Stackelberg Actor-Critic: A Game-Theoretic Perspective

Liyuan Zheng[1], **Tanner Fiez**[1], **Zane Alumbaugh**[2], **Benjamin Chasnov**[1], **Lillian J. Ratliff**[1]
[1]University of Washington [2]University of California, Santa Cruz
{liyuanz8,fiezt,bchasnov,ratliffl}@uw.edu, zanedma@gmail.com

Abstract

Actor-critic methods solve reinforcement learning problems by updating a critic to approximate the expected return of the actor and simultaneously updating actor in a direction based on the critic's estimation. The interaction between the actor and critic has an intrinsic hierarchical structure from a game-theoretic perspective. In this work, to take into account the interaction structure between the players, we formulate the actor-critic method as a two-player general-sum Stackelberg game. We propose a Stackelberg actor-critic algorithm that leverages the Stackelberg gradient update following the total derivative, where the actor optimizes utilizing the knowledge that the critic responds near-optimally to the update by the actor. Through experiments we validate that our proposed algorithms outperform the normal actor-critic method. We believe this game-theoretic perspective can be extended to general actor-critic based methods and provide more insights on a broader class of reinforcement learning algorithms.

1 Introduction

The goal of reinforcement learning is to learn an optimal policy under which an agent maximizes the obtainable cumulative reward.[1] Reinforcement learning has proven to be successful problem solving framework in a variety of domains such as video games (Mnih et al. 2015; Silver et al. 2016), robotics (Lillicrap et al. 2015; Levine et al. 2016), autonomous vehicles (Sallab et al. 2017), among many others.

The algorithmic techniques for reinforcement learning can be classified into *policy-based*, *value-based*, and *actor-critic* methods. Policy-based methods directly optimize a policy to maximize the sample approximation of the expected return. Value-based methods instead learn a value function that estimates the expected return, and they then infer an optimal policy by selecting actions that maximize the learned value function. However, there are disadvantages of pure policy-based and value-based methods when applied to continuous control problems (Duan et al. 2016). Indeed, policy-based methods are known to be sample inefficient and suffer from high variance, while value-based methods face a computational bottleneck in solving for the value-maximizing action.

Actor-critic methods combine the advantages of policy-based and value-based methods. In such methods, the parameterized policy is called the actor and the learned value function is called the critic. By learning both the actor and the critic simultaneously, actor-critic methods manage to reduce the return estimation variance (Konda and Tsitsiklis 2000; Grondman et al. 2012), and bypass the maximization problem by instead querying the actor (Silver et al. 2014; Lillicrap et al. 2015).

At a high level, actor-critic methods learn a critic that approximates the expected return of the actor, while at the same time learn an actor to optimize the expected return based on the critic's estimation. The interaction between the actor and critic has an intrinsic hierarchical structure in which the critic seeks to be at an optimum given the parameters of the actor, while the actor aims to be at an optimum knowing that the critic responds near-optimally to the parameters selected by the actor. The interaction structure between actor and critic can be viewed as a *Stackelberg game*.

Stackelberg games characterize the interaction between a leader and a follower. The leader in the game is distinguished by the ability to act before the follower. As a result of this structure, the leader optimizes accounting for how the follower responds, while the follower selects a best response to the action of the leader. The typical equilibrium concept studied in this class of games is known as a Stackelberg equilibrium. The importance of the order of play in optimization problems present in machine learning applications such as generative adversarial networks has spurred the development of local refinements of the Stackelberg equilibrium notion (Fiez, Chasnov, and Ratliff 2020; Jin, Netrapalli, and Jordan 2020) along with the design and analysis of iterative algorithms seeking to compute local Stackelberg equilibrium in nonconvex-nonconcave games (Fiez, Chasnov, and Ratliff 2020; Jin, Netrapalli, and Jordan 2020; Wang, Zhang, and Ba 2019; Fiez and Ratliff 2020).

We formulate the actor-critic method as a two-player general-sum Stackelberg game toward solving reinforcement learning problems. In this formulation, the actor seeks to solve a bilevel optimization problem in which the actor objective is a function of the critic's parameters and the critic responds optimally with respect to its own parameters.

[1]Following common terminology (Sutton and Barto 2018), we refer to the discounted cumulative reward as the return in this work.

For general Stackelberg games, Fiez, Chasnov, and Ratliff (2020) proposed a learning algorithm with a number of theoretical properties in which the leader updates by following the total derivative of its cost function defined using the implicit function theorem, while the follower descends its cost using the derivative with respect to its own parameters. We tailor this learning algorithm to the reinforcement learning problem and design a novel *Stackelberg actor-critic* algorithm that explicitly takes into account the interaction structure between the players. This is in contrast to existing actor-critic methods that do not explicitly consider the interactions between the players and only perform standard gradient descent-ascent. We demonstrate via experiments on several reinforcement learning tasks that our algorithm outperforms the normal actor-critic method. This behavior is an outcome of the careful consideration of the interaction structure. Moreover, our viewpoint has the advantage that game-theoretic equilibria are more robust to local deviations by the the follower or inner optimization problem, which is important in reinforcement learning to ensure robustness to errors from sampling bias and variance and approximations and derivatives.

1.1 Related Work

Game-theoretic frameworks have been studied extensively in multi-agent reinforcement learning (Zhang, Yang, and Başar 2019). Recently, Prajapat et al. (2020) proposed a competitive policy optimization method for multi-agent reinforcement learning that exploits the game-theoretic nature of competitive games and performs recursive reasoning about the behavior of an opponent in two-player zero-sum games. In contrast, in our Stackelberg game formulation between the actor and critic, the actor is reasoning about how the critic responds to its own update in a single agent reinforcement learning problem. The past research taking a game-theoretic viewpoint of single-agent reinforcement learning is limited despite the fact that there are often multiple players (e.g., actor and critic) in reinforcement learning algorithms. Rajeswaran, Mordatch, and Kumar (2020) propose a framework that casts model-based reinforcement learning as a game between a policy player and a model player. They construct a Stackelberg game between the two players and study different order of players. However, instead of leveraging the Stackelberg gradient update using the implicit function theorem, they only consider gradient descent-ascent to approximate the Stackelberg dynamics.

In single-agent reinforcement learning methods, algorithms using second-order information as we do in this work traces back to natural policy gradient methods (Kakade 2001) and the natural actor-critic algorithm (Peters and Schaal 2008; Bhatnagar et al. 2009). Recently, second-order methods have been proposed for both policy-based and actor-critic methods (Schulman et al. 2015a, 2017; Shen et al. 2019; Tangkaratt, Abdolmaleki, and Sugiyama 2017). The actor-critic based methods among those often use local second-order information to construct a constrained optimization problem. On the contrary, we construct a bilevel optimization, which allows us to consider the interaction between actor and critic in a Stackelberg game.

2 Preliminaries

In this section, we provide background on the actor-critic algorithm and Stackelberg games.

2.1 Actor-Critic

We consider discrete-time Markov decision processes (MDPs) with continuous state space $\mathcal{S}$ and continuous action space $\mathcal{A}$. We denote the state and action at time step t by s_t and a_t, respectively. The initial state s_0 is determined by the initial state density $s_0 \sim \rho(s)$. At time step t, the agent in state s_t takes an action a_t according to a policy $a_t \sim \pi(a|s_t)$ and obtains a reward $r_t = r(s_t, a_t)$. Then, the agent is transited into the next state s_{t+1} determined by the transition function $s_{t+1} \sim P(s'|s_t, a_t)$. A trajectory $\tau = (s_0, a_0, \ldots, s_T, a_T)$ gives us the cumulative rewards or return defined as $R(\tau) = \sum_{t=0}^{T} \gamma^t r(s_t, a_t)$, where the discount factor $0 < \tau \leq 1$ assigns weights to rewards received at different time steps. The expected return of π after executing a_t in state s_t can be expressed by the Q function defined as

$$Q^\pi(s_t, a_t) = \mathbf{E}_{\tau\sim\pi}\left[\sum_{t'=t}^{T} \gamma^{t'-t} r(s_{t'}, a_{t'})|s_t, a_t\right]. \quad (1)$$

Correspondingly, the expected return of π in state s_t can be expressed by the V function defined as

$$V^\pi(s_t) = \mathbf{E}_{\tau\sim\pi}\left[\sum_{t'=t}^{T} \gamma^{t'-t} r(s_{t'}, a_{t'})|s_t\right]. \quad (2)$$

The goal of reinforcement learning is to find an optimal policy that maximizes the expected return:

$$\begin{aligned} J(\pi) &= \mathbf{E}_{\tau\sim\pi}\left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t)\right] = \int_\tau p(\tau|\pi) R(\tau) \mathrm{d}\tau \\ &= \mathbf{E}_{s\sim\rho, a\sim\pi(\cdot|s)}\left[Q^\pi(s, a)\right], \end{aligned} \quad (3)$$

where $p(\tau|\pi) = \rho(s_0) \prod_{t=0}^{T} \pi(a_t|s_t) P(s_{t+1}|s_t, a_t)$.

The policy-based approach (Williams 1992) parameterizes π by parameter θ and finds the optimal θ^* by maximizing the expected return:

$$\max_\theta J(\theta) = \mathbf{E}_{s\sim\rho, a\sim\pi_\theta(\cdot|s)}\left[Q^\pi(s, a)\right]. \quad (4)$$

According to the policy gradient theorem (Sutton et al. 2000)

$$\nabla_\theta J(\theta) = \mathbf{E}_{s\sim\rho, a\sim\pi_\theta(\cdot|s)}\left[\nabla_\theta \log \pi_\theta(a|s) Q^\pi(s, a)\right], \quad (5)$$

and the optimization problem can be solved by gradient ascent. One way to approximate the $Q^\pi(s, a)$ in the gradient is by sampling trajectories and averaging returns. Such method is known as REINFORCE (Williams 1992).

The actor-critic method (Konda and Tsitsiklis 2000; Grondman et al. 2012) leverages another critic function $Q_w(s, a)$, parameterized by w, to approximate $Q^\pi(s, a)$. By replacing the value function in Eq. (4), we obtain the following optimization problem

$$\max_\theta J(\theta) = \mathbf{E}_{s\sim\rho, a\sim\pi_\theta(\cdot|s)}\left[Q_w(s, a)\right]. \quad (6)$$

Similarly, the optimization is solved by gradient ascent and the gradient now is

$$\nabla_\theta J(\theta) = \mathbf{E}_{s\sim\rho, a\sim\pi_\theta(\cdot|s)} \left[\nabla_\theta \log \pi_\theta(a|s) Q_w(s,a)\right]. \quad (7)$$

The critic is optimized by minimizing the error between true value functions

$$\min_w L(w) = \mathbf{E}_{s\sim\rho, a\sim\pi_\theta(\cdot|s)} \left[\left(Q_w(s,a) - Q^\pi(s,a)\right)^2\right], \quad (8)$$

where the true value function is approximated by Monte Carlo estimation or bootstrapping (Sutton and Barto 2018). Actor-critic method typically performs direct gradient descent-ascent on critic and actor, respectively (Peters and Schaal 2008; Mnih et al. 2016):

$$\theta \leftarrow \theta + \alpha_\theta \nabla_\theta J(\theta), \quad (9)$$

$$w \leftarrow w - \alpha_w \nabla_w L(w), \quad (10)$$

where α_θ and α_w are the learning rate of actor and critic.

2.2 Stackelberg Game

Stackelberg game is a game between two agents where one agent is deemed the leader and the other the follower. Each agent has an objective they want to optimize that depends on not only their own actions but also on the actions of their competitor. Specifically, the leader optimizes its objective knowing that the follower will respond optimally. Let $f_1(x_1, x_2)$ and $f_2(x_1, x_2)$ be the objective functions that the leader and follower want to minimize, respectively, where x_1 and x_2 are their decision variables. The leader aims to solve the bilevel optimization problem given by

$$\min_{x_1} \left\{ f_1(x_1, x_2) | x_2 = \arg\min_y f_2(x_1, y) \right\}. \quad (11)$$

Since the follower chooses the best response $x_2^*(x_1) = \arg\min_y f_2(x_1, y)$, the follower's decision variables are implicitly a function of the leader's. The leader utilizes this information by the total derivative of its cost function:

$$\frac{\mathrm{d}f_1(x_1, x_2^*(x_1))}{\mathrm{d}x_1} = \frac{\partial f_1(x_1,x_2)}{\partial x_1} + \frac{\mathrm{d}x_2^*(x_1)}{\mathrm{d}x_1} \frac{\partial f_1(x_1,x_2)}{\partial x_2}. \quad (12)$$

The implicit Jacobian term can be obtained using the implicit function theorem (Krantz and Parks 2012):

$$\frac{\mathrm{d}x_2^*(x_1)}{\mathrm{d}x_1} = -\left(\frac{\partial^2 f_2(x_1,x_2)}{\partial x_1 \partial x_2}\right)\left(\frac{\partial^2 f_2(x_1,x_2)}{\partial x_2^2}\right)^{-1}. \quad (13)$$

3 Stackelberg Actor-Critic

In this section, in order to capture the hierarchical interactions between value learning and policy optimization, we formulate actor-critic as a two-player general sum Stackelberg game. In this game, the actor and critic can only pick their own parameters while their objectives depend on the parameters of both. We propose Stackelberg Actor-Critic (STAC) algorithm, where the actor optimizes its objective knowing the critic responds optimally to its update.

In a two-player Stackelberg game setting of actor-critic, the critic objective is now a function of both players' parameters:

$$L(\theta, w) = \mathbf{E}_{s\sim\rho, a\sim\pi_\theta(\cdot|s)} \left[\left(Q_w(s,a) - Q^\pi(s,a)\right)^2\right]. \quad (14)$$

The critic assists to compute the policy gradient by approximating the value function of the current policy. To give an accurate approximation, the critic should be selecting a best response $w^*(\theta) = \arg\min_\phi L(\theta, \phi)$. Thus, the actor naturally plays the role of leader and the critic plays follower. Utilizing the knowledge that the critic will always choose best response while actor updates, the actor aims to solve the bilevel optimization problem given by

$$\max_\theta \quad J(\theta, w^*(\theta)) \quad (15)$$

$$\text{s.t.} \quad w^*(\theta) = \arg\min_\phi L(\theta, \phi), \quad (16)$$

where the actor objective is also a function of both actor and critic parameters:

$$J(\theta, w) = \mathbf{E}_{s\sim\rho, a\sim\pi_\theta(\cdot|s)}[Q_w(s,a)]. \quad (17)$$

According to Eq. (12) and (13), the total derivative of $J(\theta, w^*(\theta))$ is computed by

$$\frac{\mathrm{d}J(\theta, w^*(\theta))}{\mathrm{d}\theta} = \frac{\partial J(\theta,w)}{\partial \theta} + \frac{\mathrm{d}w^*(\theta)}{\mathrm{d}\theta}\frac{\partial J(\theta,w)}{\partial w} \quad (18)$$

$$= \frac{\partial J(\theta,w)}{\partial \theta} - \left(\frac{\partial^2 L(\theta,w)}{\partial \theta \partial w}\right)\left(\frac{\partial^2 L(\theta,w)}{\partial w^2}\right)^{-1}\frac{\partial J(\theta,w)}{\partial w}. \quad (19)$$

For the terms in Eq. (19), $\frac{\partial J(\theta,w)}{\partial \theta}$ can be computed by policy gradient theorem in Eq. (7); $\frac{\partial J(\theta,w)}{\partial w}$ and $\frac{\partial^2 L(\theta,w)}{\partial w^2}$ can be computed by taking the direct derivative

$$\frac{\partial J(\theta,w)}{\partial w} = \mathbf{E}_{s\sim\rho, a\sim\pi_\theta(\cdot|s)}\left[\frac{\mathrm{d}Q_w(s,a)}{\mathrm{d}w}\right], \quad (20)$$

and

$$\frac{\partial^2 L(\theta,w)}{\partial w^2} = \mathbf{E}_{s\sim\rho, a\sim\pi_\theta(\cdot|s)}\left[\frac{\partial}{\partial w^2}\left(Q_w(s,a) - Q^\pi(s,a)\right)^2\right] \quad (21)$$

$$= \mathbf{E}_{s\sim\rho, a\sim\pi_\theta(\cdot|s)}\left[2\frac{\mathrm{d}Q_w(s,a)}{\mathrm{d}w}\frac{\mathrm{d}Q_w(s,a)}{\mathrm{d}w}^\top + 2(Q_w(s,a) - Q^\pi(s,a))\frac{\mathrm{d}^2 Q_w(s,a)}{\mathrm{d}w^2}\right]. \quad (22)$$

To compute $\frac{\partial^2 L(\theta,w)}{\partial w \partial \theta}$ in Eq. (19), we first compute $\frac{\partial L(\theta,w)}{\partial \theta}$ by the following theorem:

Theorem 1. *Given a MDP and actor critic parameters θ, w,*

$$\frac{\partial L(\theta,w)}{\partial \theta} = \int_\tau \Bigg(p(\tau_0|\theta)\nabla_\theta \log \pi_\theta(a_0|s_0)(Q_w(s_0,a_0) - Q^\pi(s_0,a_0))^2 + 2\sum_{t=1}^{T}\gamma^t p(\tau_{0:t}|\theta)\nabla_\theta \log \pi_\theta(a_t|s_t) \left(Q^\pi(s_0,a_0) - Q_w(s_0,a_0)\right) Q^\pi(s_t,a_t)\Bigg)\mathrm{d}\tau. \quad (23)$$

The proof of Theorem 1 is in Appendix A.1. With this theorem, we can compute $\frac{\partial^2 L(\theta,w)}{\partial w \partial \theta}$ by further taking the derivative of $\frac{\partial L(\theta,w)}{\partial \theta}$ with respect to w. Note that if value function $V_w(s)$ is used as the critic, $\frac{\partial L(\theta,w)}{\partial \theta}$ can be computed by the following proposition.

Proposition 1. *If the objective of critic is* $L(\theta, w) = \mathbf{E}_{s\sim\rho}\left[\left(V_w(s) - V^\pi(s)\right)^2\right]$, *then*

$$\frac{\partial L(\theta,w)}{\partial \theta} = 2\int_\tau \left(\sum_{t=0}^{T} \gamma^t p(\tau_{0:t}|\theta)\nabla_\theta \log \pi_\theta(a_t|s_t) \left(V^\pi(s_0) - V_w(s_0)\right) Q^\pi(s_t)\right) \mathrm{d}\tau. \tag{24}$$

The proof of Proposition 1 is in Appendix A.2.

Once each term in Eq. (19) are computed, STAC performs the following Stackelberg gradient update (Fiez, Chasnov, and Ratliff 2020):

$$\theta \leftarrow \theta + \alpha_\theta \frac{\mathrm{d}J(\theta, w^*(\theta))}{\mathrm{d}\theta}, \tag{25}$$

$$w \leftarrow w - \alpha_w \frac{\partial L(\theta, w)}{\partial w}. \tag{26}$$

Note that the actor's update in Eq. (25) differs from that in (9) as the actor is reasoning critic's response of its owner update. In practice, in order to maintain best response in the inner level with an iterative optimization algorithm, a number of unrolling gradient steps of critic update in Eq. (26) and (10) are performed.

3.1 Hessian Regularization

In Eq. (19), we compute the inverse of critic Hessian $\frac{\partial^2 L(\theta,w)}{\partial w^2}$. However, when the critic parameter w is not in a neighborhood of critical points, the Hessian matrix might be ill-conditioned. Depending on the structure of critic objective $L(\theta, w)$ and critic function class $Q_w(s, a)$, the Hessian matrix might not be invertible or it may have eigenvalues very close to zero. In STAC, instead of computing the inverse of the Hessian matrix directly, we compute the inverse of a regularized Hessian $\frac{\partial^2 L(\theta,w)}{\partial w^2} + \lambda I$. The regularization hyperparameter λ controls the trade-off between Stackelberg and normal gradient update. When $\lambda \to \infty$, the eigenvalues of $\left(\frac{\partial^2 L(\theta,w)}{\partial w^2} + \lambda I\right)^{-1}$ becomes zero and the second term in Eq. (19) is erased. Thus, the update in Eq. (25) becomes equivalent to that in Eq. (9), and STAC resumes normal actor-critic. When $\lambda = 0$, STAC performs pure Stackelberg gradient update, and when λ takes a positive number, STAC is a mixture of Stackelberg and normal gradient update.

4 Experiments

In this section, we evaluate the performance of the STAC method on OpenAI gym platform (Brockman et al. 2016) with the Mujoco Physics simulator (Todorov, Erez, and Tassa 2012). The normal actor-critic (AC in Fig. 1) method is used as the baseline, where the actor is updated by GAE (Schulman et al. 2015b) and critic by Monte Carlo method. For fair comparison, all the hyperparameters including actor and critic neural network architectures are set the same for STAC and AC in all experiments. The actor and critic are neural networks with two hidden layers of 64×64 with tanh nonlinear activation functions follows. The only difference between STAC and AC are the update rules they adopted as defined in Eq. (25), (26) and Eq. (9) and (10).

The performance is evaluated by average episode return versus the time steps. The time steps are the number of state transitions after taking an action according to the policy. One learning epoch contains a fixed number of time steps, which may consist of several episodes depending on the environment. One step actor update and k steps unrolling critic updates are executed after each epoch. Fig. 1 shows the learning performance on several different environment tasks.

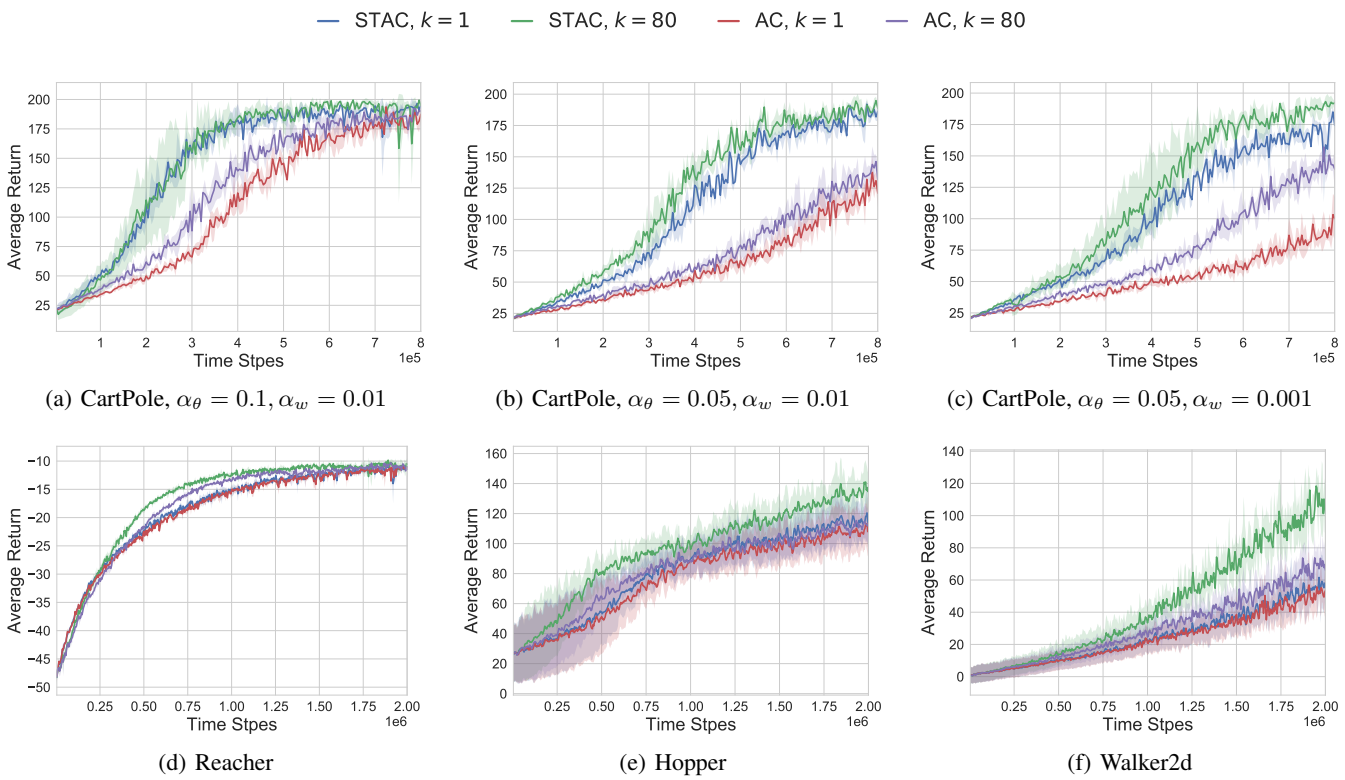
In our experiments, we compare the the performance of STAC with AC on different tasks as well as different settings of actor and critic learning rate α_θ, α_w and critic unrolling steps k. In CartPole, comparing Fig. 1(a) with 1(b), the actor learning rate α_θ has a significant influence on the convergence speed for both STAC and AC. However, the optimal actor learning rate is highly dependent on the environment and if the learning rate is too fast it results in a unstable policy. The critic learning rate has a smaller effect on the performance comparing Fig. 1(b) with 1(c). Among those three learning rate settings, 80 steps unrolling overall has better performance then only one steps critic update. This is due to the fact that better value function approximation provides the actor with more accurate policy gradient estimation. In fact, the small critic learning rate with 1 step update has the worst performance as shown by the red curve in 1(c).

The performance in Reacher, Hopper, and Walker2d environments as shown in Fig. 1(d), 1(e), and 1(f) share the same trend with CartPole that STAC has overall better performance than AC. In all those settings, the overall best performance is achieved by STAC with multiple critic unrolling steps in all the experiments.

Note that in CartPole environment, we set the regularization hyperparameter $\lambda = 0$, and in Reacher, Hopper, and Walker2d, $\lambda = 500$. The critic Hessian matrices are not invertible without setting such regularization in the later three environments. We believe this is due to the fact that the tasks are more complicated and initial actor and critic parameters are not in a neighborhood of a critical point. In fact, by setting such regularization, the implicit gradient term of the Stackelberg gradient is tempered and the performance difference between STAC and AC are not that significant comparing to that without regularization in CartPole.

5 Discussion and Future Work

In this paper we revisit the standard actor-critic algorithm from a game-theoretic perspective and formulate the problem as a Stackelberg game to capture the hierarchical interaction structure. This formulation is characterized by the actor seeking to solve a bilevel optimization problem in



(a) CartPole, $\alpha_\theta = 0.1, \alpha_w = 0.01$ (b) CartPole, $\alpha_\theta = 0.05, \alpha_w = 0.01$ (c) CartPole, $\alpha_\theta = 0.05, \alpha_w = 0.001$

(d) Reacher (e) Hopper (f) Walker2d

Figure 1: Comparison of STAC with normal actor-critic (AC) method on different tasks and learning rate settings. k represents the unrolling gradient update steps of the follower (critic).

which the objective is a function of the critic's parameters and the critic responds optimally with respect to its own parameters. To solve this problem with a gradient-based learning method, we extend the Stackelberg gradient update proposed by Fiez, Chasnov, and Ratliff (2020) to the reinforcement learning framework for the actor to follow, whereas the critic employs a standard gradient update. The novel Stackelberg actor-critic algorithm we propose outperforms the standard actor-critic algorithm in a number of environments as demonstrated in our experiments.

We believe the game-theoretic perspective of reinforcement learning presented in this paper can be extended to a broader range of algorithms and in future work we aim to provide a general Stackelberg learning meta-framework for any actor-critic based method. This will enable us to combine the Stackelberg gradient update with more advanced actor-critic methods such as DDPG (Lillicrap et al. 2015) and soft actor-critic (Haarnoja et al. 2018).

Another future direction we are pursuing is to switch the order of the leader and follower in general actor-critic based methods. When the actor is the leader such as in this paper, the learning procedure is in the form of a generalized policy iteration procedure (Sutton and Barto 2018). The critic is intended to perform policy evaluation and provide guidance for policy improvement. On the other hand, if the critic is the leader, the learning leans toward value-based methods, where the actor is intended to take actions that maximize the Q function in each iteration. We believe the comparison between the choice of roles for the actor and critic can provide insights into the trade-off between policy-based methods and value-based methods in reinforcement learning.

In future work, we plan to explore methods for decaying the amount of regularization in the Stackelberg gradient update. As shown in Section 4, regularization is necessary in some tasks where the critic problem is ill-conditioned and this often occurs when learning begins and the algorithm is far from the neighborhood of an equilibrium. However, the regularization erases the power of Stackelberg gradient update and as it grows the performance of the Stackelberg actor-critic algorithm is closer to that of the normal actor-critic algorithm. As the learning approaches the neighborhood of an equilibrium, we expect that the regularization can be decayed as the conditioning of the critic problem improves. Properly decaying the amount of regularization has the potential to significantly boost the performance of the Stackelberg gradient update. From a theoretical perspective, an interesting direction is to investigate the meaning of interpolating between equilibrium of the game with that of the regularized game. We believe a path to understand such connections is is the proximal equilibrium concept proposed by Farnia and Ozdaglar (2020), which interpolates between the set of Nash and Stackelberg equilibrium as a function of the regularization. We aim to design heuristics for decaying the regularization based on this theory.

A Appendix

A.1 Proof of Theorem 1

Proof. We derive $\frac{\partial L(\theta,w)}{\partial \theta}$ here. According to the definition of $L(\theta, w)$ in Eq. (14),

$$\frac{\partial L(\theta,w)}{\partial \theta} = \frac{\partial}{\partial \theta}\int_{s_0}\rho(s_0)\int_{a_0}\pi_\theta(a_0|s_0)\left(Q_w(s_0,a_0)-Q^\pi(s_0,a_0)\right)^2\mathrm{d}a_0\mathrm{d}s_0 \tag{27}$$

$$= \int_{s_0}\rho(s_0)\int_{a_0}\frac{\mathrm{d}\pi_\theta(a_0|s_0)}{\mathrm{d}\theta}\left(Q_w(s_0,a_0)-Q^\pi(s_0,a_0)\right)^2\mathrm{d}a_0\mathrm{d}s_0$$
$$+\int_{s_0}\rho(s_0)\int_{a_0}\pi_\theta(a_0|s_0)\frac{\mathrm{d}}{\mathrm{d}\theta}\left(Q_w(s_0,a_0)-Q^\pi(s_0,a_0)\right)^2\mathrm{d}a_0\mathrm{d}s_0 \tag{28}$$

$$= \int_{s_0}\rho(s_0)\int_{a_0}\pi_\theta(a_0|s_0)\nabla_\theta\log\pi_\theta(a_0|s_0)\left(Q_w(s_0,a_0)-Q^\pi(s_0,a_0)\right)^2\mathrm{d}a_0\mathrm{d}s_0$$
$$+2\int_{s_0}\rho(s_0)\int_{a_0}\pi_\theta(a_0|s_0)\left(Q^\pi(s_0,a_0)-Q_w(s_0,a_0)\right)\frac{\mathrm{d}Q^\pi(s_0,a_0)}{\mathrm{d}\theta}\mathrm{d}a_0\mathrm{d}s_0. \tag{29}$$

Now we compute $\frac{\mathrm{d}Q^\pi(s_0,a_0)}{\mathrm{d}\theta}$ in Eq. (29). Using Eq. (1) and (2), which define $Q^\pi(s,a)$ and $V^\pi(s)$, we have

$$Q^\pi(s,a) = r(s,a)+\gamma\int_{s'}P(s'|s,a)V^\pi(s')\mathrm{d}s', \tag{30}$$

$$V^\pi(s) = \int_a\pi_\theta(a|s)Q^\pi(s,a)\mathrm{d}a. \tag{31}$$

Hence, the gradient of the value function is

$$\frac{\mathrm{d}Q^\pi(s_0,a_0)}{\mathrm{d}\theta} = \gamma\int_{s_1}P(s_1|s_0,a_0)\frac{\mathrm{d}V^\pi(s_1)}{\mathrm{d}\theta}\mathrm{d}s_1 \tag{32}$$

$$= \gamma\int_{s_1}P(s_1|s_0,a_0)\int_{a_1}\left(\frac{\mathrm{d}\pi_\theta(a_1|s_1)}{\mathrm{d}\theta}Q^\pi(s_1,a_1)+\pi_\theta(a_1|s_1)\frac{\mathrm{d}Q^\pi(s_1,a_1)}{\mathrm{d}\theta}\right)\mathrm{d}a_1\mathrm{d}s_1 \tag{33}$$

$$= \gamma\int_{s_1}P(s_1|s_0,a_0)\int_{a_1}\pi_\theta(a_1|s_1)\nabla_\theta\log\pi_\theta(a_1|s_1)Q^\pi(s_1,a_1)\mathrm{d}a_1\mathrm{d}s_1$$
$$+\gamma^2\int_{s_1}P(s_1|s_0,a_0)\int_{a_1}\pi_\theta(a_1|s_1)\int_{s_2}P(s_2|s_1,a_1)\frac{\mathrm{d}V^\pi(s_2)}{\mathrm{d}\theta}\mathrm{d}s_2\mathrm{d}a_1\mathrm{d}s_1 \tag{34}$$

$$= \gamma\int_{s_1}P(s_1|s_0,a_0)\int_{a_1}\pi_\theta(a_1|s_1)\nabla_\theta\log\pi_\theta(a_1|s_1)Q^\pi(s_1,a_1)\mathrm{d}a_1\mathrm{d}s_1$$
$$+\gamma^2\int_{s_1}P(s_1|s_0,a_0)\int_{a_1}\pi_\theta(a_1|s_1)\int_{s_2}P(s_2|s_1,a_1)\int_{a_2}\pi_\theta(a_2|s_2)\nabla_\theta\log\pi_\theta(a_2|s_2)Q^\pi(s_2,a_2)\mathrm{d}a_2\mathrm{d}s_2\mathrm{d}a_1\mathrm{d}s_1$$
$$+\gamma^3\int_{s_1}P(s_1|s_0,a_0)\int_{a_1}\pi_\theta(a_1|s_1)\int_{s_2}P(s_2|s_1,a_1)\int_{a_2}\pi_\theta(a_2|s_2)\int_{s_3}P(s_3|s_2,a_2)\frac{\mathrm{d}V^\pi(s_3)}{\mathrm{d}\theta}\mathrm{d}s_3\mathrm{d}a_2\mathrm{d}s_2\mathrm{d}a_1\mathrm{d}s_1 \tag{35}$$

$$= \gamma\int_\tau p(\tau_{1:1}|\theta)\nabla_\theta\log\pi_\theta(a_1|s_1)Q^\pi(s_1,a_1)\mathrm{d}\tau_{1:1}$$
$$+\gamma^2\int_\tau p(\tau_{1:2}|\theta)\nabla_\theta\log\pi_\theta(a_2|s_2)Q^\pi(s_2,a_2)\mathrm{d}\tau_{1:2}$$
$$+\dots \tag{36}$$

$$= \int_\tau\sum_{t=1}^T\gamma^t p(\tau_{1:t}|\theta)\nabla_\theta\log\pi_\theta(a_t|s_t)Q^\pi(s_t,a_t)\mathrm{d}\tau. \tag{37}$$

The result in Eq. (37) is obtained by unrolling and marginalisation for the entire length of the trajectory. Substitute Eq. (37) into Eq. (29), we have

$$\begin{aligned}\frac{\partial L(\theta,w)}{\partial\theta} &= \int_{s_0}\rho(s_0)\int_{a_0}\pi_\theta(a_0|s_0)\nabla_\theta\log\pi_\theta(a_0|s_0)\left(Q_w(s_0,a_0)-Q^\pi(s_0,a_0)\right)^2\mathrm{d}a_0\mathrm{d}s_0 \\ &\quad + 2\int_{s_0}\rho(s_0)\int_{a_0}\pi_\theta(a_0|s_0)\left(Q^\pi(s_0,a_0)-Q_w(s_0,a_0)\right)\frac{\mathrm{d}Q^\pi(s_0,a_0)}{\mathrm{d}\theta}\mathrm{d}a_0\mathrm{d}s_0 \qquad (38)\\ &= \int_\tau p(\tau_0|\theta)\nabla_\theta\log\pi_\theta(a_0|s_0)\left(Q_w(s_0,a_0)-Q^\pi(s_0,a_0)\right)^2 \\ &\quad + 2\sum_{t=1}^{T}\gamma^t p(\tau_{0:t}|\theta)\nabla_\theta\log\pi_\theta(a_t|s_t)\left(Q^\pi(s_0,a_0)-Q_w(s_0,a_0)\right)Q^\pi(s_t,a_t)\mathrm{d}\tau. \qquad (39)\end{aligned}$$

□

A.2 Proof of Proposition 1

Proof. According to the critic objective definition $L(\theta,w)=\mathbf{E}_{s\sim\rho}\left[\left(V_w(s)-V^\pi(s)\right)^2\right]$,

$$\begin{aligned}\frac{\partial L(\theta,w)}{\partial\theta} &= \int_{s_0}\rho(s_0)\frac{\partial}{\partial\theta}(V_w(s_0)-V^\pi(s_0))^2\mathrm{d}s_0 \qquad (40)\\ &= 2\int_{s_0}\rho(s_0)(V^\pi(s_0)-V_w(s_0))\frac{\mathrm{d}V^\pi(s_0)}{\mathrm{d}\theta}\mathrm{d}s_0. \qquad (41)\end{aligned}$$

Now we compute $\frac{\mathrm{d}V^\pi(s_0)}{\mathrm{d}\theta}$ in Eq. (41). Use the result of Eq. (37), we have

$$\begin{aligned}\frac{\mathrm{d}V^\pi(s_0)}{\mathrm{d}\theta} &= \int_{a_0}\frac{\mathrm{d}\pi_\theta(a_0|s_0)}{\mathrm{d}\theta}Q^\pi(s_0,a_0)+\pi_\theta(a_0|s_0)\frac{\mathrm{d}Q^\pi(s_0,a_0)}{\mathrm{d}\theta}\mathrm{d}a_0 \qquad (42)\\ &= \int_\tau\pi_\theta(a_0|s_0)\left(\nabla_\theta\log\pi_\theta(a_0|s_0)Q^\pi(s_0,a_0)+\sum_{t=1}^{T}\gamma^t p(\tau_{1:t}|\theta)\nabla_\theta\log\pi_\theta(a_t|s_t)Q^\pi(s_t,a_t)\right)\mathrm{d}\tau. \qquad (43)\end{aligned}$$

Substitute Eq. (43) into Eq. (41), we have

$$\frac{\partial L(\theta,w)}{\partial\theta} = 2\int_\tau\sum_{t=0}^{T}\gamma^t p(\tau_{0:t}|\theta)\nabla_\theta\log\pi_\theta(a_t|s_t)\left(V^\pi(s_0)-V_w(s_0)\right)Q^\pi(s_t)\mathrm{d}\tau. \qquad (44)$$

□

References

Bhatnagar, S.; Sutton, R. S.; Ghavamzadeh, M.; and Lee, M. 2009. Natural actor–critic algorithms. *Automatica* 45(11): 2471–2482.

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. Openai gym. *arXiv preprint arXiv:1606.01540* .

Duan, Y.; Chen, X.; Houthooft, R.; Schulman, J.; and Abbeel, P. 2016. Benchmarking deep reinforcement learning for continuous control. In *International Conference on Machine Learning*, 1329–1338.

Farnia, F.; and Ozdaglar, A. 2020. Do GANs always have Nash equilibria? In *International Conference on Machine Learning*.

Fiez, T.; Chasnov, B.; and Ratliff, L. J. 2020. Implicit learning dynamics in stackelberg games: Equilibria characterization, convergence analysis, and empirical study. In *International Conference on Machine Learning*.

Fiez, T.; and Ratliff, L. 2020. Gradient descent-ascent provably converges to strict local minmax equilibria with a finite timescale separation. *arXiv preprint arXiv:2009.14820* .

Grondman, I.; Busoniu, L.; Lopes, G. A.; and Babuska, R. 2012. A survey of actor-critic reinforcement learning: Standard and natural policy gradients. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42(6): 1291–1307.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290* .

Jin, C.; Netrapalli, P.; and Jordan, M. I. 2020. What is local optimality in nonconvex-nonconcave minimax optimization? In *International Conference on Machine Learning*.

Kakade, S. M. 2001. A natural policy gradient. *Advances in neural information processing systems* 14: 1531–1538.

Konda, V. R.; and Tsitsiklis, J. N. 2000. Actor-critic algorithms. In *Advances in neural information processing systems*, 1008–1014.

Krantz, S. G.; and Parks, H. R. 2012. *The implicit function theorem: history, theory, and applications*. Springer Science & Business Media.

Levine, S.; Finn, C.; Darrell, T.; and Abbeel, P. 2016. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research* 17(1): 1334–1373.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* .

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, 1928–1937.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540): 529.

Peters, J.; and Schaal, S. 2008. Natural actor-critic. *Neurocomputing* 71(7-9): 1180–1190.

Prajapat, M.; Azizzadenesheli, K.; Liniger, A.; Yue, Y.; and Anandkumar, A. 2020. Competitive Policy Optimization. *arXiv preprint arXiv:2006.10611* .

Rajeswaran, A.; Mordatch, I.; and Kumar, V. 2020. A Game Theoretic Framework for Model Based Reinforcement Learning. *arXiv preprint arXiv:2004.07804* .

Sallab, A. E.; Abdou, M.; Perot, E.; and Yogamani, S. 2017. Deep reinforcement learning framework for autonomous driving. *Electronic Imaging* 2017(19): 70–76.

Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; and Moritz, P. 2015a. Trust region policy optimization. In *International conference on machine learning*, 1889–1897.

Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.; and Abbeel, P. 2015b. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438* .

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* .

Shen, Z.; Ribeiro, A.; Hassani, H.; Qian, H.; and Mi, C. 2019. Hessian aided policy gradient. In *International Conference on Machine Learning*, 5729–5738.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature* 529(7587): 484–489.

Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; and Riedmiller, M. 2014. Deterministic policy gradient algorithms. In *International Conference on Machine Learning*.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.

Sutton, R. S.; McAllester, D. A.; Singh, S. P.; and Mansour, Y. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, 1057–1063.

Tangkaratt, V.; Abdolmaleki, A.; and Sugiyama, M. 2017. Guide actor-critic for continuous control. *arXiv preprint arXiv:1705.07606* .

Todorov, E.; Erez, T.; and Tassa, Y. 2012. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5026–5033. IEEE.

Wang, Y.; Zhang, G.; and Ba, J. 2019. On solving minimax optimization locally: A follow-the-ridge approach. In *International Conference on Learning Representations*.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8(3-4): 229–256.

Zhang, K.; Yang, Z.; and Başar, T. 2019. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *arXiv preprint arXiv:1911.10635* .