TIME AND TEMPORAL ABSTRACTION IN CONTINUAL LEARNING: TRADEOFFS, ANALOGIES AND REGRET IN AN ACTIVE MEASURING SETTING

Vincent Létourneau
University of Ottawa
vletour2@uottawa.ca

Colin Bellinger
National Research Council of Canada
Colin.Bellinger@nrc-cnrc.gc.ca

Isaac Tamblyn
University of Ottawa
isaac.tamblyn@uottawa.ca

Maia Fraser
University of Ottawa
mfrase8@uottawa.ca

ABSTRACT

This conceptual paper provides theoretical results linking notions in semi-supervised learning (SSL) and hierarchical reinforcement learning (HRL) in the context of lifelong learning. Specifically, our construction sets up a direct analogy between intermediate representations in SSL and temporal abstraction in RL, highlighting the important role of factorization in both types of hierarchy and the relevance of partial labeling, resp. partial observation. The construction centres around a simple class of Partially Observed Markov Decision Processes (POMDPs) where we show tools and results from SSL imply lower bounds on regret holding for any RL algorithm without access to temporal abstraction. While our lower bound is for a restricted class of RL problems, it applies to arbitrary RL algorithms in this setting. The setting moreover features so-called "active measuring", an aspect of widespread relevance in industrial control, but - possibly due to its lifelong learning flavour - not yet well-studied in RL. Our formalization makes it possible to think about tradeoffs that apply for such control problems.

1 INTRODUCTION AND CONTRIBUTIONS

Temporal abstraction, e.g. in the form of options or macros, is nowadays common in reinforcement learning (RL) and its practical advantages undisputed; it is viewed as a form of transfer, however, there is little theoretical work studying how much it helps and when (see (Perkins & Precup, 1999) for an early example, also (Dayan & Hinton, 1998; Konidaris & Barto, 2007; van Seijen et al., 2017b;a; Vezhnevets et al., 2017; Frans et al., 2017; Brunskill & Li, 2013)). Meta-learning, or learning to learn, has also been studied for decades (Schmidhuber, 1987; Bengio et al., 1991; Thrun & Pratt, 1998), with representations playing a key role throughout - for recording inductive bias that can be transferred to subsequent tasks; transfer learning itself was formalized two decades ago (Baxter, 2000), and even earlier, related questions were considered in semi-supervised learning (SSL), exploring the value of unlabeled data for a final supervised learning (SL) problem (Castelli & Cover, 1996). These themes have continued to inform much of machine learning, while deep-learning theory and practice have recently given particular prominence to representation learning (Bengio et al., 2013), including connections with manifold learning (Alain & Bengio, 2012), itself a well-studied form of SSL (Belkin et al., 2006), (Niyogi, 2013).

The present paper cuts across technical areas to add to this conceptual body of work. We show that learning-theoretic results in SL and SSL can be used to imply analogous results in RL. Specifically, we construct a family $\mathcal{F}$ of POMDPs based on SL or SSL problems, such that intermediate representations that reduce risk in SL or SSL (including some based on manifold learning or invariant features) correspond to temporal abstractions that reduce regret in RL. The family $\mathcal{F}$ consists essentially of simplified industrial control settings with **active measuring**, namely, where an agent can choose to make more accurate but also costly measurements vs. less accurate but cheap measurements. The simplicity of our construction allows us to study tradeoffs that arise in such settings, and the fact we use a **family**, rather than a single POMPD or MDP, allows us to prove an agent-agnostic lower bound on regret. Such bounds are relatively uncommon in RL. They hold for arbitrary agents but specify worst-case regret across the family of settings and as such they are more interesting when the family is restricted to MDPs of practical interest, rather than "arbitrary MDPs" which not surprisingly may include pathological examples (Dann et al., 2017).

To improve readability, we offer two versions of our POMDP construction: (B) is based on classical results in SL, and (A) allows a stronger conclusion but is based on recent, somewhat technical, results in SSL. Let $\mathcal{H}$ denote the hypothesis class in either case. At each time step (in either version), the agent can sample a data-generating distribution by an action that makes a full but costly observation or by an action that makes a partial but cheap observation. A third possible action[1] consists in choosing a hypothesis function $h \in \mathcal{H}$, with reward equal to the risk of h. The actual rewards and numbers of episodes are specified by design settings/parameters of the *environment*, and we show there is a tradeoff such that, for certain values of these settings, an agent with access to a relevant temporal abstraction ϕ (in fact, a macro) will have higher return than an agent without. In version (A), it is possible to *learn* ϕ through inexpensive partial observation, whereas in (B), the learning of ϕ is not addressed and the focus is only on the benefits of using ϕ. Version (A) bears close resemblance to planning in many scientific or industrial control settings where precise measurements are costlier than coarse ones: so-called *active measuring* (Jaulmes et al., 2005; Armstrong-Crews & Veloso, 2007; Bellinger et al., 2021). In monitoring and adjusting an ongoing industrial process, an agent is confronted with one learning challenge after another throughout its indefinite lifetime. We are interested in studying the role of temporal abstraction in lifelong learning environments of this kind with active measuring. We formalize the environment as an infinite sequence of POMDPs from a specific class, defined in Definition 8. We then analyze in Theorem 9 the performance of a specific agent A1 with oracle access to ϕ (for example it could be expert designed, or learned in a prior phase before the agent is launched in the lifelong learning environment) vs an agent A2 without access to ϕ, or possibly even without the capability to carry out temporal abstraction at all. For the former A2 we show how initial cost may be amortized over the lifetime of the agent; for the latter A2 we obtain a lower bound on regret that applies to arbitrary agents, thus establishing theoretically the need for temporal abstraction.

The background SL and SSL results we describe in Section 2 essentially say that useful intermediate representations (e.g. manifold coordinates or group-invariant features) are ones that induce a *factorization* of the learning problem to a problem of lower complexity, and we also give specific settings where this occurs. In Section 3 we use these SL and SSL settings to define a class of POMDPs where the mentioned SL and SSL results imply an RL corollary, with a complexity-lowering representation in the original setting corresponding to a regret-lowering macro ϕ in the RL setting. With this SSL-RL correspondence, moreover, the role of factorization on the SSL side has an interesting resonance on RL side with findings of Bacon & Precup (2016) who showed that specifying a set of options that a meta-policy can use is equivalent to specifying a matrix preconditioner as done in numerical analysis. In the context of iterative solution of linear systems, such a preconditioner transforms the original problem to a new one, and the preconditioner is useful if this new problem has a lower "condition number", a quantity indicative of the number of steps to convergence, i.e., the difficulty of the approximation. Typically there are tradeoffs that arise in numerical analysis between the cost of computing a preconditioner and the benefit it may offer in terms of reduced condition number. Likewise in our thought experiment, the difficulty (cost in terms of sample complexity) to learn a useful representation ϕ is weighed against the benefit that this representation confers in terms of reduced per-episode regret for all future episodes. In particular, in a lifelong learning scenario, the benefits of the acquired representation will ideally eventually outweigh even a costly initial learning/acquisition of the needed representation. Note, we do not focus on how the representation ϕ is learned (it corresponds to an intermediate representation in SL or SSL settings that is learnable by methods relevant to the setting, for example using trajectories of a group action, or by manifold learning techniques and in our setting we use only the fact it is learnable from partially observed states) instead we focus on showing that *not* learning the representation condemns agents to incur a certain basic amount of regret on all future tasks.

The tradeoffs that we establish show the structure of the learning environment - in our case encoded by the complexity reduction of an SSL problem with vs without ϕ - is linked to the question of how many tasks or episodes are needed before learning a temporal abstraction pays off, and that this also depends on the relative costs of full vs partial observation. Our setting specifically includes active measuring, and more precisely, the tradeoffs we prove show that depending on the relative costs of full vs partial measuring and potential payoffs at the terminal state, it may be essential for an active measuring agent to either be designed with ϕ as available macro, or else designed to learn ϕ. Industrial control applications where active measuring is involved are typically ones where the control process goes on indefinitely and any lifelong learning strategy in this context would include how and when to actively take measurements, and be refined over time. Our setting is simple enough, involving only one representation ϕ and two levels of observation - partial or full - which are respectively cheap or expensive, that it serves as a description of a pattern one can find in many possible applications.

[1] Note that modifications of this set-up are also possible, e.g., making a prediction at each time step while sampling, rather than having a dedicated third action for hypotheses; there is a long tradition of expressing SL as a special case of RL and here we have chosen just one modality that keeps calculations simple.

On a conceptual level, our toy setting brings under one lens various forms of "representation" - temporal, state, action, feature, control etc.. - all of which induce *transformation* of one learning problem to another. Figure 1 illustrates this analogy.

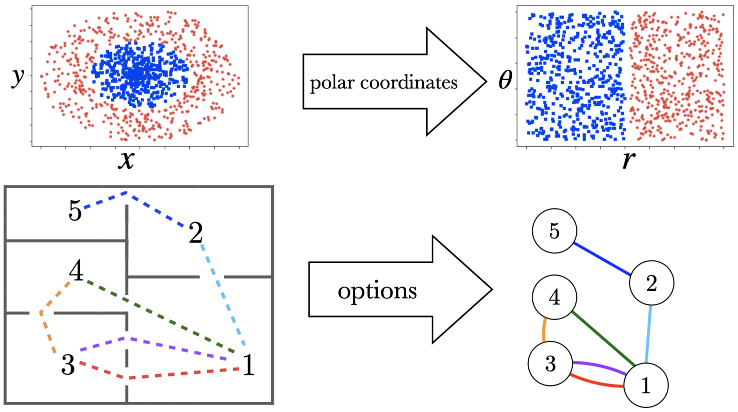

Figure 1: Providing an agent with useful learned options can have the effect of transforming the original RL problem into a new one that is "simpler" (bottom row). This is analogous to the way useful learned representations transform a supervised learning problem to another easier supervised learning problem (top row). In both cases, this implies a hierarchy: in RL this means hierarchy in time scale and control as a high-level policy calls options, which themselves invoke policies on primitive actions.

2 Background and notation

MDPs and POMDPs. Reinforcement learning (RL) problems are often formulated as *Markov Decision Processes (MDPs)*. An MDP is $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, R \rangle$, where $\mathcal{S}$ and $\mathcal{A}$ are spaces of states, resp. actions, P is the transition probability kernel, and R the reward function; see (Sutton & Barto, 2018; Puterman, 1994, 2005). The agent knows $\mathcal{S}$ and $\mathcal{A}$. It sees its current state and, upon taking an action a from $\mathcal{A}$, sees the effect of P and R as they generate respectively the next state $s' \sim P_{a,s}$ and the associated reward $R_a(s, s')$. The agent's goal is to find a (Markov) policy $\pi : \mathcal{S} \to \mathcal{A}$ that maximizes the expectation of some form of cumulative reward, which could be average reward over an infinite horizon, discounted total reward with fixed discount factor $\gamma \in (0, 1)$, or un-discounted total over a finite horizon. We consider the latter and focus on MDPs with a form of action-penalty reward such that the only positive rewards are obtained upon arrival in a terminal state s_g. Given an underlying MDP as above, one may also consider a *partially observable MDP (POMDP)*. This requires additionally specifying a set Ω of observations and a family of observation probabilities $O(o|s', a)$. At each new state s' the agent does not have access to s' itself but only to an observation $o \in \Omega$. As before, the agent seeks a policy that maximizes expected cumulative reward but policies are functions of observations not full states.

Temporal abstraction Macros and options are the most commonly considered forms of "temporal abstraction" in RL. An *option* (Sutton et al., 1999) is a triple, $o = \langle I_o, \pi_o, \beta_o \rangle$, where I_o is an initiation set of states (or observations), π_o a policy on primitive actions or else another option, β_o a termination condition. When an option o is called by a higher-level policy π, then upon termination of o, control is returned to π. Macros, i.e. finite sequences of primitive actions, are invoked in similar fashion, but termination is instead imposed after the sequence has been executed. The name *hierarchical* RL (HRL) is sometimes used to refer to such settings, and macros or options themselves called "temporal abstractions", because of the hierarchy in time and control this involves: the RL agent invoking an option or macro is "higher" in control, and this upper agent also operates at a longer time scale (time between decisions) compared to the lower agent (making multiple decisions before returning control to the upper agent).

Supervised and semi-supervised learning. A *supervised learning (SL)* task consists of a joint probability distribution p on a space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, where $\mathcal{Y}$ are "labels", typically in $[0, 1]$; classically $\mathcal{X} = \{0, 1\}^k$ for some k, but it may also be continuous as in our examples. The agent is given a sample $\overline{z} \in \mathcal{Z}^m$ and tasked with learning a specific (regression, classification or decision) function $f_p : \mathcal{X} \to \boldsymbol{R}$ determined entirely by the conditional $p(y|x)$, usually $f_p = \boldsymbol{E}_p(y|\cdot)$. *Semi-supervised learning (SSL)* considers the same situations but involves agents that use unlabeled data (from $\mathcal{X}$) as well as labeled data to improve performance of the final supervised task.

PAC framework and variants. The classic PAC framework (Valiant, 1984) and its extension to noisy y (Kearns & Schapire, 1994) assume (in the realizable case, for regression/classification) not just a specific supervised task as defined above but a *class* of these: a so-called *"(probabilistic) concept class"* $\mathcal{C}$ which is essentially a class of possible conditional distribution families $\{\rho(y|x) : x \in \mathcal{X}\}$. The agent's goal is to construct a *learning algorithm* $\mathcal{A} : \mathcal{Z}^m \to \mathcal{H}$ mapping samples to prediction functions so as to minimize *risk*, i.e., expected cumulative loss, $\boldsymbol{E}_D(\mathcal{A}(\overline{z})(x), f_p(x))$, where $\overline{z} = ((x_1, f_p(x_1)), \ldots, (x_m, f_p(x_m)))$ and each x_i, as well as x is sampled according to D. PAC theory is *distribution-free* and seeks worst-case guarantees: *"$\forall D$ on $\mathcal{X}$, the algorithm $\mathcal{A}$ returns $f = \mathcal{A}(\overline{z})$ that with high probability has risk $\mathcal{R}(f) < \epsilon$"*. The freedom in D implies unlabeled data $\overline{x}$ are useless for predicting y (for more detail on PAC learning, see for example Mohri et al. (2007)), This makes the PAC framework unsuitable in its raw form for analysis of SSL (see Balcan & Blum (2005); Fraser (2016) for discussions).

Modification of PAC framework. To address the shortcoming of PAC learning for SSL, Balcan and Blum proposed augmenting the PAC framework by the addition of a *compatibility function* $\chi : \mathcal{C} \times \mathcal{D} \to [0, 1]$, which records the amount of compatibility we believe each concept from $\mathcal{C}$ to have with each $D \in \mathcal{D}$, the class of "all" distributions on X. This function is required to be learnable by sampling D and is then used to reduce the concept class from $\mathcal{C}$ to a sub-class that is used for the supervised learning step. The idea is that χ is a useful compatibility function if this sub-class has lower complexity than $\mathcal{C}$ (Balcan & Blum, 2005). The compatibility function itself is however an additional gadget that must be specified. An alternative approach to developing learning theory for SSL considers a *distribution-constrained variant* of PAC theory (Fraser, 2016), positting a specific class $\mathcal{P}$ of joint probability distributions on $\mathcal{X} \times \mathcal{Y}$ in which the true p lives. This means $\mathcal{P}$ defines both the space $\mathcal{C}$ of conditionals $p(y|\cdot)$ for y given x, and the marginals $D = p_X$ which are allowed on $\mathcal{X}$. By contrast, in PAC-learning, $\mathcal{P}$ is de facto the product of *arbitrary* distributions D on $\mathcal{X}$ and families $\{p(\cdot|x) : x \in \mathcal{X}\}$ of conditionals from $\mathcal{C}$. For analysis in the distribution-constrained setting, Fraser (2016) defined *γ-uniform shattering dimension*, a measure of complexity similar to VC- or fat shattering dimension but applicable to joint statistical models $\mathcal{P}$; when this dimension is at least d then, as with classic measures of complexity, a lower bound proportional to γ^{m+1} applies to the risk for any learning algorithm using a sample of size $m < d$.

Semi-supervised learning and Fisher-Neyman factorization. It happens in some machine learning settings that there is an intermediate representation $\phi : \mathcal{X} \to \boldsymbol{R}^k$ such that ϕ is a sufficient statistic for the learning goal f_p in the strong sense that $f_p(x) = g_p(\phi(x))$ for some other function g_p. In other words, the learning goal can be expressed in terms of the intermediate representation or latent variable $t = \phi(x)$. In this case, an agent with access to ϕ is essentially operating in a *modified learning problem*, namely one whose underlying joint model is a Fisher-Neyman factorization Taraldsen (2022) of the model $\mathcal{P}$. Indeed, to return to concept classes, suppose the conditional distribution $p(\cdot|x)$ is entirely determined by $f_p(x)$ so the concept class under consideration is $\mathcal{C} = \{f_p : p \in \mathcal{P}\}$, and likewise $p(\cdot|t)$ is identified with g_p for the latent variable $t = \phi(x)$; then the new concept class the agent faces after using ϕ is $\mathcal{C}' = \{g_p : p \in \mathcal{P}\}$. If this new model has lower complexity, then ϕ is a useful intermediate representation. If, moreover, ϕ can be learned from unlabeled data then unlabeled data are valuable for the supervised problem because they allow the original learning problem to be transformed to an easier one where fewer labeled data will suffice. In particular, this is the case in many manifold learning- and group-invariant feature examples (Fraser, 2016), where ϕ corresponds respectively to manifold coordinates, or group-invariant features that can be learned from unlabeled data.

SL and SSL results we'll use later. The constructions of (Fraser, 2016) combined with its main theorem (Fraser, 2016, Thm 3) can be summarized:

Theorem 1 *For any $0 < s < r$ there is an $m \in \boldsymbol{N}$, a supervised learning problem (defined by a family of joint distributions p on $\mathcal{X} \times \{0, 1\}$)*[2] *and a representation $\phi : \mathcal{X} \to \mathcal{X}'$ (for simplicity, $\mathcal{X}'$ and $\mathcal{X}$ can both be taken to be $\mathbb{R}^\ell$) such that for sample complexity $\leq m$, any purely supervised algorithm has risk greater than r whereas an ERM-style learning algorithm B_ϕ with oracle knowledge of $\phi(x)$ for any $x \in \mathcal{X}$, has risk less than s. Moreover the problem construction can be such that ϕ itself is learnable using some quantity $M >> m$ of purely unlabeled data.*

The final statement of this theorem is particular to SSL, and it arises by using unlabeled data to learn manifold coordinates or invariant features (Fraser, 2016). On the other hand, for the first statement about ϕ, it is easy to obtain a

[2] This is a distribution-constrained setting introduced in Fraser (2016) to study SSL. In the classical distribution-free setting of PAC learning, the marginals p_X are arbitrary so unlabeled data are uninformative, and any ϕ learned from unlabeled data likewise. The distribution-constrained setting makes it possible to express connections between marginals p_X and concepts to study SSL, but VC dimension is no longer an adequate tool so uniform shattering dimension was introduced in Fraser (2016) to prove Theorem 1. The reader who wishes to build intuition while avoiding these technicalities may revert to the simpler Theorem 2 and its proof.

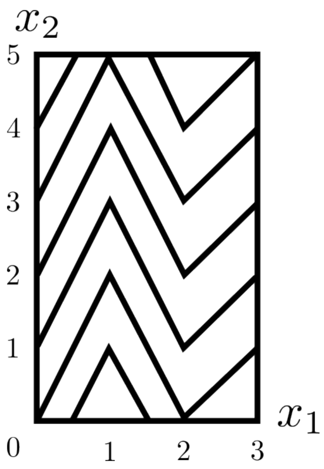

Figure 2: From Lemma 4. Points on a zigzag line L+c have the same label.

similar result without using SSL tools, just by applying classical measures of complexity like VC dimension and their known risk bounds:

Theorem 2 *For any $0 < s < r < \frac{1}{2}$, there is an $m \in \boldsymbol{N}$ and a supervised learning problem on $\mathcal{X} \times \{0, 1\}$ such that for sample complexity m, any supervised algorithm has expected risk greater than r whereas an ERM-style algorithm B_ϕ with oracle knowledge of a particular representation $\phi : \mathcal{X} \to \mathcal{X}'$ has expected risk less than s.*

We give an explicit proof of Theorem 2 below to conclude this section. The next section, Section 3, will construct a family of POMDPs in two versions; Version (B) based on Theorem 1 and Version (A) based on Theorem 2. The reader who is not familiar with SSL and/or prefers to have an explicit setting in mind when proceeding to Section 3 is invited to refer to the straightforward supervised problems we now define - see Lemma 4 and Remark 5 - for our proof of Theorem 2.

Remark 3 *The key property of these problems is that there is a Fisher-Neyman factorisation of one problem into the other, and a suitable VC-dimension gap. In this setting (see construction in the proof of Lemma 4), ϕ is learned from partially labeled data, and many different ϕ are possible; at the same time, for a given ϕ there are many concepts in $\mathcal{H}$ that benefit from using ϕ. This is also the situation for the SSL result Theorem 1 stated earlier (though its proof is not reproduced here). Consequently, in our RL construction there will be many tasks that can benefit from the same temporal abstraction defined by ϕ, This implies advantages of a macro that computes ϕ - see Theorem 9.*

Lemma 4 *For any integer $d > 1$, there are pairs of binary-function sets $\mathcal{H}, \mathcal{G}$ on $\mathcal{X}$ and $\mathcal{X}'$ respectively, with VC dimension $d_\mathcal{H} = d$ and $d_\mathcal{G} = 1$ respectively and a map $\phi : \mathcal{X} \to \mathcal{X}'$ which induces a factorization of $\mathcal{H}$ to $\mathcal{G}$.*

Remark 5 *Another example of a pair of such spaces that is perhaps easier to keep in mind is that of an artificial neural network with a single output. It is shown in Bartlett et al. (2019) that there are classes of neural network with W weights and L layers with VC dimension at least $cWL\log(W/L)$ for some constant c. So to achieve the previous result you can take $\mathcal{H}$ the set of these neural networks, ϕ the $N - 1$ first layers of a fully trained 0 loss network from this class and $\mathcal{G}$ the family of RELU functions $\sigma(x - b)$ representing the last layer of the same networks.*

Proof: [Proof of Theorem 2] Assume $0 < s < r < \frac{1}{2}$. We claim there are binary function classes $\mathcal{H}, \mathcal{G}$ with respective VC dimensions $d_\mathcal{H} > d_\mathcal{G}$, and an integer m for which the following inequalities are satisfied:

$$\frac{2d_\mathcal{G}\log(m) + 4}{m\log(2)} < s < r < \frac{1}{2}\left(1 - \frac{1}{d_\mathcal{H} - 1}\right)^m. \tag{1}$$

Indeed, suppose $d_\mathcal{G} = 1$. Since $\log(m)/m \to 0$ as $m \to \infty$, there is an m big enough so the first inequality is satisfied. Now with m fixed, $(1 - 1/(d_\mathcal{H} - 1))^m$ is increasing in $d_\mathcal{H}$ (bounded by 1) so for large enough $d_\mathcal{H}$ the inequality on the right is satisfied. The claim follows by Lemma 4. On the other hand, by classical results (see Proposition 6, and

Proposition 7 below) at sample size m, the risk of an ERM-style algorithm run on a problem with VC dimension $d_{\mathcal{G}}$ is upper bounded by the expression on the left of the inequalities in (1), while the risk of an arbitrary algorithm run on a problem with VC dimension $d_{\mathcal{H}}$ is lower bounded by the expression on the right of the inequalities. This proves Theorem 2. □

Proof: [Proof of Lemma 4] Let $\mathcal{G}$ consist of translated Heaviside functions, i.e. functions that are 1 above some threshold $\theta \in \boldsymbol{R}$, and 0 below the threshold: $\mathcal{G} = \{g_\theta : \boldsymbol{R} \mapsto \{0,1\} \mid g_\theta = \mathbf{1}_{t>\theta}, \theta \in \boldsymbol{R}\}$. This class can only shatter a singleton, so $d_{\mathcal{G}} = 1$. Now consider

$$\mathcal{H} = \{h_{L,\theta} : [0,1] \times \boldsymbol{R} \to \{0,1\} \mid h_{L,\theta} = \mathbf{1}_{x_2 > L(x_1)}, \text{ where } L : [0,1] \to \boldsymbol{R} \\ \text{is piecewise linear with } d-1 \text{ pieces, and } L(0) = 0 \in \boldsymbol{R}\}.$$

These maps assign label 1 to points above L and 0 to points below. Then $\mathcal{H}$ shatters the d-point set $\{(0,0), (1/d, 0), \ldots, (1,0)\}$ for example, and it cannot shatter any larger set. Indeed, no set S containing two distinct points (x_1, x_2) and (x_1, x_2') can be shattered by $\mathcal{H}$ because $\mathcal{G}$ cannot shatter two points $x_2 \neq x_2'$; on the other hand, if the points of S have d distinct x_1-coordinates, then a labelling that alternates (ordering the points by their x_1-coordinates) is, by induction, not achievable: one line is insufficient to shatter two points, and if $n-1$ linear pieces shattered n points with distinct x_1 then $n-2$ linear pieces would shatter the first $n-1$ points. To see the claimed Fisher-Neyman factorization, note that $h_\theta(x_1, x_2) = g_\theta(\phi(x_1, x_2))$ for $\phi : (x_1, x_2) \mapsto (x_2 - L(x_1)) \in \boldsymbol{R}$. □

The next two results establish the bounds of equation 1.

Proposition 6 *Let $\mathcal{G}$ be a class of functions $\mathcal{X} \to \{0,1\}$ with VC dimension D and let h^* be a function that is returned by an ERM algorithm for a sample $D \sim p^m$ with sample size m, then*

$$\boldsymbol{E}_{D \sim p^m}[\mathcal{R}(h^*)] \leq \frac{2d_{\mathcal{G}} \log(m) + 4}{m \log(2)}$$

This result can be derived from Devroye et al. (1996) Corollary 12.1.

Proposition 7 *Let $\mathcal{H}$ with be a class of functions with VC-dimension $d_{\mathcal{H}}$ and let $\mathcal{V}$ be the set of random variables for which the set of random variables (X, Y) for which the Bayes error is 0. Assume $\varepsilon \leq 1/4$, $n \geq 15$ and $n \leq (d-1)/(12\varepsilon)$ then for any learning algorithm $\mathcal{A}$ and sample $D = (X_1, Y_1, ..., X_m, Y_m)$*

$$\frac{1}{2}\left(1 - \frac{1}{d_{\mathcal{H}} - 1}\right)^m \leq \boldsymbol{E}_D[\mathcal{R}(\mathcal{A}(D))]$$

This is proved by a slight modification of the proof of Theorem 14.1 in Devroye et al. (1996). There are many other tighter upper and lower risk bounds, see Devroye et al. (1996). This one was chosen here for its simplicity.

3 Class of POMDPs with Active Measuring.

We will now define a class of POMDPs that involve “active measuring”, namely actions the agent can choose in order to more fully observe the state, typically at a cost. This setting is common in industrial planning and control.

Industrial/experimental control. When a chemist operates in a laboratory environment, they undertake a series of basic procedures (e.g. mixing chemicals, increasing the temperature of a heater, etc) which are aimed at modifying the molecular structure (e.g. breaking or forming new bonds) of their reactants. These elementary actions are often coupled together (through experience and training) as collective ones known as procedures. A procedure might be the distillation (separation) of two liquids which contain different chemical products, using a catalyst to drive a known reaction aimed at replacing a specific sub-unit (functionalization), or purification by some methods (e.g. filtering, centrifuge, etc).

Crucially, during a procedure, the chemist does not need to have a high fidelity view of the internal state of the molecule in order to carry out the individual steps. They will typically rely on simpler, lower information-content observations such as changes in temperature or color as feedback. This is possibly based on their experience (in the form of previous experiments and an accurate, internal physical model upon which they maintain an estimate of state). While our POMDPs are highly simplified, they nevertheless capture an important practical problem that occurs in many applications of robotics, industrial and experimental control: The situation where either high-cost, fully observable state information or lower-cost, partially observable state information can be accessed by the agent at each time step.

In these settings, a crucial aspect of planning involves deciding when to requisition a higher level of knowledge about the state of the underlying physical process and when to make do with "cheaper" partial knowledge. Sequential decision making processes of this form have previously been studied in Jaulmes et al. (2005); Armstrong-Crews & Veloso (2007); Bellinger et al. (2021), where fully observable state information is available at a cost via a special action that triggers an oracle to provide the current state of the environment. We argue the structure of the statistical model that governs complexity of the underlying process - specifically, the relative complexity of this original model vs. a factorized model that can be induced by partial observation - plays a role in the cost-benefit analysis of partial vs. full observation. Although the agent doesn't have access to this comparison, we will see how regret of various strategies employed by the agent depends on it.

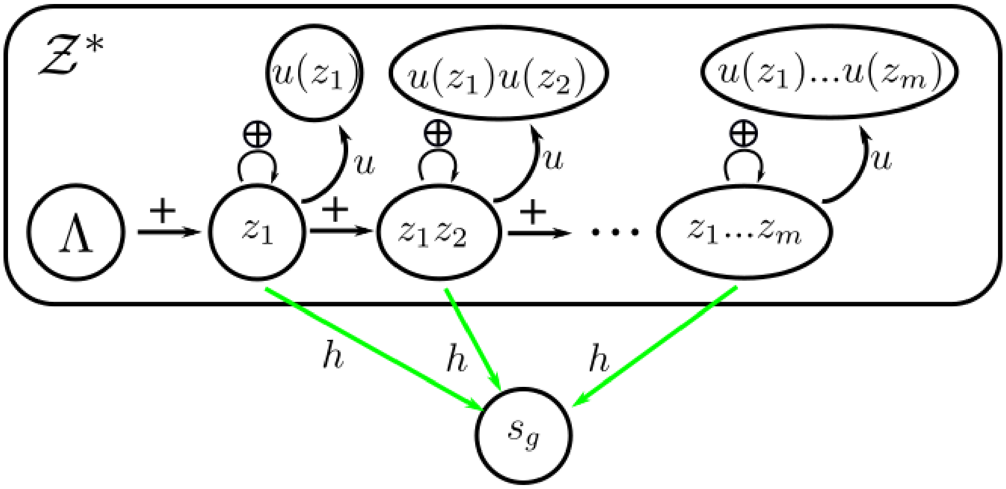

Figure 3: Class of POMDP considered. Dynamics are governed by a statistical model $\mathcal{P}$ on $\mathcal{X} \times \mathcal{Y}$ which factors to a simpler model. The underlying sufficient statistic ϕ is learnable from $x \in \mathcal{X}$. We assume $\mu < \nu$, so partial observation is cheaper than full.

To define our toy example, a class of POMDPs that formalizes a simple industrial control setting with active measuring, we will record values of relevant settings in a 7-tuple $V = (r, s, A, \epsilon, \mu, \nu, N) \in \mathbb{R}^6 \times \mathbb{N}$. The variables r, s are assumed to satisfy $0 < s < r < \frac{1}{2}$, $A \in \mathbb{R}$ will be a large number, while $\epsilon > 0$, $0 < \mu < \nu < 1$. The variable N is a natural number. Given such V, we now define an associated class $\mathcal{M}_V$ of POMDPs which we will use in the proof of Theorem 9. The values in V will serve as "knobs" that we turn during the proof, in order to highlight tradeoffs in the way relative costs and rewards affect the usefulness of temporal abstraction. More precisely, we will show that depending on the relative values of these settings, there will be a proveable gap between the performance of two agents $A1$ and $A2$ on the class $\mathcal{M}_V$, where $A1$ has oracle access to a certain macro, and $A2$ does not [3].

Definition 8 (Class of POMDPs) *See Figure 3. Let $V \in \mathbb{R}^6 \times \mathbb{N}$ be a 7-tuple as above. Then consider m, ϕ and a family $\mathcal{P}$ of underlying probability distributions p on $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ which corresponds to a specific SSL problem obtained for given r, s from Theorem 1 for Version (A), resp. Theorem 2 for Version (B). The member POMDPs in $\mathcal{M}_V$ are indexed by $p \in \mathcal{P}$ but are in all other respects identical. The state space is a disjoint union $\mathcal{S} = \{s_g\} \sqcup \mathcal{Z}^*$ with starting state $s_0 = \Lambda$, the empty string of elements from $\mathcal{Z}$. Each episode starts in state s_0 and ends at s_g. The set of observations is $\Omega = \{s_g\} \sqcup \mathcal{Z}^* \sqcup \mathcal{X}^*$, so as to allow for partial observation of states in $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ where only the $\mathcal{X}$-component is observed. The set of actions is $\mathcal{A} = \{\oplus, +\} \sqcup U \sqcup \mathcal{H}$. The* **action** *$+$ is interpreted as "pick a new partially observed sample point", meaning it has transition probability $P(sz'|s, +) = p(z')$ at any state s other than s_g (i.e. the new data point z' is appended to the existing, possibly empty, sample s), and observation probability $O(x_1 \cdots x_n | z_1 \cdots z_n), +) = 1$ at states $z_1 \cdots z_n = (x_1, y_1) \cdots (x_n, y_n) \in \mathcal{Z}^*$ (i.e. the sample's labels are hidden). It induces a reward μ. The* **action** *$\oplus$ is interpreted as "request full observation of current state", meaning it has transition probability $P(s|s, \oplus) = 1$ on any state s (i.e. $\oplus$ does not cause state transition) and observation probability $O(s|s, \oplus) = 1$ (i.e. the sample's labels are revealed). The task "pick a new fully observed sample point" can thus be accomplished by the composition of $+$ and $\oplus$. Likewise, to obtain a fully observed n-sample one may call $+$ n times*

[3] In the proof we will also comment on an even stronger assumption one could make for $A2$, namely that it does not have the control structure to carry out temporal abstraction even if it might have learned a specific sequence of actions of interest.

and then call $\oplus$. The action $\oplus$ is defined to produce a reward of $-n\nu$ when called on an n-sample; so, essentially, each revealed label costs ν (Note: there is thus no use in calling $\oplus$ early on in the sampling[4]).

Next, for every $h \in \mathcal{H}$ we define an **action h** $\in \mathcal{H}$ *which can be called from any state in $\mathcal{Z}^*$ and sends the agent to the terminal state s_g. We define $R(s_g, h) = r - \mathcal{R}(h)/A$ (this shrinks the contribution of the second term when A is a large constant). Finally,* **actions u** $\in U$ *are assumed to perform elementary computational operations on the data with transition and observation probabilities $P((u(x_1), y_1) \cdot (u(x_n), y_n)|(x_1, y_1) \cdot (x_n, y_n)) = 1$ and $O((u(x_1), y_1) \cdot (u(x_n), y_n)|(u(x_1), y_1) \cdot (u(x_n), y_n), u) = 1$. The action u is defined to produce a reward of $-n\epsilon$ when called on an n-sample; so essentially each individual computation $u(x_i)$ costs ϵ. Summarizing the reward structure:*

$$R(z, +) = -\mu, \quad R(z_1 \cdots z_n, \oplus) = -n\nu, \quad R(z_1 \cdots z_n, u) = -n\epsilon, \quad R(s, h) = r - \mathcal{R}(h)/A.$$

Thus, if ϕ is a composition of k basic computations $u \in U$, computing ϕ will cost $-nk\epsilon$, while the terminal action h has reward proportional to the negative risk of the hypothesis h.

The next theorem has two versions: (A) uses the SSL result Theorem 1 and (B) uses the SL result Theorem 2.

Theorem 9 *Fix some r, s such that $0 < s < r < \frac{1}{2}$ and let $m \in \boldsymbol{N}$ and $\phi : \mathcal{X} \to \mathcal{X}'$ be as in Theorem 1 for version (A), resp. Theorem 2 for version (B). There is a choice of ν, μ, ϵ, (A) $\in \boldsymbol{R}_{>0}$ and number of time steps N, defining the class $\mathcal{M}_V$ of POMDP, such that any agent A2 that is not given the representation ϕ as temporal abstraction will achieve negative return, whereas arbitrarily high positive return can be achieved by agent A1 that uses a macro encoding ϕ.*

Proof: Let $r > s > 0$. Consider $m \in \boldsymbol{N}$ and ϕ as stated in Theorem 1 for version (A), resp. Theorem 2 for version (B), and denote by B_ϕ an ERM-style algorithm with access to ϕ. Recall, Theorem 1 (unlike Theorem 2) says ϕ can be learned. Furthermore, suppose k is the number of basic operations needed to compute the representation $\phi(x)$ for any data point x.

At sample size m we know that r, s are lower and upper bounds respectively for the risks of the hypothesis returned by a purely supervised algorithm with no oracle knowledge vs an algorithm which first computes ϕ on the data points of the sample and then applies B_ϕ.

We wish to choose the parameters ν, μ, ϵ and N to achieve three properties:

(1) *The factored SSL problem using oracle knowledge of ϕ is easy enough that its RL version is rewarding*, i.e.

$$L := -m(\mu + \nu) - km\epsilon + r - s/A > 0. \tag{2}$$

The quantity L is a lower bound on the expected return under a policy that takes actions $+$ and then $\oplus$ m times (with reward $-m(\mu+\nu)$) to obtain labelled samples $z_1, \ldots, z_m$, and computes $\phi(z_i)$ for each of these (with reward $-km\epsilon$), then applies B_ϕ to produce a hypothesis h with $\mathcal{R}(h) < s$; this will bring the agent to the final state s_g with reward $R(s_g, h) = r - \mathcal{R}(h)/A > r - s/A$. Note: equation equation 2 is equivalent to: $m(\mu + \nu) + mk\epsilon < r - s/A$.

(2) *Full observation is expensive enough that when more than m fully labeled samples are taken the cost swamps out the possible benefit*. In other words, for even the lowest possible value of $\mathcal{R}(h)$, namely $\mathcal{R}(h) = 0$, $R(s_g, h)$ must be less than $(\mu + \nu)(m + 1)$:

$$r < (\mu + \nu)(m + 1). \tag{3}$$

(3) *Let N be a number of episodes large enough that*

$$NL >> M\mu. \tag{4}$$

Recall that M is an upper bound on the sample size of unlabeled data needed to learn ϕ, μ is the cost of each unlabelled data point and L is the per episode reward. We are requiring that N and L be large enough that after N episodes, the total reward NL exceeds the cost $M\mu$ of learning ϕ. Although $N \to \infty$ in lifelong learning, below we will discuss the return of agents after a number N of episodes satisfying (3), as well as longer term performance.

[4] This, and the strings from $\mathcal{Z}^*$ are downsides of our choice to stick with Markovian policies; if one allows non-Markovian policies then the agent can just recollect past states and observations, allowing the setting to be defined with states $z \in \mathcal{Z}$ and making the calling of $\oplus$ n times in sequence on single z_i a valid approach.

To achieve (1)-(3), we proceed as follows: By taking $\mu + \nu$ smaller and smaller, with $0 < \mu < \nu < 1$, one can arrange that $(m + 1)(\mu + \nu)$ be arbitrarily close to but just above r; in each case, by taking A sufficiently large one can simultaneously ensure that $\mu + \nu$ stays above $(1.1)s/A$ for example. Thus, eventually we can find a small enough $\mu + \nu$ and large enough A such that $m(\mu + \nu) < r - s/A$ while $(m + 1)(\mu + \nu) > r$. This guarantees (2) and if we pick ϵ sufficiently small (noting k and m are fixed) we also obtain (1). Now, since μ and M are fixed while $L > 0$ (by equation 2), we arrange (3) by taking sufficiently large N.

We can even accommodate a slow RL algorithm that takes many episodes before reaching peak performance of per-episode return L, incurring on all those episodes a total loss (negative return) of $-K$, simply by setting N even higher, namely high enough that the N' episodes at peak performance deliver $N'L >> K$.

From (1), we know the expected reward of an agent A1 that has access to a macro for computing ϕ is at least NL.

On the other hand, suppose agent A2 that doesn't know ϕ nevertheless has the control ability to execute actions in sequence as a macro. If it first applies the $+$ action M times to learn ϕ, records the needed sequence of U-actions for ϕ as m_ϕ, then takes $+$ and $\oplus$ m times and runs m_ϕ for each sampled (x, y) to obtain $(\phi(x), y)$, it can output h with reward $r - \mathcal{R}(h) > \delta > 0$. It therefore has expected cumulative reward $> -M\mu - m(\mu+\nu) - km\epsilon + \delta = -M\mu + L$ on 1st episode, and $\geq L$ on subsequent episodes, for an overall total that exceeds $-M\mu + NL$ across N episodes, which is higher than 0 by equation 4. The 1st episode reward may be $<< 0$; i.e., amount to a high cost C.

And finally, suppose A2 does not have the control structure to execute a sequence of actions, i.e. it is incapable of temporal abstraction. Notice that a macro records (represents) a sequence of actions, like a recipe, and thus cannot be replaced by a policy on primitive actions. Indeed each computational unit u involved in computing ϕ must be known in sequence and is not determined by the value the previous unit computed (i.e. the state the agent finds itself on). No agent can choose $h \in \mathcal{H}$ with risk $\mathcal{R}(h) < r$ if it has sampled m or fewer times. Indeed, if it could, then we could use it to define a supervised agent with risk less than r for sample complexity $< m$ and this is in contradiction with Theorem 1 and Theorem 2. This abstraction-less A2 agent therefore has expected cumulative reward below zero on *every* episode. On the other hand, if it instead samples more than m times it has a return $\leq -(m+1)(\mu+\nu)+r-\mathcal{R}(h)$ on the episode, and, since $\mathcal{R}(h) \geq 0$, this is $\leq -(m + 1)(\mu + \nu) + r$, which is negative by (2). Note that condition (2) means that sample size m is a point of diminishing return beyond which the gain in terms of lower risk h is swamped out by the added cost of further sampling. □

We have assumed there is a complexity-reducing sufficient statistic ϕ for all of the SSL problems in the given class. This implies having access to macros - either by being given suitable ϕ as inductive bias or being able to execute a macro m_ϕ after learning ϕ - is a provable advantage for an RL agent operating in this environment, compared to an agent without temporal abstraction. We summarize the various conclusions of the result:

- Specific agent $A1$: can achieve return at least $L > 0$ on every episode. Holds in Version (A) or (B).
- Specific agent $A2$ with temporal abstraction but no knowledge of ϕ: after learning ϕ in the first episode (at possibly great cost C), subsequent episodes (tasks) can have expected cumulative reward $L > 0$. Holds in Version (A).
- Arbitrary agent $A2$ without temporal abstraction: necessarily has expected cumulative reward < 0 on *every* episode (task). Holds in Version (A) or (B).
- Useful macro in Version (A) can be learned without calling the expensive $\oplus$ action of full observation; on the other hand, even if learning macros (in other settings) might require significant cost, they constitute a form of inductive bias that confers a benefit to agents using them and so their cost of acquisition can be amortized over future positive returns as done by the HRL-able A2.

All these phenomena occur in lifelong learning environments $\mathcal{M}_V$ for which the risk factorization r, s of dynamics is related as described to reward structure A, ϵ, μ, ν. The results for A1 and the non-HRL version of A2 are direct upper and lower bounds respectively on regret. The result for the HRL-able A2 is more subtle: the returns on future episodes need to be weighed against the initial cost C, and we consider amortizing C over N episodes of undiscounted returns (alternatively discounted computations could be done); this brings time, or rather effective lifetime, into the tradeoff.

4 Conclusions and further work

Beyond the class of POMDPs just given, we note that a form of complexity inspired by uniform shattering can be defined for general POMDP classes, not necessarily from SSL problems. This is ongoing work, that extends results in Létourneau & Fraser (2022), and it may enable a generalization of the results in the present paper. It differs from other recent RL notions of complexity such as (Lehnert et al., 2018; Jiang et al., 2015), in that it captures the difficulty of a *family* of (PO)MDPs whose transition dynamics are constrained by a statistical model (similar to, but more general than, $\mathcal{P}$ in our proof here). By contrast, the cited measures consider classes of optimal value functions, or of optimal policies, for arbitrary transition dynamics. This distinction is analogous to the one in supervised learning between distribution-constrained vs. distribution-free analysis.

In the present paper, we highlighted an analogy between options in RL that reduce regret and representations in (semi)supervised learning that reduce risk. This relationship arose by considering a class of RL problems that are built from POMDPs associated to SSL problems and we showed, for this class of RL problems, that agents with access to temporal abstraction have lower regret compared to agents that do not. Our class of RL problems also set up a correspondence between partial labeling in (semi)supervised learning and partial observation in RL. And finally, it made it possible to explore tradeoffs that govern planning when agents have access to active measuring. In our special setting, these tradeoffs showed that structure of the underlying statistical model together with relative costs of full vs partial observation, dictated the usefulness of temporal abstraction, which we expressed in agent-agnostic manner as the mentioned gap in regret with vs without temporal abstraction. Finally, another tradeoff in our active-measuring setting showed how the statistical model and cost structure also determine the scale of lifetime that a lifelong active-measuring agent would need in order to amortize the cost of acquiring a useful temporal abstraction.

References

G. Alain and Y. Bengio. What regularized auto-encoders learn from the data generating distribution. Technical report, 2012. http://arXiv:1211.4246[cs.LG].

Nicholas Armstrong-Crews and Manuela Veloso. Oracular partially observable markov decision processes: A very special case. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 2477–2482. IEEE, 2007.

Pierre-Luc Bacon and Doina Precup. A matrix splitting perspective on planning with options. *arXiv preprint arXiv:1612.00916*, 2016.

M.-F. Balcan and A. Blum. A pac-style model for learning from labeled and unlabeled data. In *Learning Theory*, volume 3559, pp. 111–126. Springer LNCS, 2005.

Peter L. Bartlett, Nick Harvey, Christopher Liaw, and Abbas Mehrabian. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. *Journal of Machine Learning Research*, 20(63):1–17, 2019. URL http://jmlr.org/papers/v20/17-612.html.

J. Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198, 2000.

M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

C. Bellinger, R. Coles, M. Crowley, and I. Tamblyn. Active measure reinforcement learning for observation cost minimization. In *Proceedings of the Canadian Conference on Artificial Intelligence*, 2021.

Y. Bengio, S. Bengio, and J. Cloutier. Learning a synaptic learning rule. In *IJCNN-91-Seattle International Joint Conference on Neural Networks*, volume ii, pp. 969 vol.2–, 1991. doi: 10.1109/IJCNN.1991.155621.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. URL http://arxiv.org/abs/1206.5538. cite arxiv:1206.5538.

E. Brunskill and L. Li. Sample complexity of multi-task reinforcement learning. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2013.

V. Castelli and T. M. Cover. The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter. *IEEE Transactions on Information Theory*, 42:2102?2117, 1996.

Christoph Dann, Tor Lattimore, and Emma Brunskill. Unifying pac and regret: Uniform pac bounds for episodic reinforcement learning, 2017. URL https://arxiv.org/abs/1703.07710.

P. Dayan and G.E. Hinton. Feudal reinforcement learning. In *NIPS 1998*, pp. 271–278, 1998.

L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*, volume 31 of *Applications of mathematics*. Springer, New York, 1996.

K. Frans, J. Ho, P. Abbeel, and J. Schulman. Meta learning shared hierarchies. Technical report, 2017. https://arxiv.org/pdf/1710.09767[cs.LG].

M. Fraser. Multi-step learning and underlying structure in statistical models. In *NIPS 2016*, pp. 4815–4823, 2016.

Robin Jaulmes, Joelle Pineau, and Doina Precup. Active learning in partially observable markov decision processes. In *European Conference on Machine Learning*, pp. 601–608. Springer, 2005.

Nan Jiang, Alex Kulesza, Satinder Singh, and Richard Lewis. The dependence of effective planning horizon on model accuracy. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '15, pp. 1181–1189, Richland, SC, 2015. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-1-4503-3413-6. URL http://dl.acm.org/citation.cfm?id=2772879.2773300.

M. Kearns and R. Schapire. Efficient distribution-free learning of probabilistic concepts. *Journal of Computer and System Sciences*, 48:464–497, 1994.

G. Konidaris and A. Barto. Building portable options: skill transfer in reinforcement learning. *IJCAI*, pp. 895–900, 2007.

Lucas Lehnert, Romain Laroche, and Harm van Seijen. On value function representation of long horizon problems. 2018.

Vincent Létourneau and Maia Fraser. Inexperienced rl agents can't get it right: lower bounds on regret at finite sample complexity. In *Proceedings of Conference on Lifelong Learning Agents - CoLLAs 2022*, 2022.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Mehryar Mohri Foundations of Machine Learning*. MIT Press, 2007. URL https://cs.nyu.edu/~mohri/mlbook/.

P. Niyogi. Manifold regularization and semi-supervised learning: Some theoretical analyses. *Journal of Machine Learning Research*, 14:1229–1250, 2013.

T.J. Perkins and D. Precup. Using options for knowledge transfer in reinforcement learning. Technical report, 1999. Technical Report UM-CS-99-34.

M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, 1994, 2005.

Jurgen Schmidhuber. Evolutionary principles in self-referential learning. on learning now to learn: The meta-meta-meta...-hook. Diploma thesis, Technische Universitat Munchen, Germany, 14 May 1987. URL http://www.idsia.ch/~juergen/diploma.html.

R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2018.

S.R. Sutton, D. Precup, and S. Singh. Between mdps and semi-mdps: a framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112:181–211, 1999.

G. Taraldsen. The factorization theorem for sufficiency. Technical report, Preprint. doi:10.13140/RG.2.2.15068.87687, 2022.

S. Thrun and L.Y. Pratt. *Learning to learn*. Kluwer Academic Publishers, 1998.

L. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

H. van Seijen, M. Fatemi, J. Romoff, R. Larcohe, T. Barnes, and J. Tsang. Hybrid reward architecture for reinforcement learning. Technical report, 2017a. https://arxiv.org/pdf/1706.04208[cs.LG].

H. van Seijen, M. Fatemi, J. Romoff, and R. Laroche. Separation of concerns in reinforcement learning. Technical report, 2017b. https://arxiv.org/pdf/1612.05159[cs.LG].

A.S. Vezhnevets, S. Osindero, T. Schaul, N. Heess, M. Jaderberg, D. Silver, and K. Kavukcuoglu. Feudal networks for hierarchical reinforcement learning. Technical report, 2017. https://arxiv.org/pdf/1703.01161[cs.LG].