'} P'(s'|s,a,k)\left(R'(s',a) - \overline{R}(s',a)\right)\right| \\
&\leq \sum_{\tau=0}^{k-1} \gamma^\tau \sum_{s'} P'(s'|s,a,k)\left|R'(s',a) - \overline{R}(s',a)\right| \\
&\leq \frac{1}{1-\gamma}\left\|R' - R\right\|_\infty \\
&\leq \frac{\epsilon'}{1-\gamma}
\end{aligned}$$

Choosing $\epsilon' = (1-\gamma)\epsilon$ gives the result. □

A.4 Proof of Proposition 1

We first decompose

$$\|Q^* - Q^{\pi_{\widehat{Q}}}\|_\infty \leq \|Q^* - \widehat{Q}\|_\infty + \|\widehat{Q} - Q^{\pi_{\widehat{Q}}}\|_\infty$$

Next we bound the first term. Using the fact that Q^* is the unique solution to the timing-as-an-action Bellman equation in (2), and the definition of $\widehat{Q}$ in (8), for a fixed (s, a, k) we have

$$\begin{aligned}
&\left|Q^*(s,a,k) - \widehat{Q}(s,a,k)\right| \\
&= \left|\mathcal{G}_{R,P}(s,a,k) - \mathcal{G}_{\widehat{R},\widehat{P}}(s,a,k) + \gamma^k \sum_{s'}\left(P(s'|s,a,k)\max_{a',k'} Q^*(s',a',k') - \widehat{P}(s'|s,a,k)\max_{a',k'}\widehat{Q}(s',a',k')\right)\right| \\
&\leq \left|\mathcal{G}_{R,P}(s,a,k) - \mathcal{G}_{\widehat{R},\widehat{P}}(s,a,k)\right| + \gamma^k \sum_{s'}\left|P(s'|s,a,k) - \widehat{P}(s'|s,a,k)\right|\max_{a',k'}|Q^*(s',a',k')| \\
&+ \gamma^k \sum_{s'}\widehat{P}(s'|s,a)\left|\max_{a',k'} Q^*(s',a',k') - \max_{a',k'}\widehat{Q}(s',a',k')\right| \\
&\leq \left\|\mathcal{G}_{R,P} - \mathcal{G}_{\widehat{R},\widehat{P}}\right\|_\infty + \frac{\gamma^k}{1-\gamma}\max_{s,a,k}\left\|P(\cdot|s,a,k) - \widehat{P}(\cdot|s,a,k)\right\|_1 + \gamma^k\left\|Q^* - \widehat{Q}\right\|_\infty
\end{aligned}$$

where we use the fact $\widehat{P}$ is a valid transition and that $\|Q^*\|_\infty \leq 1/(1-\gamma)$ in the last inequality above. Since this holds for any (s, a, k), we have

$$\left\|Q^* - \widehat{Q}\right\|_\infty \leq \left\|\mathcal{G}_{R,P} - \mathcal{G}_{\widehat{R},\widehat{P}}\right\|_\infty + \frac{\gamma}{1-\gamma}\max_{s,a,k}\left\|P(\cdot|s,a,k) - \widehat{P}(\cdot|s,a,k)\right\|_1 + \gamma^k\left\|Q^* - \widehat{Q}\right\|_\infty,$$

and rearranging the above inequality gives

$$\|Q^* - \widehat{Q}\|_\infty \leq \frac{1}{1-\gamma}\left\|\mathcal{G}_{R,P} - \mathcal{G}_{\widehat{R},\widehat{P}}\right\|_\infty + \frac{\gamma}{(1-\gamma)(1-\gamma)}\max_{s,a,k}\left\|P(\cdot|s,a,k) - \widehat{P}(\cdot|s,a,k)\right\|_1$$

For the second term, again fixing (s, a, k) and using similar techniques, we have

$$\begin{aligned}
&|\widehat{Q}(s,a,k) - Q^{\pi_{\widehat{Q}}}(s,a,k)| \\
&= \left|\mathcal{G}_{R,P}(s,a,k) - \mathcal{G}_{\widehat{R},\widehat{P}}(s,a,k) + \gamma^k \sum_{s'}\left(\widehat{P}(s'|s,a,k)\max_{a',k'}\widehat{Q}(s',a',k') - P(s'|s,a,k)Q^{\pi_{\widehat{Q}}}(s',\pi_{\widehat{Q}}(s'))\right)\right| \\
&= \left|\mathcal{G}_{R,P}(s,a,k) - \mathcal{G}_{\widehat{R},\widehat{P}}(s,a,k) + \gamma^k \sum_{s'}\left(\widehat{P}(s'|s,a,k)\widehat{Q}(s',\pi_{\widehat{Q}}(s')) - P(s'|s,a,k)Q^{\pi_{\widehat{Q}}}(s',\pi_{\widehat{Q}}(s'))\right)\right|,
\end{aligned}$$

since $\pi_{\widehat{Q}}$ is the greedy policy w.r.t. $\widehat{Q}$. Then

$$\begin{aligned}
|\widehat{Q}(s,a,k) - Q^{\pi_{\widehat{Q}}}(s,a,k)| \leq & \left|\mathcal{G}_{R,P}(s,a,k) - \mathcal{G}_{\widehat{R},\widehat{P}}(s,a,k)\right| + \gamma^k \sum_{s'} \left|P(s'|s,a,k) - \widehat{P}(s'|s,a,k)\right| \left|Q^{\pi_{\widehat{Q}}}(s', \pi_{\widehat{Q}}(s'))\right| \\
& + \gamma^k \sum_{s'} \widehat{P}(s'|s,a) \left|\widehat{Q}(s', \pi_{\widehat{Q}}(s')) - Q^{\pi_{\widehat{Q}}}(s', \pi_{\widehat{Q}}(s'))\right| \\
\leq & \left\|\mathcal{G}_{R,P} - \mathcal{G}_{\widehat{R},\widehat{P}}\right\|_\infty + \frac{\gamma^k}{1-\gamma} \max_{s,a,k} \left\|P(\cdot|s,a,k) - \widehat{P}(\cdot|s,a,k)\right\|_1 + \gamma^k \left\|\widehat{Q} - Q^{\pi_{\widehat{Q}}}\right\|_\infty
\end{aligned}$$

Then rearranging, we have

$$\|\widehat{Q} - Q^{\pi_{\widehat{Q}}}\|_\infty \leq \frac{1}{1-\gamma} \left\|\mathcal{G}_{R,P} - \mathcal{G}_{\widehat{R},\widehat{P}}\right\|_\infty + \frac{\gamma}{(1-\gamma)(1-\gamma)} \max_{s,a,k} \left\|P(\cdot|s,a,k) - \widehat{P}(\cdot|s,a,k)\right\|_1$$

Then combining the two parts of the bound, we have

$$\begin{aligned}
\|Q^* - Q^{\pi_{\widehat{Q}}}\|_\infty &\leq \|Q^* - \widehat{Q}\|_\infty + \|\widehat{Q} - Q^{\pi_{\widehat{Q}}}\|_\infty \\
&\leq \frac{2}{1-\gamma} \left\|\mathcal{G}_{R,P} - \mathcal{G}_{\widehat{R},\widehat{P}}\right\|_\infty + \frac{2\gamma}{(1-\gamma)(1-\gamma)} \max_{s,a,k} \left\|P(\cdot|s,a,k) - \widehat{P}(\cdot|s,a,k)\right\|_1
\end{aligned}$$

B Implementation Details

For the transition estimation experiments, optimization is done using SGD with a learning rate of 0.01. For the RL environments, optimization is done using the Adam optimizer with a batch size of 500 and initial learning rate of 10^{-3} for $\widehat{P}$ and 0.1 for $\widehat{R}$, and Q-value iteration is done to convergence, where convergence is defined as a change of less than 10^{-5} for at least two iterations. For additional details, see the code provided in the supplement, which reproduces all results in the paper. All experiments were conducted on a single machine with 24 CPUs and 1 Titan RTX GPU. Windy grid experiments were conducted on the GPU whereas all other experiments were conducted on CPU.

C RL Environment Details

The true one-step transition probabilities for the disease progression simulator are as follows:

```
true_P = np.array([
    [
        [0.89, 0.1, 0.01],
        [0.15, 0.8, 0.05],
        [0.0, 0.0, 1.0]
    ],
    [
        [0.1, 0.89, 0.01],
        [0.8, 0.15, 0.05],
        [0.0, 0.0, 1.0]
    ],
])
```

which is a $A \times S \times S$ matrix, and the (i, j, k)-th element is the probability of transitioning to state k conditioned on taking action i from state j.

D RL Experiment Results

Reward Estimation In the generative setting, averaged over 30 trials, we characterize the aggregate reward estimation error when the reward model is learned in conjunction with the timing-aware and timing-naive models. This estimation error is compared to when the reward model is learned but paired with an oracle transition

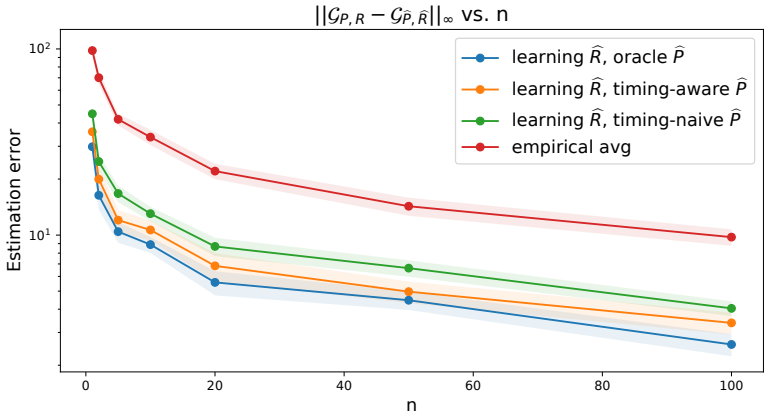

Figure 5: Average reward estimation error $||\mathcal{G}_{P,R} - \mathcal{G}_{\widehat{P},\widehat{R}}||_\infty$ in the generative setting over 30 trials, with 95% confidence intervals. The number of repetitions $n = [1, 2, 5, 10, 20, 50, 100]$ is the number of per-(s, a, k) samples drawn for all $(s, a, k) \in \mathcal{S} \times \mathcal{A} \times \mathcal{K}$.

model, as well as against simply averaging the rewards gathered from tuples of experience taking each action from each state.

Overall, simply tracking the average empirical reward is the least sample efficient, followed by learning $\widehat{R}$ in conjunction with timing-naive $\widehat{P}$, followed by timing-aware $\widehat{P}$, and finally using the oracle P when learning $\widehat{R}$.

Distribution of Actions Taken Next, we include visualizations of distribution of actions taken by the policy in each environment (Figure 6) and the policies learned in the windy grid environment (Figure 7).

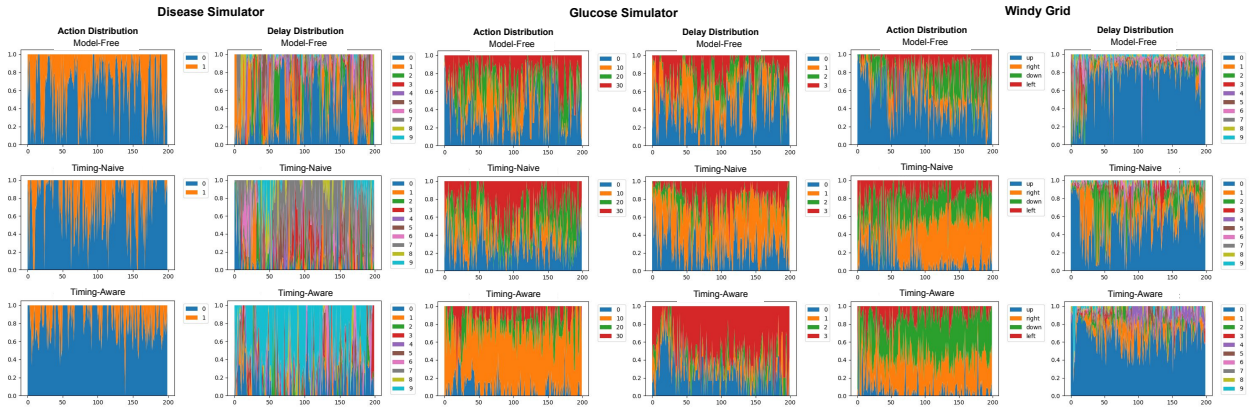

Figure 6: Distribution of actions (sub-left) and delays (sub-right) taken by the policy in the disease progression (left), glucose monitoring (middle), and windy grid (right) environments. In the disease simulator, action 0 corresponds to "don't treat" and action 1 corresponds to "treat." Delays are 0-indexed, but delay 0 corresponds to a one-timestep delay, delay 1 corresponds to a two-timestep delay, etc.

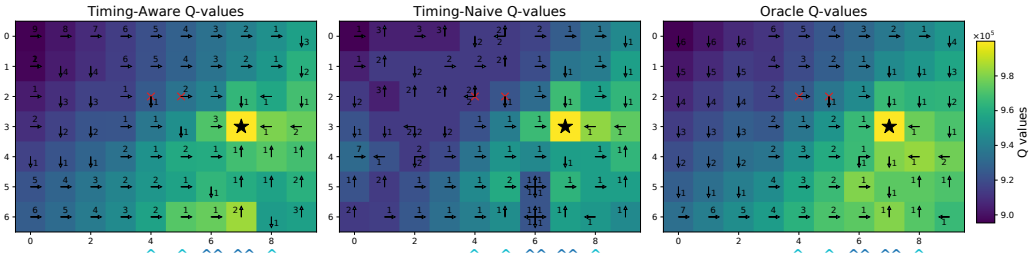

Figure 7: Maximum Q-values learned in each state from the timing-aware and timing-naive model-based methods after 200 episodes, as well as the oracle Q values. In each grid state, the arrow gives the direction of the action, the number gives the delay, and the color gives the value. ˆ or ˆˆ indicate a stochastic wind which pushes the agent up with probability 0.5 for one or two squares, respectively. The star is the goal state, and the x's are hazard states.