Appendix

A. Proof of Lemma 1

Lemma 1. *Under Assumptions* (A1)–(A2)*, which defines a function f, for all π, there exists a π_i such that*

$$v^\pi(s) = \sum_{a\in\mathcal{A}} \int_{f^{-1}(a)} \pi_i(e|s) q^\pi(s,a)\,\mathrm{d}e.$$

Proof. The Bellman equation associated with a policy, π, for any MDP, $\mathcal{M}$, is:

$$\begin{aligned} v^\pi(s) &= \sum_{a\in\mathcal{A}} \pi(a|s) q^\pi(s,a) \\ &= \sum_{a\in\mathcal{A}} \pi(a|s) \sum_{s'\in\mathcal{S}} P(s'|s,a)[\mathcal{R}(s,a) + \gamma v^\pi(s')]. \end{aligned}$$

G is used to denote $[\mathcal{R}(s,a) + \gamma v^\pi(s')]$ hereafter. Re-arranging terms in the Bellman equation,

$$\begin{aligned} v^\pi(s) &= \sum_{a\in\mathcal{A}} \sum_{s'\in\mathcal{S}} \pi(a|s) \frac{P(s',s,a)}{P(s,a)} G \\ &= \sum_{a\in\mathcal{A}} \sum_{s'\in\mathcal{S}} \pi(a|s) \frac{P(s',s,a)}{\pi(a|s)P(s)} G \\ &= \sum_{a\in\mathcal{A}} \sum_{s'\in\mathcal{S}} \frac{P(s',s,a)}{P(s)} G \\ &= \sum_{a\in\mathcal{A}} \sum_{s'\in\mathcal{S}} \frac{P(a|s,s')P(s,s')}{P(s)} G. \end{aligned}$$

Using the law of total probability, we introduce a new variable e such that:[1]

$$v^\pi(s) = \sum_{a\in\mathcal{A}} \sum_{s'\in\mathcal{S}} \int_e \frac{P(a,e|s,s')P(s,s')}{P(s)} G\,\mathrm{d}e.$$

After multiplying and dividing by $P(e|s)$, we have:

$$\begin{aligned} v^\pi(s) &= \sum_{a\in\mathcal{A}} \sum_{s'\in\mathcal{S}} \int_e P(e|s) \frac{P(a,e|s,s')P(s,s')}{P(e|s)P(s)} G\,\mathrm{d}e \\ &= \sum_{a\in\mathcal{A}} \int_e P(e|s) \sum_{s'\in\mathcal{S}} \frac{P(a,e|s,s')P(s,s')}{P(s,e)} G\,\mathrm{d}e \\ &= \sum_{a\in\mathcal{A}} \int_e P(e|s) \sum_{s'\in\mathcal{S}} \frac{P(a,e,s,s')}{P(s,e)} G\,\mathrm{d}e \\ &= \sum_{a\in\mathcal{A}} \int_e P(e|s) \sum_{s'\in\mathcal{S}} P(s',a|s,e) G\,\mathrm{d}e \\ &= \sum_{a\in\mathcal{A}} \int_e P(e|s) \sum_{s'\in\mathcal{S}} P(s'|s,a,e) P(a|s,e) G\,\mathrm{d}e. \end{aligned}$$

Since the transition to the next state, S_{t+1}, is conditionally independent of E_t, given the previous state, S_t, and the action taken, A_t,

$$v^\pi(s) = \sum_{a\in\mathcal{A}} \int_e P(e|s) \sum_{s'\in\mathcal{S}} P(s'|s,a) P(a|s,e) G\,\mathrm{d}e.$$

[1] Note that a and e are from a joint distribution over a discrete and a continuous random variable. For simplicity, we avoid measure-theoretic notations to represent its joint probability.

Similarly, using the Markov property, action A_t is conditionally independent of S_t given E_t,

$$v^\pi(s) = \sum_{a\in\mathcal{A}} \int_e P(e|s) \sum_{s'\in\mathcal{S}} P(s'|s,a)P(a|e)G \,\mathrm{d}e.$$

As $P(a|e)$ evaluates to 1 for representations, e, that map to a and 0 for others (Assumption A2),

$$\begin{aligned} v^\pi(s) &= \sum_{a\in\mathcal{A}} \int_{f^{-1}(a)} P(e|s) \sum_{s'\in\mathcal{S}} P(s'|s,a)G \,\mathrm{d}e \\ &= \sum_{a\in\mathcal{A}} \int_{f^{-1}(a)} P(e|s)\; q^\pi(s,a) \,\mathrm{d}e. \end{aligned} \tag{8}$$

In (8), note that the probability density, $P(e|s)$ is the internal policy, $\pi_i(e|s)$. Therefore,

$$v^\pi(s) = \sum_{a\in\mathcal{A}} \int_{f^{-1}(a)} \pi_i(e|s)\; q^\pi(s,a) \,\mathrm{d}e.$$

□

B. Proof of Lemma 2

Lemma 2. *For all deterministic functions, f, which map each point, $e \in \mathbb{R}^d$, in the representation space to an action, $a \in \mathcal{A}$, the expected updates to θ based on $\frac{\partial J_i(\theta)}{\partial\theta}$ are equivalent to updates based on $\frac{\partial J_o(\theta,f)}{\partial\theta}$. That is,*

$$\frac{\partial J_o(\theta,f)}{\partial\theta} = \frac{\partial J_i(\theta)}{\partial\theta}.$$

Proof. Recall from (1) that the probability of an action given by the overall policy, π_o, is

$$\pi_o(a|s) \coloneqq \int_{f^{-1}(a)} \pi_i(e|s) \,\mathrm{d}e.$$

Using Lemma 1, we express the performance function of the overall policy, π_o, as:

$$\begin{aligned} J_o(\theta,f) &= \sum_{s\in\mathcal{S}} d_0(s) v^{\pi_o}(s) \\ &= \sum_{s\in\mathcal{S}} d_0(s) \sum_{a\in\mathcal{A}} \int_{f^{-1}(a)} \pi_i(e|s) q^{\pi_o}(s,a) \mathrm{d}e. \end{aligned}$$

The gradient of the performance function is therefore

$$\frac{\partial J_o(\theta,f)}{\partial\theta} = \frac{\partial}{\partial\theta}\left[\sum_{s\in\mathcal{S}} d_0(s) \sum_{a\in\mathcal{A}} \int_{f^{-1}(a)} \pi_i(e|s) q^{\pi_o}(s,a) \mathrm{d}e\right].$$

Using the policy gradient theorem (Sutton et al., 2000) for the overall policy, π_o, the partial derivative of $J_o(\theta,f)$ w.r.t. θ is,

$$\begin{aligned} \frac{\partial J_o(\theta,f)}{\partial\theta} &= \sum_{t=0}^{\infty} \mathbf{E}\left[\sum_{a\in\mathcal{A}} \gamma^t q^{\pi_o}(S_t,a) \frac{\partial}{\partial\theta}\left(\int_{f^{-1}(a)} \pi_i(e|S_t) \mathrm{d}e\right)\right] \\ &= \sum_{t=0}^{\infty} \mathbf{E}\left[\sum_{a\in\mathcal{A}} \gamma^t \int_{f^{-1}(a)} \frac{\partial}{\partial\theta}\left(\pi_i(e|S_t)\right) q^{\pi_o}(S_t,a) \mathrm{d}e\right] \\ &= \sum_{t=0}^{\infty} \mathbf{E}\left[\sum_{a\in\mathcal{A}} \gamma^t \int_{f^{-1}(a)} \pi_i(e|S_t) \frac{\partial}{\partial\theta} \ln\left(\pi_i(e|S_t)\right) q^{\pi_o}(S_t,a) \mathrm{d}e\right]. \end{aligned}$$

Note that since e deterministically maps to a, $q^{\pi_o}(S_t, a) = q^{\pi_i}(S_t, e)$. Therefore,

$$\frac{\partial J_o(\theta, f)}{\partial \theta} = \sum_{t=0}^{\infty} \mathbf{E}\left[\gamma^t \sum_{a \in \mathcal{A}} \int_{f^{-1}(a)} \pi_i(e|S_t) \frac{\partial}{\partial \theta} \ln\left(\pi_i(e|S_t)\right) q^{\pi_i}(S_t, e) \mathrm{d}e\right].$$

Finally, since each e is mapped to a unique action by the function f, the nested summation over a and its inner integral over $f^{-1}(a)$ can be replaced by an integral over the entire domain of e. Hence,

$$\begin{aligned}
\frac{\partial J_o(\theta, f)}{\partial \theta} =& \sum_{t=0}^{\infty} \mathbf{E}\left[\gamma^t \int_e \pi_i(e|S_t) \frac{\partial}{\partial \theta} \ln\left(\pi_i(e|S_t)\right) q^{\pi_i}(S_t, e) \mathrm{d}e\right] \\
=& \sum_{t=0}^{\infty} \mathbf{E}\left[\gamma^t \int_e q^{\pi_i}(S_t, e) \frac{\partial}{\partial \theta} \pi_i(e|S_t) \, \mathrm{d}e\right] \\
=& \frac{\partial J_i(\theta)}{\partial \theta}.
\end{aligned}$$

□

C. Convergence of PG-RA

To analyze the convergence of PG-RA, we first briefly review existing two-timescale convergence results for actor-critics. Afterwards, we present a general setup for stochastic recursions of three dependent parameter sequences. Asymptotic behavior of the system is then discussed using three different timescales, by adapting existing multi-timescale results by Borkar (2009). This lays the foundation for our subsequent convergence proof. Finally, we prove convergence of the PG-RA method, which extends standard actor-critic algorithms using a new action prediction module, using a three-timescale approach. This technique for the proof is not a novel contribution of the work. We leverage and extend the existing convergence results of actor-critic algorithms (Borkar & Konda, 1997) for our algorithm.

C.1. Actor-Critic Convergence Using Two-Timescales

In the actor-critic algorithms, the updates to the policy depends upon a critic that can estimate the value function associated with the policy at that particular instance. One way to get a good value function is to fix the policy temporarily and update the critic in an inner-loop that uses the transitions drawn using only that fixed policy. While this is a sound approach, it requires a possibly large time between successive updates to the policy parameters and is severely sample-inefficient. Two-timescale stochastic approximation methods (Bhatnagar et al., 2009; Konda & Tsitsiklis, 2000) circumvent this difficulty. The faster update recursion for the critic ensures that asymptotically it is always a close approximation to the required value function before the next update to the policy is made.

C.2. Three-Timescale Setup

In our proposed algorithm, to update the action prediction module, one could have also considered an inner loop that uses transitions drawn using the fixed policy for supervised updates. Instead, to make such a procedure converge faster, we extend the existing two-timescale actor-critic results and take a three-timescale approach.

Consider the following system of stochastic ordinary differential equations (ODE):

$$X_{t+1} = X_t + \alpha_t^x (F_x(X_t, Y_t, Z_t) + \mathcal{N}_{t+1}^1), \tag{9}$$

$$Y_{t+1} = Y_t + \alpha_t^y (F_y(Y_t, Z_t) + \mathcal{N}_{t+1}^2), \tag{10}$$

$$Z_{t+1} = Z_t + \alpha_t^z (F_z(X_t, Y_t, Z_t) + \mathcal{N}_{t+1}^3), \tag{11}$$

where, F_x, F_y and F_g are Lipschitz continuous functions and $\{\mathcal{N}_t^1\}, \{\mathcal{N}_t^2\}, \{\mathcal{N}_t^3\}$ are the associated martingale difference sequences for noise w.r.t. the increasing σ-fields $\mathcal{F}_t = \sigma(X_n, Y_n, Z_n, \mathcal{N}_n^1, \mathcal{N}_n^2, \mathcal{N}_n^3, n \leq t), t \geq 0$, satisfying

$$\mathbf{E}[||N_{t+1}^i||^2 | \mathcal{F}_t] \leq D_1(1 + ||X_t||^2 + ||Y_t||^2 + ||Z_t||^2),$$

for $i = 1, 2, 3$, $t \geq 0$ and any constant $D < \infty$ such that the quadratic variation of noise is always bounded. To study the asymptotic behavior of the system, consider the following standard assumptions,

Assumption B1 (Boundedness). $\sup_t (||X_t|| + ||Y_t|| + ||Z_t||) < \infty$, *almost surely.*

Assumption B2 (Learning rate schedule). *The learning rates* α_t^x, α_t^y *and* α_t^z *satisfy:*

$$\sum_t \alpha_t^x = \infty, \sum_t \alpha_t^y = \infty, \sum_t \alpha_t^z = \infty,$$
$$\sum_t (\alpha_t^x)^2 < \infty, \sum_t (\alpha_t^y)^2 < \infty, \sum_t (\alpha_t^z)^2 < \infty,$$
$$As \;\; t \to \infty, \quad \frac{\alpha_t^z}{\alpha_t^y} \to 0, \; \frac{\alpha_t^y}{\alpha_t^x} \to 0. \tag{12}$$

Assumption B3 (Existence of stationary point for Y). *The following ODE has a globally asymptotically stable equilibrium* $\mu_1(Z)$*, where* $\mu_1(\cdot)$ *is a Lipschitz continuous function.*

$$\dot{Y} = F_y(Y(t), Z) \tag{13}$$

Assumption B4 (Existence of stationary point for X). *The following ODE has a globally asymptotically stable equilibrium* $\mu_2(Y, Z)$*, where* $\mu_2(\cdot, \cdot)$ *is a Lipschitz continuous function.*

$$\dot{X} = F_x(X(t), Y, Z), \tag{14}$$

Assumption B5 (Existence of stationary point for Z). *The following ODE has a globally asymptotically stable equilibrium* $Z^\star$*,*

$$\dot{Z} = F_z(\mu_2(\mu_1(Z(t)), Z(t)), \mu_1(Z(t)), Z(t)). \tag{15}$$

Assumptions B1–B2 are required to bound the values of the parameter sequence and make the learning rate well-conditioned, respectively. Assumptions B3-B4 ensure that there exists a global stationary point for the respective recursions, individually, when other parameters are held constant. Finally, Assumption B5 ensures that there exists a global stationary point for the update recursion associated with Z, if between each successive update to Z, X and Y have converged to their respective stationary points.

Lemma 3. *Under Assumptions B1-B5,* $(X_t, Y_t, Z_t) \to (\mu_2(\mu_1(Z^\star), Z^\star), \mu_1(Z^\star), Z^\star)$ *as* $t \to \infty$*, with probability one.*

Proof. We adapt the multi-timescale analysis by Borkar (2009) to analyze the above system of equations using three-timescales. First we present an intuitive explanation and then we formalize the results.

Since these three updates are not independent at each time step, we consider three step-size schedules: $\{\alpha_t^x\}$, $\{\alpha_t^y\}$ and $\{\alpha_t^z\}$, which satisfy Assumption B2. As a consequence of (12), the recursion (10) is 'faster' than (11), and (9) is 'faster' than both (10) and (11). In other words, Z moves on the slowest timescale and the X moves on the fastest. Such a timescale is desirable since Z_t converges to its stationary point if at each time step the value of the corresponding converged X and Y estimates are used to make the next Z update (Assumption B5).

To elaborate on the previous points, first consider the ODEs:

$$\dot{Y} = F_y(Y(t), Z(t)), \tag{16}$$
$$\dot{Z} = 0. \tag{17}$$

Alternatively, one can consider the ODE

$$\dot{Y} = F_y(Y(t), Z),$$

in place of (16), because Z is fixed (17). Now, under Assumption B3 we know that the iterative update (10) performed on Y, with a fixed Z, will eventually converge to a corresponding stationary point.

Now, with this converged Y, consider the following ODEs:

$$\dot{X} = F_x(X(t), Y(t), Z(t)), \tag{18}$$
$$\dot{Y} = 0, \tag{19}$$
$$\dot{Z} = 0. \tag{20}$$

Alternatively, one can consider the ODE

$$\dot{X} = F_x(X(t), Y, Z),$$

in place of (18), as Y and Z are fixed (19)-(20). As a consequence of Assumption B4, X converges when both Y and Z are held fixed.

Intuitively, as a result of Assumption B2, in the limit, the learning-rate, α_t^z becomes very small relative to α_t^y. This makes Z 'quasi-static' compared to Y and has an effect similar to fixing Z_t and running the iteration (10) forever to converge at $\mu_1(Z_t)$. Similarly, both α_t^y and α_t^z become very small relative to α_t^x. Therefore, both Y and Z are 'quasi-static' compared to X, which has an effect similar to fixing Y_t and Z_t, and running the iteration (9) forever. In turn, this makes Z_t see X_t as a close approximation to $\mu_2(\mu_1(Z(t)), Z(t))$ always, and thus Z_t converges to $Z^\star$ due to Assumption B5.

Formally, define three real-valued sequences $\{i_t\}, \{j_t\}$ and $\{k_t\}$ as $i_t = \sum_{n=0}^{t-1} \alpha_t^y, j_t = \sum_{n=0}^{t-1} \alpha_t^x$ and $k_t = \sum_{n=0}^{t-1} \alpha_t^z$, respectively. These are required for tracking the continuous time ODEs, in the limit, using discretized time. Note that $(i_t - i_{t-1}), (j_t - j_{t-1}), (k_t - k_{t-1})$ almost surely converge to 0 as $t \to \infty$.

Define continuous time processes $\bar{Y}(i), \bar{Z}(i), i \geq 0$ as $\bar{Y}(i_t) = Y_t, \bar{Z}(i_t) = Z_t$, respectively with linear interpolations in between. For $s \geq 0$, let $Y^s(i), Z^s(i), i \geq s$ denote the trajectories of (16)–(17) with $Y^s(s) = \bar{Y}(s)$ and $Z^s(s) = \bar{Z}(s)$. Note that because of (17), $\forall_i \geq s\ Z^s(i) = \bar{Z}(s)$. Now consider re-writing (10)–(11) as,

$$\begin{aligned}
Y_{t+1} &= Y_t + \alpha_t^y (F_y(Y_t, Z_t) + \mathcal{N}_{t+1}^2), \\
Z_{t+1} &= Z_t + \alpha_t^y \left(\frac{\alpha_t^z}{\alpha_t^y} (F_z(X_t, Y_t, Z_t) + \mathcal{N}_{t+1}^3) \right).
\end{aligned}$$

When the time discretization corresponds to $\{i_t\}$, this shows that (10)–(11) can be seen as 'noisy' Euler discretizations of the ODE (13) (or, equivalently of ODEs (16)–(17)), but as $\dot{Z} = 0$ this ODE has an approximation error of $\frac{\alpha_t^z}{\alpha_t^y}(F_z(X_t, Y_t, Z_t) + \mathcal{N}_{t+1}^3)$. However, asymptotically, this error vanishes as $\frac{\alpha_t^z}{\alpha_t^y} \to 0$. Now using results by Borkar (2009), it can be shown that, for any given $T \geq 0$, as $s \to \infty$,

$$\begin{aligned}
\sup_{i \in [s, s+T]} ||\bar{Y}(i) - Y^s(i)|| &\to 0, \\
\sup_{i \in [s, s+T]} ||\bar{Z}(i) - Z^s(i)|| &\to 0,
\end{aligned}$$

with probability one. Hence, in the limit, the discretization error also vanishes and $(Y(t), Z(t)) \to (\mu_1(Z(t)), Z(t))$. Similarly for (9)–(11), with $\{j_t\}$ as time discretization, and using the fact that both $\frac{\alpha_t^z}{\alpha_t^y} \to 0$, and $\frac{\alpha_t^y}{\alpha_t^x} \to 0$, a noisy Euler discretization can be obtained for ODE (14) (or equivalently for ODEs (18)–(20)). Hence, in the limit, $(X(t), Y(t), Z(t)) \to (\mu_2(\mu_1(Z(t)), Z(t)), \mu_1(Z(t)), Z(t))$.

Now consider re-writing (11) as:

$$\begin{aligned}
Z_{t+1} = Z_t & \\
&+ \alpha_t^z (F_z(\mu_2(\mu_1(Z(t)), Z(t)), \mu_1(Z(t)), Z(t))) \\
&- \alpha_t^z (F_z(\mu_2(\mu_1(Z(t)), Z(t)), \mu_1(Z(t)), Z(t))) \\
&+ \alpha_t^z (F_z(X_t, Y_t, Z_t)) \\
&+ \alpha_t^z (\mathcal{N}_{t+1}^3).
\end{aligned} \tag{21}$$

This can be seen as a noisy Euler discretization of the ODE (15), along the time-line $\{k_t\}$, with the error corresponding to the third, fourth and fifth terms on the RHS of (21). We denote these error terms as I, II and III, respectively. In the limit, using the result that $(X(t), Y(t), Z(t)) \to (\mu_2(\mu_1(Z(t)), Z(t)), \mu_1(Z(t)), Z(t))$ as $t \to \infty$, the error $I + II$ vanishes. Similarly, martingale noise error, III, vanishes asymptotically as a consequence of bounded Z values and $\sum_t (\alpha_t^z)^2 < \infty$. Now using sequence of approximations using Gronwall's inequality, it can be shown that (21) converges to Z^* asymptotically (Borkar, 2009).

Therefore, under the Assumptions B1-B5, $(X_t, Y_t, Z_t) \to (\mu_2(\mu_1(Z^\star), Z^\star), \mu_1(Z^\star), Z^\star)$ as $t \to \infty$. □

C.3. PG-RA Convergence Using Three- Timescales:

Let the parameters of the critic and the internal policy be denoted as ω and θ respectively. Also, let ϕ denote all the parameters of $\hat{f}$ and $\hat{g}$. Similar to prior work (Bhatnagar et al., 2009; Degris et al., 2012; Konda & Tsitsiklis, 2000), for analysis of the updates to the parameters, we consider the following standard assumptions required to ensure existence of gradients and bound the parameter ranges.

Assumption A3. *For any state action-representation pair (s,e), internal policy, $\pi_i(e|s)$, is continuously differentiable in the parameter θ.*

Assumption A4. *The updates to the parameters, $\theta \in \mathbb{R}^{d_\theta}$, of the internal policy, π_i, includes a projection operator $\Gamma : \mathbb{R}^{d_\theta} \to \mathbb{R}^{d_\theta}$ that projects any $x \in \mathbb{R}^{d_\theta}$ to a compact set $\mathcal{C} = \{x|c_i(x) \leq 0, i = 1, ..., n\} \subset \mathbb{R}^{d_\theta}$, where $c_i(\cdot), i = 1, ..., n$ are real-valued, continuously differentiable functions on $\mathbb{R}^{d_\theta}$ that represents the constraints specifying the compact region. For each x on the boundary of $\mathcal{C}$, the gradients of the active c_i are considered to be linearly independent.*

Assumption A5. *The iterates ω_t and ϕ_t satisfy $\sup_t (||\omega_t||) < \infty$ and $\sup_t (||\phi_t||) < \infty$.*

Let $v(\cdot)$ be the gradient vector field on $\mathcal{C}$. We define another vector field operator $\hat{\Gamma}$,

$$\hat{\Gamma}(v(\theta)) \coloneqq \lim_{h \to 0} \frac{\Gamma(\theta + hv(\theta)) - \theta}{h},$$

that projects any gradients leading outside the compact region, $\mathcal{C}$, back to $\mathcal{C}$.

Theorem 2. *Under Assumptions (A1)-(A5), the internal policy parameters θ_t converge to $\hat{\mathcal{Z}} = \left\{x \in \mathcal{C}|\hat{\Gamma}\left(\frac{\partial J_i(x)}{\partial \theta}\right) = 0\right\}$ as $t \to \infty$, with probability one.*

Proof. PG-RA algorithm considers the following stochastic update recursions for the critic, action representation modules, and the internal policy, respectively:

$$\begin{aligned}
\omega_{t+1} =& \omega_t + \alpha_t^\omega \delta_t \frac{\partial v(s)}{\partial \omega} \\
\phi_{t+1} =& \phi_t + \alpha_t^\phi \frac{-\partial \log \hat{P}(a|s, s')}{\partial \phi} \\
\theta_{t+1} =& \theta_t + \alpha_t^\theta \hat{\Gamma}\left(\delta_t \frac{\partial \log \pi_i(e|s)}{\partial \theta}\right),
\end{aligned}$$

where, δ_t is the TD-error and is given by:

$$\delta_t = r + \gamma v(s') - v(s).$$

We now establish how these updates can be mapped to the three ODEs (9)–(11) satisfying Assumptions (B1)–(B5), so as to leverage the result from Lemma 3. To do so, we must consider how the recursions are dependent on each other. Since the reward observed is a consequence of the action executed using the internal policy and the action representation module, it makes δ dependent on both ϕ and θ. Due to the use of bootstrapping and a baseline, δ is also dependent on ω. As a result, the updates to both ω and θ are dependent on all three sets of parameters. In contrast, notice that the updates to the action representation module is independent of the rewards/critic and is thus dependent only on θ and ϕ. Therefore, the ODEs that govern the update recursions for PG-RA parameters are of the form (9)–(11), where (ω, ϕ, θ) correspond directly to (X, Y, Z). The functions F_x, F_y and F_z in (9)–(11) correspond to the semi-gradients of TD-error, gradients of self-supervised loss, and policy gradients, respectively. Lemma 3 can now be leveraged if the associated assumptions are also satisfied by our PG-RA algorithm.

For requirement B1, as a result of the projection operator Γ, the internal policy parameters, θ, remain bounded. Further, by assumption, ω and ϕ always remain bounded as well. Therefore, we have that $\sup_t (||\omega_t|| + ||\phi_t|| + ||\theta_t||) < \infty$.

For requirement B2, the learning rates $\alpha_t^\omega, \alpha_t^\phi$ and α_t^θ are hyper-parameters and can be set such that as $t \to \infty$,

$$\frac{\alpha_t^\theta}{\alpha_t^\phi} \to 0, \frac{\alpha_t^\phi}{\alpha_t^\omega} \to 0,$$

to meet the three-timescale requirement in Assumption B2.

For requirement B3, recall that when the internal policy has fixed parameters, θ, the updates to the action representation component follows a supervised learning procedure. For linear parameterization of estimators $\hat{f}$ and $\hat{g}$, the action prediction module is equivalent to a bi-linear neural network. Multiple works have established that for such models, there are no spurious local minimas and the Hessian at every saddle point has at least one negative eigenvalue (Kawaguchi, 2016; Haeffele & Vidal, 2017; Zhu et al., 2018). Further, the global minima can be achieved by stochastic gradient descent. This ensures convergence to the required critical point and satisfies Assumption B3.

For requirement B4, given a fixed policy (fixed action representations and fixed internal policy) the proof of convergence of a linear critic to the stationary point $\mu_2(\phi, \theta)$ using TD(λ) is a well established result (Tsitsiklis & Van Roy, 1996). We use $\lambda = 0$ in our algorithm, the proof however carries through for $\lambda > 0$ as well. This satisfies Assumption B4.

For requirement B5, the condition can be relaxed to a local rather than global asymptotically stable fixed point, because we only need convergence. Under the optimal critic and action representations for every step, the internal policy follows its internal policy gradient. Using Lemma 2, we established that this is equivalent to following the policy gradient of the overall policy and thus the internal policy converges to its local fixed point as well.

This completes the necessary requirements, the remaining proof now follows from Lemma 3. □

D. Implementation Details

D.1. Parameterization

In our experiments, we consider a parameterization that minimizes the computational complexity of the algorithm. Learning the parameters of the action representation module, as in (5), requires computing the value $\hat{P}(a|s, s')$ in (3). This involves a complete integral over e. Due to the absence of any closed form solution, we need to rely on a stochastic estimate. Depending on the dimensions of e, an extensive sample based evaluation of this expectation can be computationally expensive. To make this more tractable, we approximate (3) by mean-marginalizing it using the estimate of the mean from $\hat{g}$. That is, we approximate (3) as $\hat{f}(a|\hat{g}(s, s'))$. We then parameterize $\hat{f}(a|\hat{g}(s, s'))$ as,

$$\text{where,} \quad \begin{aligned} \hat{f}(a|\hat{g}(s, s')) =& \frac{\mathbf{e}^{z_a/\tau}}{\sum_{a'} \mathbf{e}^{z_{a'}/\tau}}, \\ z_a =& W_a^\top \hat{g}(s, s'). \end{aligned} \tag{22}$$

This estimator, $\hat{f}$, models the probability of any action, a, based on its similarity with a given representation e. In (22), $W \in \mathbb{R}^{d_e \times |\mathcal{A}|}$ is a matrix where each column represents a learnable action representation of dimension $\mathbb{R}^{d_e}$. $W_a^\top$ is the transpose of the vector corresponding to the representation of the action a, and z_a is its measure of similarity with the embedding from $\hat{g}(s, s')$. To get valid probability values, a Boltzmann distribution is used with τ as a temperature variable. In the limit when $\tau \to 0$ the conditional distribution over actions becomes the required deterministic estimate for $\hat{f}$. That is, the entire probability mass would be on the action, a, which has the most similar representation to e. To ensure empirical stability during training, we relax τ to 1. During execution, the action, a, which has the most similar representation to e, is chosen for execution. In practice, the linear decomposition in (22) is not restrictive as $\hat{g}$ can still be any differentiable function approximator, like a neural network.

D.2. Hyper-parameters

For the maze domain, single layer neural networks were used to parameterize both the actor and critic, and the learning rates were searched over $\{1e-2, 1e-3, 1e-4, 1e-5\}$. State features were represented using the 3^{rd} order coupled Fourier basis (Konidaris et al., 2011). The discounting parameter γ was set to 0.99 and λ to 0.9. Since it was a toy domain, the dimensions of action representations were fixed to 2. 2000 randomly drawn trajectories were used to learn an intial representation for the actions. Action representations were only trained once in the beginning and kept fixed from there on.

For the real-world environments, 2 layer neural networks were used to parameterize both the actor and critic, and the learning rates were searched over $\{1e-2, 1e-3, 1e-4, 1e-5\}$. Similar to prior work, the module for encoding state features was shared to reduce the number of parameters, and the learning rate for it was additionally searched over $\{1e-2, 1e-3, 1e-4, 1e-5\}$. The dimension of the neural network's hidden layer was searched over $\{64, 128, 256\}$.

The discounting parameter γ was set to 0.9. For actor-critic based results λ was set to 0.9 and for DPG the target actor and policy update rate was fixed to its default setting of 0.001. The dimension of action representations were searched over $\{16, 32, 64\}$. Initial $10{,}000$ randomly drawn trajectories were used to learn an initial representation for the actions. The action prediction component was continuously improved on the fly, as given in PG-RA algorithm.

For all the results of the PG-RA based algorithms, since π_i was defined over a continuous space, it was parameterized as the isotropic normal distribution. The value for variance was searched over $\{0.1, 0.25, 1, -1\}$, where -1 represents learned variance. Function $\hat{g}$ was parameterized to concatenate the state features of both s and s' and project to the embedding space using a single layer neural network with *Tanh* non-linearity. To keep the effective range of action representations bounded, they were also transformed by *Tanh* non-linearity before computing the similarity scores. Though the similarity metric is naturally based on the dot product, other distance metrics are also valid. We found squared Euclidean distance to work best in our experiments. The learning rates for functions $\hat{f}$ and $\hat{g}$ were jointly searched over $\{1e-2, 1e-3, 1e-4\}$. All the results were obtained for 10 different seeds to get the variance.

As our proposed method decomposes the overall policy into two components, the resulting architecture resembles that of a one layer deeper neural network. Therefore, for the baselines, we ran the experiments with a hyper-parameter search for policies with additional depths $\{1, 2, 3\}$, each with different combinations of width $\{2, 16, 64\}$. The remaining architectural aspects and properties of the hyper-parameter search for the baselines were performed in the same way as mentioned above for our proposed method. All the results presented in Figure 5 corresponds to the hyper-parameter setting that performed the best.