7. Appendix

7.1. Why SVRPG does not work

While this *importance sampling* technique removes the bias, the variance of estimator $\mathbf{g}_{vr}^t$ cannot be properly bounded since

$$\begin{aligned}&\mathbb{E}_{\mathcal{M}}\|\mathbf{g}^t - \nabla J(\theta^t)\|^2\\ \leq&\frac{1}{|\mathcal{M}|}\mathbb{E}_\tau\|\mathbf{g}(\theta^t;\{\tau\}) - w(\theta^t,\tilde{\theta};\tau)\mathbf{g}(\theta^t;\{\tau\})\|^2\\ =&\frac{1}{|\mathcal{M}|}\int_\tau \frac{\mathcal{R}(\tau)}{p_\tau^t}\|p_\tau^t\log\nabla p_\tau^t - \tilde{p}_\tau\log\nabla\tilde{p}_\tau\|^2\mathbf{d}\tau,\end{aligned}$$

where $p_\tau^t := p(\tau;\pi_{\theta^t})$, $\tilde{p}_\tau := p(\tau;\pi_{\tilde{\theta}})$, and the term $\frac{1}{p_\tau^t}$ in the integral can be infinity large. The lack of proper variance control deprives SVRPG of its high sample-efficiency: by scrutinizing the convergence result, $\mathcal{O}(\frac{1}{\epsilon^4})$ random trajectories are still required to achieve an ϵ-FOSP (4), which is the same as the original policy-gradient type method.

7.2. Derivation of Policy Gradient and Policy Hessian

Let $\tau = \{s_1, a_1, \ldots, s_H, a_H\}$ be a trajectory sampled according to $p(\tau;\pi_\theta)$ and define $\tau_h := \{s_1, a_1, \ldots, s_h, a_h\}$ for any $h \in [H]$. For simplicity of notation we will denote

$$\ell_\theta^{\tau_h} := \log p(\tau_h;\pi_\theta), \quad \bar{\mathcal{R}}_\gamma^{\tau_h} := \gamma^h\bar{\mathcal{R}}(a_h|s_h)$$

in the following discussion. From (3) and (2), we have

$$J(\theta) = \sum_{h=1}^H \mathbb{E}_{\tau\sim p(\tau;\pi_\theta)}[\bar{\mathcal{R}}_\gamma^{\tau_h}] = \sum_{h=1}^H \mathbb{E}_{\tau_h\sim p(\tau_h;\pi_\theta)}[\bar{\mathcal{R}}_\gamma^{\tau_h}],$$

where we replace τ by τ_h since $\bar{\mathcal{R}}_\gamma^{\tau_h}$ is independent of the randomness after a_h. To compute the policy gradient

$$\begin{aligned}\nabla J(\theta) &= \sum_{h=1}^H\int_{\tau_h}\bar{\mathcal{R}}_\gamma^{\tau_h}\nabla p(\tau_h;\pi_\theta)\mathbf{d}\tau_h\\ &= \sum_{h=1}^H\int_{\tau_h}\bar{\mathcal{R}}_\gamma^{\tau_h}p(\tau_h;\pi_\theta)\nabla\ell_\theta^{\tau_h}\mathbf{d}\tau_h,\end{aligned}$$

where we use the log-trick in the second equation

$$\nabla p(\tau_h;\pi_\theta) = p(\tau_h;\pi_\theta)\nabla\log p(\tau_h;\pi_\theta) = p(\tau_h;\pi_\theta)\nabla\ell_\theta^{\tau_h}.$$

The policy gradient can be further simplified:

$$\begin{aligned}\nabla J(\theta) &= \sum_{h=1}^H\int_{\tau_h}\bar{\mathcal{R}}_\gamma^{\tau_h}p(\tau_h;\pi_\theta)\nabla\ell_\theta^{\tau_h}\mathbf{d}\tau_h\\ &= \sum_{h=1}^H\mathbb{E}_{\tau_h\sim p(\tau_h;\pi_\theta)}[\bar{\mathcal{R}}_\gamma^{\tau_h}\sum_{i=1}^h\nabla\log\pi_\theta(a_i|s_i)]\\ &= \sum_{h=1}^H\sum_{i=1}^h\mathbb{E}_{\tau_h\sim p(\tau_h;\pi_\theta)}[\bar{\mathcal{R}}_\gamma^{\tau_h}\nabla\log\pi_\theta(a_i|s_i)]\\ &= \sum_{h=1}^H\sum_{i=1}^h\mathbb{E}_{\tau\sim p(\tau;\pi_\theta)}[\bar{\mathcal{R}}_\gamma^{\tau_h}\nabla\log\pi_\theta(a_i|s_i)],\end{aligned}$$

where in the last equality we use that $\bar{\mathcal{R}}_\gamma^{\tau_h}\nabla\log\pi_\theta(a_i|s_i)$ with $i \leq h$ is independent of the randomness after a_h. Exchange the summation over i and h to obtain

$$\begin{aligned}\nabla J(\theta) &= \sum_{i=1}^H\sum_{h=i}^H\mathbb{E}_{\tau\sim p(\tau;\pi_\theta)}[\bar{\mathcal{R}}_\gamma^{\tau_h}\nabla\log\pi_\theta(a_i|s_i)]\\ &= \sum_{i=1}^H\mathbb{E}_{\tau\sim p(\tau;\pi_\theta)}[\left(\sum_{h=i}^H\bar{\mathcal{R}}_\gamma^{\tau_h}\right)\nabla\log\pi_\theta(a_i|s_i)]\\ &= \sum_{i=1}^H\mathbb{E}_{\tau\sim p(\tau;\pi_\theta)}[\Psi_i(\tau)\nabla\log\pi_\theta(a_i|s_i)],\end{aligned}$$

where $\Psi_i := \sum_{h=i}^H\gamma^h\bar{\mathcal{R}}(a_h|s_h)$ is the discounted reward after action a_i given state s_i. Let

$$\Phi(\theta;\tau) = \sum_{i=1}^H\Psi_i(\tau)\log p(a_i|s_i;\pi_\theta).$$

Using such notation, we have

$$\begin{aligned}\nabla J(\theta) &= \mathbb{E}_{\tau\sim p(\tau;\pi_\theta)}\nabla\Phi(\theta;\tau)\\ &= \int_\tau p(\tau;\pi_\theta)\nabla\Phi(\theta;\tau)\mathbf{d}\tau.\end{aligned}$$

The second order derivative can be computed by

$$\begin{aligned}&\nabla^2 J(\theta)\\ =&\int_\tau\nabla\Phi(\theta;\tau)\nabla p(\tau;\pi_\theta)^\top + p(\tau;\pi_\theta)\nabla^2\Phi(\theta;\tau)\mathbf{d}\tau\\ =&\int_\tau p(\tau;\pi_\theta)\left[\nabla\Phi(\theta;\tau)\nabla\log p(\tau;\pi_\theta)^\top + \nabla^2\Phi(\theta;\tau)\right]\mathbf{d}\tau\\ =&\mathbb{E}_{\tau\sim p(\tau;\pi_\theta)}\left[\nabla\Phi(\theta;\tau)\nabla\log p(\tau;\pi_\theta)^\top + \nabla^2\Phi(\theta;\tau)\right].\end{aligned}$$

7.3. Detail Hyper-parameter Settings

We present the Hyper-parameter settings in Table 1. The code for our experiments are available in https://github.com/m1zju/HAPG.

Table 1. Hyper-parameter Settings

	CartPole	Swimmer	Reacher	Walker2d	Humanoid	HumanoidStandup
Horizon	100	500	50	500	500	500
Baseline	No	Linear	Linear	Linear	Linear	Linear
Number of timesteps	$5 \cdot 10^5$	10^7	10^7	10^7	10^7	10^7
NN sizes	8	32x32	32x32	64x64	64x64	64x64
REINFORCE learning rate	0.01	0.01	0.01	0.01	0.01	0.01
REINFORCE batchsize	50	100	100	100	100	100
HAPG learning rate	0.01	0.01	0.01	0.01	0.01	0.01
HAPG $\|\mathcal{M}_0\|$	50	100	100	100	100	100
HAPG $\|\mathcal{M}\|$	10	10	10	10	10	10
HAPG p	5	10	10	10	10	10