A Parametric Class of Approximate Gradient Updates for Policy Optimization

Ramki Gummadi [1] **Saurabh Kumar** [1] [2] **Junfeng Wen** [3] **Dale Schuurmans** [1] [4]

Abstract

Approaches to policy optimization have been motivated from diverse principles, based on how the parametric model is interpreted (e.g. value versus policy representation) or how the learning objective is formulated, yet they share a common goal of maximizing expected return. To better capture the commonalities and identify key differences between policy optimization methods, we develop a unified perspective that re-expresses the underlying updates in terms of a limited choice of gradient form and scaling function. In particular, we identify a parameterized space of approximate gradient updates for policy optimization that is highly structured, yet covers both classical and recent examples, including PPO. As a result, we obtain novel yet well motivated updates that generalize existing algorithms in a way that can deliver benefits both in terms of convergence speed and final result quality. An experimental investigation demonstrates that the additional degrees of freedom provided in the parameterized family of updates can be leveraged to obtain non-trivial improvements both in synthetic domains and on popular deep RL benchmarks.

1. Introduction

Policy optimization in reinforcement learning considers the problem of learning a parameterized policy that maximizes some notion of expected return, either via direct interactions with the environment (on-policy) or via learning from a fixed dataset (off-policy). A direct approach to policy optimization is based on the policy gradient (PG) theorem (Sutton et al., 1999), which shows how a parameterized policy can be locally improved from unbiased Monte Carlo estimates of the gradient of the expected return. Alternately,

value-based methods attempt to solve a proxy task of fitting a parameterized action-value function, which explicitly models returns in the environment.

Even though the learning objectives appear different, they share a common goal of constructing an agent that maximizes returns in an unknown environment. Moreover, they share a low level algorithmic primitive of incrementally updating model parameters based on learning signals derived from random experience, which suggests a deeper underlying connection. In this paper, we consider learning a single parametric model while being agnostic about whether it is a policy or a value model. A key contribution in Section 3 is to derive tight constraints on gradient estimates that correspond to standard action value or policy gradient updates on a parametric model. These results highlight the shared sample dependent components of the updates while also emphasizing key differences, which we use to organize a *form-axis* of update variations. This leads us to propose a novel form of policy gradient (PG) in Theorem 3.1, *Policy Gradient With Policy Baseline*, that leverages a specific form of state-action baseline (Gu et al., 2017; Wu et al., 2018; Tucker et al., 2018), typically used to provide a control variate for additional variance reduction beyond state only baselines (Weaver & Tao, 2001; Greensmith et al., 2001).

In Section 4, an alternate axis of variation among updates is identified, the *scale-axis*, which covers gradients for robust alternatives to the MSE loss, including Huber loss. Notably, the only difference between updates along this axis lies in how return estimates are rescaled in the gradient update. In Section 4.2, we show the Maximum-Likelihood gradient can also be approximated as a scale variation.

Prior work has shown that accounting for distribution shift from the behavior policy is valuable (Espeholt et al., 2018; Schulman et al., 2017) in the case of off-policy learning. In Section 4.3, we consider off-policy corrections as another instance of *scale*-axis variation. A key insight is the central role of a simple scaling function whose two scalar inputs are sample-dependent learning signals: (1) the return error estimate and (2) the log probability ratio of importance weight. In Section 4.4, we prove that the gradient update arising from the surrogate objective defined for Proximal Policy Optimization (PPO) (Schulman et al., 2017) also corresponds to a particular non-trivial instantiation of this

---

[1] Google Research, Brain Team [2] Stanford University [3] Layer 6 AI [4] University of Alberta. Correspondence to: Ramki Gummadi <gsrk@google.com>.

scaling function.

Section 5 then summarizes the range of updates we identify for both discrete and continuous action spaces, and lists some shared properties of the scaling function that underlie viable update rules. In Section 5.1, we extrapolate these shared properties as constraints to guide the design of a simple class of scaling functions that includes both a numerically stable approximation to the maximum likelihood scaling function as well as the default scaling function used in policy gradient and Q-learning.

Section 6 conducts an experimental evaluation of novel update rules in the expanded family, in settings ranging from a synthetic 2D bandit benchmark (Section 6.1), a tabular environment (Section 6.2) and the MuJoCo continuous control benchmark versus the PPO baseline (Section 6.3). These results demonstrate that the form-axis and scale-axis simplify the identification of new updates that are able to improve policy optimization algorithms in a variety of situations.

2. Background

Consider the standard infinite horizon discounted Markov Decision Process model with states indexed by $s \in \mathcal{S}$, actions by $a \in \mathcal{A}$ and a reward $r(s, a)$. A policy is a conditional distribution over the action space $\mathcal{A}$ for a given state, $\pi(.|s)$, satisfying the normalization constraint $\sum_a \pi(a|s) = 1$ for all $s \in \mathcal{S}$. Given policy π, let $Q^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{t=0}^\infty \gamma^t r_t | s_0 = s, a_0 = a\right]$ denote the expected return starting from (s, a) and then following the policy π. The goal of policy optimization, loosely speaking, is to find a policy π that achieves high expected returns. In the general case of a parametric policy, however, different policies can be optimal for different starting states. To resolve this ambiguity, a fixed but arbitrary initial state distribution $\mu(s)$ over $\mathcal{S}$ is typically used to define a concrete objective as the expected return, $J_\mu(\pi) = \mathbb{E}_{s_0 \sim \mu} \left[\sum_{t=0}^\infty \gamma^t r_t\right]$. An alternate way to handle $J_\mu(\pi)$ is by defining a key distribution $d_\mu^\pi(s)$, the discounted state visitation distribution when sampling the initial state from μ: $d_\mu^\pi(s) = (1 - \gamma) \sum_{t=0}^\infty \gamma^t \mathbb{E}_{s_0 \sim \mu} \left[\Pr^\pi(s_t = s | s_0)\right]$, where $\Pr^\pi$ represents the (unknown) transition kernel on the MDP corresponding to a given policy π. With this definition, it can be shown that $J_\mu(\pi) = \mathbb{E}_{(s,a) \sim D_\mu^\pi} [r(s, a)]$ (Puterman, 2014), where

$$D_\mu^\pi(s, a) = d_\mu^\pi(s) \pi(a|s). \qquad (1)$$

When the policy is parameterized by θ as π_θ, the classical policy gradient theorem (Sutton et al., 1999) shows that $\nabla_\theta J_\mu(\pi_\theta)$ can also be expressed as an expectation of gradient samples over D_μ^π: $\nabla_\theta J_\mu(\theta) = \mathbb{E}_{(s,a) \sim D_\mu^\pi}[Q^\pi(s, a) \nabla_\theta \log \pi_\theta(a|s)]$. In practice, Q^π is generally not available to compute the gradients so **policy-based methods** usually replace it with an estimate, $\hat{T}(s, a)$,

based on, e.g., discounted Monte Carlo returns. Such an estimator, however, can perform poorly in practice. To address this, a *baseline* is typically subtracted from the target estimate $\hat{T}(s, a)$, i.e., introducing a control variate without bias:

$$\hat{G}_{PG}(s, a, \theta) = (\hat{T}(s, a) - b(s)) \nabla_\theta \log \pi_\theta(a|s) \qquad (2)$$

On the other hand, **value-based methods** like Q-learning rely on temporal differences and perform updates that do not explicitly follow the gradient of an objective function. Instead, they are motivated as *semi*-gradients, defined by updates in a pseudo-objective that ignores certain terms in the gradient. These are typically considered off-policy methods and do not explicitly specify the sampling distribution for (s, a) when estimating an update direction. For instance, given a parametric model $q_\theta(s, a)$, the Q-learning update, denoted by $\hat{G}_{QL}$ can be written as:

$$\hat{G}_{QL}(s, a, \theta) = (\hat{T}(s, a) - q_\theta(s, a)) \nabla_\theta q_\theta(s, a) \qquad (3)$$

where $\hat{T}(s, a) = r(s, a) + \gamma \max_u q_\theta(s', u)$ and s' is a random sample for the transition kernel from (s, a). TD-learning with SARSA (Sutton & Barto, 2018) on a policy π uses a slightly different target: $\hat{T}(s, a) = r(s, a) + \gamma q_\theta(s', u)$, where $u \sim \pi(.|s')$, which is called the 1-step bootstrap. This estimate can have a bias when the value function is an approximation. Therefore, generalizations of the target definition based on n-step returns can also be considered, which smoothly interpolate between the unbiased Monte Carlo rollouts and the bootstrap estimates. However, such variations on the target are orthogonal to our analysis, which focuses on the structure of the gradient updates once an estimation procedure for $\hat{T}(s, a)$ is fixed.

3. Policy-based versus Value-based Updates

Consider a learnable parametric function approximator, $q_\theta(s, a) : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ for which the parameters θ are updated using gradient estimates constructed from stochastic samples of experience data. For value-based methods (e.g. DQN or TD-learning), $q_\theta(s, a)$ explicitly models some notion of an infinite horizon discounted return value starting from state s and taking action a. For a policy-based method, $q_\theta(s, a)$ could instead represent the energy function that encodes the action conditional distribution for a stochastic policy π_θ. In the case of discrete actions, $q_\theta(s, a)$ can be interpreted as the action logits or preferences (Sutton & Barto, 2018). Taking this perspective, we begin our analysis by studying the gradients for a stochastic policy $\pi_\theta(a|s)$, that denotes a properly normalized conditional likelihood for taking action a at state s. For discrete actions, which we assume through this section, this is equivalent to the softmax transformation that links $\pi_\theta(a|s)$ with $\{q_\theta(s, u)\}_{u \in \mathcal{A}}$:

$$\pi_\theta(a|s) \triangleq \exp\left(q_\theta(s, a)\right) / \sum_u \exp\left(q_\theta(s, u)\right) \qquad (4)$$

3.1. Policy Gradient with Policy Baseline (PGPB)

In (2), $\hat{G}_{PG}$ can often be impractical for non-trivial problems due to its high variance, which motivates baselines, which are generally parameterized functions that can depend on both the state and the action. The parameters that define baselines are typically independent of the policy parameters, however. We now consider a particular form of state-action baseline that uses the policy logits as the baseline under the softmax parameterization of (4). The primary motivation for introducing this is to clarify the relationship between policy-based and value-based gradients. We begin with a result that characterizes this form of the policy gradient estimator, whose relation to value-based gradients would be evident later in Section 3.2 by contrasting the terms in their respective gradients. All proofs are provided in the Appendix.

Theorem 3.1. *[Policy Gradient with Policy Baseline] Given $q_\theta(s,a) : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$, let $\pi_\theta(a|s)$ be defined by (4). Consider the sample based gradient estimator for θ:*

$$\hat{G}_{PGPB}(s,a,\theta) \triangleq \tag{5}$$
$$\left(\hat{T}(s,a) - q_\theta(s,a)\right)\nabla_\theta \log \pi_\theta(a|s) - \nabla_\theta H(\pi_\theta(.|s))$$

where $H(\pi_\theta(.|s)) = \mathbb{E}_{a \sim \pi_\theta(.|s)}\left[-\log \pi_\theta(a|s)\right]$ denotes the entropy of $\pi_\theta(.|s)$. Then, $\hat{G}_{PGPB}$ is an unbiased estimate of the policy gradient if $\hat{T}$ is an unbiased estimate of return:

$$\nabla_\theta J_\mu^\pi(\theta) = \mathop{\mathbb{E}}_{(s,a) \sim D_\mu^\pi}\left[\hat{G}_{PGPB}(s,a,\theta)\right] \tag{6}$$

Remark 3.2 (**Entropy Term**). Note that unlike MaxEnt-RL (Haarnoja et al., 2017), the objective $J_\mu^\pi(\theta)$ does not include entropy regularization in its formulation, yet an entropy term intriguingly shows up in (5).

3.2. Connecting Policy-based and Value-based Updates

Consider two gradient estimators that can be viewed as fitting an action-value function by minimizing an objective w.r.t. a target estimator $\hat{T}(s,a)$, where the first objective is the common Mean Squared Error (MSE), while the second is the less common Mean Variance Error (MVE) (Flet-Berliac et al., 2021).

Theorem 3.3. *Consider the value-based gradients following the MSE and MVE objectives respectively defined as:*

$$G_{MSE}(\theta) = -\nabla_\theta \frac{1}{2} \mathop{\mathbb{E}}_{(s,a)}\left[(\hat{T}(s,a) - q_\theta(s,a))^2\right] \tag{7}$$

$$G_{MVE}(\theta) = -\nabla_\theta \frac{1}{2} \mathop{\mathbb{E}}_{s} \mathop{\mathbb{V}}_{a}\left[\hat{T}(s,a) - q_\theta(s,a)\right], \tag{8}$$

where, $\mathbb{V}_a$ denotes the variance of the distribution over a, conditioned on the state s. Then, $\hat{G}_{MSE}$ and $\hat{G}_{MVE}$ below

are unbiased gradient[1] estimators for G_{MSE} and G_{MVE}.

$$\hat{G}_{MSE}(s,a,\theta) \triangleq \left(\hat{T}(s,a) - q_\theta(s,a)\right)\nabla_\theta q_\theta(s,a) \tag{9}$$

$$\hat{G}_{MVE}(s,a,\theta) \triangleq \left(\hat{T}(s,a) - q_\theta(s,a)\right)\nabla_\theta \log \pi_\theta(a|s) \tag{10}$$

Note that value-based methods typically do not specify the replay buffer sampling distribution, in contrast to PG, which comes with a specific on-policy sampling distribution, D_μ^π defined in (1). However, this distinction is not relevant when considering the updates conditioned on a given (s,a) unless off-policy corrections are considered. We are now ready to relate the policy gradient $\hat{G}_{PGPB}$ with $\hat{G}_{MSE}$ and $\hat{G}_{MVE}$.

Corollary 3.4. *Given any fixed target return estimator, $\hat{T}(s,a)$, the following relations hold between the sample-based gradient estimators at every sample (s,a) and θ.*

$$\hat{G}_{MVE} - \hat{G}_{PGPB} = \nabla_\theta H(\pi_\theta(.|s)) \tag{11}$$

$$\hat{G}_{MSE} - \hat{G}_{MVE} \propto \sum_u \pi_\theta(u|s)\nabla_\theta q_\theta(s,u) \tag{12}$$

$$\hat{G}_{MSE} - \hat{G}_{PGPB} - \nabla_\theta H(\pi_\theta(.|s)) \tag{13}$$
$$\propto \sum_u \pi_\theta(u|s)\nabla_\theta q_\theta(s,u)$$

Proof. Equation (11) follows directly from combining (5) with (10). Equation (12) follows from relating $\nabla_\theta \log \pi_\theta(a|s)$ with $\nabla_\theta q_\theta(s,a)$ using Equation (4). Finally, Equation (13) follows from the previous two observations. $\square$

The significance of Corollary 3.4 is that Equations (11), (12), (13) express the relationship between the three gradient forms in terms of quantities that have *no dependence on the action a or the return estimate $\hat{T}(s,a)$*. The main signal indicating quality of the chosen action a, quantified by $\hat{T}(s,a)$, is aggregated identically for all three updates via:

$$\Delta_R \triangleq \hat{T}(s,a) - q_\theta(s,a) \tag{14}$$

To summarize the three forms of updates above more explicitly in terms of the underlying parametric model $q_\theta(s,a)$, we now state a helper theorem:

Theorem 3.5. *Let $\hat{q}_\theta$ indicate a stop-gradient operator on q_θ with respect to θ. Given the softmax parameterization of π_θ (Equation (4)), we have:*

$$\nabla_\theta H(\pi_\theta(.|s) + \nabla_\theta \mathop{\mathbb{E}}_{u|s \sim \pi_\theta}[\hat{q}_\theta(s,u)] = 0 \tag{15}$$

---

[1]These are technically semi-gradients because any dependence of $\hat{D}_\mu^\pi$, the assumed on-policy sampling distribution for (s,a) and $\hat{T}(s,a)$ on θ are ignored, similar to most TD learning methods.

Theorem 3.5, Equation (4) and Equation (14), together allow for a summary of the three updates directly in terms of $q_\theta(s, a)$ as:

$$\hat{G}_{PGPB} = \Delta_R \left(\nabla_\theta q_\theta(s, a) - \mathop{\mathbb{E}}_{u|s\sim\pi} \left[\nabla_\theta q_\theta(s, u) \right] \right)$$
$$+ \nabla_\theta \mathop{\mathbb{E}}_{u|s\sim\pi_\theta} \left[\hat{q}_\theta(s, u) \right] \quad (16)$$

$$\hat{G}_{MVE} = \Delta_R \left(\nabla_\theta q_\theta(s, a) - \mathop{\mathbb{E}}_{u|s\sim\pi} \left[\nabla_\theta q_\theta(s, u) \right] \right) \quad (17)$$

$$\hat{G}_{MSE} = \Delta_R \left(\nabla_\theta q_\theta(s, a) \right) \quad (18)$$

We refer to the gradients listed in Equations (16), (17), (18) as update variations along the *form-axis* since the key changes required to span these updates have to do with what are effectively the *bias* vectors, $\mathbb{E}_{u|s\sim\pi} \left[\nabla_\theta q_\theta(s, u) \right]$ and $\nabla_\theta \mathbb{E}_{u|s\sim\pi_\theta} \left[\hat{q}_\theta(s, u) \right]$, that can be computed exclusively from the model q_θ without any query to the data sample, and in particular, Δ_R.

4. Update Variations: The Scale-Axis

In the previous section, we discussed update variations along the *form-axis*. In this section, an orthogonal variation along the *scale-axis* is proposed, which considers how the gradient terms are scaled based on the return estimate.

4.1. Error Loss Function Updates

Consider value fitting objectives arising from aggregating individual state action prediction errors via an *error loss function*, l which is assumed to be smooth and convex, $l : \mathbb{R} \mapsto \mathbb{R}_+$ such that $l(0) = 0$. Under these assumptions, the derivative of l is given by a non-decreasing function $\lambda : \mathbb{R} \mapsto \mathbb{R}$ with $\lambda(0) = 0$. Given any such loss criterion l on the prediction error Δ_R, let $G_{E(l)}(\theta)$ denote a generalization of $G_{MSE}(\theta)$ in Equation (7) that generalizes to the error loss $l(.)$ from the MSE:

$$G_{E(l)}(\theta) = -\nabla_\theta \mathop{\mathbb{E}}_{(s,a)} \left[l \left(\hat{T}(s, a) - q_\theta(s, a) \right) \right] \quad (19)$$

The sample gradient estimator, $\hat{G}_\lambda$ for the scaling function λ (or equivalently, its loss l) is:

$$\hat{G}_\lambda(s, a, \theta) = -\nabla_\theta \, l \left(\hat{T}(s, a) - q_\theta(s, a) \right) \quad (20)$$
$$= \lambda(\Delta_R) \nabla_\theta q_\theta(s, a) \quad (21)$$

Unlike the *form-axis* variations which involve the previously discussed sample independent *bias* vectors, we refer to this type of variation as the *scale-axis*. The MSE update is equivalent to $l(x) = \frac{1}{2}x^2$ and $\lambda(x) = x$. An alternative, the Huber loss (Huber, 1964), has an $l(x)$ parameterized by a hyperparameter $\delta > 0$ as:

$$l(x) = \begin{cases} \frac{1}{2}x^2, & \text{if } |x| < \delta \\ \delta(|x| - \frac{1}{2}\delta), & \text{otherwise} \end{cases}$$

In this case, $\lambda(x) = \text{clip}(x, -\delta, \delta)$ is a clipped version of the identity scaling function that arises from the MSE loss. Clipping the TD-errors is a common heuristic in deep RL (Mnih et al., 2015), and corresponds to a simple scale variation as shown above.

4.2. A Maximum-Likelihood Update

Next, we derive a novel scale-axis update that is inspired by a maximum likelihood objective. Given a target estimator, $\hat{T}$, consider an action target distribution $\hat{p}_T(a|s) \triangleq \exp \left(\hat{T}(s, a) - F(\hat{T})(s) \right)$, where $F(T)(s) \triangleq \log \sum_a \exp T(s, a)$. The gradient of the log-likelihood for action conditionals given this target distribution is:

$$G_{ML}(\theta) \triangleq \mathop{\mathbb{E}}_s \left[\mathop{\mathbb{E}}_{a|s\sim\hat{p}_T} \nabla_\theta \log \pi_\theta(a|s) \right]$$

$$= \mathop{\mathbb{E}}_s \left[\mathop{\mathbb{E}}_{a|s\sim\hat{p}_T} \left[\nabla_\theta q_\theta(s, a) \right] - \sum_u \pi_\theta(u|s) \nabla_\theta q_\theta(s, u) \right]$$

$$= \mathop{\mathbb{E}}_s \left[\mathop{\mathbb{E}}_{a|s\sim\pi} \left[\frac{\hat{p}_T(a|s)}{\pi_\theta(a|s)} \nabla_\theta q_\theta(s, a) \right] - \mathop{\mathbb{E}}_{u|s\sim\pi} \left[\nabla_\theta q_\theta(s, u) \right] \right]$$

$$= \mathop{\mathbb{E}}_{(s,a)\sim\pi} \left[\left(\frac{\hat{p}_T(a|s)}{\pi_\theta(a|s)} - 1 \right) \nabla_\theta q_\theta(s, a) \right] \quad (22)$$

Note that the log-likelihood ratio between $\hat{p}_T$ and π_θ is:

$$\log \hat{p}_T(a|s) - \log \pi_\theta(a|s)$$
$$= \hat{T}(s, a) - F(\hat{T})(s) - q_\theta(s, a) + F(q_\theta)(s)$$
$$= \Delta_R + \xi(s), \text{ where } \xi(s) = F(q_\theta)(s) - F(\hat{T})(s),$$

which gives an unbiased sample estimator for (22) in the form $(e^{\Delta_R + \xi(s)} - 1) \nabla_\theta q_\theta(s, a)$. However, computing $F(\hat{T})(s)$ explicitly requires defining $\hat{T}(s, a)$ for all possible actions and not just the chosen action. To avoid this issue, we make an approximation that $\xi(s) \approxeq 0$, which may also be justified as an implicit constraint on the estimated target values. This motivates an alternate update:

$$\hat{G}_{ML}(s, a, \theta) \triangleq \left(e^{\Delta_R} - 1 \right) \nabla_\theta q_\theta(s, a) \quad (23)$$

This update can be recovered by $\lambda(x) = e^x - 1$, which also satisfies $\lambda(0) = 0$ and is increasing.

In Oh et al. (2018), a surrogate objective derived from modifying the MSE loss for Q-learning by clipping the prediction error below at 0 is proposed, which is referred to as self imitation learning (SIL). We can write the gradient expression corresponding to this objective as

$$\hat{G}_{SIL}(s, a, \theta) = \max(\Delta_R, 0) \nabla_\theta q_\theta(s, a) \quad (24)$$

which reveals that it could be considered a piece-wise linear approximation to the Maximum Likelihood gradient $\hat{G}_{ML}$ derived above since $\lambda(x) = e^x - 1 \approxeq \max(x, 0)$.

4.3. Off-Policy Corrected Updates

In Section 3, we made an implicit assumption that the sampled actions for each update are chosen from the policy encoded by the current parameters, which is an on-policy assumption that is typically not satisfied in practice. Accounting for a different behavior policy requires re-weighting the updates with off-policy importance ratio for each sample, which is an action dependent scalar signals, similar to Δ_R. Let π_b denote the behavior policy. Denote the log probability ratio at a given (s, a) sample as:

$$\Delta_O \triangleq \log \frac{\pi_\theta(a|s)}{\pi_b(a|s)} \tag{25}$$

For practical reasons, we only consider importance correction of the action conditional distribution at each state and implicitly assume that the state marginals match, an assumption which is also common in prior practical methods which make use of off-policy corrections (Chen et al., 2019a; Schulman et al., 2017). Note that the two scalar quantities Δ_O and Δ_R, defined in Equations (25) and (14) summarize all the action dependent *learning signals* needed for gradient estimation on any given (s, a) sample. Accounting for importance weight correction, the updates in Equations (16), (17), (18), (23) can be generalized to:

$$\hat{G}_{PGPB} = e^{\Delta_O} \Delta_R \Big(\nabla_\theta q_\theta(s, a) - \mathbb{E}_{u|s \sim \pi} [\nabla_\theta q_\theta(s, u)] \Big)$$
$$+ \nabla_\theta \mathbb{E}_{u|s \sim \pi_\theta} [\hat{q}_\theta(s, u)] \tag{26}$$

$$\hat{G}_{MVE} = e^{\Delta_O} \Delta_R \Big(\nabla_\theta q_\theta(s, a) - \mathbb{E}_{u|s \sim \pi} [\nabla_\theta q_\theta(s, u)] \Big) \tag{27}$$

$$\hat{G}_\lambda = e^{\Delta_O} \lambda(\Delta_R) \Big(\nabla_\theta q_\theta(s, a) \Big) \tag{28}$$

Remark 4.1. Equation (28) spans an entire class of updates by choosing $\lambda : \mathbb{R} \mapsto \mathbb{R}$ to be any particular non-decreasing function which vanishes at 0. Setting $\Delta_O = 0$ recovers the on-policy special case that ignores the importance weights.

Remark 4.2. For (26) the term $\nabla_\theta \mathbb{E}_{u|s \sim \pi_\theta} [\hat{q}_\theta(s, u)]$ does not include an off-policy correction because it can be computed explicitly without estimating the return for the chosen action sample. This is in contrast to the terms involving Δ_R, which are specific to the sampled action and need importance correction whenever the actions are chosen by a divergent behavior policy.

With off-policy corrections, the scale-axis variation can now include a joint dependence of the gradient estimates on Δ_O and Δ_R in a single expression, $f(\Delta_O, \Delta_R)$. Note that the form-axis variations considered in Section 3 can be systematically generalized by replacing Δ_R with $f(\Delta_O, \Delta_R) = e^{\Delta_O} \Delta_R$. So far, we have only demonstrated cases of

$f(\Delta_O, \Delta_R)$ that are multiplicatively factored into terms that depend only on Δ_O and Δ_R. However, this need not be a universal assumption, as shown next for PPO.

4.4. The Proximal Policy Optimization (PPO) update

PPO (Schulman et al., 2017) optimizes a surrogate objective derived from policy gradient to learn a policy parameterization directly without an explicit state-action value function. To relate to our framework, we need to reconstruct an explicit state-action preference q_θ from π_θ. While this can be done using the softmax assumption in the discrete action case, the more general continuous action setting is not straightforward, as π_θ may only support sampling the conditional actions and estimating the conditional probability for a given state-action pair (e.g. from a closed form Gaussian). Nevertheless, if we have an estimate for $V^\pi(s)$, the state value function, then O'Donoghue et al. (2017) suggests that we can consider an implied estimate of the state-action values for some constant α of the form $\hat{q}(s, a) \cong \alpha(\log \pi_\theta(a|s) + H(\pi(.|s)) + \hat{V}^\pi(s)$. Using the notation for the advantage function, $\hat{A} = \hat{T} - V^\pi(s)$, we define:

$$\Delta_R = \hat{T} - \hat{q} = \hat{A} - \alpha \big(\log \pi(a|s) + H(\pi(.|s)) \big) \tag{29}$$

Given any advantage estimator (e.g., GAE(λ) (Schulman et al., 2016)), this extends our generalized update rule starting from PGPB using the value notation to a policy parameterized update rule that also supports continuous actions as:

$$\hat{G}_{f,\pi} \triangleq f(\Delta_O, \Delta_R) \nabla_\theta \log \pi_\theta(a|s) + \beta \nabla_\theta H(\pi_\theta(.|s)) \tag{30}$$

In the above equation, β is an additional entropy hyperparameter similar to the final term in $\hat{G}_{PGPB}$ for the special case of discrete actions from Section 3.

Theorem 4.3. *The PPO surrogate objective gradient can be recovered as $\hat{G}_{f,\pi}$ in (30) for the gradient scaling function, $f(x, y) = e^x y \, \tau_\epsilon(x, y)$, where $\tau_\epsilon(x, y) = \mathbb{1}_{y>0} \mathbb{1}_{x<\log(1+\epsilon)} + \mathbb{1}_{y<0} \mathbb{1}_{x>\log(1-\epsilon)}$ while choosing $\alpha = 0$ in (29). Note that $\tau_\epsilon(x, y) \in \{0, 1\} \, \forall x, y, \epsilon$.*

5. A Parametrized Class of Updates

In Sections 3 and 4, two separate axes of variation for standard policy optimization updates were identified. These were referred to as the *form*-axis and the *scale*-axis to highlight the systematic structure shared between the updates.

For a given $f : \mathbb{R}^2 \mapsto \mathbb{R}$, the form variations include:

$$\mathcal{U}_Q(f) \triangleq f(\Delta_O, \Delta_R)\Big(\nabla_\theta q_\theta(s,a)\Big) \tag{31}$$

$$\mathcal{U}_V(f) \triangleq f(\Delta_O, \Delta_R)\Big(\nabla_\theta q_\theta(s,a) - \underset{u|s\sim\pi}{\mathbb{E}}\left[\nabla_\theta q_\theta(s,u)\right]\Big) \tag{32}$$

$$\mathcal{U}_P(f) \triangleq f(\Delta_O, \Delta_R)\Big(\nabla_\theta q_\theta(s,a) - \underset{u|s\sim\pi}{\mathbb{E}}\left[\nabla_\theta q_\theta(s,u)\right]\Big)$$
$$+ \nabla_\theta \underset{u|s\sim\pi_\theta}{\mathbb{E}}\left[\hat{q}_\theta(s,u)\right] \tag{33}$$

$$\mathcal{U}_\pi(f) \triangleq f(\Delta_O, \Delta_R)\nabla_\theta \log \pi_\theta(a|s) + \beta\nabla_\theta H(\pi_\theta(.|s)) \tag{34}$$

In the above equations, the first three rules apply specifically to discrete action settings where there is an explicit state-action parameterized model. The final update, $\mathcal{U}_\pi(f)$, in contrast, applies to a direct policy parameterization π_θ that is unavoidable in continuous action settings due to the need for efficient sampling and normalization of the action distribution.

Section 4 outlined a range of possibilities for the scale-axis variation of gradient updates. In Assumption 5.1 below, we consider natural constraints that restrict the possibilities while investigating the design of new approximate scaling functions $f : \mathbb{R}^2 \mapsto \mathbb{R}$ that may not directly correspond to any prior known principle.

Assumption 5.1. Let $f : \mathbb{R}^2 \mapsto \mathbb{R}$ denote a function that captures the dependence of a gradient update on Δ_O, Δ_R. Then, we consider f to be a valid scaling function for a gradient update rule only if:

1. $f(\Delta_O, \Delta_R)$ is non-decreasing in Δ_R, with $f(\Delta_O, 0) = 0$ for any fixed Δ_O. This implies that $f(\Delta_O, \Delta_R)$ has the same sign as Δ_R, i.e. $\Delta_R f(\Delta_O, \Delta_R) \geq 0$.

2. $|f(\Delta_O, \Delta_R)|$ is non-decreasing in Δ_O around a neighborhood of 0 for any fixed Δ_R.

Constraint 1 ensures that the update direction increases $q_\theta(s,a)$ when it underestimates $\hat{T}$ and decreases $q_\theta(s,a)$ when it overestimates $\hat{T}$. Constraint 2 ensures that samples that are more off-policy have a weaker update strength in comparison to on-policy samples. This is only assumed around some neighborhood of 0 to allow for the trust region constraints (Schulman et al., 2015) that inspire the clipping surrogate optimized by PPO. Both constraints are clearly satisfied by all of the specific updates derived in Section 4.

5.1. A Parametric Class of Scaling Functions

We now leverage the unified view including the constraints of Assumption 5.1 to identify a numerically stable approximation, f_{MLA} to the maximum likelihood scaling function,

which was derived as $f_{ML}(\Delta_O, \Delta_R) = e^{\Delta_O}(e^{\Delta_R} - 1)$, but suffers from an exponential dependence on both Δ_O and Δ_R. This new approximation, derived in the appendix, is:

$$f_{MLA}(x,y) = \tag{35}$$
$$\begin{cases} -\frac{1}{2}(1+x)^2, & \text{if } y \leq -(1+x) \leq 0 \\ y\max\left(1+x+\frac{y}{2}, 0\right) & \text{otherwise} \end{cases}$$

Remark 5.2 (Qualitative similarity of f_{ML} to f_{MLA}). When highly off-policy ($\Delta_O \ll 0$), $f_{MLA}(\Delta_O, \Delta_R) = 0$ for all values of Δ_R up to some positive threshold that depends on Δ_O. This threshold increases as the sample becomes more off-policy (i.e. $\Delta_O \to -\infty$). For large values of Δ_R, the update strength is quadratic in Δ_R. Conversely, it saturates to a negative value as $\Delta_R \to -\infty$.

We now introduce a class of scaling functions that is derived by further approximations to f_{MLA}. This class spans a range of behaviors, all satisfying the constraints of Assumption 5.1.

$$f_{MLA(\alpha_o, \alpha_r)}(x, y) = \tag{36}$$
$$y\max\left(1 + \alpha_o x + \alpha_r y, \frac{(1+\alpha_o x)_+}{2}\right)$$

In Equation (36), the constants α_o, α_r denote hyper-parameters[2] that can be used to span a class of update rules corresponding to a two-dimensional search space of scaling functions.

Observation 5.3. [Recovery of Specific Scale Functions] With particular values of α_o, α_r, $f_{MLA(\alpha_o, \alpha_r)}$ includes[3] the following important special cases of scaling functions that were explicitly considered during the unification.

1. $\alpha_o = 1, \alpha_r = 0.5$ approximately recovers f_{MLA}, and therefore f_{ML}.

2. $\alpha_o = 1, \alpha_r = 0$ approximately recovers the baseline scaling function for policy gradient with importance weight corrections.

3. $\alpha_o = 0, \alpha_r = 0$ recovers the baseline identity scaling function for action value methods without importance weight corrections (e.g. Q-learning).

Justification of Observation 5.3. For claim 1, a visualization for the projections of f_{ML} and $f_{ML(1,0.5)}$ along various slices of Δ_O and Δ_R is shown in Figures 4 and 5 in the Appendix.
To justify claim 2, note that $f_{MLA(1,0)}(x,y) = y\max(1 + x, \frac{(1+x)_+}{2}) = y\max(1+x, 0) \approx ye^x$.
Claim 3 follows trivially as $f_{MLA(0,0)}(x,y) = y$. $\qquad\square$

---
[2]$(x)_+$ denotes $\max(x, 0)$
[3]In a non-vanishing neighborhood of $(0,0)$

6. Empirical Evaluation

The goal of the experiments is to investigate the practical utility of the the parametric approximate gradient updates proposed. Through a diverse set of evaluations, we find that structure identified in the updates is narrow enough to demonstrate the possibility of improvements to the vanilla versions both in simple settings with few confounding effects as well as a more complex deep RL benchmark setting.

6.1. A Synthetic 2D Contextual Bandit

The experiments in this Section are designed to enable an exhaustive study of the dynamics of several update rules and their convergence to the optimum under the following constraints: (1) An easily visualizable 2D parameter space (2) Under-parameterization that makes value fitting non-trivial (3) A unique and tractable optimum parameter setting to analyze the gap from the optimum solution (4) Avoid confounders related to the target estimation procedures.

Setting: State space is two-dimensional corresponding to a degenerate single step MDP, denoted using $x = (x_0, x_1) \in \mathbb{R}^2$. Action space is discrete with $a \in \{0, \ldots, 7\}$ embedded onto a unit circle with embeddings $\Psi(a) \triangleq (\cos(2\pi a/8), \sin(2\pi a/8))$. Reward is set as $r(x, a) = \sigma(\langle x, \Psi(a) \rangle) \in (0, 1)$, where $\langle a, b \rangle$ denotes the dot product of a, b; and σ is the sigmoid. The dataset consists of (x, a, r), with x sampled from the 2D standard Gaussian and a sampled uniformly in each batch. The model considered[4] is $q_\theta(x, a) = \langle (\theta_0(1 + x_0) - 1, \theta_1(1 + x_1) - 1), \Psi(a) \rangle$. The true reward cannot be fit exactly under the given parameterization, but the optimal parameters for maximizing returns of the implied policy are $\theta^* = (1, 1)$. Figure 8 in the Appendix visualizes the policy return objective landscape across the two-dimensional search space traversed by the various gradient update rules.

Results: To compare the various update rules, trajectories of θ starting from an initialization of $(0, 0)$ are generated for each of the twelve gradient updates listed in Table 1, with more details about the setup provided in the Appendix. Figure 1 shows that the best performing update is the combination of the scaling function f_{MLA} with the PGPB style update. More detailed learning curves including the dynamics of the Euclidean distance to the optimum parameters are provided in the Appendix in Figures 6, 7. Within each form of the gradient update, the maximum likelihood scaling functions performed better than the other updates in terms of both the speed of convergence as well as the final objective value. By contrast, the vanilla squared error objective saturates at a sub-optimal solution though it gets there much

---

[4]The simpler alternative of $q_\theta(x, a) = \langle (\theta_0 x_0, \theta_1 x_1), \Psi(a) \rangle$ results in a policy parameterization that is scale invariant wrt θ, and hence without a proper solution.
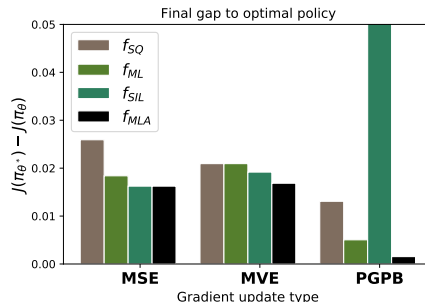

Figure 1: Comparison of update rules on a 2D synthetic bandit problem: Regret ($J(\pi_{\theta^*}) - J(\pi_\theta)$) for each of the 12 update rules considered (lower is better) at the end of 10k iterations. The updates are grouped by the three forms (MSE, MVE, PGPB), and across four scaling functions as listed in Table 1 in the Appendix.

faster compared to policy gradient, while the policy gradient version converges to a good final solution, but much slower. These experiments show evidence that the proposed class of updates demonstrate the potential to improve both speed and final solution compared to other natural baselines.

6.2. FourRoom Environment

To complement the other experiments, we show that the parametric scaling function can deliver benefits even while ablating the off-policy correction in the FourRoom domain (Precup, 2000).

Setting: The agent obtains a reward of 10 when they reach a pre-defined goal cell, and 0 otherwise in an environment with discount factor 0.9 from a random initial state. We consider a batch/offline setting in which a dataset is collected from a uniformly random behaviour policy. The dataset is sufficient to cover the whole state and action space as exploration is not the focus here. In each gradient update, a minibatch of 64 transitions is uniformly sampled from the dataset to compute the gradients. The policy is defined via the logits $q_\theta(s, a)$. Additional details are provided in the Appendix.

Algorithms and Scaling Functions: We investigate the performance of the $f_{MLA(0, \alpha_r)}$ scaled update derived from two baseline updates corresponding to policy gradient and Q-learning. For policy gradient, the target $\tilde{T}$ is set to be a critic learned together with the policy/actor as commonly done in the literature (Wen et al., 2021). For Q-learning, the target is bootstrapped as $r(s, a) + \gamma \max_u q_\theta(s', u)$.

Results and Discussions: The results are shown in Figure 2, with mean and standard deviation over 5 runs. The number following MLA indicates the value of α_r (α_o is set to zero). One can see that MLA consistently improves over

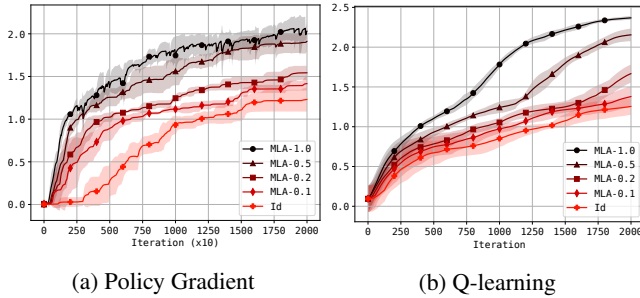(a) Policy Gradient   (b) Q-learning

Figure 2: FourRoom Environment: Return curves for $f_{MLA(0,\alpha)}$ across $\alpha \in \{0.1, 0.2, 0.5, 1.0\}$ against baseline for PG and Q-learning updates
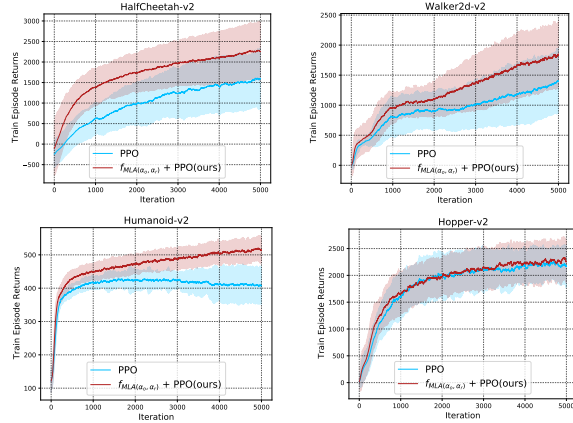

Figure 3: Results on MuJoCo Environments with mean and standard deviation across 100 seeds comparing $f_{MLA+PPO}$ with vanilla PPO. The x-axis corresponds to a total of 5 million environment steps.

the identity scaling for both policy gradient and Q-learning, which are ubiquitous in the RL literature. As discussed in Sec. 5.1, MLA corresponds to the identity scaling when $\alpha_o = \alpha_r = 0$. Figure 2 shows that the agent learns faster as α_r deviates from zero. Recall that $\alpha_r = 0$ recovers the baseline as discussed in Sec. 5.1. The consistent improvements of MLA for policy gradient and Q-learning with different targets indicate the significance of the scaling function for the gradient update.

6.3. Continuous Control Experiments

In this section, benchmarks from the MuJoCo suite are evaluated with Proximal Policy Optimization (PPO) as the baseline algorithm. Using Theorem 4.3, which shows that PPO can be equivalently implemented as a special case of the more general update rule, the experiments in this section are designed to validate the possibility that a systematic search over the structured scaling function space identified in Section 5 can deliver non-trivial gains. Specifically, we consider the continuous action update rule (30) for PPO and replace the policy gradient scale component, $e^{\Delta_O} \Delta_R$, with the parameterized variant, $f_{MLA(\alpha_o,\alpha_r)}(\Delta_O, \Delta_R)$ from Equation (36) that can span more behaviors. The complete expressions for the general parameterized update rule are provided in Appendix C.2. Under this setting, $(\alpha, \alpha_o, \alpha_r) = (0, 1, 0)$ recovers an update approximately equivalent to vanilla PPO. However, our experiments demonstrate that this is generally not the best choice as seen by the improvements in Figure 3. These results suggest that proposed update form with a flexible scaling parameterization can be a significant lever to improve state of the art algorithms. We used the single process agent baseline of PPO in the open source ACME framework (Hoffman et al., 2020) for the experiments.

7. Related Work

Unification of policy and value approaches: (Schulman et al., 2018; Nachum et al., 2017; O'Donoghue et al., 2017;

Ghosh et al., 2020) make several connections between policy and value based methods, and propose algorithms that leverage those relationships by interpreting the model as both a policy and a value parameterization. Traditional Actor-Critic methods (Peters & Schaal, 2008; Konda & Tsitsiklis, 2000) use value-based methods to learn a critic during policy optimization, maintaining two separate models in the process. Recent progress (Wen et al., 2021) has also led to an improved understanding of how such Actor-Critic algorithms are related to pure PG approaches.

Surrogate objectives and gradients: Several prior works have also considered surrogate objectives in policy optimization (Chen et al., 2019b; Oh et al., 2018; Liang et al., 2018; Schulman et al., 2017; Kumar et al., 2020; Serrano et al., 2021). Directly taking the gradient perspective instead of focusing on objectives has also been a focus in several prior works including O'Donoghue et al. (2017); Maei et al. (2009). Kumar et al. (2019) has also proposed and studied a general class of gradient updates for policy optimization, instances of which demonstrated empirical improvements over value-based methods. Vieillard et al. (2020b;a) demonstrate that applying certain form of momentum/averaging to the gradient scale can be beneficial. These are orthogonal to our results, which concern novel variations for the dependence of the scale factor on the sample prediction errors.

8. Conclusion

In this work, we identify structural similarities among the gradient updates for policy optimization algorithms motivated using various principles, ranging from classical policy gradients and value estimation procedures to modern exam-

ples like PPO. We show that policy based gradients can be directly contrasted with value based gradients when considering itself as a state-action baseline. In particular, they share the same dependence on the sample dependent learning signals that indicate the quality of a given action, but differ in terms of how they make use of certain other sample independent bias terms. We also identify a critical gradient scaling dependence function on the sample prediction errors and importance weights that can be used as a lever to span the updates corresponding to a wide variety of policy optimization procedures. Based on this insight, we design a simple parametric search space for valid scaling functions that can deliver reliable wins over strong baselines in a range of experimental studies from simple synthetic settings to deep RL benchmarks.

The proposed framework has broad applicability and relevance to various policy optimization techniques. Exploring connections to adaptive step size methods (Dabney, 2014) is an interesting direction for further research. Applying the proposed parametric updates in applied settings where off policy corrections play an important role in policy optimization (Chen et al., 2019a) is another useful direction for future work. Another interesting direction is to leverage the newly discovered structure in the approximate update space to guide the search space for Auto-RL (Parker-Holder et al., 2022) techniques.

9. Acknowledgements

References

Chen, M., Beutel, A., Covington, P., Jain, S., Belletti, F., and Chi, E. H. Top-k off-policy correction for a REINFORCE recommender system. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019*, pp. 456–464. ACM, 2019a.

Chen, M., Gummadi, R., Harris, C., and Schuurmans, D. Surrogate objectives for batch policy optimization in one-step decision making. In *Advances in Neural Information Processing Systems*, 2019b. URL.

Dabney, W. *Adaptive step-sizes for reinforcement learning*. PhD thesis, 2014.

Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Doron, Y., Firoiu, V., Harley, T., Dunning, I., Legg, S., and Kavukcuoglu, K. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. *ArXiv:1802.01561*, 2018. URL.

Flet-Berliac, Y., Ouhamma, R., Maillard, O.-A., and Preux, P. Learning value functions in deep policy gradients using residual variance. In *ICLR*, 2021.

Ghosh, D., Machado, M. C., and Roux, N. L. An operator view of policy gradient methods. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Greensmith, E., Bartlett, P. L., and Baxter, J. Variance reduction techniques for gradient estimates in reinforcement learning. In *Advances in Neural Information Processing Systems*, 2001. URL.

Gu, S., Lillicrap, T., Ghahramani, Z., Turner, R. E., and Levine, S. Q-prop: Sample-efficient policy gradient with an off-policy critic. In *ICLR*, 2017. URL.

Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning*, pp. 1352–1361. PMLR, 2017.

Hoffman, M., Shahriari, B., Aslanides, J., Barth-Maron, G., Behbahani, F., Norman, T., Abdolmaleki, A., Cassirer, A., Yang, F., Baumli, K., Henderson, S., Novikov, A., Colmenarejo, S. G., Cabi, S., Gülçehre, Ç., Paine, T. L., Cowie, A., Wang, Z., Piot, B., and de Freitas, N. Acme: A research framework for distributed reinforcement learning. *ArXiv:2006.00979*, 2020. URL.

Huber, P. J. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.

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

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. *ArXiv:2006.04779*, 2020. URL.

Kumar, S., Dadashi, R., Ahmed, Z., Schuurmans, D., and Bellemare, M. G. Generalized policy updates for policy optimization. In *NeurIPS 2019 Optimization Foundations for Reinforcement Learning Workshop*, 2019. URL.

Liang, C., Norouzi, M., Berant, J., Le, Q. V., and Lao, N. Memory augmented policy optimization for program synthesis and semantic parsing. In *Advances in Neural Information Processing Systems*, 2018. URL.

Maei, H. R., Szepesvari, C., Bhatnagar, S., Precup, D., Silver, D., and Sutton, R. S. Convergent temporal-difference learning with arbitrary smooth function approximation. In *NIPS*, pp. 1204–1212, 2009.

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. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533, 2015.

Nachum, O., Norouzi, M., Xu, K., and Schuurmans, D. Bridging the gap between value and policy based reinforcement learning. In *Advances in Neural Information Processing Systems*, 2017. URL.

O'Donoghue, B., Munos, R., Kavukcuoglu, K., and Mnih, V. Combining policy gradient and q-learning. In *ICLR*, 2017. URL.

Oh, J., Guo, Y., Singh, S., and Lee, H. Self-imitation learning. In *ICML*, 2018. URL.

Parker-Holder, J., Rajan, R., Song, X., Biedenkapp, A., Miao, Y., Eimer, T., Zhang, B., Nguyen, V., Calandra, R., Faust, A., Hutter, F., and Lindauer, M. Automated reinforcement learning (autorl): A survey and open problems, 2022.

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

Precup, D. *Temporal abstraction in reinforcement learning*. University of Massachusetts Amherst, 2000.

Puterman, M. L. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

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

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. High-dimensional continuous control using generalized advantage estimation. In *ICLR*, 2016.

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

Schulman, J., Chen, X., and Abbeel, P. Equivalence between policy gradients and soft q-learning. *ArXiv:1704.06440*, 2018. URL.

Serrano, J. B., Curi, S., Krause, A., and Neu, G. Logistic q-learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 3610–3618. PMLR, 2021.

Sutton, R., McAllester, D., Singh, S., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, 1999. URL.

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

Tucker, G., Bhupatiraju, S., Gu, S., Turner, R. E., Ghahramani, Z., and Levine, S. The mirage of action-dependent baselines in reinforcement learning. In *ICML*, 2018. URL.

Vieillard, N., Kozuno, T., Scherrer, B., Pietquin, O., Munos, R., and Geist, M. Leverage the average: an analysis of kl regularization in reinforcement learning. *Advances in Neural Information Processing Systems*, 33, 2020a.

Vieillard, N., Scherrer, B., Pietquin, O., and Geist, M. Momentum in reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 2529–2538. PMLR, 2020b.

Weaver, L. and Tao, N. The optimal reward baseline for gradient-based reinforcement learning. In *UAI*, 2001. URL.

Wen, J., Kumar, S., Gummadi, R., and Schuurmans, D. Characterizing the gap between actor-critic and policy gradient. In *ICML*, 2021.

Wu, C., Rajeswaran, A., Duan, Y., Kumar, V., Bayen, A. M., Kakade, S. M., Mordatch, I., and Abbeel, P. Variance reduction for policy gradient with action-dependent factorized baselines. In *ICLR*, 2018. URL.

Appendix

A. Proofs

Theorem 3.1. *[Policy Gradient with Policy Baseline] Given $q_\theta(s, a) : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$, let $\pi_\theta(a|s)$ be defined by (4). Consider the sample based gradient estimator for θ:*

$$\hat{G}_{PGPB}(s, a, \theta) \triangleq \tag{5}$$
$$\left(\hat{T}(s, a) - q_\theta(s, a)\right) \nabla_\theta \log \pi_\theta(a|s) - \nabla_\theta H(\pi_\theta(.|s))$$

where $H(\pi_\theta(.|s)) = \mathbb{E}_{a \sim \pi_\theta(.|s)} [-\log \pi_\theta(a|s)]$ denotes the entropy of $\pi_\theta(.|s)$. Then, $\hat{G}_{PGPB}$ is an unbiased estimate of the policy gradient if $\hat{T}$ is an unbiased estimate of return:

$$\nabla_\theta J_\mu^\pi(\theta) = \mathop{\mathbb{E}}_{(s,a) \sim D_\mu^\pi} \left[\hat{G}_{PGPB}(s, a, \theta)\right] \tag{6}$$

Proof. Under the given assumptions on $\hat{T}(s, a)$, we have:

$$\nabla_\theta J_\mu^\pi(\theta) = \mathop{\mathbb{E}}_{(s,a) \sim D_\pi} \left[\hat{T}(s, a) \nabla_\theta \log \pi_\theta(a|s)\right] \tag{37}$$

$$= \mathop{\mathbb{E}}_{s \sim d_\mu^\pi} \left[\mathop{\mathbb{E}}_{a|s \sim \pi} \left[(\hat{T}(s, a) - q_\theta(s, a)) \nabla_\theta \log \pi_\theta(a|s)\right] + \mathop{\mathbb{E}}_{u|s \sim \pi} [q_\theta(s, u) \nabla_\theta \log \pi_\theta(u|s)]\right] \tag{38}$$

$$= \mathop{\mathbb{E}}_{s \sim d_\mu^\pi} \left[\mathop{\mathbb{E}}_{a|s \sim \pi} \left[(\hat{T}(s, a) - q_\theta(s, a)) \nabla_\theta \log \pi_\theta(a|s)\right] + \nabla_\theta \mathop{\mathbb{E}}_{u|s \sim \pi_\theta} [\hat{q}_\theta(s, u)]\right] \tag{39}$$

$$= \mathop{\mathbb{E}}_{(s,a) \sim D_\pi} \left[\left(\hat{T}(s, a) - q_\theta(s, a)\right) \nabla_\theta \log \pi_\theta(u|s) + \nabla_\theta \mathop{\mathbb{E}}_{u|s \sim \pi_\theta} [\hat{q}_\theta(s, u)]\right] \tag{40}$$

$$= \mathop{\mathbb{E}}_{(s,a) \sim D_\pi} \left[\left(\hat{T}(s, a) - q_\theta(s, a)\right) \nabla_\theta \log \pi_\theta(u|s) - \nabla_\theta H(\pi_\theta(.|s))\right] \text{ using Theorem 3.5} \tag{41}$$

$$= \mathop{\mathbb{E}}_{(s,a) \sim D_\pi} \left[\hat{G}_{PGPB}(s, a, \theta)\right] \tag{42}$$

$\square$

Theorem 3.5. *Let $\hat{q}_\theta$ indicate a stop-gradient operator on q_θ with respect to θ. Given the softmax parameterization of π_θ (Equation (4)), we have:*

$$\nabla_\theta H(\pi_\theta(.|s)) + \nabla_\theta \mathop{\mathbb{E}}_{u|s \sim \pi_\theta} [\hat{q}_\theta(s, u)] = 0 \tag{15}$$

Proof.

$$\nabla_\theta \mathop{\mathbb{E}}_{u|s\sim\pi_\theta} [\hat{q}_\theta(s,u)] = \sum_u q_\theta(s,u)\nabla_\theta\pi_\theta(u|s)$$

$$= \sum_u \pi_\theta(u|s)q_\theta(s,u)\nabla_\theta\log\pi_\theta(u|s)$$

$$= \sum_u \pi_\theta(u|s)(\log\pi_\theta(u|s) + F(q_\theta(s)))\nabla_\theta\log\pi_\theta(u|s)$$

$$= \sum_u \pi_\theta(u|s)\log\pi_\theta(u|s)\nabla_\theta\log\pi_\theta(u|s) + F(q_\theta(s))\sum_u \pi_\theta(u|s)\nabla_\theta\log\pi_\theta(u|s)$$

$$= \sum_u \pi_\theta(u|s)\log\pi_\theta(u|s)\nabla_\theta\log\pi_\theta(u|s) + F(q_\theta(s))\sum_u \nabla_\theta\pi_\theta(u|s)$$

$$= \sum_u \pi_\theta(u|s)\log\pi_\theta(u|s)\nabla_\theta\log\pi_\theta(u|s) + F(q_\theta(s))\nabla_\theta\sum_u \pi_\theta(u|s)$$

$$= \sum_u \pi_\theta(u|s)\log\pi_\theta(u|s)\nabla_\theta\log\pi_\theta(u|s) + F(q_\theta(s))\nabla_\theta 1$$

$$= \sum_u \pi_\theta(u|s)\log\pi_\theta(u|s)\nabla_\theta\log\pi_\theta(u|s) + \nabla_\theta 1$$

$$= \sum_u \pi_\theta(u|s)\log\pi_\theta(u|s)\nabla_\theta\log\pi_\theta(u|s) + \sum_u \pi_\theta(u|s)\nabla_\theta\log\pi_\theta(u|s)$$

$$= \sum_u \log\pi_\theta(u|s)\nabla_\theta\pi_\theta(u|s) + \sum_u \pi_\theta(u|s)\nabla_\theta\log\pi_\theta(u|s)$$

$$= \sum_u \nabla_\theta\left(\pi_\theta(u|s)\log\pi_\theta(u|s)\right)$$

$$= -\nabla_\theta H(\pi_\theta(.|s))$$

□

Theorem 3.3. *Consider the value-based gradients following the MSE and MVE objectives respectively defined as:*

$$G_{MSE}(\theta) = -\nabla_\theta\frac{1}{2}\mathop{\mathbb{E}}_{(s,a)}\left[(\hat{T}(s,a) - q_\theta(s,a))^2\right] \tag{7}$$

$$G_{MVE}(\theta) = -\nabla_\theta\frac{1}{2}\mathop{\mathbb{E}}_{s}\mathop{\mathbb{V}}_{a}\left[\hat{T}(s,a) - q_\theta(s,a)\right], \tag{8}$$

where, $\mathbb{V}_a$ denotes the variance of the distribution over a, conditioned on the state s. Then, $\hat{G}_{MSE}$ and $\hat{G}_{MVE}$ below are unbiased gradient[5] estimators for G_{MSE} and G_{MVE}.

$$\hat{G}_{MSE}(s,a,\theta) \triangleq \left(\hat{T}(s,a) - q_\theta(s,a)\right)\nabla_\theta q_\theta(s,a) \tag{9}$$

$$\hat{G}_{MVE}(s,a,\theta) \triangleq \left(\hat{T}(s,a) - q_\theta(s,a)\right)\nabla_\theta\log\pi_\theta(a|s) \tag{10}$$

Proof.

$$G_{MSE}(\theta) = -\frac{1}{2}\mathop{\mathbb{E}}_{(s,a)\sim\hat{D}_\mu^\pi}\nabla_\theta\left[(\hat{T}(s,a) - q_\theta(s,a))^2\right]$$

$$= \mathop{\mathbb{E}}_{(s,a)\sim\hat{D}_\mu^\pi}\left[\left(\hat{T}(s,a) - q_\theta(s,a)\right)\nabla_\theta q_\theta(s,a)\right]$$

$$= \mathop{\mathbb{E}}_{(s,a)\sim\hat{D}_\mu^\pi}\left[\hat{G}_{MSE}(s,a,\theta)\right] \tag{43}$$

---

[5]These are technically semi-gradients because any dependence of $\hat{D}_\mu^\pi$, the assumed on-policy sampling distribution for (s,a) and $\hat{T}(s,a)$ on θ are ignored, similar to most TD learning methods.

This proves the statement about $\hat{G}_{MSE}$. For the claim regarding $\hat{G}_{MVE}$:

$$G_{MVE}(\theta) = -\nabla_\theta \frac{1}{2} \mathop{\mathbb{E}}_s \mathop{\mathbb{V}}_a \left[\hat{T}(s,a) - q_\theta(s,a) \right] \tag{44}$$

$$= -\frac{1}{2} \mathop{\mathbb{E}}_s \nabla_\theta \mathop{\mathbb{V}}_{a|s\sim\hat{\pi}} \left[\hat{T}(s,a) - q_\theta(s,a) \right] \tag{45}$$

$$= -\frac{1}{2} \mathop{\mathbb{E}}_s \left[\nabla_\theta \mathop{\mathbb{E}}_{a|s\sim\hat{\pi}} \left[(\hat{T}(s,a) - q_\theta(s,a))^2 \right] - \nabla_\theta \left(\mathop{\mathbb{E}}_{a|s\sim\hat{\pi}} \left[\hat{T}(s,a) - q_\theta(s,a) \right] \right)^2 \right] \tag{46}$$

$$= \mathop{\mathbb{E}}_s \left[\mathop{\mathbb{E}}_{a|s\sim\pi} \left[\left(\hat{T}(s,a) - q_\theta(s,a) \right) \nabla_\theta q_\theta(s,a) \right] - \mathop{\mathbb{E}}_{a|s\sim\pi} \left[\hat{T}(s,a) - q_\theta(s,a) \right] \mathop{\mathbb{E}}_{u|s\sim\pi} \left[\nabla_\theta q_\theta(s,u) \right] \right] \tag{47}$$

$$= \mathop{\mathbb{E}}_{(s,a)} \left[\left(\hat{T}(s,a) - q_\theta(s,a) \right) \left(\nabla_\theta q_\theta(s,a) - \mathop{\mathbb{E}}_{u|s\sim\pi} \left[\nabla_\theta q_\theta(s,u) \right] \right) \right] \tag{48}$$

$$= \mathop{\mathbb{E}}_{(s,a)} \left[\left(\hat{T}(s,a) - q_\theta(s,a) \right) \nabla_\theta \log \pi_\theta(a|s) \right] \tag{49}$$

$$= \mathop{\mathbb{E}}_{(s,a)} \left[\hat{G}_{MVE}(s,a,\theta) \right] \tag{50}$$

$\square$

Theorem 4.3. *The PPO surrogate objective gradient can be recovered as $\hat{G}_{f,\pi}$ in (30) for the gradient scaling function,* $f(x,y) = e^x y\, \tau_\epsilon(x,y)$, *where* $\tau_\epsilon(x,y) = \mathbb{1}_{y>0}\mathbb{1}_{x<\log(1+\epsilon)} + \mathbb{1}_{y<0}\mathbb{1}_{x>\log(1-\epsilon)}$ *while choosing* $\alpha = 0$ *in (29). Note that* $\tau_\epsilon(x,y) \in \{0,1\}\ \forall x,y,\epsilon.$

Proof. Assume that the batch of sample data is generated from a policy π_b with parameters b. Let $\hat{G}_{PPO}$ denote the PPO surrogate objective sample gradient. This gradient can be simplified as follows to recover the scaling function equivalence. The PPO surrogate objective is defined for a given clipping parameter ϵ as:

$$J_{PPO}(s,a,\theta) = \left(\frac{\pi_\theta(a|s)}{\pi_b(a|s)} \hat{A}, \mathrm{clip}\left(\frac{\pi_\theta(a|s)}{\pi_b(a|s)}, 1-\epsilon, 1+\epsilon \right) \hat{A} \right)$$

We will focus on computing the gradient, $\hat{G}_{PPO}(s,a,\theta) = \nabla_\theta J_{PPO}(s,a,\theta)$ below. Note that $\Delta_O = \log \frac{\pi_\theta(a|s)}{\pi_b(a|s)}$, and $\Delta_R = \hat{A}$ under the specified assumptions. Therefore,

$$\hat{G}_{PPO}(s,a,\theta) = \nabla_\theta \left(\min\left(\frac{\pi_\theta(a|s)}{\pi_b(a|s)} \Delta_R, \mathrm{clip}\left(\frac{\pi_\theta(a|s)}{\pi_b(a|s)}, 1-\epsilon, 1+\epsilon \right) \Delta_R \right) \right) \tag{51}$$

$$= \nabla_\theta \left(\mathbb{1}_{\Delta_R>0} \min\left(\frac{\pi_\theta(a|s)}{\pi_b(a|s)}, 1+\epsilon \right) \Delta_R + \mathbb{1}_{\Delta_R<0} \max\left(1-\epsilon, \frac{\pi_\theta(a|s)}{\pi_b(a|s)} \right) \Delta_R \right) \tag{52}$$

$$= \nabla_\theta \Bigg(\left(\mathbb{1}_{\Delta_R>0} \mathbb{1}_{\Delta_O<\log(1+\epsilon)} + \mathbb{1}_{\Delta_R<0} \mathbb{1}_{\Delta_O>\log(1+\epsilon)} \right) \frac{\pi_\theta(a|s)}{\pi_b(a|s)} \Delta_R\ +$$

$$(1+\epsilon)\mathbb{1}_{\Delta_R>0}\mathbb{1}_{\Delta_O\geq\log(1+\epsilon)}\Delta_R + (1-\epsilon)\mathbb{1}_{\Delta_R<0}\mathbb{1}_{\Delta_O\leq\log(1-\epsilon)}\Delta_R \Bigg) \tag{53}$$

$$= \left(\mathbb{1}_{\Delta_R>0}\mathbb{1}_{\Delta_O<\log(1+\epsilon)} + \mathbb{1}_{\Delta_R<0}\mathbb{1}_{\Delta_O>\log(1+\epsilon)} \right) \nabla_\theta \frac{\pi_\theta(a|s)}{\pi_b(a|s)} \Delta_R + \mathbf{0} \quad (\because \text{ no gradient from the second term.}) \tag{54}$$

$$= \tau_\epsilon(\Delta_O, \Delta_R) \frac{\pi_\theta(a|s)}{\pi_b(a|s)} \nabla_\theta \log \pi_\theta(a|s) \Delta_R \tag{55}$$

$$= \tau_\epsilon(\Delta_O, \Delta_R) e^{\Delta_O} \Delta_R \nabla_\theta \log \pi_\theta(a|s) \tag{56}$$

$\square$

A.1. Design for the parametric scaling function approximations

Consider the following lower bound to $e^{\Delta_O}(e^{\Delta_R} - 1)$ which is also exact upto second order around $(\Delta_O, \Delta_R) = (0, 0)$. Let $X \sim U[\Delta_O, \Delta_O + \Delta_R]$ be a continuous RV with uniform density. Then:

$$\mathbb{E}[e^X] = \int_{\Delta_O}^{\Delta_O + \Delta_R} \frac{e^x}{\Delta_R} dx = \frac{e^{\Delta_O + \Delta_R} - e^{\Delta_O}}{\Delta_R} \text{ and } e^{\mathbb{E}[X]} = e^{\Delta_O + \frac{\Delta_R}{2}}$$

Jensen's inequality implies that

$$0 < e^{\Delta_O + \frac{\Delta_R}{2}} \leq \frac{e^{\Delta_O + \Delta_R} - e^{\Delta_O}}{\Delta_R} \ \forall \ \Delta_O, \Delta_R \in \mathbb{R}$$

Consider the following approximation inspired by the above inequality:

$$e^{\Delta_O + \frac{\Delta_R}{2}} \approx \frac{e^{\Delta_O + \Delta_R} - e^{\Delta_O}}{\Delta_R} \tag{57}$$

We then consider the following sequence of approximations:

$$e^{\Delta_O}(e^{\Delta_R} - 1) = e^{\Delta_O + \Delta_R} - e^{\Delta_O} \tag{58}$$

$$\approx \Delta_R e^{\Delta_O + \frac{\Delta_R}{2}} \quad \text{Using Eq 57} \tag{59}$$

$$\geq \Delta_R \max\left(1 + \Delta_O + \frac{\Delta_R}{2}, 0\right) \quad \text{since } e^x \geq \max(1 + x, 0) \ \forall x \in \mathbb{R} \tag{60}$$

Unfortunately, $f_0(\Delta_O, \Delta_R) \triangleq \Delta_R \max\left(1 + \Delta_O + \frac{\Delta_R}{2}, 0\right)$ does not satisfy the monotonicity constraint in Δ_R listed in 5.1 in all cases. To check this, we can verify that it is non-decreasing (and also convex) in Δ_R if and only if $1 + \Delta_O \leq 0$. Equation (35) makes a fix to f_0 to ensure that the monotonicity constraints from Assumption 5.1 hold while also remaining a good second order approximation when Δ_O, Δ_R are small.

Fix for non-monotonicity(f_{MLA}): First notice that the non-monotonicity only happens when $1 + \Delta_O > 0$. In that case, the minimum of the quadratic in y, $q(x, y) = y(1 + x + y/2)$ occurs for $y_{min}(x) \triangleq -(1 + x) < 0$. However, in this case $1 + x + y_{min}(x)/2 = (1 + x)/2 > 0$. So $f(x, y) = q(x, y)$ in a left neighborhood of $y_{min}(x) = -(1 + x)$, which means it must be decreasing in y in that range. To avoid this, we make a fix $f(x, y) \triangleq q(x, y_{min}(x)) = -(1 + x)^2/2$, when $y < y_{min}(x) = -(1 + x)$ and $1 + x > 0$. The resulting $f(x, y)$ is the definition of $f_{MLA}(x, y)$ in Equation (35).

Derivation of $f_{MLA(\alpha_0, \alpha_r)}$: In the policy parameterization, we did not include any temperature parameter for simplicity. However, by considering a generalization of these as free parameters, we may consider scaling functions where the inputs are replaced with $\alpha_o \Delta_O$ and $\alpha_r \Delta_R$ respectively. The same sequence of approximations resulting in Equation (60) now result in the constraint violating function[6] $f_1(x, y) = \alpha_r y(1 + \alpha_o x + \alpha_r y/2)_+$. Without loss of generality, this is equivalent to $f_1(x, y) = y(1 + \alpha_o x + \alpha_r y)_+$, because (1) the constant α_r scaling the outer y has no impact on the overall update, and (2) we can redefine the remaining α_r inside to include the 1/2 for simplicity. We are now left with fixing the non-monotonicity of $f_1(x, y) = y \max(1 + \alpha_0 x + \alpha_r y, 0)$ for certain ranges of y when $1 + \alpha_0 x < 0$. However, we do this slightly differently compared to how it was done for f_{MLA} above, in order to preserve the linear component in y, as this allows for a more intuitive understanding as being the component that matches the identity scaling function.

Fix for non-monotonicity: As before, consider the quadratic in y, $q(x, y) \triangleq y(1 + \alpha_o x + \alpha_r y)$ and its argmin $y_{min}(x) = \frac{-(1 + \alpha_o x)}{2\alpha_r}$ in the problematic case when $1 + \alpha_o x > 0$. Observe that when $y = y_{min}$, we have $1 + \alpha_o x + \alpha_r y = \frac{1 + \alpha_o x}{2}$. Recall that the problem occurs due to the quadratic dependence on y from the second term for $y < y_{min}$. Instead of completely eliminating the dependence on y, we can continue a monotonic linear relationship by simply clipping the second term below by the value it takes at y_{min} which is $1 + \alpha_o x + \alpha_r y = \frac{1 + \alpha_o x}{2}$. However, recall that we should only perform this clipping of the second term below by $\frac{1 + \alpha_o x}{2}$ in the case when $1 + \alpha_o x > 0$. This can be done simply by clipping the second term by $\frac{(1 + \alpha_o x)_+}{2}$ as this reduces to the original floor of 0 in the case where no fix was needed. This results in the proposed $f_{MLA(\alpha_o, \alpha_r)}(x, y) = y \max\left(1 + \alpha_o x + \alpha_r y, \frac{(1 + \alpha_o x)_+}{2}\right)$.
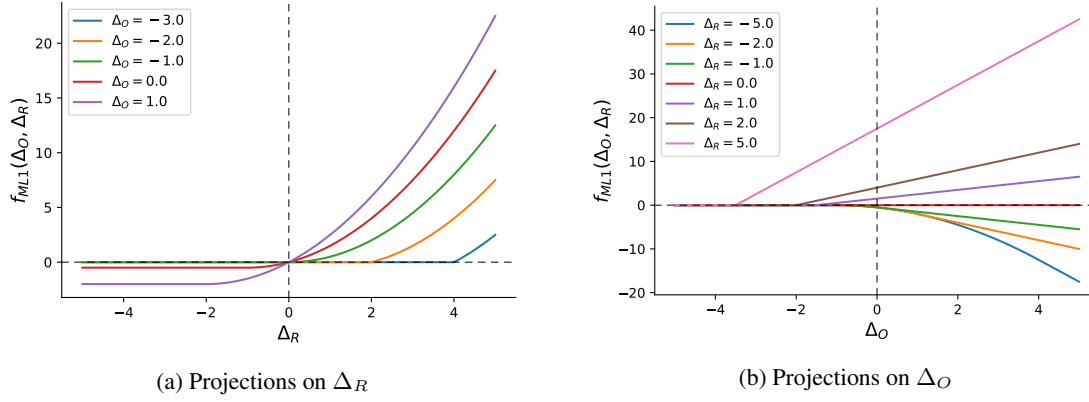
---

[6] $(x)_+$ denotes $\max(x, 0)$

(a) Projections on Δ_R

(b) Projections on Δ_O

Figure 4: Visualization for $f_{MLA}(\Delta_O, \Delta_R)$ (Equation 35)

(a) Projections on Δ_R
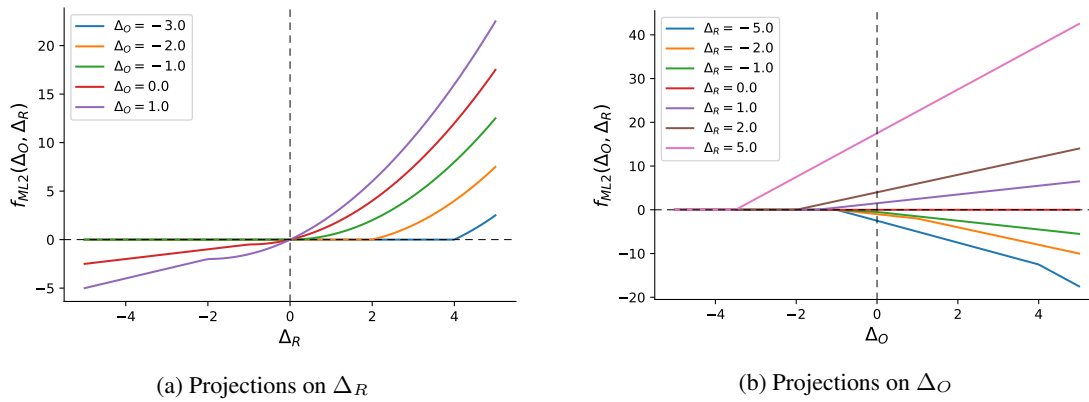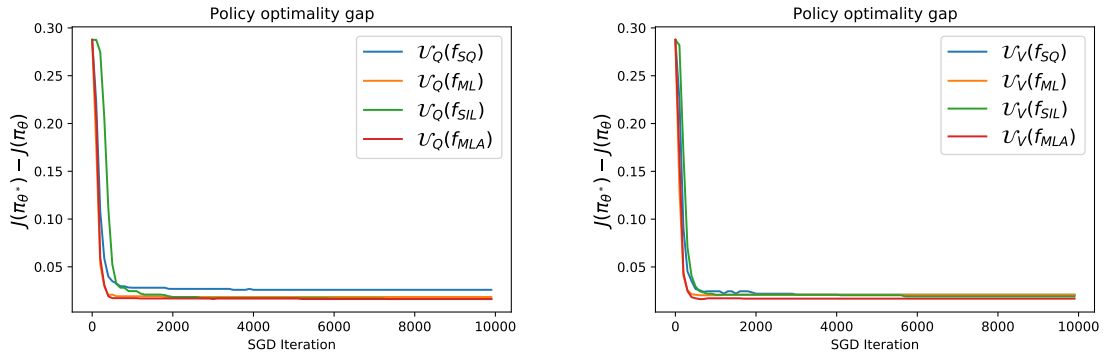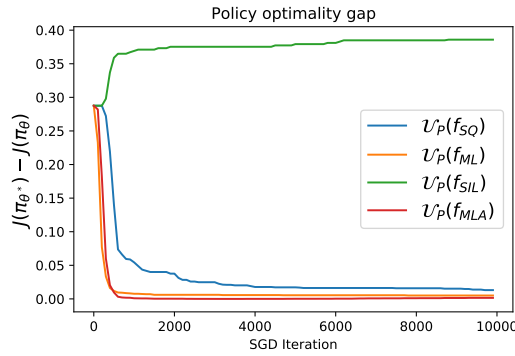
(b) Projections on Δ_O

Figure 5: Visualization for $f_{MLA(1,0.5)}(\Delta_O, \Delta_R)$ (Equation 36 for $\alpha_o = 1, \alpha_r = 0.5$.)

(a) Learning curves for $\mathcal{U}_Q(f)$: Col 1, Table 1

(b) Learning curves for $\mathcal{U}_V(f)$: Col 2, Table 1

(c) Learning curves for $\mathcal{U}_P(f)$: Col 3, Table 1

Figure 6: Gap to the optimal policy reward for the twelve updates listed in Table 1 grouped by the update type in each column, and compared across scaling functions $f \in \{f_{SQ}, f_{ML}, f_{SIL}, f_{MLA}\}$
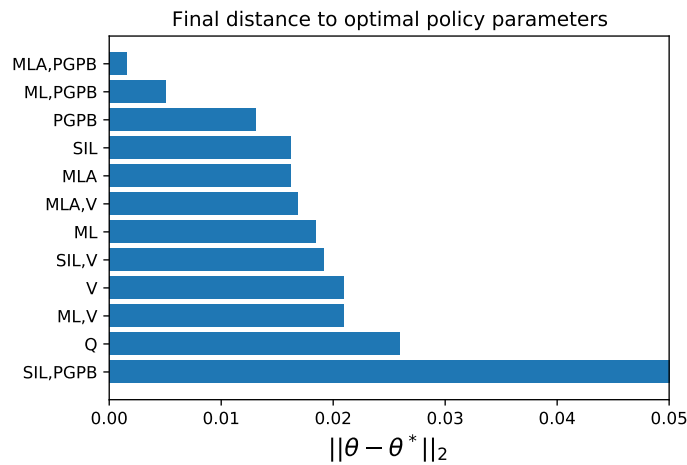

Figure 7: Final Euclidean distance to the optimal parameters, $\theta^* = (1,1)$ for the twelve update rules considered.
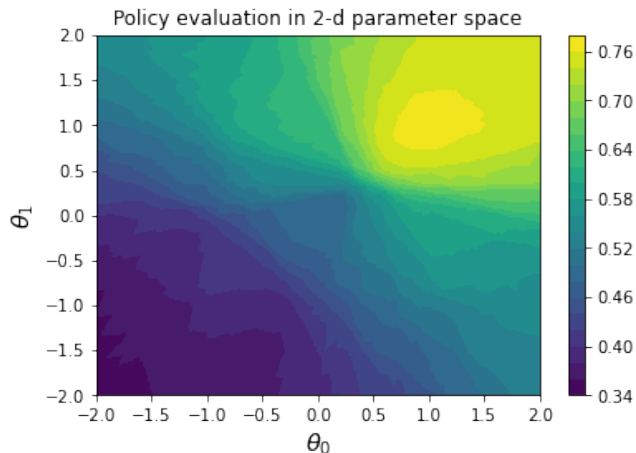
Figure 8: Objective landscape

B. Table of update rule combinations evaluated for the 2D synthetic experiment

The full range of gradients characterized are summarized in Table 1, which considers the form-axis along the columns for the three types of updates (squared error minimization, variance minimzation and PGPB), in conjunction with several scaling functions along the row axis.

Table 1: A summary of the various updates evaluated in the synthetic 2D experiment.

$$\mathcal{U}_Q(f) \equiv f(\Delta_O, \Delta_R)\Big(\nabla_\theta q_\theta(s,a)\Big)$$
$$\mathcal{U}_V(f) \equiv f(\Delta_O, \Delta_R)\Big(\nabla_\theta q_\theta(s,a) - \mathbb{E}_{u|s\sim\pi}\left[\nabla_\theta q_\theta(s,u)\right]\Big)$$
$$\mathcal{U}_P(f) \equiv f(\Delta_O, \Delta_R)\Big(\nabla_\theta q_\theta(s,a) - \mathbb{E}_{u|s\sim\pi}\left[\nabla_\theta q_\theta(s,u)\right]\Big) + \nabla_\theta \mathbb{E}_{u|s\sim\pi_\theta}\left[\hat{q}_\theta(s,u)\right]$$

f	$\mathcal{U}_Q(f)$	$\mathcal{U}_V(f)$	$\mathcal{U}_P(f)$
$f_{SQ}(x,y) \triangleq e^x y$	$\hat{G}_Q$	$\hat{G}_V$	$\hat{G}_{PGPB}$
$f_{ML}(x,y) \triangleq e^x(e^y - 1)$	$\hat{G}_{ML}$	$\hat{G}_{ML,V}$	$\hat{G}_{ML,PGPB}$
$f_{SIL}(x,y) \triangleq e^x \max(y, 0)$	$\hat{G}_{SIL}$	$\hat{G}_{SIL,V}$	$\hat{G}_{SIL,PGPB}$
$f_{MLA}(x,y) \triangleq \begin{cases} -\frac{1}{2}(1+x)^2, & \text{if } 1+x+y \le 0 < 1+x \\ y\max\left(1+x+\frac{y}{2}, 0\right), & \text{otherwise} \end{cases}$	$\hat{G}_{MLA}$	$\hat{G}_{MLA,V}$	$\hat{G}_{MLA,PGPB}$

C. Experiment Details

C.1. Tabular Experiments

The FourRoom environment is shown in Figure 9. Both policy gradient (PG) and Q-learning (QL) use SGD as the optimizer. The learning rate for PG is 0.1 for both the actor and the critic, while the learning rate for QL is 0.01. Figure 2 reports the exponential running average return with exponent discount of 0.01.
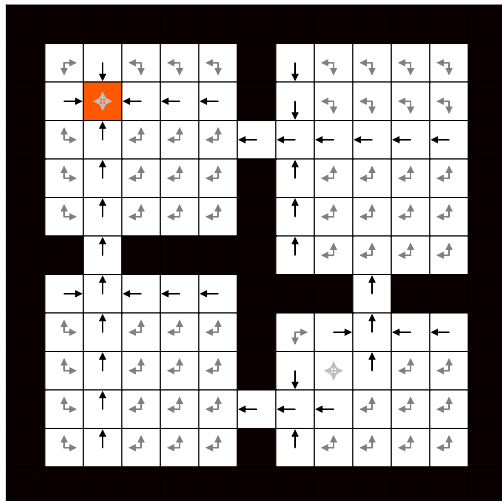
Figure 9: The FourRoom Environment

C.2. Continuous Control Experiments

The full set of update equations used are summarized below:

$$\hat{G}_{MLA-PPO}(s, a, \theta) = f_{MLA-PPO}(\Delta_O, \Delta_R)\nabla_\theta \log \pi_\theta(a|s) + \beta\nabla_\theta H(\pi_\theta(.|s)), \text{ where:}$$
$$f_{MLA-PPO}(x, y) = f_{MLA(\alpha_o, \alpha_r)}(x, y)\tau_\epsilon(x, y), \text{ where:}$$
$$f_{MLA(\alpha_o, \alpha_r)}(x, y) = y \max\left(1 + \alpha_o x + \alpha_r y, \frac{(1 + \alpha_o x)_+}{2}\right)$$
$$\tau_\epsilon(x, y) = \mathbb{1}_{y>0}\mathbb{1}_{x<\log(1+\epsilon)} + \mathbb{1}_{y<0}\mathbb{1}_{x>\log(1-\epsilon)}$$
$$\Delta_R = \hat{A} - \alpha\big(\log \pi(a|s) + H(\pi(.|s))\big)$$

Table 2: Optimal hyper-parameter configurations for PPO - MLA on the MuJoCo Tasks. Note that the performance reported for the baseline PPO implementation uses its own independently tuned clipping ϵ.

Environment	α_r	α_o	α	ϵ
Walker2d-v2	0.05	0.1	0.0	0.2
Hopper-v2	1.0	0.1	0.01	0.2
HalfCheetah-v2	0.5	0.1	0.1	0.2
Humanoid-v2	1.0	1.0	0.0	0.3