In this work, we consider the reinforcement learning setting where an agent interacts with an environment ε over a number of discrete time steps. At each time step t, the agent receives a state s_t and selects an action a_t according to its policy π, which maps states s_t to actions a_t. In return, the agent receives a scalar reward r_t and the next state s_{t+1}. This process continues until the agent reaches a terminal state. The return $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$ is the total accumulated and discounted reward from time step t. The objective of the agent is to maximize the expected return from each state s_t.

Policy based models parametrize the policy $\pi(a|s;\theta)$. They work by computing an estimator

$$\hat{g} = \hat{\mathbb{E}}_t[\nabla_\theta \; log \; \pi_\theta(a_t|s_t)\hat{A}_t] \tag{1}$$

and plugging it into a stochastic gradient ascent algorithm. $\hat{A}_t$ corresponds to an estimator of the advantage function at time step t.

The initial objective of the agent to learn the policy $\pi(a|s;\theta)$ is augmented so that it will be robust to the value of nuisance parameters $z \in Z$ which remain unknown at test time. A formal way of enforcing this is to require that

$$\pi(a|s, z; \theta) = \pi(a|s, z'; \theta) \tag{2}$$

for all z, $z' \in Z$, all values of $a \in A$ and all values of $s \in S$.