Asymmetric Actor-Critic with Approximate Information State

Amit Sinha and Aditya Mahajan

***Abstract*— Applying reinforcement learning (RL) to partially observable Markov decision processes (POMDPs) is a challenging problem because decisions need to be made based on the entire history of observations and actions. However, in several scenarios, we may have state information available during the training phase only. We are interested in exploiting the availability of this state information during the training phase to efficiently learn a history-based policy using RL. Specifically, we consider actor-critic algorithms in RL to incorporate this paradigm. This involves using only history information for the actor, and history and state information for the critic. For this reason, it is referred to in the literature as an asymmetric actor-critic approach. Motivated by the recent success of using representation losses in RL for POMDPs [1], we derive similar theoretical results for the asymmetric actor-critic case and are interested in evaluating whether adding such auxiliary losses in experiments helps in performance. Towards this goal, we learn a representation for the history—called an approximate information state (AIS)—that we prove is approximately optimal to the degree of approximation in the model by certain specified metrics. Additionally, we provide an explanation on why the policy gradient using a state-history critic has superior performance compared to that of a history-only critic.**

I. Introduction

A. Motivation

Partially observable Markov decision processes (POMDPs) are a more powerful modeling tool than Markov decision processes (MDPs) as they allow for the possibility of a hidden state which is not seen by the decision maker. This feature is useful while modeling many real-world applications such as autonomous driving [2], quantitative trading [3], energy systems [4], robotics [5] etc. However, this modeling power comes at a cost, as solving POMDPs is computationally much harder than solving MDPs [6]–[10]. Recently, reinforcement learning (RL) has emerged as a powerful tool to solve high-dimensional POMDP models [11]–[20].

In many instances, the RL algorithm uses a simulation environment. In such settings, the state of the system is available during the learning phase and can be used to speed up learning as long as the learned policy does not use the additional state information. Typically, the additional state is exploited in actor-critic algorithms by constructing a critic which has access to the additional state, but restricting attention to policies which do not use the additional state. Such algorithms are called *asymmetric* actor-critic due to the asymmetry of the information available to the actor and critic. There have been a series of recent papers which have shown that such asymmetric actor-critic algorithms significantly speed up the learning process [14], [15], [21]–[26].

It was shown in [21], [23] that using critics with just the state information significantly improves empirical performance. However, it was shown in [14] that the value functions defined in [21] are generally ill-defined, and that even when they are well defined, then the policy gradient may be biased. A stronger theoretical basis for incorporating state information into the critic was presented in [14], where a variation of asymmetric actor-critic with well-defined value functions and unbiased policy gradient was presented.

There have also been some recent results for vanilla actor-critic methods, which show that adding representation learning losses as an auxiliary loss in RL for POMDPs improves learning [1], [27]–[30]. So, a natural question is whether adding similar representation learning losses in asymmetric actor-critic improves learning. We investigate this question in this paper.

Our main contribution is to develop a theoretical framework for characterizing the representation loss in state-based dynamic programs for POMDPs. To do so, we propose a notion of approximate information state (AIS), which is motivated by the notion of AIS presented in [1], but has some differences because of the presence of state in the action-value function. We also provide an explanation of why having additional information at the critic improves performance of actor-critic algorithms. Such an explanation was missing from the literature. Finally, we propose a reinforcement learning algorithm which uses the AIS losses as auxiliary losses and present a detailed experimental study to compare the performance of the proposed algorithm with vanilla asymmetric actor-critic. Our experiments show that there is no significant improvement in performance due to the addition of AIS-losses. This suggests that unlike symmetric actor-critic, where adding AIS losses provided significant performance improvement, adding AIS losses does not provide significant improvement when the full state information is available to the critic.

B. Notation Used

We use uppercase letters to denote random variables (e.g. X, Y, etc.), lowercase letters to denote their realizations (e.g. x, y, etc.) and sans serif letters to denote sets (e.g. X, Y, etc.). We also use subscripts (e.g. X_t, Y_t, etc.) to a variable at time t. We use $\Delta(\mathsf{X})$ to denote the space of probability measures on a set X and use $\mathbb{P}$ and $\mathbb{E}$ to denote the probability of an event and the expectation of a random variable, respectively. We use $\|x\|$ to denote the norm of a vector x.

The authors are with the Department of Electrical and Computer Engineering, McGill University, Montreal amit.sinha@mail.mcgill.ca, aditya.mahajan@mcgill.ca

This was supported in part by the NSERC International Catalyst Grant AALRP 571054-21.

Given a set S and a function $f\colon \mathsf{S} \to \mathbb{R}$, we use $\operatorname{span}(f)$ to denote the span of f, i.e., $\operatorname{span}(f) = \sup_{s,s' \in \mathsf{S}} |f(s) - f(s')|$ and we use $\|f\|_\infty$ to denote the supremum norm of function f, i.e., $\|f\|_\infty = \sup_{s \in \mathsf{S}} f(s)$.

Given a metric space (S, d) and a function $f\colon \mathsf{S} \to \mathbb{R}$, we use $\operatorname{Lip}(f)$ to denote the Lipschitz constant of f, i.e.,

$$\operatorname{Lip}(f) = \sup_{s,s' \in \mathsf{S}} \frac{|f(s) - f(s')|}{d(s,s')}.$$

II. Background

A. Background on POMDPs

A partially observable Markov decision process (POMDP) is a tuple $\langle \mathsf{S}, \mathsf{Y}, \mathsf{A}, P^{\mathsf{S}}, P^{\mathsf{Y}}, r, \mathsf{T}, \gamma \rangle$, where

- S denotes the state space, Y denotes the observation space and A denotes the action space. Moreover, $S_t \in \mathsf{S}$, $Y_t \in \mathsf{Y}$, $A_t \in \mathsf{A}$ denote the state, action and observation, respectively, at time t.
- $P^{\mathsf{S}}\colon \mathsf{S} \times \mathsf{A} \to \Delta(\mathsf{S})$ is the transition dynamics of the state, i.e., for any realization $s_{1:t}$ of $S_{1:t}$, $y_{1:t}$ of $Y_{1:t}$, $a_{1:t}$ of $A_{1:t}$ and any Borel subset B of S, we have

$$\begin{aligned}&\mathbb{P}(S_{t+1} \in \mathsf{B} \mid S_{1:t} = s_{1:t}, Y_{1:t} = y_{1:t}, A_{1:t} = a_{1:t}) \\ &\quad = P^{\mathsf{S}}(\mathsf{B} \mid s_t, a_t).\end{aligned}$$

- $P^{\mathsf{Y}}\colon \mathsf{S} \times \mathsf{A} \to \Delta(\mathsf{Y})$ is the observation channel, i.e., for any realization $s_{1:t}$ of $S_{1:t}$, $y_{1:t-1}$ of $Y_{1:t-1}$, $a_{1:t-1}$ of A_{t-1} and any Borel subset B of Y, we have

$$\begin{aligned}&\mathbb{P}(Y_t \in \mathsf{B} \mid S_{1:t} = s_{1:t}, Y_{1:t-1} = y_{1:t-1}, A_{1:t-1} = a_{1:t-1}) \\ &\quad = P^{\mathsf{Y}}(\mathsf{B} \mid s_t, a_{t-1}).\end{aligned}$$

- $r : \mathsf{S} \times \mathsf{A} \mapsto \mathbb{R}$ is the per-step reward function. The reward at time step t is a random variable $R_t = r(S_t, A_t)$.
- T denotes the horizon for which the system runs.
- $\gamma \in (0, 1]$ denotes the discount factor.

It is sometimes useful to work with the conditional distribution of observation given the state and the actions, which we denote by $P^{\mathsf{S},\mathsf{Y}}$ and which is given as follows: for Borel subset B of Y, we have

$$\begin{aligned}&P^{\mathsf{S},\mathsf{Y}}(Y_{t+1} \in \mathsf{B} \mid s_t, a_t) \\ &\quad \coloneqq \int_{\mathsf{S}} P^{\mathsf{Y}}(Y_{t+1} \in \mathsf{B} \mid s_t, s_{t+1}, a_t) P^{\mathsf{S}}(ds_{t+1} \mid s_t, a_t) \\ &\quad = \int_{\mathsf{S}} P^{\mathsf{Y}}(Y_{t+1} \in \mathsf{B} \mid s_{t+1}, a_t) P^{\mathsf{S}}(ds_{t+1} \mid s_t, a_t).\end{aligned}$$

The standard solution method for POMDPs is to construct a belief space and write a dynamic program in terms of the belief space. It is well established that belief is a sufficient statistic for optimality [10], [31]. However, for our results, it is more convenient to work with the entire history instead of the belief space. For that matter, let $H_t = (Y_{1:t}, A_{1:t-1})$ denote the history of observations and actions until time t and let $\mathsf{H}_t = \mathsf{Y}^t \times \mathsf{A}^{t-1}$ denote the space of realizations of all histories until time t. Let $\pi = (\pi_1, \dots, \pi_{\mathsf{T}})$ be any history dependent randomized policy, i.e., $\pi_t\colon \mathsf{H}_t \to \Delta(\mathsf{A})$ and the action at time t is chosen according to $A_t \sim \pi_t(H_t)$. Let

$$V_t^\pi(h_t) \coloneqq \mathbb{E}^\pi \left[\sum_{\tau=t}^{\mathsf{T}} \gamma^{\tau-t} R_\tau \,\middle|\, H_t = h_t \right] \tag{1}$$

denote the performance of policy π from time t on wards, when starting at history $h_t \in \mathsf{H}_t$. The function V^π is also called the value function of policy π and it satisfies the following dynamic program: $V^\pi_{\mathsf{T}+1} \equiv 0$ and for $t \in \{\mathsf{T}, \dots, 1\}$, we have

$$\begin{aligned}Q_t^\pi(h_t, a_t) &= \int_{\mathsf{S}} r(s_t, a_t) \mathbb{P}(ds_t \mid h_t) \\ &\quad + \gamma \int_{\mathsf{S}} \int_{\mathsf{Y}} V^\pi_{t+1}(h_{t+1}) P^{\mathsf{S},\mathsf{Y}}(dy_{t+1} \mid s_t, a_t) \mathbb{P}(ds_t \mid h_t)\end{aligned} \tag{2}$$

and

$$V_t^\pi(h_t) = \sum_{a_t \in \mathsf{A}} \pi(a_t | h_t) Q_t^\pi(h_t, a_t). \tag{3}$$

Let Π denote the set of all randomized history dependent policies. A policy $\pi^\star \in \Pi$ is called *optimal* if

$$V_1^{\pi^\star}(h_1) \geq V_1^\pi(h_1), \quad \forall \pi \in \Pi, \forall h_1 \in \mathsf{H}_1.$$

Let $V_t^\star\colon H_t \to \mathbb{R}$ denote the performance of any optimal policy. The function $V_t^\star$ is also called the optimal value function and it satisfies the following dynamic program: $V^\star_{\mathsf{T}+1} \equiv 0$ and for $t \in \{\mathsf{T}, \dots, 1\}$, we have

$$\begin{aligned}Q_t^\star(h_t, a_t) &= \int_{\mathsf{S}} r(s_t, a_t) \mathbb{P}(ds_t \mid h_t) \\ &\quad + \gamma \int_{\mathsf{S}} \int_{\mathsf{Y}} V^\star_{t+1}(h_{t+1}) P^{\mathsf{S},\mathsf{Y}}(dy_{t+1} \mid s_t, a_t) \mathbb{P}(ds_t \mid h_t)\end{aligned} \tag{4}$$

and

$$V_t^\star(h_t) = \max_{a_t \in \mathsf{A}} Q_t^\star(h_t, a_t). \tag{5}$$

B. Background on integral probability metrics

Our results rely on a class of metrics on probability spaces known as integral probability metrics (IPMs) [32]. So, we present an overview of them here.

Definition 1: Let $(\mathsf{X}, \mathscr{G})$ be a measurable space and $\mathfrak{F}$ denote a class of uniformly bounded measurable functions on $(\mathsf{X}, \mathscr{G})$. The integral probability metric (IPM) between two probability distributions $\mu, \nu \in \mathscr{P}(\mathsf{X})$ with respect to the function class $\mathfrak{F}$ is defined as

$$d_{\mathfrak{F}}(\mu, \nu) \coloneqq \sup_{f \in \mathfrak{F}} \left| \int_{\mathsf{X}} f d\mu - \int_{\mathsf{X}} f d\nu \right|.$$

Different forms of IPMs that can be used in this paper are as follows

1) **Total variation distance**: If $\mathfrak{F}$ is chosen as $\mathfrak{F}^{\mathrm{TV}} \coloneqq \{f : \operatorname{span}(f) \leq 1\}$, then $d_{\mathfrak{F}}$ is the total variation distance.
2) **Wasserstein distance**: If X is a metric space and $\mathfrak{F}$ is chosen as $\mathfrak{F}^{\mathrm{W}} \coloneqq \{f : \operatorname{Lip}(f) \leq 1\}$ (where the Lipschitz constant is computed with respect to the metric on X), then $d_{\mathfrak{F}}$ is the Wasserstein distance.

3) **Maximum mean discrepancy (MMD)** Let $\mathcal{H}$ be a reproducing kernel Hilbert space (RKHS) of real valued functions on X and let $\mathfrak{F}$ be chosen as $\mathfrak{F}^{\mathrm{MMD}} := \{f \in \mathcal{H} \colon \|f\|_{\mathcal{H}} \le 1\}$, then $d_{\mathfrak{F}}$ is the maximum mean discrepancy.

Our approximation results are stated in terms of the Minkowski functional of a function f (not necessarily in $\mathfrak{F}$) with respect to a function class $\mathfrak{F}$, which is defined as follows:

$$\rho_{\mathfrak{F}}(f) := \inf\{\rho \in \mathbb{R}_{>0} : \rho^{-1} f \in \mathfrak{F}\}. \tag{6}$$

A key implication of this definition is that for any function f,

$$\left| \int_{\mathsf{X}} f d\mu - \int_{\mathsf{X}} f d\nu \right| \le \rho_{\mathfrak{F}}(f) \cdot d_{\mathfrak{F}}(\mu, \nu), \tag{7}$$

The Minkowski functional of the IPMs considered above are as follows

1) **Total variation distance**: If $\mathfrak{F}$ is chosen as $\mathfrak{F}^{\mathrm{TV}}$, $\left|\int_{\mathsf{X}} f d\mu - \int_{\mathsf{X}} f d\nu\right| \le \mathrm{span}(f) d_{\mathfrak{F}}(\mu, \nu)$. Thus, for total variation, $\rho_{\mathfrak{F}^{\mathrm{TV}}}(f) = \mathrm{span}(f)$.
2) **Wasserstein distance**: If $\mathfrak{F}$ is chosen as $\mathfrak{F}^{\mathrm{W}}$, $\left|\int_{\mathsf{X}} f d\mu - \int_{\mathsf{X}} f d\nu\right| \le \mathrm{Lip}(f) \cdot d_{\mathfrak{F}}(\mu, \nu)$. Thus, for the Wasserstein distance, $\rho_{\mathfrak{F}^{\mathrm{W}}}(f) = \mathrm{Lip}(f)$.
3) **Maximum mean discrepancy (MMD)**: If $\mathfrak{F}$ is chosen as $\mathfrak{F}^{\mathrm{MMD}}$, $\left|\int_{\mathsf{X}} f d\mu - \int_{\mathsf{X}} f d\nu\right| \le \|f\|_{\mathcal{H}} \cdot d_{\mathfrak{F}}(\mu, \nu)$. Thus, for the maximum mean discrepancy, $\rho_{\mathfrak{F}^{\mathrm{MMD}}}(f) = \|f\|_{\mathcal{H}}$.

III. Asymmetric actor-critic

A. Using state information in dynamic programs for POMDPs

We previously introduced value functions which only use the history h_t in their dynamic programs in Sec. II-A. Next, we introduce similar value functions that use the state s_t along with the history h_t in their dynamic programs.

$$\tilde{V}_t^{\pi}(s_t, h_t) := \mathbb{E}^{\pi}\left[\sum_{\tau=t}^{\mathsf{T}} \gamma^{\tau-t} R_\tau \mid S_t = s_t, H_t = h_t\right], \tag{8}$$

$$\begin{aligned}\tilde{Q}_t^{\pi}(s_t, h_t, a_t) &:= \mathbb{E}^{\pi}\left[\sum_{\tau=t}^{\mathsf{T}} \gamma^{\tau-t} R_\tau \mid S_t = s_t, H_t = h_t, A_t = a_t\right] \\ &= r(s_t, a_t) + \gamma \int_{\mathsf{Y}} V_{t+1}^{\pi}(h_{t+1}) P^{\mathsf{S},\mathsf{Y}}(dy_{t+1} \mid s_t, a_t). \end{aligned} \tag{9}$$

We call $\tilde{V}_t^{\pi}$ and $\tilde{Q}_t^{\pi}$ as the *augmented* value and action-value functions. We can retrieve the original value functions V_t^{π} and Q_t^{π} from the augmented ones, $\tilde{V}_t^{\pi}$ and $\tilde{Q}_t^{\pi}$, as follows:

$$V_t^{\pi}(h_t) = \int_{\mathsf{S}} \tilde{V}_t^{\pi}(s_t, h_t) \mathbb{P}(ds_t \mid h_t), \tag{10}$$

$$Q_t^{\pi}(h_t, a_t) = \int_{\mathsf{S}} \tilde{Q}_t^{\pi}(s_t, h_t, a_t) \mathbb{P}(ds_t \mid h_t). \tag{11}$$

Similarly, the optimal value function $Q_t^{\star}$ can be obtained from $\tilde{Q}_t^{\star}$ as follows

$$Q_t^{\star}(h_t, a_t) = \int_{\mathsf{S}} \tilde{Q}_t^{\star}(s_t, h_t, a_t) \mathbb{P}(ds_t \mid h_t), \tag{12}$$

$$\begin{aligned}\tilde{Q}_t^{\star}(s_t, h_t, a_t) &= r(s_t, a_t) \\ &\quad + \gamma \int_{\mathsf{Y}} V_{t+1}^{\star}(h_{t+1}) P^{\mathsf{S},\mathsf{Y}}(dy_{t+1} \mid s_t, a_t).\end{aligned} \tag{13}$$

The augmented value and action-value functions defiend above are useful to understand the asymmetric actor-critic algorithm, which we explain in the next section.

B. Asymmetric actor-critic algorithm

An actor-critic algorithm involves an actor function π_θ with parameters θ which takes the history as input and gives a probability distribution over actions as output; and a critic function Q_ζ with parameters ζ which takes the history and action as input and gives a real number denoting the value as an output [33], [34]. In settings where the state information is available during training, an asymmetric critic function $\tilde{Q}_\zeta$ (which corresponds to the augmented action-value function defined in (9)) with parameters ζ can be used which takes the state, history and action as input and gives a real number denoting the value as an output. As proposed in [14], the actor loss $\mathcal{L}_\theta^{\text{actor}}$ and the critic loss $\mathcal{L}_\zeta^{\text{critic}}$ for the asymmetric actor-critic algorithm are optimized simultaneously using the following gradient equations:

$$\begin{aligned}\nabla_\theta \mathcal{L}_\theta^{\text{actor}}(h_t) = -\sum_{\tau=t}^{\mathsf{T}} \mathbb{E}^{\pi_\theta}\big[\gamma^{\tau-t} \nabla_\theta \log \pi_\theta(A_\tau \mid H_\tau) \\ \tilde{Q}_\zeta(S_\tau, H_\tau, A_\tau) \mid H_t = h_t\big]\end{aligned} \tag{14}$$

and

$$\begin{aligned}\nabla_\zeta \mathcal{L}_\zeta^{\text{critic}}(s_t, h_t) = \nabla_\zeta \mathbb{E}^{\pi_\theta}\Bigg[\tilde{Q}_\zeta(S_t, H_t, A_t) \\ - \sum_{\tau=t}^{\mathsf{T}} \gamma^{\tau-t} R_\tau(S_\tau, A_\tau) \;\Bigg|\; S_t = s_t, H_t = h_t, A_t = a_t\Bigg]^2,\end{aligned} \tag{15}$$

C. Benefits of using state-history critic over history critic

It is shown in [14] that the policy gradient used in actor-critic and asymmetric actor-critic are the same in expectation. However, there is no discussion on why one expects asymmetric actor-critic to do better than symmetric actor-critic. In this section, we provide such an explanation.

Let's consider the training setup for a standard actor-critic implementation. The agent starts with an initial policy and generates trajectories $\{s_\tau, h_\tau, a_\tau, r_\tau\}_{\tau=1}^{T}$ which are stored in a buffer. Next, an empirical estimate of the policy gradient is constructed (as described below) and gradient descent on the policy parameters θ is performed based on this estimate. This process of performing rollouts and gradient descent happens iteratively till the policy converges.

When only observation and action history is used, the empirical policy gradient is constructed as follows

$$\nabla V_t^{\pi_\theta} \approx \sum_{m=1}^{M} \gamma^{\tau_m - t} \nabla \log \pi_\theta(a_{\tau_m} \mid h_{\tau_m}) Q_{\tau_m}^{\pi_\theta}(h_{\tau_m}, a_{\tau_m})$$

where M is the size of the mini-batch. We call this estimate as **history-only critic policy gradient (HOPG).**

When state information is available, the empirical policy gradient can be constructed as follows

$$\nabla V_t^{\pi_\theta} \approx \sum_{m=1}^{M} \gamma^{\tau_m - t} \nabla \log \pi_\theta(a_{\tau_m} \mid h_{\tau_m}) \tilde{Q}_{\tau_m}^{\pi_\theta}(s_{\tau_m}, h_{\tau_m}, a_{\tau_m}).$$

We call this estimate as **state-history critic policy gradient (SHPG)**.

Note that we are not using the state information from the buffer for the HOPG expression, but only in the SHPG expression. The TD(0) estimates for the Q-functions can be obtained as follows

$$\begin{aligned} Q_{\tau_m}^{\pi_\theta}(h_{\tau_m}, a_{\tau_m}) &\approx r_{\tau_m}(s_{\tau_m}, a_{\tau_m}) \\ &\quad + \gamma Q_{\tau_m+1}^{\pi_\theta}(h_{\tau_m+1}, a_{\tau_m+1}), \\ \tilde{Q}_{\tau_m}^{\pi_\theta}(s_{\tau_m}, h_{\tau_m}, a_{\tau_m}) &\approx r_{\tau_m}(s_{\tau_m}, a_{\tau_m}) \\ &\quad + \gamma \tilde{Q}_{\tau_m+1}^{\pi_\theta}(s_{\tau_m+1}, h_{\tau_m+1}, a_{\tau_m+1}). \end{aligned}$$

In both policy gradient expressions, we are sampling from a joint distribution of $\mathbb{P}(s_{\tau_m}, h_{\tau_m})$. In the HOPG expression, we discard the state information, so we marginalize s_{τ_m} out and sample over $\mathbb{P}(h_{\tau_m})$. But now, to estimate the HOPG critic we need to sample the reward $r_{\tau_m}(s_{\tau_m}, a_{\tau_m})$, which indirectly requires sampling the state through the reward. This means that we require sampling from $\mathbb{P}(s_{\tau_m} \mid h_{\tau_m})$ after we have already discarded state information from $\mathbb{P}(s_{\tau_m}, h_{\tau_m})$ to get $\mathbb{P}(h_{\tau_m})$. Effectively, we are combining this distribution $\mathbb{P}(h_{\tau_m})$ and the conditional distribution $\mathbb{P}(s_{\tau_m} \mid h_{\tau_m})$ to get a joint distribution $\mathbb{P}(s_{\tau_m}, h_{\tau_m})$ which is just a reconstruction of the original distribution. But since we are taking a practical batch size M which is not too large, it will be very unlikely that we will get more than a single sample from each unique history trajectory. Thus the variance due to the single sample estimate of $\mathbb{P}(s_{\tau_m} \mid h_{\tau_m})$ will be very high which means the variance for the final policy gradient expression which samples from $\mathbb{P}(s_{\tau_m}, h_{\tau_m})$ will be very high. A high variance policy gradient update can lead to slow or even unstable learning and would also be less sample efficient.

In contrast to this, we have the SHPG policy gradient expression which samples directly from $\mathbb{P}(s_{\tau_m}, h_{\tau_m})$. In this case, we pay the cost of requiring extra state information for each sample but this removes the requirement of sampling from $\mathbb{P}(s_{\tau_m} \mid h_{\tau_m})$ which is the main source of problems in the HOPG case (single sample issue). It would not be a requirement to encounter the same histories over different samples, since we are sampling from the joint distribution. There is no reconstruction of the joint distribution $\mathbb{P}(s_{\tau_m}, h_{\tau_m})$ required, and thus the variance from this directly sampled distribution is lower. A lower variance policy gradient update can lead to faster and more stable learning since there is less noise in the updates. It would also be more sample efficient since fewer samples would be required to construct the batch gradient.

IV. Representation loss for state-based dynamic program for POMDPs

In practice, one does not implement the history-based asymmetric actor-critic described in (14) and (15). Rather, the history is compressed via a recurrent neural network, and the compressed version of the history is used as a state. Using such a history compression leads to a loss in performance, which we call *representation loss*. In this section, we present a bound of representation loss for state-based dynamic programs.

A. Approximate Information State for POMDPs with state information

Next, we present an alternate form of the concept of an approximate information state (AIS) [1] that incorporates state information.

Definition 2: Let Z be a pre-specified Banach space, $\mathfrak{F}$ be a function class for IPMs and $\{\varepsilon, \delta^{\mathsf{Z}}, \delta^{\mathsf{S}}\}$ be pre-specified positive real numbers. A history compression function $\sigma \colon \mathsf{H}_t \to \mathsf{Z}$, reward approximation function $\hat{\hat{r}} \colon \mathsf{S} \times \mathsf{Z} \times \mathsf{A} \to \mathbb{R}$, approximate update kernel $\hat{P}^{\mathsf{Z}} \colon \mathsf{S} \times \mathsf{Z} \times \mathsf{A} \to \Delta(\mathsf{Z})$ and an approximate state distribution kernel $\hat{P}^{\mathsf{S}} \colon \mathsf{Z} \to \Delta(\mathsf{S})$, is called a $\{\varepsilon, \delta^{\mathsf{Z}}, \delta^{\mathsf{S}}\}$-*AIS generator* if the process $Z_t = \sigma(H_t)$ satisfies the following properties for all $t \in \{1, \dots, \mathsf{T}\}$:

(P1) **Sufficient for approximate reward evaluation**. For any realization s_t of S_t, h_t of H_t and any choice a_t of A_t, we have

$$|r(s_t, a_t) - \hat{\hat{r}}(s_t, z_t, a_t)| \le \varepsilon.$$

(P2) **Sufficient to predict itself approximately**. For any realization s_t of S_t, h_t of H_t and any choice a_t of A_t, and for any Borel subset B of Z, define $\mu_t^{\mathsf{Z}}(\mathsf{B}) \coloneqq \mathbb{P}(Z_{t+1} \in \mathsf{B} \mid S_t = s_t, H_t = h_t, A_t = a_t)$ and $\nu^{\mathsf{Z}}(\mathsf{B}) \coloneqq \hat{P}^{\mathsf{Z}}(\mathsf{B} \mid s_t, \sigma(h_t), a_t)$; then,

$$d_{\mathfrak{F}}(\mu_t^{\mathsf{Z}}, \nu^{\mathsf{Z}}) \le \delta^{\mathsf{Z}}.$$

(P3) **Sufficient to generate the belief over the state approximately**. For any realization h_t of H_t, and for any Borel subset B of S, define $\mu_t^{\mathsf{S}}(\mathsf{B}) \coloneqq \mathbb{P}(S_t \in \mathsf{B} \mid H_t = h_t)$ and $\nu^{\mathsf{S}}(\mathsf{B}) \coloneqq \hat{P}^{\mathsf{S}}(\mathsf{B} \mid \sigma(h_t))$; then,

$$d_{\mathfrak{F}}(\mu_t^{\mathsf{S}}, \nu^{\mathsf{S}}) \le \delta^{\mathsf{S}}.$$

Using this $\{\varepsilon, \delta^{\mathsf{Z}}, \delta^{\mathsf{S}}\}$-AIS Z_t, we can construct an approximate dynamic program that uses the approximate functions and kernels associated with the AIS, which tries to approximate the functions and kernels of the original dynamics. For all $s_{\mathsf{T}+1}, z_{\mathsf{T}+1}$ and $a_{\mathsf{T}+1}$, we initialize $\hat{V}_{\mathsf{T}+1}(z_{\mathsf{T}+1})$, $\hat{Q}_{\mathsf{T}+1}(z_{\mathsf{T}+1}, a_{\mathsf{T}+1})$ and $\hat{\hat{Q}}_{\mathsf{T}+1}(s_{\mathsf{T}+1}, z_{\mathsf{T}+1}, a_{\mathsf{T}+1})$ to zero and set:

$$\hat{V}_t(z_t) \coloneqq \max_{a_t \in \mathsf{A}} \hat{Q}_t(z_t, a_t), \tag{16}$$

$$\hat{Q}_t(z_t, a_t) \coloneqq \int_{\mathsf{S}} \hat{\hat{Q}}_t(s_t, z_t, a_t) \hat{P}^{\mathsf{S}}(ds_t \mid z_t), \tag{17}$$

$$\hat{\tilde{Q}}_t(s_t, z_t, a_t) \coloneqq \hat{\tilde{r}}(s_t, z_t, a_t) + \gamma \int_{\mathsf{Z}} \hat{V}_{t+1}(z_{t+1}) \hat{P}^{\mathsf{Z}}(dz_{t+1} \mid s_t, z_t, a_t). \tag{18}$$

Next, given any AIS dependent time-homogeneous recursively updateable policy $\hat{\pi}(z_t) = \hat{\pi}(\sigma(h_t)) = \pi(h_t)$, we can write down the associated value functions to evaluate the performance of that policy in the approximated setup. For all $s_{\mathsf{T}+1}, z_{\mathsf{T}+1}$ and $a_{\mathsf{T}+1}$, we initialize $\hat{V}^{\hat{\pi}}_{\mathsf{T}+1}(z_{\mathsf{T}+1})$ $\hat{Q}^{\hat{\pi}}_{\mathsf{T}+1}(z_{\mathsf{T}+1}, a_{\mathsf{T}+1})$ and $\hat{\tilde{Q}}^{\hat{\pi}}_{\mathsf{T}+1}(s_{\mathsf{T}+1}, z_{\mathsf{T}+1}, a_{\mathsf{T}+1})$ to zero and set:

$$\hat{V}^{\hat{\pi}}_t(z_t) \coloneqq \sum_{a_t \in \mathsf{A}} \hat{\pi}(a_t \mid z_t) \hat{Q}^{\hat{\pi}}_t(z_t, a_t), \tag{19}$$

$$\hat{Q}^{\hat{\pi}}_t(z_t, a_t) \coloneqq \int_{\mathsf{S}} \hat{\tilde{Q}}^{\hat{\pi}}_t(s_t, z_t, a_t) \hat{P}^{\mathsf{S}}(ds_t \mid z_t), \tag{20}$$

$$\hat{\tilde{Q}}^{\hat{\pi}}_t(s_t, z_t, a_t) \coloneqq \hat{\tilde{r}}(s_t, z_t, a_t) + \gamma \int_{\mathsf{Z}} \hat{V}^{\hat{\pi}}_{t+1}(z_{t+1}) \hat{P}^{\mathsf{Z}}(dz_{t+1} \mid s_t, z_t, a_t). \tag{21}$$

It is relevant to discuss some common aspects between [35] and our work. The degree of approximation in the approximate representation, which we refer to as an approximate information state (AIS) [1], can be quantified and performance guarantees can be obtained for the AIS representation, which are of a different nature from the regret based bounds in [35]. The regret bounds still hold for the POMDP model with the AIS representation considered in this paper.

B. An upper bound on loss when using an AIS

Theorem 1: Suppose we have for all s_t, h_t and a_t, an AIS $\{Z_t\}_{t=1}^{\mathsf{T}}$ that satisfies (P1), (P2) and (P3); and any $\hat{\pi}$ and $\pi = \hat{\pi} \circ \sigma$, then

$$|Q^\star_t(h_t, a_t) - \hat{Q}_t(\sigma(h_t), a_t)| \le \alpha_t, \tag{22}$$

$$|V^\star_t(h_t) - \hat{V}_t(\sigma(h_t))| \le \alpha_t, \tag{23}$$

$$|Q^\pi_t(h_t, a_t) - \hat{Q}^{\hat{\pi}}_t(\sigma(h_t), a_t)| \le \alpha^{\hat{\pi}}_t, \tag{24}$$

$$|V^\pi_t(h_t) - \hat{V}^{\hat{\pi}}_t(\sigma(h_t))| \le \alpha^{\hat{\pi}}_t, \tag{25}$$

$$|\tilde{Q}^\pi_t(s_t, h_t, a_t) - \hat{\tilde{Q}}^{\hat{\pi}}_t(s_t, \sigma(h_t), a_t)| \le \tilde{\alpha}^{\hat{\pi}}_t, \tag{26}$$

where

$$\alpha_t = \sum_{\tau=t}^{\mathsf{T}} \gamma^{\tau-t} \left[\varepsilon + \delta^{\mathsf{S}} \rho_{\mathfrak{F}}(\hat{\tilde{r}}) \right] + \gamma^{\tau-t+1} (\delta^{\mathsf{Z}} + \delta^{\mathsf{S}}) \rho_{\mathfrak{F}}(\hat{V}_{\tau+1}),$$

$$\alpha^{\hat{\pi}}_t = \sum_{\tau=t}^{\mathsf{T}} \gamma^{\tau-t} \left[\varepsilon + \delta^{\mathsf{S}} \rho_{\mathfrak{F}}(\hat{\tilde{r}}) \right] + \gamma^{\tau-t+1} (\delta^{\mathsf{Z}} + \delta^{\mathsf{S}}) \rho_{\mathfrak{F}}(\hat{V}^{\hat{\pi}}_{\tau+1}),$$

$$\tilde{\alpha}^{\hat{\pi}}_t = \sum_{\tau=t}^{\mathsf{T}} \gamma^{\tau-t} \left[\varepsilon + \gamma \delta^{\mathsf{Z}} \rho_{\mathfrak{F}}(\hat{V}^{\hat{\pi}}_{\tau+1}) \right].$$

Furthermore, if we have $\hat{\pi}$ such that for all h_t, $\operatorname{supp}(\hat{\pi}(\sigma(h_t))) \subseteq \arg\max_{a_t \in \mathsf{A}} \hat{Q}_t(\sigma(h_t), a_t)$, then

$$|Q^\star_t(h_t, a_t) - Q^\pi_t(h_t, a_t)| \le 2\alpha_t, \tag{27}$$

$$|V^\star_t(h_t) - V^\pi_t(h_t)| \le 2\alpha_t. \tag{28}$$

See Appendix I for proof.

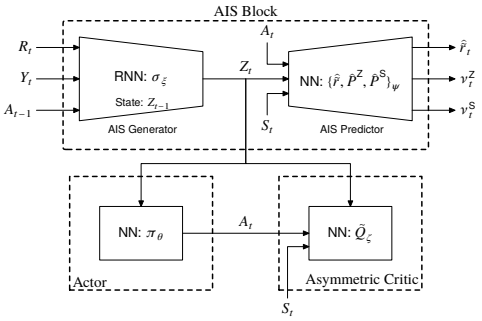


Fig. 1: Block diagram of the proposed RL algorithm

V. Reinforcement learning with representation losses

Following the main idea of [1], one may conjecture that adding the representation losses of Theorem 1 in the standard implementation of asymmetric actor-critic may improves performance. To test this conjecture, we modify the asymmetric actor-critic algorithm described in Sec. III-B by adding representation losses as an auxiliary loss. We describe this algorithm below.

A. Asymmetric actor-critic with AIS losses

The main idea is to add an "AIS-block" to the existing asymmetric actor-critic architecture, as shown in Fig. 1. The AIS-block consists of two parts: (i) an AIS generator σ_ξ with parameters ξ, which is a recurrent neural network such as an LSTM or a GRU and (ii) an AIS-predictor $\{\hat{\tilde{r}}, \hat{P}^{\mathsf{Z}}, \hat{P}^{\mathsf{S}}\}_\psi$ with parameters ψ, which is a feed-forward neural network. The loss of the AIS-block is chosen as $\varepsilon^2 + (\delta^{\mathsf{Z}})^2 + (\delta^{\mathsf{S}})^2$, where $(\varepsilon, \delta^{\mathsf{Z}}, \delta^{\mathsf{S}})$ are as defined in Def. 2. Note that δ^{Z} and δ^{S} depend on the choice of an IPM. See [1] for a discussion on the choice of IPM. In our experiments, we choose MMD as the IPM. For this choice, we have the following gradient for the AIS loss:

$$\nabla_{\xi,\psi} \mathcal{L}^{\text{AIS}} = \mathbb{E}^{\pi_\theta}\left[(r - \hat{\tilde{r}})^2\right] + \mathbb{E}^{\pi_\theta}\left[(\mathbb{E}[\nu^{\mathsf{Z}}_t] - 2Z_{t+1})^\intercal \mathbb{E}[\nu^{\mathsf{Z}}_t]\right] + \mathbb{E}^{\pi_\theta}\left[(\mathbb{E}[\nu^{\mathsf{S}}_t] - 2S_t)^\intercal \mathbb{E}[\nu^{\mathsf{S}}_t]\right], \tag{29}$$

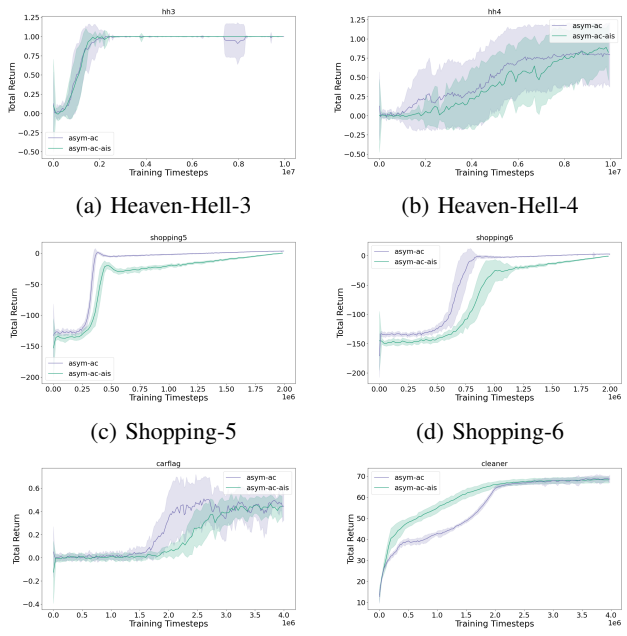
where the expression for the last two terms follows from [1, Prop 35]. We update the AIS parameters (ξ, ψ) by back-propagating the above gradient.

The AIS generator generates an approximate information state Z_t. We use Z_t as a compression for the history in the auxiliary action-value function $\tilde{Q}_t$ and the policy π and update the parameters of the critic and actor in a manner similar to (14) and (15).

B. Numerical experiments

To test the performance of the asymmetric actor-critic with AIS losses described above, we compare with the vanilla asymmetric actor-critic, as presented in [14]. We use the following environments, which were also used in [14]:

1) **Heaven-Hell-**3 and **Heaven-Hell-**4 are corridor-like gridworld environments with three dead-ends. One dead-end has a *priest* that can help the agent identify the



(a) Heaven-Hell-3 (b) Heaven-Hell-4

(c) Shopping-5 (d) Shopping-6

(e) Car-Flag (f) Cleaner

Fig. 2: Comparison of AIS-based A2C with history only critic and state-based critic for 6 benchmarking environments used in [14] (for 20 random seeds).

correct exit from the other two dead-ends. Heaven-Hell-3 has 28 states, 15 observations and 4 actions, while Heaven-Hell-4 has 36 states, 19 observations and 4 actions.

2) **Shopping-**5 and **Shopping-**6 are gridworld environments which contains an agent that can move around and *query* in a 5×5 or 6×6 grid for a randomly placed *item*. The *item* is observed only when *queried* and the agent needs to *move* to the correct location and *buy* the item. Shopping-5 has 625 states, 50 observations and 6 actions, while Shopping-6 has 1296 states, 72 observations and 6 actions.
3) **Carflag** is a continuous state and observation version of the Heaven-Hell environments which involves a *car* moving along a single dimension with some *position* and *velocity* that are observed. There are 3 flags: an *info flag* which tells the agent which of the reamining two flags is the *good flag* and *bad flag*. The 7 possible actions involve applying different levels of force in either direction.
4) **Cleaner** is a centralized two agent problem treated as a single agent problem. Two agents must cover an entire 13×13 maze-like environment in order to clean it. Every single grid cell must be visited at least once. Each agent can move in one of the 4 directions (total 16 actions) and receives a $3\times3\times3$ binary tensor around it as an observation. This observation indicates whether it contains a *wall* or *no wall*, a *dirty* or *clean cell*, the *first* or *second agent*.

We compare the following two algorithms:

1) **asym-ac** which is the asymmetric actor-critic as proposed in [14]. We use the code provided in [14] to run our experiments.
2) **asym-ac-ais** which is the asymmetric actor-critic with AIS losses presented in Sec. V-A.

We train both algorithms on the six environments described above for 10^6 to 10^7 steps. Each experiment is repeated for 20 sample paths and the mean and standard deviation are shown in Fig. 2. The resuls show that adding AIS losses (or representation losses) slightly slows down learning and does not lead to an improvement in the converged value. Based on these results, there is no advantage (or rather, there is a slight disadvantage) in adding AIS losses as an auxiliary loss in asymmetric actor-critic. This is in sharp contrast to the drastic improvement in performance obtained by adding AIS losses in symmetric actor-critic demonstrated in [1]. These results suggest that representation learning is not as important when full state information is available.

VI. Conclusions

The main contribution of this work is to establish the theoretical guarantees on performance for the case of asymmetric actor-critic using an approximate information state that is offered by Theorem 1. We also provide an explanation for why asymmetric actor-critc performs better than actor-critic algorithms. Motivated by recent successes in actor-critic algorithms for POMDPs [1], [27]–[30] which establishes similar bounds and provides experiments in RL, we aim to study the effectiveness of the concept of approximate information state for the case of asymmetric actor-critic methods. This is mainly because several situations arise where state information is available on a temporary basis (during training only) following execution without this state information. Our empirical results perform comparably with the existing state-of-the-art actor-critic method. However, the empirical validation of this theory is of secondary importance and we do not claim that using an AIS with asymmetric actor-critic always improves performance, rather we aim to provide a systematic rationale for RL algorithms for this particular class of problems. Another benefit of using an AIS is that it allows us to learn a meaningful common representation for the actor, critic and the AIS generator and predictor. Such a representation also has utility in the form of interpretability in terms of why an autonomous decision maker makes certain types of decisions. Future work involves formally showing that such an algorithm converges.

References

[1] J. Subramanian, A. Sinha, R. Seraj, and A. Mahajan, "Approximate information state for approximate planning and reinforcement learning in partially observed systems," *Journal of Machine Learning Research*, vol. 23, no. 12, pp. 1–83, 2022.
[2] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 4909–4926, 2021.
[3] B. An, S. Sun, and R. Wang, "Deep reinforcement learning for quantitative trading: Challenges and opportunities," *IEEE Intelligent Systems*, vol. 37, no. 2, pp. 23–26, 2022.

[4] A. Perera and P. Kamalaruban, "Applications of reinforcement learning in energy systems," *Renewable and Sustainable Energy Reviews*, vol. 137, p. 110618, 2021.
[5] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.
[6] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.
[7] C. H. Papadimitriou and J. N. Tsitsiklis, "The complexity of optimal queueing network control," in *Proceedings of IEEE 9th Annual Conference on Structure in Complexity Theory*. IEEE, 1994, pp. 318–322.
[8] A. R. Cassandra, M. L. Littman, and N. L. Zhang, "Incremental pruning: A simple, fast, exact method for partially observable markov decision processes," *arXiv preprint arXiv:1302.1525*, 2013.
[9] A. R. Cassandra, L. P. Kaelbling, and M. L. Littman, "Acting optimally in partially observable stochastic domains," in *Aaai*, vol. 94, 1994, pp. 1023–1028.
[10] R. D. Smallwood and E. J. Sondik, "The optimal control of partially observable markov processes over a finite horizon," *Operations research*, vol. 21, no. 5, pp. 1071–1088, 1973.
[11] M. Hausknecht and P. Stone, "Deep recurrent q-learning for partially observable mdps," in *2015 aaai fall symposium series*, 2015.
[12] N. Heess, J. J. Hunt, T. P. Lillicrap, and D. Silver, "Memory-based control with recurrent neural networks," *arXiv preprint arXiv:1512.04455*, 2015.
[13] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap, "Mastering diverse domains through world models," *arXiv preprint arXiv:2301.04104*, 2023.
[14] A. Baisero and C. Amato, "Unbiased asymmetric actor-critic for partially observable reinforcement learning," *arXiv preprint arXiv:2105.11674*, 2021.
[15] A. Baisero, B. Daley, and C. Amato, "Asymmetric DQN for partially observable reinforcement learning," in *Uncertainty in Artificial Intelligence*. PMLR, 2022, pp. 107–117.
[16] D. Ha and J. Schmidhuber, "World models," *arXiv preprint arXiv:1803.10122*, 2018.
[17] P. Zhu, X. Li, P. Poupart, and G. Miao, "On improving deep reinforcement learning for pomdps," *arXiv preprint arXiv:1704.07978*, 2017.
[18] D. Wierstra, A. Foerster, J. Peters, and J. Schmidhuber, "Solving deep memory pomdps with recurrent policy gradients," in *Artificial Neural Networks–ICANN 2007: 17th International Conference, Porto, Portugal, September 9-13, 2007, Proceedings, Part I 17*. Springer, 2007, pp. 697–706.
[19] A. Baisero and C. Amato, "Learning internal state models in partially observable environments," in *Reinforcement Learning under Partial Observability, NeurIPS Workshop*, 2018.
[20] M. Igl, L. Zintgraf, T. A. Le, F. Wood, and S. Whiteson, "Deep variational reinforcement learning for pomdps," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2117–2126.
[21] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel, "Asymmetric actor critic for image-based robot learning," *arXiv preprint arXiv:1710.06542*, 2017.
[22] W. Yue, Y. Zhou, X. Zhang, Y. Hua, Z. Wang, and G. Kou, "Aacc: Asymmetric actor-critic in contextual reinforcement learning," *arXiv preprint arXiv:2208.02376*, 2022.
[23] A. Dionigi, A. Devo, L. Guiducci, and G. Costante, "E-vat: An asymmetric end-to-end approach to visual active exploration and tracking," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4259–4266, 2022.
[24] A. Warrington, J. W. Lavington, A. Scibior, M. Schmidt, and F. Wood, "Robust asymmetric learning in pomdps," in *International Conference on Machine Learning*. PMLR, 2021, pp. 11 013–11 023.
[25] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl, "Learning by cheating," in *Conference on Robot Learning*. PMLR, 2020, pp. 66–75.
[26] H. Nguyen, B. Daley, X. Song, C. Amato, and R. Platt, "Belief-grounded networks for accelerated robot learning under partial observability," *arXiv preprint arXiv:2010.09170*, 2020.
[27] G. Patil, A. Mahajan, and D. Precup, "On learning history based policies for controlling markov decision processes," *arXiv preprint arXiv:2211.03011*, 2022.
[28] S. Bhatt, W. Mao, A. Koppel, and T. Başar, "Semiparametric information state embedding for policy search under imperfect information," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 4501–4506.
[29] A. Dave, N. Venkatesh, and A. A. Malikopoulos, "Approximate information states for worst-case control of uncertain systems," in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 4945–4950.
[30] L. Yang, K. Zhang, A. Amice, Y. Li, and R. Tedrake, "Discrete approximate information states in partially observable environments," in *2022 American Control Conference (ACC)*. IEEE, 2022, pp. 1406–1413.
[31] K. J. Åström, "Optimal control of markov processes with incomplete state information," *Journal of mathematical analysis and applications*, vol. 10, no. 1, pp. 174–205, 1965.
[32] A. Müller, "Integral probability metrics and their generating classes of functions," *Advances in Applied Probability*, vol. 29, no. 2, pp. 429–443, 1997.
[33] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
[34] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*. PMLR, 2016, pp. 1928–1937.
[35] S. Dong, B. Van Roy, and Z. Zhou, "Simple agent, complex environment: Efficient reinforcement learning with agent states," *arXiv preprint arXiv:2102.05261*, 2021.

APPENDIX I
PROOF OF THEOREM 1

First, note that

$$\begin{aligned}
&\mathbb{E}[\hat{V}_{t+1}(Z_{t+1}) \mid S_t = s_t, H_t = h_t, A_t = a_t] \\
&= \int_{\mathsf{Z}} \hat{V}_{t+1}(z_{t+1}) \mathbb{P}(dz_{t+1} \mid S_t = s_t, H_t = h_t, A_t = a_t) \\
&= \int_{\mathsf{Y}} \hat{V}_{t+1}(z_{t+1}) P^{\mathsf{S},\mathsf{Y}}(dy_{t+1} \mid s_t, a_t). \qquad (30)
\end{aligned}$$

The proof follows by backward induction. At time $\mathsf{T}+1$, the induction hypothesis is true. Now, consider for time t

$$\begin{aligned}
&|Q_t^\star(h_t, a_t) - \hat{Q}_t(z_t, a_t)| \\
&\overset{(a)}{\leq} \left| \int_{\mathsf{S}} \left[r(s_t, a_t) \mathbb{P}(ds_t \mid h_t) - \hat{\hat{r}}(s_t, z_t, a_t) \hat{P}^{\mathsf{S}}(ds_t \mid z_t) \right] \right| \qquad (31) \\
&\quad + \gamma \left| \int_{\mathsf{S}} \int_{\mathsf{Y}} V_{t+1}^\star(h_{t+1}) P^{\mathsf{S},\mathsf{Y}}(dy_{t+1} \mid s_t, a_t) \mathbb{P}(ds_t \mid h_t) \right. \\
&\qquad \left. - \int_{\mathsf{S}} \int_{\mathsf{Z}} \hat{V}_{t+1}(z_{t+1}) \hat{P}^{\mathsf{Z}}(dz_{t+1} \mid s_t, z_t, a_t) \hat{P}^{\mathsf{S}}(ds_t \mid z_t) \right|, \qquad (32)
\end{aligned}$$

where (a) follows from (12), (17) and the triangle inequality. First, we consider (31) separately:

$$\begin{aligned}
(31) &\overset{(b)}{\leq} \left| \int_{\mathsf{S}} \left[r(s_t, a_t) \mathbb{P}(ds_t \mid h_t) - \hat{\hat{r}}(s_t, z_t, a_t) \mathbb{P}(ds_t \mid h_t) \right] \right| \\
&\quad + \left| \int_{\mathsf{S}} \left[\hat{\hat{r}}(s_t, z_t, a_t) \mathbb{P}(ds_t \mid h_t) - \hat{\hat{r}}(s_t, z_t, a_t) \hat{P}^{\mathsf{S}}(ds_t \mid z_t) \right] \right| \\
&\overset{(c)}{\leq} \varepsilon + \delta^{\mathsf{S}} \rho_{\mathfrak{F}}(\hat{\hat{r}}),
\end{aligned}$$

where (b) follows from adding and subtracting $\int_{\mathsf{S}} \hat{\hat{r}}(s_t, z_t, a_t) \mathbb{P}(ds_t \mid h_t)$ and the triangle inequality

and (c) follows from (P1) and (P3). Next, we consider (32):

$$(32) \overset{(d)}{\leq} \gamma \Big| \int_{\mathsf{S}} \int_{\mathsf{Y}} V^{\star}_{t+1}(h_{t+1}) P^{\mathsf{S},\mathsf{Y}}(dy_{t+1} \mid s_t, a_t) \mathbb{P}(ds_t \mid h_t) - \int_{\mathsf{S}} \int_{\mathsf{Y}} \hat{V}_{t+1}(z_{t+1}) P^{\mathsf{S},\mathsf{Y}}(dy_{t+1} \mid s_t, a_t) \mathbb{P}(ds_t \mid h_t) \Big| \tag{33}$$

$$+ \gamma \Big| \int_{\mathsf{S}} \int_{\mathsf{Z}} \hat{V}_{t+1}(z_{t+1}) \mathbb{P}(dz_{t+1} \mid s_t, h_t, a_t) \mathbb{P}(ds_t \mid h_t) - \int_{\mathsf{S}} \int_{\mathsf{Z}} \hat{V}_{t+1}(z_{t+1}) \hat{P}^{\mathsf{Z}}(dz_{t+1} \mid s_t, z_t, a_t) \hat{P}^{\mathsf{S}}(ds_t \mid z_t) \Big|, \tag{34}$$

where (d) follows from adding and subtracting (30) and the triangle inequality. Since the induction hypothesis is true for $t+1$, we have

$$(33) \leq \gamma \alpha_{t+1}.$$

Finally, for the remaining part in (34), we have

$$\begin{aligned}(34) &\overset{(e)}{\leq} \gamma \Big| \int_{\mathsf{S}} \int_{\mathsf{Z}} \hat{V}_{t+1}(z_{t+1}) \mathbb{P}(dz_{t+1} \mid s_t, h_t, a_t) \mathbb{P}(ds_t \mid h_t) - \int_{\mathsf{S}} \int_{\mathsf{Z}} \hat{V}_{t+1}(z_{t+1}) \hat{P}^{\mathsf{Z}}(dz_{t+1} \mid s_t, z_t, a_t) \mathbb{P}(ds_t \mid h_t) \Big| \\ &+ \gamma \Big| \int_{\mathsf{S}} \int_{\mathsf{Z}} \hat{V}_{t+1}(z_{t+1}) \hat{P}^{\mathsf{Z}}(dz_{t+1} \mid s_t, z_t, a_t) \mathbb{P}(ds_t \mid h_t) - \int_{\mathsf{S}} \int_{\mathsf{Z}} \hat{V}_{t+1}(z_{t+1}) \hat{P}^{\mathsf{Z}}(dz_{t+1} \mid s_t, z_t, a_t) \hat{P}^{\mathsf{S}}(ds_t \mid z_t) \Big| \\ &\overset{(f)}{\leq} \gamma \delta^{\mathsf{S}} \rho_{\mathfrak{F}}(\hat{V}_{t+1}) + \gamma \delta^{\mathsf{Z}} \rho_{\mathfrak{F}}(\hat{V}_{t+1}),\end{aligned}$$

where (e) follows from adding and subtracting a term and the triangle inequality and (f) follows from (P2) and (P3). Thus, we have shown that

$$|Q^{\star}_t(h_t, a_t) - \hat{Q}_t(z_t, a_t)| \leq \varepsilon + \delta^{\mathsf{S}} \rho_{\mathfrak{F}}(\hat{\tilde{r}}) + \gamma \alpha_{t+1} + \gamma(\delta^{\mathsf{Z}} + \delta^{\mathsf{S}}) \rho_{\mathfrak{F}}(\hat{V}_{t+1}) = \alpha_t.$$

To complete the induction argument at time t, notice that

$$|V^{\star}_t(h_t) - \hat{V}_t(z_t)| = \Big| \max_{a_t \in \mathsf{A}} Q^{\star}_t(h_t, a_t) - \max_{a_t \in \mathsf{A}} \hat{Q}_t(z_t, a_t) \Big| \leq \max_{a_t \in \mathsf{A}} |Q^{\star}_t(h_t, a_t) - \hat{Q}_t(z_t, a_t)| \leq \alpha_t.$$

This proves (22) and (23). Now, to show the next part, we consider any arbitrary AIS dependent policy $\hat{\pi}_t(z_t)$. Note that this can also be represented as a history dependent policy $\pi_t(h_t)$ if we take $\pi_t = \hat{\pi}_t \circ \sigma_t$. The proof follows by backward induction. At time $\mathsf{T}+1$, the induction hypothesis is true. Now, consider for time t

$$\begin{aligned}&|Q^{\pi}_t(h_t, a_t) - \hat{Q}^{\hat{\pi}}_t(z_t, a_t)| \\ &\overset{(g)}{\leq} \Big| \int_{\mathsf{S}} \Big[r(s_t, a_t) \mathbb{P}(ds_t \mid h_t) - \hat{\tilde{r}}(s_t, z_t, a_t) \hat{P}^{\mathsf{S}}(ds_t \mid z_t) \Big] \Big| \\ &+ \gamma \Big| \int_{\mathsf{S}} \int_{\mathsf{Y}} V^{\pi}_{t+1}(h_{t+1}) P^{\mathsf{S},\mathsf{Y}}(dy_{t+1} \mid s_t, a_t) \mathbb{P}(ds_t \mid h_t) - \int_{\mathsf{S}} \int_{\mathsf{Z}} \hat{V}^{\hat{\pi}}_{t+1}(z_{t+1}) \hat{P}^{\mathsf{Z}}(dz_{t+1} \mid s_t, z_t, a_t) \hat{P}^{\mathsf{S}}(ds_t \mid z_t) \Big| \\ &\overset{(h)}{\leq} \varepsilon + \delta^{\mathsf{S}} \rho_{\mathfrak{F}}(\hat{\tilde{r}}) + \gamma \alpha^{\hat{\pi}}_{t+1} + \gamma(\delta^{\mathsf{Z}} + \delta^{\mathsf{S}}) \rho_{\mathfrak{F}}(\hat{V}^{\hat{\pi}}_{t+1}) = \alpha^{\hat{\pi}}_t,\end{aligned}$$

where (h) follows from exactly the same steps $(a) - (f)$ mentioned previously. The only difference is that in step (f), we use $\hat{V}^{\hat{\pi}}_{t+1}$ instead of $\hat{V}_{t+1}$. To complete the induction argument at time t, notice that

$$\begin{aligned}&|V^{\pi}_t(h_t) - \hat{V}^{\hat{\pi}}_t(z_t)| \\ &= \Big| \sum_{a_t \in \mathsf{A}} \hat{\pi}_t(a_t \mid z_t) Q^{\pi}_t(h_t, a_t) - \sum_{a_t \in \mathsf{A}} \hat{\pi}_t(a_t \mid z_t) \hat{Q}^{\hat{\pi}}_t(z_t, a_t) \Big| \\ &\leq \sum_{a_t \in \mathsf{A}} \hat{\pi}_t(a_t \mid z_t) |Q^{\pi}_t(h_t, a_t) - \hat{Q}^{\hat{\pi}}_t(z_t, a_t)| \leq \alpha^{\hat{\pi}}_t.\end{aligned}$$

This proves (24) and (25). Finally, if $\hat{\pi}$ is such that $\operatorname{supp}(\hat{\pi}(\sigma(h_t))) \subseteq \arg\max_{a_t \in \mathsf{A}} \hat{Q}_t(\sigma(h_t), a_t)$, so that $\hat{Q}_t(z_t, a_t) = \hat{Q}^{\hat{\pi}}_t(z_t, a_t)$, and also $\pi = \hat{\pi} \circ \sigma$, then (27) and (28) are obtained by the triangle inequality as follows

$$\begin{aligned}&|Q^{\star}_t(h_t, a_t) - Q^{\pi}_t(h_t, a_t)| \\ &\quad \leq |Q^{\star}_t(h_t, a_t) - \hat{Q}_t(z_t, a_t)| + |Q^{\pi}_t(h_t, a_t) - \hat{Q}^{\hat{\pi}}_t(z_t, a_t)| \\ &\quad \leq \alpha_t + \alpha^{\hat{\pi}}_t = 2\alpha_t.\end{aligned}$$

To show 26, first note that

$$\begin{aligned}&\mathbb{E}[\hat{V}^{\hat{\pi}}_{t+1}(Z_{t+1}) \mid S_t = s_t, H_t = h_t, A_t = a_t] \\ &= \int_{\mathsf{Z}} \hat{V}^{\hat{\pi}}_{t+1}(z_{t+1}) \mathbb{P}(dz_{t+1} \mid s_t, h_t, a_t) \\ &= \int_{\mathsf{Y}} \hat{V}^{\hat{\pi}}_{t+1}(z_{t+1}) P^{\mathsf{S},\mathsf{Y}}(dy_{t+1} \mid s_t, a_t).\end{aligned} \tag{35}$$

Consider

$$\begin{aligned}&|\tilde{Q}^{\pi}_t(s_t, h_t, a_t) - \hat{\tilde{Q}}^{\hat{\pi}}_t(s_t, z_t, a_t)| \\ &\quad \overset{(a)}{\leq} |r(s_t, a_t) - \hat{\tilde{r}}(s_t, z_t, a_t)| \\ &\qquad + \gamma \Big| \int_{\mathsf{Y}} V^{\pi}_{t+1}(h_{t+1}) P^{\mathsf{S},\mathsf{Y}}(dy_{t+1} \mid s_t, a_t) - \int_{\mathsf{Y}} \hat{V}^{\hat{\pi}}_{t+1}(z_{t+1}) P^{\mathsf{S},\mathsf{Y}}(dy_{t+1} \mid s_t, a_t) \Big| \\ &\qquad + \gamma \Big| \int_{\mathsf{Z}} \hat{V}^{\hat{\pi}}_{t+1}(z_{t+1}) \mathbb{P}(dz_{t+1} \mid s_t, h_t, a_t) - \int_{\mathsf{Z}} \hat{V}^{\hat{\pi}}_{t+1}(z_{t+1}) \hat{P}^{\mathsf{Z}}(dz_{t+1} \mid s_t, z_t, a_t) \Big| \\ &\quad \overset{(b)}{\leq} \varepsilon + \gamma \alpha^{\hat{\pi}}_{t+1} + \gamma \delta^{\mathsf{Z}} \rho_{\mathfrak{F}}(\hat{V}^{\hat{\pi}}_{t+1}) \overset{(c)}{\leq} \alpha^{\hat{\pi}}_t\end{aligned}$$

where (a) follows from (9), (21), adding and subtracting (35) and the triangle inequality; (b) follows from (P1), (P2) and (25); and (c) follows from adding a few extra positive terms.