Learning Interpretable Policies in Hindsight-Observable POMDPs through Partially Supervised Reinforcement Learning

February 15, 2024

Michael Lanier, Ying Xu, Nathan Jacobs, Chongjie Zhang, Yevgeniy Vorobeychik
Department of Computer Science & Engineering, Washington University in St. Louis

Abstract

Deep reinforcement learning has demonstrated remarkable achievements across diverse domains such as video games, robotic control, autonomous driving, and drug discovery. Common methodologies in partially-observable domains largely lean on end-to-end learning from high-dimensional observations, such as images, without explicitly reasoning about true state. We suggest an alternative direction, introducing the Partially Supervised Reinforcement Learning (PSRL) framework. At the heart of PSRL is the fusion of both supervised and unsupervised learning. The approach leverages a state estimator to distill supervised semantic state information from high-dimensional observations which are often fully observable at training time. This yields more interpretable policies that compose state predictions with control. In parallel, it captures an unsupervised latent representation. These two—the semantic state and the latent state—are then fused and utilized as inputs to a policy network. This juxtaposition offers practitioners a flexible and dynamic spectrum: from emphasizing supervised state information to integrating richer, latent insights. Extensive experimental results indicate that by merging these dual representations, PSRL offers a potent balance, enhancing model interpretability while preserving, and often significantly outperforming, the performance benchmarks set by traditional methods in terms of reward and convergence speed.

1 Introduction

The realm of deep reinforcement learning has been transformed through a series of advances that utilize deep neural networks for policy or value function approximations. Dominating the current Reinforcement Learning (RL) landscape are methodologies that view the world as a Markov Decision Process

(MDP)—treating sequences of observations as states and often learning policies directly from observable data, such as images translating to controls [Carlucho et al., 2018]. Such an end-to-end (E2E) approach has indeed set the benchmark in many domains, like robot navigation [Shi et al., 2019], Atari games [Mnih et al., 2013], human-level driving in Gran Turismo [Wurman et al., 2022], and numerous others.

However, many environments, especially those with high-dimensional sensory inputs, are better modeled as partially observable Markov decision processes (POMDPs). The so-called "true" or "semantic" state in these settings is often a relatively low-dimensional representation which encapsulates semantic information. For example, in autonomous driving, the true state includes critical variables such as the ego-vehicle position, velocity, acceleration, or locations of nearby obstacles. If we are to learn policies that map such semantic information to control, they are naturally more interpretable. However, we do not observe this state directly, but instead have a high-dimensional sensory data stream, such as from camera, LiDAR, and radar. An alternative to typical end-to-end learning that aims to capture some of this intuition has been to learn a low-dimensional latent state representation that is implicit in the raw sensory data [Guo et al., 2023b]. However, such representations are typically not interpretable, and don't meet the regulatory bar on explainable artificial intelligence in many countries [Yannella and Kagan, 2018].

Significantly, what all conventional end-to-end approaches, including those which infer latent state, fail to capture is that in many domains true semantic state information is *available during training, even while it is unobservable at execution time*. For example, this is a common setting in robotics and control research [Bansal et al., 2020, Bowman et al., 2017, Dean et al., 2020, Dong et al., 2017, Godbole and Subbarao, 2019, Kumar et al., 2013, Morton and Kochenderfer, 2017, Rafailov et al., 2021, Tang et al., 2018], and true state information is available by construction in any simulation-based learning environment. Several recent efforts have attempted to exploit this structure to improve the efficiency of reinforcement learning in POMDPs. Lee et al. [2023] consider a closely related model of *hindsight observability* in which information about semantic state is observed at the end of each episode, and develop model-based reinforcement learning approaches that achieve sample complexity that is linear in the number of actions and logarithmic in the size of the transition and observation function spaces. Pinto et al. [2018] and Baisero and Amato [2022] term this problem *asymmetric reinforcement learning* and leverage state information by training critics—but not actors—on states rather than observations. However, while Lee et al. offer a general-purpose algorithm, theirs is a custom model-based approach; further, they assume that once the POMDP has been learned, they can quickly solve it to optimality, making their approach impractical in settings with high-dimensional observations, such as images. On the other hand, while both Pinto et al. and Baisero and Amato offer practical algorithms, these are specific to actor-critic reinforcement learning frameworks. Recently, Baisero et al. [2022] proposed an asymmetric RL algorithm for deep Q-learning. However, this approach is specific to DQN-style algorithms [Farebrother et al., 2018, Wang et al.,

2016]. Thus, asymmetric RL methods are at the moment distinct for Q-learning and actor critic methods, and no practical unified (general-purpose) method exists for making use of known state information in reinforcement learning for POMDPs with high-dimensional observations. Moreover, existing asymmetric RL approaches ultimately still train policies that are end-to-end in the sense of mapping raw sensory inputs to controls. Consequently, concerns about the lack of interpretability of conventional E2E approaches, which yield complex "black-box" policies, apply to these as well.

To address this gap, we introduce a Partially Supervised Reinforcement Learning (PSRL) framework. In PSRL, we assume that training time trajectories include both observation and true state information, whereas only the former is available at execution time. PSRL has two main building blocks: state predictor g, which maps (finite) observation history to predicted state, and policy π, which maps predicted state to action. At training time, both g and π are jointly trained using a combination of supervised and reinforcement learning loss. Crucially, we train π with *predicted*, rather than actual state as an input, ensuring that the learned policy is robust to prediction errors.

At execution time, actions are chosen by composing π and g, thereby effecting an end-to-end policy that relies only on observations. Nevertheless, this explicit composition preserves a high degree of interpretability, as the policy makes a clear distinction between the prediction of a semantically meaningful state, and actions taken with such predictions as an input. Further, the PSRL framework admits a natural generalization that allows for state to be incompletely observed even at training time, whereby we additionally learn a very low-dimensional representation of the latent part of the state which is concatenated with predicted state as an input to π. We refer to this as PSRL-K, where K is the number of such latent dimensions. This generalization, in turn, induces a space of algorithmic approaches that span from fully end-to-end (ignoring supervised loss in training, with K just the dimension of latent state in end-to-end RL) to PSRL-0, with its added advantage of being interpretable due to the semantic nature of the state predicted by g. Thus, varying K induces a tradeoff between interpretability and ability to capture additional information from sensor data that may be relevant for control.

We investigate the efficacy of PSRL-K for varying K for both deep Q-learning (DQN) and actor-critic (PPO) approaches that instantiate the policy learning architecture and loss, on six domains in OpenAI Gym. Broadly, we find that PSRL-0 is usually, although not always, more sample efficient than state-of-the-art end-to-end counterparts, and the added value of increasing K is typically low. Moreover, our experiments demonstrate that both pre-training the state predictor g or the policy π—both common ways state information is used in specific applications—tends to perform poorly, in the former case because PSRL makes more efficient use of supervised data, and in the latter case because of fragility of policies learned using true, rather than predicted, state to prediction errors. Finally, our experiments demonstrate interpretability of composite policies learned by PSRL, showing that state prediction errors are typically far lower than unsupervised embeddings learned in conventional

3

end-to-end reinforcement learning paradigms.

Related Work Deep reinforcement learning has demonstrated remarkable success in settings in which input information is high-dimensional sensory data, such as images [Arulkumaran et al., 2017, Li, 2017, Mnih et al., 2013, Wang et al., 2018, Wurman et al., 2022]. Settings of this kind are naturally modeled as partially-observable Markov decision processes. However, since true state is often observable (albeit, not necessarily obviously) from sensory data, it has proved quite effective to simply treat such high-dimensional input as state, and apply standard deep RL methods to learn end-to-end (that is, raw data to control) policies [Mnih et al., 2013, Wurman et al., 2022]. Recently, however, this general tendency has come under scrutiny from two directions. First, in numerous applied settings, such as involving robotics and control applications, it is often natural to decompose end-to-end control into state inference and state-based control [Bansal et al., 2020, Bowman et al., 2017, Dean et al., 2020, Dong et al., 2017, Godbole and Subbarao, 2019, Kumar et al., 2013, Ma et al., 2021, Morton and Kochenderfer, 2017, Rafailov et al., 2021, Tang et al., 2018], although this decomposition is typically ad hoc and domain specific. For example, one may simply learn to predict state from observational data first (i.e., independently of control), and then uses such predictions to synthesize a controller (e.g., using conventional RL). However, such approaches can often fail due to prediction errors [Pinto et al., 2018]. The proposed PSRL framework can be viewed as unifying these disparate application-driven approaches, as well as providing a systematic means for studying them. Moreover, the joint supervised and RL training in PSRL overcomes the fragility of many such approaches to prediction errors.

Second, several lines of RL research have observed that one commonly observes true state of a POMDP during training. On the more theoretical side, this has been modeled as *hindside-observability* in POMDPs (HOMDPs), that is, true state can be observed after an action has been taken, or after an episode, during training, inference, or both [Guo et al., 2023a, Lee et al., 2023]. This work has demonstrated that observability of state (at least) during training can yield theoretical advantages, but did not yield practical algorithms for high-dimensional settings (e.g., most approaches assume a finite set of states). On the more practical side, several recent approaches proposed *asymmetric RL* [Baisero and Amato, 2022, Baisero et al., 2022, Pinto et al., 2017]. The key idea in asymmetric RL is to train a critic that depends directly on state, and an actor that is end-to-end. However, such approaches need to be designed separately for deep Q-learning and actor-critic RL frameworks, and still ultimately rely on learning end-to-end actors (for example, in the Q-learning context, the goal is to learn an actor which is, approximately, the expectation of the critic Q-function). PSRL provides a simple unified framework that is conceptually independent of the particular flavor of model-free RL, making use of supervised information provided by observability of state at training time.

Our work is closely related to ideas of representation learning [Bengio et al., 2013]. These works seek to understand the theoretical and practical impacts of the way networks use underlying representations of their input data, and the

4

way these representations are leverged by downstream tasks. He et al. [2023] demonstrated that for optimal policies learned through DQN, the primary Q network needs to learn distinguishable representation of the underlying input data. Le Lan et al. [2023] looked at how the latent representations change when auxiliary objectives are incorporated into the training process of temporal difference learning. Their work hypothesises the potential case where the representation is parameterized by a neural network, as we have done here.

Finally, our work is loosely related to model-based reinforcement learning, such as DREAMER, which also uses supervised techniques [Jin et al., 2019, 2021, Okada and Taniguchi, 2021, Wu et al., 2023, Yang and Wang, 2019]. However, in model-based RL, supervised techniques focus primarily on learning dynamics and rewards, whereas the propose PSRL framework uses supervision as a means to take advantage of observed state at training time in learning how to act in POMDPs.

2 Preliminaries

Partially Observable MDPs A partially-observable Markov decision process (POMDP) is characterized by a tuple: $\{S, A, T, O, \Omega, r, \gamma, \rho\}$, where S is the state space, A the action space, T the transition function where $T(s'|s, a)$ is the probability distribution over $s' \in S$ given true state s and action a, Ω the set of observations, O the observation function with $O(o|s)$ the distribution of observations $o \in \Omega$ given true state $s \in S$, $r(s, a)$ the reward function, $\gamma \in [0, 1)$ the temporal discount factor, and ρ the distribution over initial state s_0. Throughout, we assume that $S \subseteq \mathbb{R}^n$ and $\Omega \subseteq \mathbb{R}^m$, and that typically $m \gg n$, that is, the dimension of the raw sensory observation space is far greater than the dimension of the true state s, which furthermore is semantically meaningful (for example, position, velocity, and acceleration of a car). We allow A to be either finite or infinite and multi-dimensional.

In general, a policy in a POMDP can be a function of an arbitrary observation history. However, in practice it is typical to keep track of only a finite sequence of observations (say, the last L). For simplicity, and since an observation o_t can embed such a finite sequence, we aim to learn policies $\pi(o)$ that map an observation $o \in \Omega$ to an action $a \in A$. Formally, an optimal policy of this form solves the following problem:

$$\max_\pi \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | a_t = \pi(o_t) \right],$$

where the expectation is with respect to ρ as well as the distribution of trajectories induced by a policy π and transition function T.

Deep Reinforcement Learning When the POMDP model is given and the state, observation, and action sets are finite and relatively small, it is possible to compute an optimal policy relatively effectively Shani et al. [2013]. However, in many problem domains we are not given the POMDP model, but instead

have an environment—commonly, a simulator—that we can experiment with by taking actions a and observing rewards $r(s,a)$. In such cases, reinforcement learning (RL) provides a general framework for *learning* how to act, that is, for learning a policy π from experience. Modern approaches for reinforcement learning with states represented by real vectors rely on function approximation in which the value function, policy, or both are represented using deep neural networks. While most such approaches have been developed for MDPs (i.e., where state s is fully observable at both training and execution time), a common adaptation to POMDPs is to replace states s with (a sequence of) observations o, effectively treating o as state, and apply standard RL methods as if it where an MDP; prototypical examples of this are Mnih et al. [2013] and Wurman et al. [2022].

We consider two classes of reinforcement learning algorithms. The first involves variants of Q learning that are most common when the set of actions A is relatively small. The second class are policy gradient methods, including actor-critic approaches, most common when the set of actions is large or infinite.

In relatively general terms, deep reinforcement learning algorithms can be viewed as minimizing a composite loss function comprised of up to three elements: 1) critic loss $\mathcal{L}_c(\theta_c)$, 2) actor loss $\mathcal{L}_a(\theta_a)$, and 3) entropy $\mathcal{H}_e$:

$$\mathcal{L}_{RL}(\theta_c, \theta_a) = \mathbb{E}_{r,s,a,s',a'}[\mathcal{L}_c(\theta_c) + \alpha_1 \mathcal{L}_a(\theta_a) + \alpha_2 \mathcal{H}]. \tag{1}$$

In the case of deep Q-learning, $\alpha_1 = \alpha_2 = 0$, and in the most basic variant of deep Q network (DQN) Mnih et al. [2015], we learn the parametric action-value function $Q_{\theta_c}(s,a)$ using loss $\mathcal{L}_c(\theta_c) = (\bar{Q} - Q_{\theta_c}(s,a))^2$, where $\bar{Q} = r + \max_{a'} Q_\theta(s,a')$. Variations, such as double DQN (DDQN), involve a distinct Q network as a target in $\bar{Q}$ van Hasselt et al. [2015]. In the case of actor-critic methods, one commonly learns a value function $V_{\theta_c}(s)$ as the critic. In particular, in this case it is common to define the advantage function $A(s,a) = (\bar{V} - V_{\theta_c})$, where $\bar{V} = r + V_{\theta_c}(s')$, and the critic loss is just $\mathcal{L}_c(\theta_c) = A(s,a)^2$, while the actor loss is $\mathcal{L}_a(\theta_a) = A(s,a)\log \pi_{\theta_a}$, where π_{θ_a} is the policy (often represented as a neural network), as in the case of approaches as such advantage actor critic (A2C) Mnih et al. [2016], or

$$\mathcal{L}_a(\theta_a) = -A(s,a)\min\left\{\frac{\log \pi_{\theta_a}}{\log \pi_{\theta_{old}}},\right.$$
$$\left.\text{clip}(\frac{\log \pi_{\theta_a}}{\log \pi_{\theta_{old}}}, 1-\eta, 1+\eta)\right\},$$

as in PPO Schulman et al. [2017]. Finally, the entropy term $\mathcal{H}$ aims to ensure that the actor π remains stochastic during training to ensure exploration.

3   Approach

In a broad array of domains, it is reasonable to assume that even though the domain is partially observable, true state is observed *during training*, though
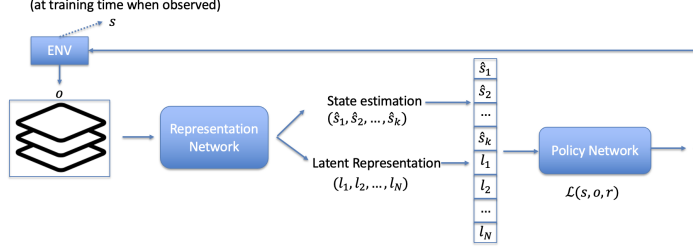
Figure 1: The PSRL-K framework.

not at execution time. For example, any simulation, by construction, must generate both true state and observations, while only emitting observations to mimic whatever domain is being simulated. Moreover, common robotic settings involve highly instrumented training runs (besides training using simulations) in which one can indeed closely approximate whatever actual state information is important. Formally, we assume that training trajectories $\tau = (s_0, o_0, a_0, s_1, o_1, a_1, \ldots, s_t, o_t, a_t, \ldots)$ include *both* observations o_t and states s_t at each time step t.

We propose to leverage knowledge of true state at training time by combining reinforcement learning, which learns how to act, with supervised learning, which learns to predict state s from observations o. Let $g_\phi(o)$ denote a parametric model that predicts state given observation o. In general, g_ϕ can be either a point prediction or a distribution. Here, we assume it makes a point prediction of a state s given an observation o as input. Recall that policy $\pi_{\theta_a}(o)$ takes an observation (or a sequence thereof) as an input. We refer to this as an *end-to-end* policy. The key idea, which is either implicitly or explicitly common in numerous particular cases , but has not previously been systematically investigated on its own, is to have a policy with an architecture that explicitly composes state prediction with decision. Formally, let $\tilde{\pi}_{\theta_a}(s)$ be a policy that maps the *true* state s (albeit, unobserved at decision time) to actions. We can then define $\pi(o) = \tilde{\pi}_{\theta_a}(g_\phi(o))$.

If both $\tilde{\pi}$ and g are differentiable, we can train the resulting policy end-to-end using the conventional RL loss $\mathcal{L}_{RL}(\theta_c, \theta_a)$ in Equation (1). And, in conventional RL settings, there is little else to do. However, when we observe state s during training, we can do more. Recent asymmetric RL approaches take advantage of such information by learning a critic value or action-value (Q) function as a function of true state, rather than observation, with the actor (policy) Baisero and Amato [2022], Baisero et al. [2022], Pinto et al. [2018]. However, actor policies in these methods are still learned end-to-end. We suggest that this is a missed opportunity for two reasons: first, because it does not take full advantage of the knowledge of true state at training time, and second, because distinct approaches are needed for actor-critic and DQN settings. We propose *partially-observable reinforcement learning (PSRL)* as a general framework for RL that leverages knowledge of true state of a POMDP during training. In its

most basic variant, we jointly train $\tilde{\pi}$ and g using a combination of supervised loss (for g) and RL loss (for both):

$$\mathcal{L}_{PSRL}(\theta_c, \theta_a, \phi) = \mathcal{L}_{RL}(\theta_c, \theta_a) + \beta \mathcal{L}_S(\phi), \qquad (2)$$

where $\mathcal{L}_S(\phi)$ is the supervised (e.g., cross-entropy) loss and β a parameter that determines the relative weight of the supervised loss to RL loss. Note that since the policy architecture composes the state-level policy $\tilde{\pi}$ and predictions g, effectively both the actor and the critic in actor-critic method is still trained with respect to observation input o, unlike asymmetric actor-critic, in which the critic takes true state as an input. A corresponding asymmetric variation of PSRL is straightforward. A key advantage of PSRL over asymmetric actor-critic, however, is its simplicity and generality; in particular, it applies equally directly to both Q learning and policy gradient methods. However, we now generalize it still further.

Suppose that we are in a situation where true state is only *partly* observed during training, or some of its variables are difficult to accurately predict from a finite sequence of observations. In such settings, we would expect end-to-end methods to be far better than PSRL. To address this, we propose a generalization of PSRL, PSRL-K, in which the policy $\tilde{\pi}$ takes as input (predicted) state s along with K latent variables trained solely using the RL loss. Formally, let $\tilde{\pi}(s, z)$, where z represents a latent low-dimensional observation embedding, and let $z = h_\psi(o)$ be a neural network architecture that captures this embedding. The full policy architecture is then $\pi(o) = \tilde{\pi}_{\theta_a}(g_\phi(o), h_\psi(o))$, still trained using the PSRL loss in Equation (2). We provide a schematic illustration of PSRL-K in Figure 1.

Note that PSRL-K generalizes both PSRL and end-to-end RL. In the former case, we omit the dependence on h_ψ (equivalently, set $K = 0$), while in the latter case, we omit the dependence on g_ϕ (and, therefore, the supervised part of PSRL loss). Consequently, it enables us to modulate between the two extremes, which our experiments show can be beneficial.

Finally, the proposed joint supervised and RL training in PSRL and its generalization is unlike a rather natural idea common in applied settings, where we train a state predictor g and independently a state-conditional policy $\tilde{\pi}$, and compose these at decision time. However, this approach is sensitive to state prediction errors. An improvement is to first pre-train g, and then train $\tilde{\pi} \circ g$ using either RL or PSRL loss. We explore these alternatives in the experiments below.

4   Experiments

We evaluate the performance of PSRL-K on several common benchmark environments from OpenAI Gym, comparing to state of the art end-to-end and asymmetric actor critic approaches, as detailed below. In three environments featuring small action sets, we evaluate the implementation of PSRL-K in deep Q-learning-based approaches (specifically, double deep Q-network (DDQN) [van
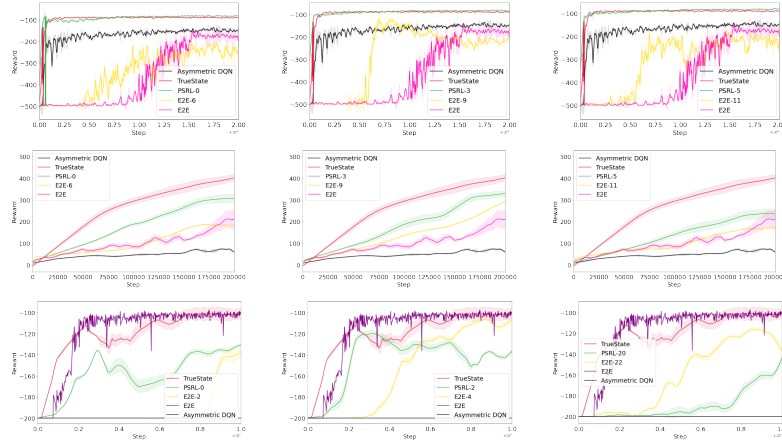
Figure 2: Experiments in Acrobot (top row), Cart Pole (with finite action sets; middle row), and Mountain Car (bottom row), in finite-action environments using DDQN approaches.
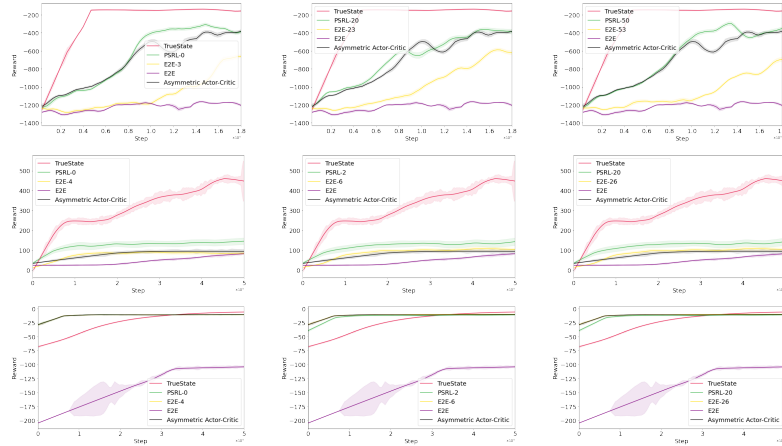

Figure 3: Experiments in Pendulum (top row), Cart Pole (with continuous action sets; middle row), and Reacher (bottom row), in continuous-action environments using PPO approaches.

Hasselt et al., 2015]). In three others that feature continuous actions, we combine PSRL-K with PPO.

4.1 Experiment Setup

Environments We consider five OpenAI Gym environments: Acrobot, Cart Pole, Mountain Car, Reacher, and Pendulum [OpenAI, 2023a,b,c,e,d]. Of these,
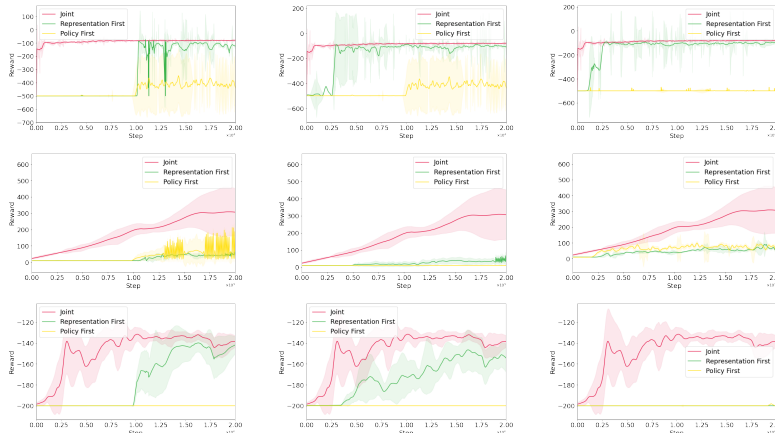
Figure 4: Experiments comparing PSRL-0 (joint RL and supervised) learning with either representation first or policy first approaches. Top row, left-to-right: Acrobot 50% pretrained, Acrobot 25% pretrained, Acrobot 10% pretrained. Middle row: Cartpole 50% pretrained, Cartpole 25% pretrained, Cartpole 10% pretrained. Bottom row: Mountain Car 50% pretrained, Mountain Car 25% pretrained, Mountain Car 10% pretrained.

Acrobot and Mountain Car have finite action sets, Reacher and Pendulum involve continuous actions, and we used two versions of Cart Pole, one with finite and another with continuous action sets. In Acrobot, (true) state is a 6-dimensional vector providing information about the two rotational joints and two links. In Mountain Car, state is represented by a 4-dimensional vector, comprising the cart position, cart velocity, pole angle, and pole angular velocity. In Pendulum state is a 3-dimensional vector, representing the cosine and sine of the pendulum's angle and its angular velocity. In Cart Pole is a 4-dimensional vector, comprising the cart position, cart velocity, pole angle, and pole angular velocity. Finally, in Reacher the state is given by an 11-dimensional vector comprising the cosine and sine values for the angles of both arms, the coordinates of the target, the angular velocities of the arms, and the vector between the target and the reacher's fingertip. In all cases, observations o are short sequences of 2D images simulated by OpenAI Gym.

Baselines Our first baseline is largely for calibration, and assumes that the environment is fully observable. This *true state* baseline simply learns the policy $\pi(s)$ using either DDQN (in finite-action settings) or PPO (in continuous-action settings). The second baseline is end-to-end (*E2E*), for which we learn policies $\pi(o)$ directly mapping image inputs to control using either DDQN or PPO as conventionally done. For each environment, we use the best performing E2E baseline available. To ensure a fair comparison with PSRL-K, we additionally provide results for E2E-$K + n$, where the architecture of $\pi(o)$ mirrors that of PSRL-K, except the policy is learned using solely the RL loss. Finally, we
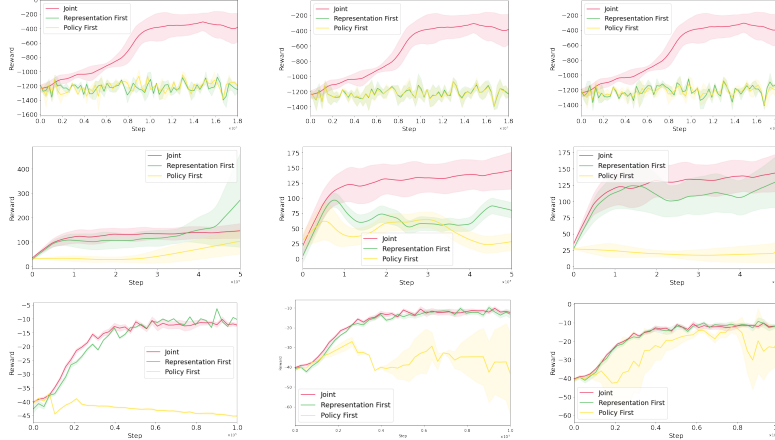
Figure 5: Experiments comparing PSRL-0 (joint RL and supervised) learning with either representation first or policy first approaches. Top row, left-to-right: Pendulum 50% pretrained, Pendulum 25% pretrained, Pendulum 10% pretrained. Middle row: Continuous Cartpole 50% pretrained, Continuous Cartpole 25% pretrained, Continuous Cartpole 10% pretrained. Bottom row: Reacher 50% pretrained, Reacher 25% pretrained, Reacher 10% pretrained.

compare with a state-of-the-art asymmetric DQN ($ADQN$) [Baisero et al., 2022] in the finite-action settings and asymmetric PPO ($APPO$) [Baisero and Amato, 2022] in the continuous-action settings.

4.2   Results

Q-Learning Settings Our first set of experiments considers the DDQN framework, comparing PSRL with the baselines. The results are presented in Figure 2. Consider first the top row, corresponding to Acrobot. In this setting, PSRL is essentially as effective as learning in its fully observable counterpart. In contrast, both variants of end-to-end learning considerably slower, even as they ultimately approach a near-optimal policy. Finally, we observe that ADQN significantly outperforms E2E approaches, but remains clearly below PSRL in terms of performance.

The second row of Figure 2 presents DDQN results for the finite-action Cart Pole environment. Here, PSRL again exhibits a clear advantage over end-to-end approaches as well as asymmetric DQN, although in this domain there is a clear advantage in the knowledge of true state. Notably, in both the Acrobot and Cart Pole domain, there appears to be little advantage to PSRL-K for $K > 0$, with PSRL-0 already exhibiting strong performance.

Finally, the last row of Figure 2 presents the results for Mountain Car. In this domain, we find that end-to-end performs extremely well, essentially no different from using the true state. PSRL-0 is tangibly worse, but in this case

PSRL-2 performs better.

Actor-Critic Settings Next, we consider settings with continuous actions, combining PSRL and baselines with the PPO actor-critic approach. The results are provided in Figure 3. In the Pendulum environment (top row), both asymmetric actor-critic and PSRL outperform end-to-end approaches by a large margin, although learn tangibly slower than true-state PPO. Moreover, PSRL-0 learns somewhat faster than asymmetric actor-critic. In the Cart Pole environment (bottom row), the advantage of PSRL and asymmetric methods over end-to-end is smaller. Indeed, in this case, asymmetric actor critic no longer outperforms one of the end-to-end baselines. PSRL, however, maintains an advantage over both. In both of these domains, PSRL-0 performance is comparable to PSRL-K for several values of K; that is, no added value is provided by including latent features. Finally, in the Reacher environment, PSRL and APPO are both comparable and actually learn somewhat faster than the approach which knows the true state (perhaps because of the implicit increase in exploration that results from imperfect state predictions), whereas end-to-end approaches are considerably worse.

Ablation Experiments Now, we consider two natural ablations of PSRL. The first ablation involves pretraining state prediction (representation network) g_ϕ using only supervised loss for the first T iterations. Thereafter, we freeze the representation network g_ϕ, and train only the policy network $\tilde{\pi}(g_\phi(o))$ using conventional RL loss. This roughly mirrors the typical way that state information in reinforcement learning is used in applied settings, such as robotic control [Bansal et al., 2020, Godbole and Subbarao, 2019, Morton and Kochenderfer, 2017, Tang et al., 2018]. We refer to this as *representation-first PSRL*. The second ablation first pretrains a policy mapping true state to actions for the first T iterations using RL loss. Thereafter, we freeze the policy network $\tilde{\pi}$, and train the state prediction g_ϕ using supervised loss. We refer to this as *policy-first PSRL*. This corresponds to an effective decomposition of prediction and policy training, also common in applied settings.

The results are provided in Figure 4 for discrete action environments and Figure 5 for continuous action environments. We can observe that neither the representation first nor the policy first baselines are as effective as PSRL in the majority of cases. However, there are a few interesting exceptions. While the policy first baseline is nearly always poor, representation first is effective in Continuous Cart Pole and Reacher, indistinguishable from PSRL in the latter, and actually outperforming it under one configuration in the former. Interestingly, these exceptions are only in the PPO setting, whereas the joint representation and policy training in PSRL appears uniformly more advantageous than these variations in the DQN setting.

Interpretability Finally, we consider the issue of interpretability, which we quantify simply as the quality of true (semantic) state estimation, comparing PSRL (that is, the predictions by g) with E2E with respect to the embedding of the identical dimension. One may expect that since E2E approaches typically learn to be quite effective, they do so by learning a good estimate of true state despite not having any explicit supervision. Our results in Table 1 dispel this

Table 1: Comparison of PSRL-0 and E2E test mean squared error for different environments.

Environment	PSRL-0	E2E
Acrobot	0.78 ± 0.27	16260 ± 7435
Mountain Car	0.0003 ± 0.0001	337 ± 37
Cartpole	0.20 ± 0.005	56 ± 5.6
Reacher	0.30 ± 0.13	0.88 ± 0.13
Cont Cartpole	0.05 ± 0.001	805 ± 146
Pendulum	1.04 ± 0.18	134 ± 21

notion, showing that while PSRL yields highly accurate state predictions, latent embedding in E2E is entirely uninterpretable, with distance to true state often many orders of magnitude higher than that for PSRL.

5   Conclusion

The PSRL framework reveals its effectiveness as a compelling alternative for addressing the challenges posed by important classes of partially observable domains, striking a balance between performance and interpretability. Through the fusion of supervised and unsupervised learning components, PSRL leverages a state estimator to distill semantic information from high-dimensional observations, providing interpretability in the inferred state. The dynamic spectrum created by the juxtaposition of the semantic and latent states offers practitioners a flexible approach, allowing them to navigate between emphasizing supervised state information and integrating richer latent insights. Our experimental results underscore the versatility and efficacy of PSRL across diverse domains, commonly outperforming traditional methods in terms of reward and convergence speed.

In the broader context of AI, where interpretability and performance often present conflicting objectives, PSRL emerges as a promising methodology. As we continue to advance our understanding and capabilities in complex, partially observable environments, further research and refinement of PSRL could contribute significantly to the evolution of deep reinforcement learning paradigms in practical applications. Our findings suggest that PSRL represents an important step towards achieving a more interpretable and effective approach to deep reinforcement learning.

Acknowledgement

References

Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017.

Andrea Baisero and Christopher Amato. Unbiased asymmetric reinforcement learning under partial observability. In *International Conference on Autonomous Agents and Multiagent Systems*, 2022.

Andrea Baisero, Brett Daley, and Christopher Amato. Asymmetric DQN for partially observable reinforcement learning. In James Cussens and Kun Zhang, editors, *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pages 107–117. PMLR, 01–05 Aug 2022. URL `https://proceedings.mlr.press/v180/baisero22a.html`.

Somil Bansal, Varun Tolani, Saurabh Gupta, Jitendra Malik, and Claire Tomlin. Combining optimal control and learning for visual navigation in novel environments. In *Conference on Robot Learning*, pages 420–429, 2020.

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. doi: 10.1109/TPAMI.2013.50.

Sean L Bowman, Nikolay Atanasov, Kostas Daniilidis, and George J Pappas. Probabilistic data association for semantic slam. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 1722–1729. IEEE, 2017.

Ignacio Carlucho, Mariano De Paula, Sen Wang, Yvan Petillot, and Gerardo G Acosta. Adaptive low-level control of autonomous underwater vehicles using deep reinforcement learning. *Robotics and Autonomous Systems*, 107:71–86, 2018.

Sarah Dean, Nikolai Matni, Benjamin Recht, and Vickie Ye. Robust guarantees for perception-based control. In *Learning for Dynamics and Control*, pages 350–360, 2020.

Chiyu Dong, John M Dolan, and Bakhtiar Litkouhi. Intention estimation for ramp merging control in autonomous driving. In *2017 IEEE intelligent vehicles symposium (IV)*, pages 1584–1589, 2017.

Jesse Farebrother, Marlos C Machado, and Michael Bowling. Generalization and regularization in dqn. *arXiv preprint arXiv:1810.00123*, 2018.

Ameya R Godbole and Kamesh Subbarao. Nonlinear control of unmanned aerial vehicles with cable suspended payloads. *Aerospace Science and Technology*, 93:105299, 2019.

Jiacheng Guo, Minshuo Chen, Huan Wang, Caiming Xiong, Mengdi Wang, and Yu Bai. Sample-efficient learning of pomdps with multiple observations in hindsight, 2023a.

Jiacheng Guo, Zihao Li, Huazheng Wang, Mengdi Wang, Zhuoran Yang, and Xuezhou Zhang. Provably efficient representation learning with tractable planning in low-rank pomdp. In *International Conference on Machine Learning*, 2023b.

Q. He, H. Su, J. Zhang, and X. Hou. Frustratingly easy regularization on representation can boost deep reinforcement learning. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20215–20225, Los Alamitos, CA, USA, jun 2023. IEEE Computer Society. doi: 10.1109/CVPR52729.2023.01936. URL `https://doi.ieeecomputersociety.org/10.1109/CVPR52729.2023.01936`.

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I. Jordan. Provably efficient reinforcement learning with linear function approximation, 2019.

Chi Jin, Qinghua Liu, and Sobhan Miryoosefi. Bellman eluder dimension: New rich classes of rl problems, and sample-efficient algorithms. *Advances in Neural Information Processing Systems*, 2021.

Puneet Kumar, Mathias Perrollaz, Stéphanie Lefevre, and Christian Laugier. Learning-based approach for online lane change intention prediction. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 797–802, 2013.

Charline Le Lan, Stephen Tu, Mark Rowland, Anna Harutyunyan, Rishabh Agarwal, Marc G Bellemare, and Will Dabney. Bootstrapped representations in reinforcement learning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 18686–18713. PMLR, 23–29 Jul 2023. URL `https://proceedings.mlr.press/v202/le-lan23a.html`.

Jonathan Lee, Alekh Agarwal, Christoph Dann, and Tong Zhang. Learning in pomdps is sample-efficient with hindsight observability. In *International Conference on Machine Learning*, pages 18733–18773, 2023.

Yuxi Li. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*, 2017.

Xiaobai Ma, Jiachen Li, Mykel J Kochenderfer, David Isele, and Kikuo Fujimura. Reinforcement learning for autonomous driving with latent state inference and spatial-temporal relationships. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6064–6071. IEEE, 2021.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning, 2016.

Jeremy Morton and Mykel J Kochenderfer. Simultaneous policy learning and latent state inference for imitating driver behavior. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, 2017.

Masashi Okada and Tadahiro Taniguchi. Dreaming: Model-based reinforcement learning by latent imagination without reconstruction. In *2021 ieee international conference on robotics and automation (icra)*, pages 4209–4215, 2021.

OpenAI. Acrobot-v1. `https://www.gymlibrary.dev/environments/classic_control/acrobot`, 2023a. Accessed: 2023-04-12.

OpenAI. CartPole-v1. `https://gym.openai.com/envs/CartPole-v1/`, 2023b. Accessed: 2023-04-12.

OpenAI. MountainCar-v0. `https://www.gymlibrary.dev/environments/classic_control/mountain_car/`, 2023c. Accessed: 2023-04-12.

OpenAI. Pendulum-v1. `https://www.gymlibrary.dev/environments/classic_control/pendulum/`, 2023d. Accessed: 2023-04-12.

OpenAI. Reacher-v2. `https://www.gymlibrary.dev/environments/mujoco/reacher/`, 2023e. Accessed: 2023-04-12.

Lerrel Pinto, Marcin Andrychowicz, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. Asymmetric actor critic for image-based robot learning, 2017.

Lerrel Pinto, Marcin Andrychowicz, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. Asymmetric actor critic for image-based robot learning. In *Robotics: Science and Systems*, 2018.

Rafael Rafailov, Tianhe Yu, Aravind Rajeswaran, and Chelsea Finn. Offline reinforcement learning from images with latent space models. In *Learning for Dynamics and Control*, pages 1154–1168, 2021.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

Guy Shani, Joelle Pineau, and Robert Kaplow. A survey of point-based pomdp solvers. *Autonomous Agents and Multi-Agent Systems*, 27:1–51, 2013.

Haobin Shi, Lin Shi, Meng Xu, and Kao-Shing Hwang. End-to-end navigation strategy with deep reinforcement learning for mobile robots. *IEEE Transactions on Industrial Informatics*, 16(4):2393–2402, 2019.

Sarah Tang, Valentin Wüest, and Vijay Kumar. Aggressive flight with suspended payloads using vision-based control. *IEEE Robotics and Automation Letters*, 3(2):1152–1159, 2018.

Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning, 2015.

Sen Wang, Daoyuan Jia, and Xinshuo Weng. Deep reinforcement learning for autonomous driving. *arXiv preprint arXiv:1811.11329*, 2018.

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*, pages 1995–2003. PMLR, 2016.

Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. Daydreamer: World models for physical robot learning. In *Conference on Robot Learning*, pages 2226–2240, 2023.

Peter R Wurman, Samuel Barrett, Kenta Kawamoto, James MacGlashan, Kaushik Subramanian, Thomas J Walsh, Roberto Capobianco, Alisa Devlic, Franziska Eckert, Florian Fuchs, et al. Outracing champion gran turismo drivers with deep reinforcement learning. *Nature*, 602(7896):223–228, 2022.

Lin Yang and Mengdi Wang. Sample-optimal parametric q-learning using linearly additive features. In *International Conference on Machine Learning*, pages 6995–7004. PMLR, 2019.

Philip N. Yannella and Odia Kagan. Analysis: Article 29 working party guidelines on automated decision making under GDPR. *Cyber Adviser*, 2018.