Is the Policy Gradient a Gradient?

Chris Nota
College of Information and Computer Sciences
University of Massachusetts Amherst
cnota@cs.umass.edu

Philip S. Thomas
College of Information and Computer Sciences
University of Massachusetts Amherst
pthomas@cs.umass.edu

ABSTRACT
The policy gradient theorem describes the gradient of the expected discounted return with respect to an agent’s policy parameters. However, most policy gradient methods drop the discount factor from the state distribution and therefore do not optimize the discounted objective. What do they optimize instead? This has been an open question for several years, and this lack of theoretical clarity has lead to an abundance of misstatements in the literature. We answer this question by proving that the update direction approximated by most methods is not the gradient of any function. Further, we argue that algorithms that follow this direction are not guaranteed to converge to a “reasonable” fixed point by constructing a counterexample wherein the fixed point is globally pessimal with respect to both the discounted and undiscounted objectives. We motivate this work by surveying the literature and showing that there remains a widespread misunderstanding regarding discounted policy gradient methods, with errors present even in highly-cited papers published at top conferences.

1 INTRODUCTION
Reinforcement learning (RL) is a subfield of machine learning in which computational agents learn to maximize a numerical reward signal through interaction with their environment. Policy gradient methods encode an agent’s behavior as a parameterized stochastic policy and update the policy parameters according to an estimate of the gradient of the expected sum of rewards (the expected return) with respect to those parameters. In practice, estimating the effect of a particular action on rewards received far in the future can be difficult, so almost all state-of-the-art implementations instead consider an exponentially discounted sum of rewards (the discounted return), which shortens the effective horizon considered when selecting actions. The policy gradient theorem [25] describes the appropriate update direction for this discounted setting. However, almost all modern policy gradient algorithms deviate from the original theorem by dropping one of the two instances of the discount factor that appears in the theorem. It has been an open question for several years as to whether these algorithms are unbiased with respect to a different, related objective [26]. In this paper, we answer this question and prove that most policy gradient algorithms, including state-of-the-art algorithms, do not follow the gradient of any function. Further, we show that for some tasks, the fixed point of the update direction followed by these algorithms is pessimal, regardless of whether the discounted or undiscounted objective is considered.

The analysis in this paper applies to nearly all state-of-the-art policy gradient methods. In Section 6, we review all of the policy gradient algorithms included in the popular stable-baselines repository [9] and their associated papers, including A2C/A3C [13], ACER [28], ACKTR [30], DDPG [11], PPO [18], TD3 [6], TRPO [16], and SAC [8]. We motivate this choice in Section 6, but we note that all of these papers were published at top conferences and have received hundreds or thousands of citations. We found that all of the implementations of the algorithms used the “incorrect” policy gradient that we discuss in this paper. While this is a valid algorithmic choice if properly acknowledged, we found that only one of the eight papers acknowledged this choice, while three of the papers made erroneous claims regarding the discounted policy gradient and others made claims that were misleading. The purpose of identifying these errors is not to criticize the authors or the algorithms, but to draw attention to the fact that confusion regarding the behavior of policy gradient algorithm exists at the very core of the RL community and has gone largely unnoticed by reviewers. This has led to a proliferation of errors in the literature. We hope that by providing definitive answers to the questions associated with these errors we are able to improve the technical precision of the literature and contribute to the development of a better theoretical understanding of the behavior of reinforcement learning algorithms.

2 NOTATION
RL agents learn through interactions with an environment. An environment is expressed mathematically as a Markov decision process (MDP). An MDP is a tuple, \((S, \mathcal{A}, P, R, d_0, \gamma)\), where \(S\) is the set of possible states of the environment, \(\mathcal{A}\) is the set of actions available to the agent, \(P : S \times \mathcal{A} \times S \rightarrow [0, 1]\) is a transition function that determines the probability of transitioning between states given an action, \(R : S \times \mathcal{A} \rightarrow [-R_{\text{max}}, R_{\text{max}}]\) is the expected reward from taking an action in a particular state, bounded by some \(R_{\text{max}} \in \mathbb{R}\), \(d_0 : S \rightarrow [0, 1]\) is the initial state distribution, and \(\gamma \in [0, 1]\) is the discount factor which decreases the utility of rewards received in the future. In the episodic setting, interactions with the environment are broken into independent episodes. Each episode is further broken into individual timesteps. At each timestep, \(t\), the agent observes a state, \(S_t\), takes an action, \(A_t\), transitions to a new state, \(S_{t+1}\), and receives a reward, \(R_t\). Each episode begins with \(t = 0\) and ends when the agent enters a special state called the terminal absorbing state, \(s_\infty\). Once \(s_\infty\) is entered, the agent can never leave and receives a reward of 0 forever. We assume that

\[\text{ICML, NeurIPS, or ICLR, with the exception of PPO, which appears to have been published only on arXiv.}\]
The discounted objective has some convenient mathematical properties, differs from the true discounted policy gradient in that it drops the most conventional "policy gradient" algorithms instead directly or appropriately step-size schedule and smoothness assumptions, guarantees of stochastic gradient descent (namely, that given an appropriate step-size schedule and smoothness assumptions, convergence to a locally optimal policy is almost sure [4]). However, most conventional "policy gradient" algorithms instead directly or indirectly estimate the expression:

$$
\nabla J_f(\theta) = \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t \psi(S_t, A_t) Q^\theta(S_t, A_t) \right].
$$

Because $\nabla J_f$ is the true gradient of the discounted objective, algorithms that follow unbiased estimates of it are given the standard guarantees of stochastic gradient descent (namely, that given an appropriate step-size schedule and smoothness assumptions, convergence to a locally optimal policy is almost sure [4]). However, most conventional "policy gradient" algorithms instead directly or indirectly estimate the expression:

$$
\nabla J_f(\theta) = \mathbb{E} \left[ \sum_{t=0}^{\infty} \psi(S_t, A_t) Q^\theta(S_t, A_t) \right].
$$

Note that this expression includes the $\gamma^t$ contained in $Q^\theta$, but differs from the true discounted policy gradient in that it drops the outer $\gamma^t$. We label this expression $\nabla J_f(\theta)$ because the question of whether or not it is the gradient of some objective function, $J_f$, was left open by Thomas [26]. Thomas [26] was only able to construct $J_f$ in an impractically restricted setting where $\pi$ did not affect the state distribution. The goal of this paper is to provide answers to the following questions:

- Is $\nabla J_f(\theta)$ the gradient of some objective function?
- If not, does $\nabla J_f(\theta)$ at least converge to a reasonable policy?

### 4 $\nabla J_f(\theta)$ IS NOT A GRADIENT

In this section, we answer the first of our two questions and show that the update direction used by almost all policy gradient algorithms, $\nabla J_f(\theta)$, is not the gradient of any function using a proof by contraposition with the Clairaut-Schwarz theorem on mixed partial derivatives [19]. First, we present this theorem (Theorem 4.1) and its contrapositive (Corollary 4.2). Next, we present Lemma 4.3, which allows us to rewrite $\nabla J_f(\theta)$ in a new form. Finally, in Theorem 4.4 we apply Corollary 4.2 and Lemma 4.3 and derive a counterexample proving that $J_f$ does not, in general, exist, and therefore that the "policy gradient" given by $\nabla J_f(\theta)$ is not, in fact, a gradient.

**Theorem 4.1. (Clairaut-Schwarz theorem):** If $f : \mathbb{R}^n \rightarrow \mathbb{R}$ exists and is continuously twice differentiable in some neighborhood of the point $(a_1, a_2, \ldots, a_n)$, then its second derivative is symmetric:

$$
\forall i, j : \frac{\partial^2 f(a_1, a_2, \ldots, a_n)}{\partial x_i \partial x_j} = \frac{\partial^2 f(a_1, a_2, \ldots, a_n)}{\partial x_j \partial x_i}.
$$

**Proof.** The first complete proof was given by Schwarz [19]. English proofs can be found in many advanced calculus and analysis textbooks [15, p. 236].

**Corollary 4.2. (Contrapositive of Clairaut-Schwarz):** If at some point $(a_1, a_2, \ldots, a_n) \in \mathbb{R}^n$ there exists an $i$ and $j$ such that

$$
\frac{\partial^2 f(a_1, a_2, \ldots, a_n)}{\partial x_i \partial x_j} \neq \frac{\partial^2 f(a_1, a_2, \ldots, a_n)}{\partial x_j \partial x_i},
$$

then $f$ does not exist or is not continuously twice differentiable in any neighborhood of $(a_1, a_2, \ldots, a_n)$.

**Proof.** Contrapositive of Theorem 4.1. As a reminder, the contrapositive of a statement, $P \implies Q$, is $\neg Q \implies \neg P$. The contrapositive is always implied by the original statement. Additionally, recall that for any function $g$, $\forall i : g(i) = \exists i : \neg g(i)$. 

If we can find an example where $\nabla^2 J_f(\theta)$ is continuous but asymmetric, that is, $\exists i, j : \frac{\partial^2 J_f(\theta)}{\partial x_i \partial x_j} \neq \frac{\partial^2 J_f(\theta)}{\partial x_j \partial x_i}$, then we may apply Corollary 4.2 and conclude that $J_f$ does not exist. To this end, we present a new lemma that allows us to rewrite $\nabla J_f(\theta)$ in a form that is more amenable to computing the second derivatives by hand. The result of this lemma is of some theoretical interest in itself, but further interpretation is left as future work. We do not leverage it here for any purpose except to aid in our proof of Theorem 4.4.

**Lemma 4.3.** Let $d^\theta_0$ be the unnormalized, weighted state distribution given by:

$$
d^\theta_0(s) := d_0(s) + (1 - \gamma) \sum_{t=1}^{\infty} \Pr(S_t = s \mid \theta).
$$
Then:
\[ \nabla f_j(\theta) = \sum_{s \in S} d_j^\theta(s) \frac{\partial}{\partial \theta} V_j^\theta(s). \]

**Proof.** See appendix. \(\Box\)

In this form, we begin to see the root of the issue: In the above expression, \(d_j^\theta(s)\) is not differentiated, meaning that \(\nabla f_j(\theta)\) does not consider the effect updates to \(\theta\) have on the state distribution. We will show that this is in fact the source of the asymmetry in \(\nabla^2 f_j(\theta)\). With this in mind, we present our main theorem.

**Theorem 4.4.** Let \(M\) be the set of all MDPs with rewards bounded by \([-R_{\text{max}}, R_{\text{max}}]\) satisfying \(\forall \pi : \sum_{t=0}^{\infty} \Pr(S_t \neq s_\infty) < \infty\). Then, for all \(y < 1\):

\[ -\exists f_j : \forall M \in M : \nabla f_j(\theta) = \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t \pi(S_t, A_t) Q^\theta_j(S_t, A_t) \right]. \]

**Proof.** Theorem 4.1 states that if \(f_j\) is a twice continuously differentiable function, then its second derivative is symmetric. It is easy to show that this is not true informally using Lemma 4.3:

\[ \frac{\partial^2 f_j(\theta)}{\partial \theta_i \partial \theta_j} = \frac{\partial}{\partial \theta_i} \left( \sum_{s \in S} d_j^\theta(s) \frac{\partial}{\partial \theta_j} V_j^\theta(s) \right) \]
\[ = \sum_{s \in S} d_j^\theta(s) \frac{\partial^2}{\partial \theta_i \partial \theta_j} V_j^\theta(s) + \sum_{s \in S} \frac{\partial}{\partial \theta_i} d_j^\theta(s) \frac{\partial}{\partial \theta_j} V_j^\theta(s). \]

Therefore, we have by Corollary 4.2 that so long as \(\nabla^2 f_j(\theta)\) is continuous, \(f_j\) does not exist. In order to rigorously complete the proof, we must provide as a counterexample an MDP for which the above asymmetry is present. We provide such a counterexample in Figure 1. While we defer the full proof to the appendix, we describe the intuition behind the counterexample below.

Assume that for the example given in Figure 1, \(f_j\) exists and \(\nabla f_j(\theta)\) is its gradient. Consider in particular the case where \(y = 0\). In this case, \(\nabla^2 f_j(\theta)\) is asymmetric because \(\theta_1\) affects the value function and the state distribution, whereas \(\theta_2\) affects the value function but not the state distribution. Therefore, the second term in \(\partial^2 f_j(\theta)/\partial \theta_i \partial \theta_j\) labeled "asymmetric" in the above equation) is non-zero when \(i = 1\) and \(j = 2\), and zero when \(i = 2\) and \(j = 1\). In the appendix, we show that the above expression is symmetric if and only if \(y = 1\). Therefore, for \(y < 1\), Corollary 4.2 applies and we may conclude that \(f_j\) either does not exist or is not continuously twice differentiable. In this example, the \(\nabla^2 f_j(\theta)\) is continuous everywhere. Therefore, we conclude that \(f_j\) does not exist. This completes the counterexample. \(\Box\)

**5 THE FIXED POINT OF \(\nabla f_j(\theta)\) IS SOMETIMES PESSIMAL**

Having established that \(\nabla f_j(\theta)\) is not the gradient of any function for choices of \(y < 1\), we move on to the question of whether or not \(\nabla f_j(\theta)\) converges to some reasonable policy in the general case. For instance, consider the case of temporal difference (TD) methods. While the expected update of TD is not a gradient update [22], in the on-policy setting with a linear function approximator, TD has been shown to converge to a unique point close to the global optimum of the mean-squared projected Bellman error (MSPBE) [23, 24, 27], which is called the "TD fixed point." Through the geometric interpretation of the MSPBE, it may be said that the TD fixed point is "reasonable" in that it is close to the best possible estimate of the mean squared Bellman error (MSBE) under a particular linear parameterization of the value function.

We ask the question of whether or not a similar reasonable fixed point exists in the case of the update given by \(\nabla f_j(\theta)\). While it is not clear in what sense the fixed point should be "reasonable," we propose a very minimal criterion: any reasonable policy fixed point should at least surpass the worst possible (pessimal) policy under either the discounted or undiscounted objective. That is, if the fixed point is pessimal under both objectives, this suggests that it will be difficult to come up with a satisfactory justification. Surprisingly, it can be shown that \(\nabla f_j(\theta)\) fails to pass even this low bar.

To demonstrate this, we contrast two examples, given in Figures 2 and 3. In the former example, \(\nabla f_j(\theta)\) behaves in a way that is (perhaps) expected: it converges to the optimal policy under the discounted objective, while failing to optimize the undiscounted objective. This is a well understood trade-off of discounting that can be explained as "short-sightedness" by the agent. The latter example, however, shows a case where an agent following \(\nabla f_j(\theta)\) behaves in a manner that is apparently irrational: it achieves the smallest possible discounted return and undiscounted return, despite the fact that it is possible to maximize either within the given policy parameterization. We therefore suggest that for at least some MDPs, \(\nabla f_j(\theta)\) may not be a reasonable choice.

**6 LITERATURE REVIEW**

We previously claimed that \(\nabla f(\theta)\) is the direction followed by state-of-the-art policy gradient methods and that the lack of theoretical clarity on the behavior of algorithms following \(\nabla f(\theta)\) has resulted in a multiplicity of errors in the literature. In this section, we substantiate this point by surveying a subset of popular policy gradient...
algorithms and their associated papers. We show that the majority of them include erroneous or misleading statements directly relating to arguments made in this paper. We emphasize that we do not pose this as a criticism of the authors, but rather a symptom of the ambiguity in the literature that we hope to address with this paper.

6.1 Methodology

Rather than manually selecting which papers to review, which may have introduced an unacceptable degree of bias, we chose to review the papers associated with every policy gradient algorithm included in stable-baselines [9], a fork of the popular OpenAI baselines [5] library. We chose this particular subset of algorithms because inclusion in the library generally indicates that the algorithms have achieved a certain level of popularity and relevance. The papers corresponding to these algorithms have received hundreds or thousands of citations each, and, with the exception of PPO [18], were published at top machine learning conferences. It therefore seems reasonable to claim that the papers were impactful in the field and are representative of high-quality research. While we acknowledge that this sampling of algorithms is heavily biased towards the subfield of “deep” RL, we argue that this is not unreasonable given the immense popularity of this area and its impact on the broader field.

For each algorithm, we examined the pseudocode for the algorithm itself, the background and theoretical sections of the associated paper, and several publicly available implementations, including the implementation created by the authors where available. We tried to answer the following three questions for each algorithm:

(1) Does the algorithm use \( \nabla J(\theta) \) rather than an unbiased estimator?

(2) If so, did the authors note that \( \nabla J(\theta) \) is not an unbiased estimator of the policy gradient?

(3) Did the paper include any erroneous or misleading claims about \( \nabla J(\theta) \)?

For questions (2) and (3), we support our evaluation of the papers with quotations from the text in cases where there were errors or ambiguities. While this approach is verbose, we felt that paraphrasing the original papers would not allow the readers understand the errors in their appropriate context. For 5 out of 8 of the algorithms, the original authors or organizations provided code, allowing us to directly answer (1). For all eight papers, we examined the stable-baselines [9] implementation as well as several other implementations including tf-agents [20], garage [7], spinning-up [1], and the autonomous-learning-library [14]. Finally, for each paper we note the conference, year, and citation count estimated by Google Scholar on February 23, 2020.

6.2 Results

Eight policy gradient algorithms are included in stable-baselines. Our high-level results to each question are as follows:

(1) All eight of the algorithms use \( \nabla J(\theta) \) instead of an unbiased estimator, both in their pseudocode and implementations.

(2) Only one out of the eight papers calls attention to the fact that \( \nabla J(\theta) \) is a biased estimator.

\[\begin{align*}
\text{Figure 2: An example where the fixed point of } \nabla J(\theta) \text{ is optimal with respect to the discounted objective but pessimal with respect to the undiscounted objective. The agent starts in state } s_1, \text{ and can achieve a reward of } +1 \text{ by transitioning from } s_1 \text{ to } s_2, \text{ and a reward of } +2 \text{ by transitioning from } s_6 \text{ to } s_7, \text{ regardless of the action taken. In order to maximize the undiscounted return, the agent should choose } a_1. \text{ However, the discounted return is higher from choosing } a_2. \text{ Only the return in } s_1 \text{ will affect the policy gradient as the advantage is zero in every other state. Thus, algorithms following } \nabla J(\theta) \text{ methods will eventually choose } a_1. \text{ If the researcher is concerned with the undiscounted return, the agent should choose } a_2. \text{ Choosing a larger value of } \gamma \text{ trivially fixes the problem in this particular example, but for any value of } \gamma < 1, \text{ similar problems will arise given a sufficiently long horizon, and thus the problem is not truly eliminated. Nevertheless, this is well-understood as the trade-off of discounting.}
\end{align*}\]
Three out of the eight papers included claims that are clearly erroneous. Two additional papers made misleading claims in that they presented the discounted policy gradient ($\nabla J_\pi(\theta)$), and then proposed an algorithm that uses $\nabla J_\pi(\theta)$, without making note of this discrepancy. A sixth paper included claims that we argue are misleading, but not strictly false.

We reproduce the three explicitly erroneous claims below in order to give the reader a sense of common misunderstandings. Notice that the quotations sometimes use slightly different notation than this paper. We try to supply the appropriate context such that the meaning intended by the authors can be understood. Additional analysis can be found in the appendix.

**A3C (ICML 2016, 2859 citations)**

Asynchronous Advantage Actor-Critic (A3C) [13] is an actor-critic method that generates sample batches by running multiple actors in parallel. The original version of the algorithm achieved state-of-the-art performance on many Atari games when it was published. The background section of the main paper includes the text:

The return $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$ is the total accumulated return from time step $t$ with discount factor $\gamma \in (0, 1]$. The goal of the agent is to maximize the expected return from each state $s_t$.

[...] In contrast to value-based methods, policy based methods directly parameterize the policy $\pi(a|s; \theta)$ and update the parameters $\theta$ by performing, typically approximate, gradient ascent on $\mathbb{E}[R_t]$. One example of such a method is the REINFORCE family of algorithms due to Williams [29]. Standard REINFORCE updates the policy parameters $\theta$ in the direction $\nabla_\theta \log \pi(a_t|s_t; \theta) R_t$, which is an unbiased estimate of $\nabla_\theta \mathbb{E}[R_t]$. It is possible to reduce the variance of this estimate while keeping unbiased by subtracting a learned function of the state $b_t(s_t)$, known as a baseline [29], from the return. The resulting gradient is $\nabla_\theta \log \pi(a_t|s_t; \theta) (R_t - b_t(s_t))$.

It is falsely claimed that $\nabla_\theta \pi(a_t|s_t; \theta) R_t$ (which is an unbiased sample estimate of $\nabla J_\pi(\theta)$) is an unbiased estimate of $\nabla_\theta \mathbb{E}[R_t]$, where $R_t$ is the discounted return at timestep $t$ rather than the individual reward. We showed that this is not the case.

**ACKTR (ICLR 2017, 235 citations)**

Actor Critic using Kronecker-Factored Trust Region (ACKTR) [30] is a variation of A3C that attempts to efficiently estimate the natural policy gradient, which uses an alternate notion of the direction of steepest ascent. The resulting algorithm is considerably more sample efficient than A3C. The background section contains:

The goal of the agent is to maximize the expected $\gamma$-discounted cumulative return $J(\theta) = \mathbb{E}_\pi [R_t] = \mathbb{E}_\pi [\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}]$ with respect to the policy parameters $\theta$. Policy gradient methods directly parameterize a policy $\pi_\theta(a|s)$ and update parameter $\theta$ so as to maximize the objective $J(\theta)$. In its general form [17], the policy gradient is defined as:

$$\nabla_\theta J(\theta) = \mathbb{E}_\pi \left[ \sum_{t=0}^{\infty} \Psi^t \nabla_\theta \log \pi_\theta(a_t|s_t) \right],$$

where $\Psi^t$ is often chosen to be the advantage function $A^\pi(s_t, a_t)$, which provides a relative measure of value of each action $a_t$ at a given state $s_t$.

The authors assert that the gradient, $\nabla_\theta J(\theta)$, is equal to the given expression, which is false for the proposed choice of $A^\pi(s_t, a_t)$. They did not note that above expression is an approximation or otherwise clarify. Therefore, the definition of the policy gradient presented above is erroneous.

**SAC (ICML 2018, 446 citations)**

Soft Actor-Critic (SAC) [8] is an algorithm in the deterministic policy gradient family [21]. The deterministic policy gradient gives an expression for updating the parameters of a deterministic policy and is derived from the conventional policy gradient theorem, meaning that the arguments presented in this paper apply. SAC is considered state-of-the-art on continuous control tasks. The appendix states:

The exact definition of the discounted maximum entropy objective is complicated by the fact that, when using a discount factor for policy gradient methods, we typically do not discount the state distribution, only the rewards. In that sense, discounted policy gradients typically do not optimize the true discounted objective. Instead, they optimize average reward, with the discount serving to reduce variance, as discussed by Thomas [26].

The relationship between the average reward objective and $\nabla J_\pi(\theta)$ was discussed by Kakade [10] and elaborated on by Thomas [26]. However, the claim that $\nabla J_\pi(\theta)$ optimizes the average reward objective is erroneous by Theorem 4.4.

**6.3 Discussion**

The three erroneous claims all involved misinterpreting $\nabla J_\pi(\theta)$ as the gradient of some function. Two of the remaining papers failed to acknowledge the difference between the gradient of the discounted objective, $\nabla J_\pi(\theta)$, and the gradient direction followed by the presented algorithm, typically an estimate of $\nabla J_\pi(\theta)$. Even among the papers where we did not find explicit errors, errors were avoided largely through the use of hedged language and ambiguity, rather than technical precision. For examples of this, we encourage the reader to refer to the appendix. While for the purposes of this review we only sampled a small subset of the literature on policy gradients, we found the results sufficient to support our claim that there exists a widespread misunderstanding regarding $\nabla J_\pi(\theta)$.

**7 Conclusions**

We conclude by emphasizing the while $\nabla J_\pi(\theta)$ is not a gradient (Section 4), can in some cases result in pessimal behavior (Section 5), and is commonly misrepresented in the literature (Section 6), it has remained the most popular estimator of the policy gradient due to its effectiveness when applied to practical problems. The precise reason for this effectiveness, especially in the episodic setting, remains an open question.
We then must prove that this holds for some choice of $w$. We begin by hypothesizing that

$$\Pr(S_t = s|\theta) = \frac{\prod_{k=0}^{\infty} \Pr(S_{t+k} = x|S_t = s, \theta) \prod_{a \in A} Q^\theta(x, a) \frac{\partial^n \theta(x, a)}{\partial \theta}}{\sum_{s \in S} \prod_{k=0}^{\infty} \Pr(S_{t+k} = x|S_t = s, \theta) \prod_{a \in A} Q^\theta(x, a) \frac{\partial^n \theta(x, a)}{\partial \theta}}.$$

Substituting this into our expression for $\nabla f_i(\theta)$ gives us:

$$\sum_{s \in S} \frac{d_j^\theta(s)}{\partial \theta} V^\theta(s) = \sum_{s \in S} \sum_{t=0}^{\infty} \sum_{k=0}^{\infty} \sum_{a \in A} w(t) \Pr(S_t = s|\theta) \prod_{k=0}^{\infty} \Pr(S_{t+k} = x|S_t = s, \theta) \prod_{a \in A} Q^\theta(x, a) \frac{\partial^n \theta(x, a)}{\partial \theta} \frac{\partial \theta}{\partial \theta}.$$

In order to derive $\nabla f_i(\theta)$, we simply need to choose a $w(t)$ such that $\forall i: \sum_{t=0}^{\infty} w(t)^{i-t} = 1$. This is satisfied by the choice: $w(t) = 1$ if $t = 0$, and $1 - y$ otherwise. This trivially holds for $i = 0$, as $w(0)^{i-0} = (1)(1) = 1$. For $i > 0$:

$$\sum_{t=0}^{i} w(t)^{i-t} = w(0)^{i} + \sum_{t=1}^{i} w(t)^{i-t} = y^i + \sum_{t=1}^{i} (1 - y)^{i-t} = y^i + \sum_{t=1}^{i} (y^{i-t} - y^{i-t+1})$$

telescoping series

$$y^i - y^{i+1} = y^i + 1 - y^i = 1.$$

Thus, for this choice of $w(t)$:

$$\sum_{s \in S} \frac{d_j^\theta(s)}{\partial \theta} V^\theta(s) = \mathbb{E} \left[ \sum_{i=0}^{\infty} Q^\theta(S_i, A_i) \psi(S_i, A_i) \right] = \nabla f_j(\theta).$$

Finally, we see that this choice of $w(t)$ also gives us the expression for $d_j^\theta$ stated in Lemma 4.3:
Continuation of Proof of Theorem 4.2

We continue from the example given in Figure 1. First, we compute $d_T^θ$ in terms of $θ$ for each state using the definition of the MDP and π:

$$d_T^θ(s) = \sum_{t=0}^{∞} w(t) Pr(S_t = s|θ)$$

$$= w(0) Pr(S_0 = s|θ) + \sum_{t=1}^{∞} w(t) Pr(S_t = s|θ)$$

$$= d_0(s) (1 - γ) \sum_{t=1}^{∞} Pr(S_t = s|θ).$$

Next, we compute $V_T^θ$ in each state in terms of $θ$. Note that $Q_T^θ(s_1, a_1) = γ V_T^θ(s_2)$ because taking $a_1$ in $s_1$ leads to $s_2$ and has zero reward.

$$V_T^θ(s_2) = \pi(s_2, a_1) Q_T^θ(s_2, a_1) + \pi(s_2, a_2) Q_T^θ(s_2, a_2)$$

$$= \pi(s_2, a_1)(1) + \pi(s_2, a_2)(0)$$

$$= \sigma(θ_2)$$

$$V_T^θ(s_1) = \pi(s_1, a_1) Q_T^θ(s_1, a_1) + \pi(s_1, a_1) Q_T^θ(s_1, a_1)$$

$$= \pi(s_1, a_1)(γ V_T^θ(s_2)) + \pi(s_1, a_1)(0)$$

$$= γ \sigma(θ_1) \sigma(θ_2)$$

Recall that by the definition of our policy and substitution, we have:

$$\frac{∂π^θ(s_1, a_1)}{∂θ_1} = \frac{∂σ(θ_1)}{∂θ_1}$$

$$\frac{∂π^θ(s_2, a_1)}{∂θ_2} = \frac{∂σ(θ_1)}{∂θ_2}$$

Next, we compute each partial derivative, $∂V_T^θ(s)/∂θ_1$:

$$\frac{∂V_T^θ(s_1)}{∂θ_1} = γ \sigma(θ_1) \frac{∂σ(θ_1)}{∂θ_1}$$

$$\frac{∂V_T^θ(s_2)}{∂θ_2} = γ \sigma(θ_1) \frac{∂σ(θ_2)}{∂θ_2}$$

With the necessary components in place, we can apply Lemma 4.3 to compute each partial derivative of $J_f$:

$$\frac{∂J_f(θ)}{∂θ_1} = \frac{∂V_T^θ(s_1)}{∂θ_1} + d_T^θ(s_2) \frac{∂V_T^θ(s_2)}{∂θ_2}$$

$$= γ \sigma(θ_1) \frac{∂σ(θ_1)}{∂θ_1} + d_T^θ(s_2) \frac{∂σ(θ_2)}{∂θ_2}$$

Finally, we compute the second order partial derivatives:

$$\frac{∂}{∂θ_1} \left( \frac{∂J_f(θ)}{∂θ_1} \right) = \frac{∂}{∂θ_1} \left( γ \sigma(θ_1) \frac{∂σ(θ_1)}{∂θ_1} \right)$$

$$= γ \sigma(θ_1) \frac{∂σ(θ_1)}{∂θ_1}$$

$$\frac{∂}{∂θ_2} \left( \frac{∂J_f(θ)}{∂θ_2} \right) = \frac{∂}{∂θ_2} \left( γ \sigma(θ_1) \frac{∂σ(θ_2)}{∂θ_2} \right)$$

$$= γ \sigma(θ_1) \frac{∂σ(θ_2)}{∂θ_2}$$

Thus we see that the following holds for any $θ_1$ and $θ_2$:

$$\forall γ < 1 : \frac{∂}{∂θ_2} \left( \frac{∂J_f(θ)}{∂θ_1} \right) \neq \frac{∂}{∂θ_1} \left( \frac{∂J_f(θ)}{∂θ_2} \right).$$

The consequence of this asymmetry is that the contrapositive of the Clairaut-Schwarz theorem [19] implies that if $J_f$ exists, it must not be continuously twice differentiable. However, consider the remaining terms in the second order partial derivative:
\[ \frac{\partial}{\partial \theta_1} \left( \frac{\partial J_1(\theta)}{\partial \theta_1} \right) = \frac{\partial}{\partial \theta_1} \left( \gamma \sigma(\theta_2) \frac{\partial \sigma(\theta_1)}{\partial \theta_1} \right) = \gamma \sigma(\theta_2) \frac{\partial^2 \sigma(\theta_1)}{\partial \theta_1^2} \]

\[ \frac{\partial}{\partial \theta_2} \left( \frac{\partial J_1(\theta)}{\partial \theta_2} \right) = \frac{\partial}{\partial \theta_2} \left( \sigma(\theta_1) \frac{\partial \sigma(\theta_1)}{\partial \theta_2} \right) = \sigma(\theta_1) \frac{\partial^2 \sigma(\theta_1)}{\partial \theta_2^2}. \]

Thus, we have constructed all of the second order partial derivatives. As the sigmoid function, \( \sigma \), is itself continuously twice differentiable, we see that \( \forall \theta \in \mathbb{R}^2, i, j : \partial^2 J_i(\theta)/\partial \theta_i \partial \theta_j \) is continuous. Therefore, if \( J_f \) exists, it is continuously twice differentiable. However, we showed using the Clairaut-Schwarz theorem [19] that \( J_f \) is not continuously twice differentiable. Therefore, we have derived a contradiction.

**Additional Literature Review**

Here we include our analysis of the papers associated with algorithms from stable-baselines which did not include clearly erroneous claims regarding \( \nabla J_f(\theta) \). However, several of the papers included claims that we argue are nevertheless misleading, more so than others. Due to space limitations, we are forced to omit our analysis of DDPG and TRPO.

**ACER (ICLR 2017) (299 Citations)**

ACER [30] combines experience replay with actor-critic methods, improving sample efficiency. The background section contains the following text:

The parameters \( \theta \) of the differentiable policy \( \pi_\theta(a_t|x_t) \) can be updated using the discounted approximation to the policy gradient [25], which borrowing notation from Schulman et al. [17], is defined as:

\[ g = \mathbb{E}_{x_t,\infty, a_t, \infty} \left[ \sum_{i=0}^{\infty} A^\pi(x_t, a_t) \nabla_\theta \log \pi_\theta(a_t|x_t) \right]. \]

Following Proposition 1 of Schulman et al. [17], we can replace \( A^\pi(x_t, a_t) \) in the above expression with the state-action value \( Q^\pi(x_t, a_t) \), the discounted return \( R_t \), or the temporal difference residual \( r_t + \gamma V^\pi(x_{t+1}) - V^\pi(x_t) \), without introducing bias.

We note that \( g \) is exactly \( \nabla J_f(\theta) \). The authors claim that any of the given choices may be used "without introducing bias." We would argue that a naive reader is likely to assume that this means that \( g \) is unbiased with respect to some objective. The authors were actually making the subtler point that the given choices do not introduce bias relative to the choice of \( A^\pi \), which is itself biased. This claim is not erroneous, but at the same time, information is left out that is important for a clear understanding. The correctness hinges on ambiguity rather than precision, and a reader is likely to come away with the opposite impression. For this reason, we argue that this section is still misleading.

**PPO (arXiv 2017, 1769 citations)**

Proximal Policy Optimization (PPO) [18] is arguably the most popular deep actor-critic algorithm at this time due to its speed and sample efficiency. The paper contains the text:

Policy gradient methods work by computing an estimator of the policy gradient and plugging it into a stochastic gradient ascent algorithm. The most commonly used gradient estimator has the form

\[ \hat{g} = \mathbb{E}[\nabla_\theta \log \pi_\theta(a_t|x_t) A_t] \]

where \( \pi_\theta \) is a stochastic policy and \( A_t \) is an estimator of the advantage function at timestep \( t \). Here the expectation \( \mathbb{E}[\ldots] \) indicates the empirical average over a finite batch of samples, in an algorithm that alternates between sampling and optimization.

In this case, the authors considered a very specific setup where an algorithm "alternates between sampling and optimization." They construct an objective that operates on a given batch of data and is used only for a single optimization step. They do not relate this objective to a global objective. The final algorithm does follow a direction resembling \( \hat{g} \), with a number of optimization tricks. For this reason, we did not consider the claims above made to be misleading. However, we note again that the issues with \( \nabla J_f(\theta) \) were sidestepped rather than addressed directly.

**TD3 (ICML 2018, 247 citations)**

Twin Delayed Deep Deterministic policy gradient (TD3) Fujimoto et al. [6] is another paper in the DDPG family. The paper was published concurrently with SAC and contains similar enhancements. It contains the text:

The return is defined as the discounted sum of rewards

\[ R_t = \sum_{i=1}^{T} y^{i-t} r(s_i, a_i), \]

where \( y \) is a discount factor determining the priority of short-term rewards. In reinforcement learning, the objective is to find the optimal policy \( \pi_\phi \), with parameters \( \phi \), which maximizes the expected return \( J(\pi) = \mathbb{E}_{s_0, a_0, \pi} [R_0] \). For continuous control, parameterized policies \( \pi_\phi \) can be updated by taking the gradient of the expected return \( \nabla_\phi J(\pi) \). In actor-critic methods, the policy, known as the actor, can be updated through the deterministic policy gradient algorithm:

\[ \nabla_\phi J(\pi) = \mathbb{E}_{s_0, \pi} [\nabla_a Q^\pi(s, a)]_{a = \pi(s)} \nabla_\phi \pi(s). \]

\[ Q^\pi(s, a) = \mathbb{E}_{s_{1:t}, a_{1:t} \sim \pi, [R_t | s, a]}, \]

the expected return when performing action \( a \) in state \( s \) and following \( \pi \) after, is known as the critic or the value function.

The authors did not define \( p_\pi \), leaving the above expression ambiguous. In the original DQN paper, it was defined as the discounted state distribution. It is misused here in the definitions of \( J \) and \( Q \), in that it is not clear what role the discounted state distribution plays in the definition of \( Q \) in either case. The algorithm eventually computes the sample gradient by averaging over samples drawn from a replay buffer: \( \nabla J_f(\phi) = N^{-1} \sum \nabla_\phi Q_\phi(s, a)_{a = \pi(s)} \nabla_\phi \pi(s), \) the deterministic policy gradient form of \( \nabla J_f(\theta). \)
REFERENCES