Safely Exploring Policy Gradient

Matteo Papini MATTEO.PAPINI@POLIMI.IT
Andrea Battistello ANDREA.BATTISTELLO@MAIL.POLIMI.IT
Marcello Restelli MARCELLO.RESTELLI@POLIMI.IT
Politecnico di Milano

Abstract

Exploration is a fundamental aspect of reinforcement learning (RL) agents. In real-world applications, such as the control of industrial processes, exploratory behavior may have immediate costs that are only repaid in the long run. Selecting the right amount of exploration is a challenging task that can significantly improve overall performance. Safe RL algorithms focus on limiting the immediate costs. This conservative approach, however, carries the risk of sacrificing too much in terms of learning speed and exploration. Starting from the idea that a practical algorithm should be as safe as needed, but not more, we identify an interesting safety scenario and propose Safely-Exploring Policy Gradient (SEPG) to solve it. To do this, we generalize the existing bounds on performance improvement for Gaussian policies to the case of adaptive variance and propose policy updates that are both safe and exploratory. We evaluate our algorithm on simulated continuous control tasks.

Keywords: Reinforcement Learning, Policy Gradient, Safe Learning, Exploration

1. Introduction

Reinforcement learning (RL) (Sutton and Barto, 1998) is an approach to adaptive intelligence that employs a reward signal to train an autonomous agent on a general task by direct interaction with the surrounding environment. The results recently achieved by RL in games (Mnih et al., 2013; Silver et al., 2017) are astounding. However, when RL is applied to real-world scenarios (e.g., industrial process control, robot learning, autonomous driving), we have to face further challenges. First of all, most games are naturally modeled with discrete states and actions. Instead, real-world problems are often better modeled as continuous control tasks. For this reason, we will focus on policy gradient (PG) (Sutton et al., 2000; Deisenroth et al., 2013), an RL technique that employs stochastic gradient ascent to optimize parametric controllers. PG is particularly suited for continuous control tasks due to its robustness to noise, convergence properties, and versatility in policy design (Peters and Schaal, 2008b). Another advantage of games is that they are easily simulated. Simulation requires a reliable model of the environment, which is often not available in real-world tasks. This means we must learn on a real system (e.g., a manufacturing plant), facing the *safe exploration* problem (Amodei et al., 2016). Exploration can be defined, in very general terms, as the execution of unprecedented behaviors in order to gather useful information, and is a fundamental aspect of RL. The performance of exploratory behavior is either knowingly suboptimal or unpredictable. This results in immediate costs that can only be repaid in the long run by using the gathered information to improve the agent's policy. The need for *safe* exploration arises whenever the actions of our agent have concrete consequences. In the context of RL, the word *safety* has been used with several different meanings (Garcıa

and Fernández, 2015). When exploratory behavior can harm machines and people, we need to ensure safety in the traditional sense of avoiding dangerous situations, e.g., accidents in autonomous driving. In other cases, safety has an economic connotation: we must ensure that exploration costs are limited, or that they are fully repaid by long-term revenues. Although many cases of interest show both types of safety requirements (think of an industrial process), in this paper we will focus on the economic side. Several works have addressed this safety issue (Kakade and Langford, 2002; Pirotta et al., 2013b; Thomas et al., 2015; Schulman et al., 2015; Ghavamzadeh et al., 2016). In the case of PG, safety can be guaranteed through a conservative choice of meta-parameters (Pirotta et al., 2013a, 2015; Papini et al., 2017). Current safe PG algorithms focus exclusively on immediate costs. Without taking into account the long-term benefits of exploration, they run the risk of adopting unnecessarily conservative measures, resulting in slow learning. Moreover, they address monotonic improvement, i.e., the requirement that, during learning, the new policy is never worse than the previous one. We believe that this is too strict for most practical applications. In this paper, we adopt a more general definition of economic safety, based on *High Confidence Policy Improvement* (Thomas et al., 2015). We study ways of updating a Gaussian policy that are safe while explicitly taking into account exploration. We then focus on a more specific safety scenario that we think is of broad interest: to improve an existing controller without ever worsening compared to the original design. The structure of the paper is as follows: in Section 2, we provide the necessary PG fundamentals. In Section 3, we give our definition of a safe policy update and derive safe-exploratory updates for the Gaussian case. In Section 4, we describe the fine-tuning problem as a special case of safe exploration. In Section 5, we propose rigorous and heuristic algorithms to solve this problem. In Section 6, we empirically evaluate our algorithms on simulated control tasks.

2. Preliminaries

In this section, we provide an essential background on continuous Markov decision processes and policy gradient methods. A continuous Markov Decision Process (MDP) (Puterman, 2014) $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, \rho \rangle$ is defined by a continuous state space $\mathcal{S}$; a continuous action space $\mathcal{A}$; a Markovian transition kernel $\mathcal{P}$, where $\mathcal{P}(s'|s,a)$ is the transition density from state s to s' under action a; a reward function $\mathcal{R}$, where $\mathcal{R}(s,a) \in [-R, R]$ is the reward for state-action pair (s,a) and R is the maximum absolute-value reward; a discount factor $\gamma \in [0,1)$; and an initial state distribution ρ on $\mathcal{S}$. An agent's behavior is modeled as a policy π, where $\pi(\cdot|s)$ is the density distribution over $\mathcal{A}$ in state s. We study episodic MDPs with indefinite horizon. In practice, we consider episodes of length H, a number of steps sufficient to reach steady optimal behavior. A trajectory τ is a sequence of states and actions $(s_0, a_0, s_1, a_1, \ldots, s_{H-1}, a_{H-1})$ observed by following a stationary policy, where $s_0 \sim \rho$ and $s_{h+1} \sim \mathcal{P}(\cdot|s_h, a_h)$. The policy induces a measure p_π over trajectories. We denote with $\mathcal{R}(\tau)$ the total discounted reward provided by trajectory τ: $\mathcal{R}(\tau) = \sum_{h=0}^{H-1} \gamma^h \mathcal{R}(s_h, a_h)$. Policies can be ranked based on their expected total reward $J(\pi) = \mathbb{E}_{\tau \sim p_\pi}[\mathcal{R}(\tau)]$. Solving the MDP means finding an optimal policy $\pi^* \in \arg\max_\pi \{J(\pi)\}$. Policy gradient (PG) methods restrict this optimization problem to a class of parametrized policies $\Pi_{\boldsymbol{\theta}} = \{\pi_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \mathbb{R}^{m+1}\}$, so that $\pi_{\boldsymbol{\theta}}$ is differentiable w.r.t. $\boldsymbol{\theta}$. A widely used policy class is the Gaussian, i.e.,

$\pi_{\boldsymbol{\theta}}(a|s) \sim \mathcal{N}(\mu_{\boldsymbol{\theta}}(s), \sigma^2_{\boldsymbol{\theta}})$. We focus on the following, common parametrization:

$$\mu_{\boldsymbol{\theta}}(s) = \boldsymbol{v}^T \boldsymbol{\phi}(s), \qquad \sigma_{\boldsymbol{\theta}} = e^w, \tag{1}$$

where $\boldsymbol{\phi}(\cdot)$ is a vector of m state-features. Parameters $\boldsymbol{\theta} = [\boldsymbol{v}|w]$ consist of the weights $\boldsymbol{v} \in \mathbb{R}^m$ for the mean and a scalar $w \in \mathbb{R}$ for the policy log-variance. We also assume the state features to be bounded, i.e., $\sup_{s \in \mathcal{S}} |\phi_i(s)| \leq M_{\boldsymbol{\phi}}$ for $i = 1, \ldots, m$.

We denote the performance of a parametric policy $\pi_{\boldsymbol{\theta}}$ with $J(\boldsymbol{\theta})$. A locally optimal policy can be found via gradient ascent on the performance measure:

$$\boldsymbol{\theta}^{t+1} \leftarrow \boldsymbol{\theta}^t + \alpha \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^t), \qquad \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\tau \sim p_{\boldsymbol{\theta}}} \left[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\tau) \mathcal{R}(\tau) \right], \tag{2}$$

where t denotes the current iteration, $p_{\boldsymbol{\theta}}$ is short for $p_{\pi_{\boldsymbol{\theta}}}$ and α is a step size. This policy search technique is known as policy gradient (PG) (Sutton et al., 2000; Peters and Schaal, 2008a). In the following, we employ a variant of PG that uses *greedy coordinate ascent* (Nutini et al., 2015):

$$\theta_k^{t+1} \leftarrow \theta_k^t + \alpha \nabla_{\theta_k} J(\boldsymbol{\theta}^t) \quad \text{if} \quad k = \arg\max_i |\nabla_{\theta_i} J(\boldsymbol{\theta}^t)| \text{ else } \theta_k^t. \tag{3}$$

For the sake of brevity, we write the coordinate ascent update simply as:

$$\boldsymbol{\theta}^{t+1} \leftarrow \boldsymbol{\theta}^t \oplus \alpha \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^t), \tag{4}$$

where the $\oplus$ sign accounts for the fact that only the component with the maximum gradient magnitude is updated (ties are broken uniformly at random). For Gaussian policies, greedy coordinate ascent guarantees better one-step improvements than gradient ascent (Papini et al., 2017). In practice, $\nabla_{\boldsymbol{\theta}} J$ is often not available, but can be estimated from a batch of trajectories $\mathcal{D}_N = \{\tau_1, \ldots, \tau_N\}$. We discuss gradient estimation in Appendix B.

3. Safe Updates

In this section, we define what a safe policy update is and provide safe methods for updating the parameters of a Gaussian policy that explicitly take exploration into account. We assume here to have exact gradients, in order to decouple optimization and approximation issues. Our definition of safe update is based on the one proposed by (Thomas et al., 2015) in the context of batch RL.

Definition 1 *Given a parametric policy $\pi_{\boldsymbol{\theta}}$ with current parameter $\boldsymbol{\theta}^t$, we say that update $\Delta\boldsymbol{\theta} \in \mathbb{R}$ is δ-safe w.r.t. requirement $C^t \in \mathbb{R}$ if:*

$$J(\boldsymbol{\theta}^{t+1}) - J(\boldsymbol{\theta}^t) \geq C^t \quad \text{with probability at least } 1 - \delta,$$

where $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t + \Delta\boldsymbol{\theta}$.

When requirement C^t is positive, we talk of a *required performance improvement*, otherwise of a *bounded worsening*. Note that this is a generalization of the monotonic improvement constraint (Pirotta et al., 2013a), which corresponds to $C^t = 0$. First, we provide a way to safely update the mean parameter via greedy coordinate ascent:

Theorem 2 (Adapted from Theorem 3.3 in Papini et al. (2017)) *Assuming w is fixed, the guaranteed performance improvement yielded by (4) is maximized by step size $\alpha^* = \frac{1}{2c}$, where $c = \frac{RM_\phi^2}{(1-\gamma)^2\sigma^2}\left(\frac{|\mathcal{A}|}{\sqrt{2\pi}\sigma} + \frac{\gamma}{2(1-\gamma)}\right)$ and $|\mathcal{A}|$ is the volume of the action space. Moreover, the largest step size guaranteeing $J(\boldsymbol{v}^{t+1}, w) - J(\boldsymbol{v}^t, w) \geq C^t$ is $\overline{\alpha} = \alpha^*(1 + \lambda_{\boldsymbol{v}})$, where $\lambda_{\boldsymbol{v}} = \sqrt{1 - \frac{4cC^t}{\|\nabla_{\boldsymbol{v}} J(\boldsymbol{v}^t, w^t)\|_\infty^2}}$.*

Next, we provide a way to update the variance parameter (w-update) greedily, but safely:

Theorem 3 *Assuming $\boldsymbol{v}$ is fixed and $o(|\Delta w|^3)$ terms can be neglected,[1] the guaranteed performance improvement yielded by (4) is maximized by step size $\beta^* = \frac{1}{2d}$, where $d = \frac{R}{(1-\gamma)^2}\left(\frac{\psi|\mathcal{A}|}{2\sigma_w} + \frac{\gamma}{1-\gamma}\right)$ and $\psi = 4(\sqrt{7}-2)\exp(\sqrt{7}/2-2)/\sqrt{2\pi} \simeq 0.524$. Moreover, the largest step size guaranteeing $J(\boldsymbol{v}, w^{t+1}) - J(\boldsymbol{v}, w^t) \geq C^t$ is $\overline{\beta} = \beta^*(1 + \lambda_w)$, where $\lambda_w = \sqrt{1 - \frac{4dC^t}{\|\nabla_w J(\boldsymbol{v}^t, w^t)\|_\infty^2}}$.*

Finally, we need also a way to update the variance parameter in an exploratory fashion. The key intuition is that, most of the time, larger values of σ_w can positively affect the guaranteed improvement yielded by the next $\boldsymbol{v}$-update. Hence, we define a surrogate objective encoding this advantage of exploration:

Corollary 4 *The optimal step size α^* from Theorem 2, guarantees:*

$$J(\boldsymbol{v}^{t+1}, w) - J(\boldsymbol{v}^t, w) \geq \frac{\left\|\nabla_{\boldsymbol{v}} J(\boldsymbol{v}^t, w)\right\|_2^2}{4mc} := \mathcal{L}(\boldsymbol{v}^t, w),$$

where c is from Theorem 2 and m is the size of $\boldsymbol{v}$.

Regardless of the step size α that will be actually used in the following $\boldsymbol{v}$-update, we can optimistically update w in the direction of $\nabla_w \mathcal{L}$. This can be done safely as well:

Theorem 5 *Assuming $\boldsymbol{v}$ is fixed and $o(|\Delta w|^3)$ terms can be neglected, the guaranteed performance improvement yielded by $w^{t+1} \leftarrow w^t + \eta \nabla_w \mathcal{L}(\boldsymbol{v}, w^t)$ is maximized by step size $\eta^* = \frac{\nabla_w J(\boldsymbol{v}, w^t)}{2d\nabla_w \mathcal{L}(\boldsymbol{v}, w^t)}$. Moreover, the largest step size guaranteeing $J(\boldsymbol{v}, w^{t+1}) - J(\boldsymbol{v}, w^t) \geq C^t$ is $\overline{\eta} = \eta^* + |\eta^*|(1 + \lambda_w)$, where d and λ_w are from Theorem 3.*

Note that η^* can be negative and that $\overline{\eta}$ can have an opposite sign. This reflects the fact that the exploratory update is often in the opposite direction w.r.t. a greedy update. When the gradients must be estimated from data, high-probability variants of these theorems can be derived (see Appendix C).

4. Fine Tuning

The single-step definition of safety given in Section 3 can be applied to longer learning scenarios by specifying the safety requirement C^t for each update. In this section, we describe a special case of this general framework that we consider particularly relevant for applications: the fine-tuning of an existing controller.

1. This approximation is not critical since the steps produced by safe algorithms tend to be very small.

Imagine that an initial controller, or policy, is provided, e.g., designed by human experts. It has been thoroughly tested and is known to perform reasonably well. Now, we want to improve performance by tuning its parameters on the go, with online RL. If the costs of exploration are not repaid soon enough, the whole learning operation may be deemed ruinous and suspended. Hence, we need guarantees that the updated policies will at least not perform worse than the initial one.

Let us denote with $\boldsymbol{\theta}^t$ (short for $\pi_{\boldsymbol{\theta}^t}$) the t-th update of our policy, the initial version being $\boldsymbol{\theta}^0$. We call it *baseline policy* and $J(\boldsymbol{\theta}^0)$ *baseline performance*. The requirement of not to worsen w.r.t. the initial setting, as described above, can easily be formulated as follows:

$$J(\boldsymbol{\theta}^t) \geq J(\boldsymbol{\theta}^0) \quad \text{for all } t > 0, \tag{5}$$

where we assume a fixed batch size $N^t = N$ for all t. To use the theoretical tools developed in Section 3, we need to rewrite constraint (5) in accordance with Definition 1. To do so, we introduce the *exploration budget*:

Definition 6 *The exploration budget B^t following the evaluation of policy $\boldsymbol{\theta}^t$ is defined incrementally as:*

$$B^0 \coloneqq 0 \quad \textit{(initial budget)} \quad ; \quad B^{t+1} \coloneqq B^t + J(\boldsymbol{\theta}^{t+1}) - J(\boldsymbol{\theta}^t) \quad \textit{for all } t. \tag{6}$$

The budget simply cumulates all the performance changes up to the current policy, measuring the total improvement w.r.t. the baseline. This allows to reformulate constraint (5) as follows:

Theorem 7 *Constraint (5) is equivalent to the following:*

$$J(\boldsymbol{\theta}^{t+1}) - J(\boldsymbol{\theta}^t) \geq -B^t \quad \textit{for all } t.$$

Since the constraints of Theorem 12 are in the form of Definition 1, the safe updates devised in Section 3 can be performed to guarantee constraint (5) in the case of a Gaussian policy. It is enough to set the safety requirement to the negative budget, i.e., $C^t \leftarrow -B^t$ and use the suggested step size. A safe update can worsen performance as long as the loss does not exceed the budget. Any performance improvement increases the budget, allowing for larger steps and encouraging further exploration.

5. Algorithms

In this section we design an algorithm to solve the problem defined in Section 4 for the special case of Gaussian policies parameterized as in (1). We provide both a rigorous, safe version and a heuristic, semi-safe version.

5.1 SEPG

The initial controller is parameterized by $\boldsymbol{\theta}^0 = [\boldsymbol{v}^0 | w^0]$. It can be seen as a deterministic controller $a = \boldsymbol{\phi}(s)^T \boldsymbol{v}^0$ (e.g., based on some domain knowledge or learned by imitation) plus a tolerable noise $\mathcal{N}(0, e^{2w^0})$. Starting from this initial design, we update the mean parameter $\boldsymbol{v}$ and the variance parameter w alternately. The mean parameter is updated via coordinate

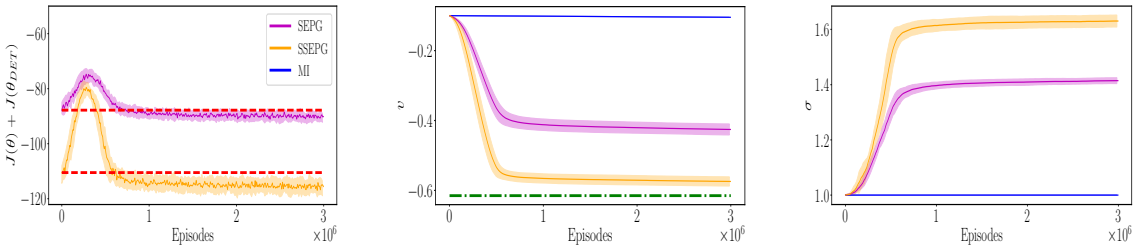
Algorithm 1 (S)SEPG

1: **input:** $\boldsymbol{v}^0$, w^0
2: **initialize:** $B \leftarrow B^0$
3: **for** t $= 0 \ldots$ **do**
4: $\quad \boldsymbol{v}^{t+1} \leftarrow \boldsymbol{v}^t \oplus \overline{\alpha} \nabla_{\boldsymbol{v}} J(\boldsymbol{v}^t, w^t)$
5: $\quad$ **evaluate** $J(\boldsymbol{v}^{t+1}, w^t)$
6: $\quad B \leftarrow B + J(\boldsymbol{v}^{t+1}, w^t) - J(\boldsymbol{v}^t, w^t)$
7: $\quad$ **evaluate** $J(\boldsymbol{v}^{t+1}, -\infty)$
8: $\quad B \leftarrow B + J(\boldsymbol{v}^{t+1}, -\infty) - J(\boldsymbol{v}^t, -\infty)$
9: $\quad w^{t+1} \leftarrow w^t \oplus \overline{\eta} \nabla_w \mathcal{L}(\boldsymbol{v}^{t+1}, w^t)$
10: $\quad$ **evaluate** $J(\boldsymbol{v}^{t+1}, w^{t+1})$
11: $\quad B \leftarrow B + J(\boldsymbol{v}^{t+1}, w^{t+1}) - J(\boldsymbol{v}^{t+1}, w^t)$
12: **end for**

ascent on $J(\boldsymbol{\theta})$ using the largest safe step size $\overline{\alpha}$ from Theorem 2. The variance parameter is then updated via coordinate ascent on the exploratory objective $\mathcal{L}(\boldsymbol{\theta})$ using the largest safe step size $\overline{\eta}$ from Theorem 5. The pseudo-code for SEPG is reported in Algorithm 1 (ignoring lines 2, 7, 8). We identify a learning iteration t with a full pair of updates, but the budget is still updated after each parameter update. The safety requirement used to compute the step sizes is always the current negative budget. The performance of the initial policy is assumed to be known. We call this algorithm Safely Exploring Policy Gradient (SEPG), since it explicitly favors exploration while satisfying a natural safety constraint.

5.2 SSEPG

In practice, what has been done so far may not be sufficient to account for the long-term advantages of exploration. In many applications, the final goal of learning is a deterministic controller. The stochasticity of the policy is necessary for exploration, but should be dropped in the end to fully exploit the optimized controller. Hence, the advantages of updating w^t should be measured not only on $\boldsymbol{\theta}^{t+1} = [\boldsymbol{v}^{t+1} \mid w^{t+1}]$, but also on the corresponding deterministic policy. We call it *test policy* and denote it with $\boldsymbol{\theta}^{t+1}_{\text{DET}} = [\boldsymbol{v}^{t+1} \mid -\infty]$, since $w = -\infty$ corresponds to zero variance. Based on this intuition, we develop a heuristic SEPG variant , called SSEPG (the double 'S' stands for *Semi-Safe*). In this version, the test policy is also evaluated. The safety condition becomes $J(\boldsymbol{\theta}^t) + J(\boldsymbol{\theta}^t_{\text{DET}}) \geq J(\boldsymbol{\theta}^0) + J(\boldsymbol{\theta}^0_{\text{DET}})$. The improvements of the deterministic policy are added to the budget, encouraging exploration even further. The pseudo-code for SSEPG is reported in Algorithm 1. The performance of the initial policy and its deterministic counterpart are assumed to be known. We also include the possibility of providing an initial "exploration grant" $B^0 > 0$. The additions w.r.t. SEPG are highlighted. Unfortunately, the evaluation of the test policy has no safety guarantees, since $w = -\infty$ represents a degenerate case for the bounds of Section 3. This means SSEPG cannot be considered safe in the sense of constraint (5). However, in many practical cases, we expect the deterministic policy to perform much better than the stochastic counterpart. Moreover, if the budget actually becomes negative, the safe updates will try to restore it by selecting very conservative step sizes, although this may require several iterations.



(a) Total performance (b) Mean parameter (c) Policy variance

Figure 1: LQG experiment averaged over 5 runs, with 95% confidence intervals.

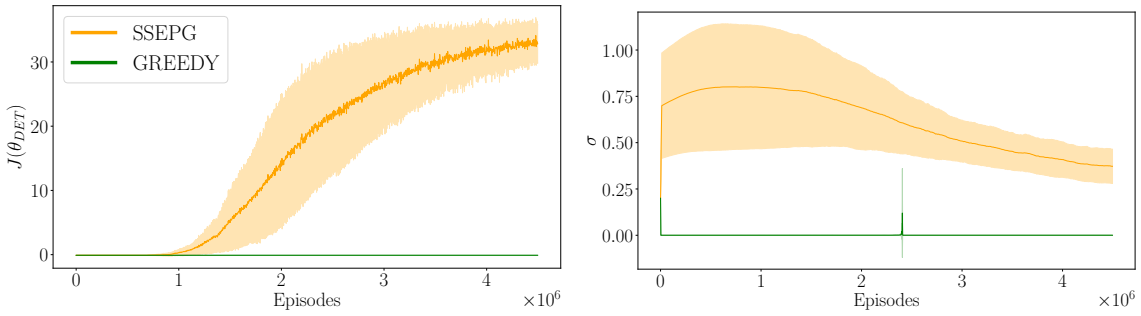
For this reason, SSEPG can (informally) be deemed safe over a long time horizon. The appropriateness of employing this semi-safe variant strongly depends on the application.

6. Empirical Evaluation

In this section, we evaluate the algorithms proposed in Section 5 on simulated control problems. First, we use the Linear Quadratic Regulator task to get empirical evidence about the safety properties of our algorithms and to compare them. Then, we use the Mountain Car task to highlight the importance of the exploratory variance update. In all our experiments we use a batch size of $N = 100$ episodes for policy evaluations and gradient estimations.

6.1 Linear Quadratic Regulator

The Linear-Quadratic Regulator (LQR) problem has been extensively studied in the optimal control literature. Recently, its importance as a benchmark for RL has been stressed (Recht, 2018), while (Fazel et al., 2018) have shown some non-trivial aspects of this problem. Figure 1 shows the results of our one-dimensional LQG experiments. Both SEPG and SSEPG keep the total performance (Figure 1a) well above the baseline (dashed line) in the early learning iterations, and slightly below when approaching convergence. We believe this latter phenomenon is due to approximations. In Figure 1b we can see the evolution of the mean parameter, where the optimal value is marked with a dotted line. We can see that SSEPG achieves optimality much faster than SEPG. A monotonically improving (MI) algorithm (Pirotta et al., 2013a) is also reported as a reference, and is extremely slow in comparison. This supports our initial claim that unnecessarily strict safety requirements can sacrifice too much in terms of learning speed. In Figure 1c, we show the evolution of policy variance σ_w. We can see that SSPG selects higher values of σ_w. This increased exploration explains the faster convergence of SSPG. Similarly, the MI algorithm performs negligible variance updates, resulting in very slow learning.



(a) (b)

Figure 2: Mountain Car experiment averaged over 5 runs, with 95% confidence intervals

6.2 Mountain Car

In the continuous Mountain Car task[2] a vehicle must reach a high place from the bottom of a valley. The limited power of the engine requires the vehicle to build momentum via oscillations. The agent observes a positive reward only when it reaches the top, which makes the task challenging from the point of view of exploration. We start from a Gaussian policy with $\boldsymbol{v} = [0, 0]$ and $w = \log(0.2)$. With this policy, the agent never reaches the top, so it has no clue about the goal. Only through exploration it can gather enough knowledge to make any progress in the task. We use this setting to show the importance of the exploratory surrogate objective $\mathcal{L}$ introduced in Corollary 4 and used in our algorithms. We compare SSEPG with a variant that employs the naïve gradient $\nabla_w J$ in place of $\nabla_w \mathcal{L}$. We call this variant GREEDY in the figure. To encourage initial exploration, we provide a nominal initial budget $B^0 = 1$ (about 1% of optimal performance) in both cases. Figure 2a shows the performance of the deterministic policies. Only SSEPG learns to collect a positive reward, while GREEDY shows no significant improvement. In Figure 2b we can see that SSEPG is able to increase the variance σ_w enough to allow the agent to reach the goal state at least once. Without the exploratory objective, GREEDY does not have any incentive to increase variance and gets stuck.

7. Conclusions

We provided a definition of economic safety in the context of policy gradient. We extended the existing improvement guarantees for Gaussian policies to the non-trivial adaptive-variance case. Given the special role of this parameter, we proposed a surrogate objective that explicitly takes exploration into account and can be pursued in a safe way. We used our safety framework to model the problem of fine-tuning an existing controller and proposed algorithms to solve it. We empirically evaluated our solutions on simulated control tasks.

Future work should try to extend performance guarantees to a broader class of policies. Another important extension would be to develop more informative metrics in order to better evaluate the costs and benefits of exploration.

2. `gym.openai.com`

References

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. 2016.

Jonathan Baxter and Peter L Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15, 2001.

Marc Peter Deisenroth, Gerhard Neumann, Jan Peters, et al. A survey on policy search for robotics. *Foundations and Trends® in Robotics*, 2(1–2):1–142, 2013.

Maryam Fazel, Rong Ge, Sham M Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for linearized control problems. *arXiv preprint arXiv:1801.05039*, 2018.

Javier Garcıa and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1), 2015.

Mohammad Ghavamzadeh, Marek Petrik, and Yinlam Chow. Safe policy improvement by minimizing robust baseline regret. In *Advances in Neural Information Processing Systems*, 2016.

Manuel Gil, Fady Alajaji, and Tamas Linder. Rényi divergence measures for commonly used univariate continuous distributions. *Information Sciences*, 249, 2013.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *ICML*, volume 2, 2002.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Julie Nutini, Mark Schmidt, Issam Laradji, Michael Friedlander, and Hoyt Koepke. Coordinate descent converges faster with the gauss-southwell rule than random selection. In *International Conference on Machine Learning*, 2015.

Matteo Papini, Matteo Pirotta, and Marcello Restelli. Adaptive batch size for safe policy gradients. In *Advances in Neural Information Processing Systems*, 2017.

Jan Peters and Stefan Schaal. Reinforcement learning of motor skills with policy gradients. *Neural Networks*, 21(4), 2008a.

Jan Peters and Stefan Schaal. Natural actor-critic. *Neurocomputing*, 71(7-9), 2008b.

Mark S Pinsker. Information and information stability of random variables and processes. 1960.

Matteo Pirotta, Marcello Restelli, and Luca Bascetta. Adaptive step-size for policy gradient methods. In *Advances in Neural Information Processing Systems*, pages 1394–1402, 2013a.

Matteo Pirotta, Marcello Restelli, Alessio Pecorino, and Daniele Calandriello. Safe policy iteration. In *International Conference on Machine Learning*, 2013b.

Matteo Pirotta, Marcello Restelli, and Luca Bascetta. Policy gradient in lipschitz markov decision processes. *Machine Learning*, 100(2-3), 2015.

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

Benjamin Recht. A tour of reinforcement learning: The view from continuous control. *arXiv preprint arXiv:1806.09460*, 2018.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, 2015.

David Silver, Julian Schrittwieser, Karen Simonyan, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676), 2017.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, 2000.

Philip Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High confidence policy improvement. In *International Conference on Machine Learning*, 2015.

George Tucker, Surya Bhupatiraju, Shixiang Gu, Richard E Turner, Zoubin Ghahramani, and Sergey Levine. The mirage of action-dependent baselines in reinforcement learning. *arXiv preprint arXiv:1802.10031*, 2018.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4), 1992.

Table of Supplementary Contents

Appendix A. Proofs

In this section we provide formal proofs of all the formal statements made in the main text.

Theorem 2 is a slight adaptation of existing results from (Papini et al., 2017):

Theorem 2 (Adapted from Theorem 3.3 in Papini et al. (2017)) *Assuming w is fixed, the guaranteed performance improvement yielded by (4) is maximized by step size $\alpha^* = \frac{1}{2c}$, where $c = \frac{RM_\phi^2}{(1-\gamma)^2\sigma^2}\left(\frac{|\mathcal{A}|}{\sqrt{2\pi}\sigma} + \frac{\gamma}{2(1-\gamma)}\right)$ and $|\mathcal{A}|$ is the volume of the action space. Moreover, the largest step size guaranteeing $J(\boldsymbol{v}^{t+1}, w) - J(\boldsymbol{v}^t, w) \geq C^t$ is $\overline{\alpha} = \alpha^*(1 + \lambda_{\boldsymbol{v}})$, where $\lambda_{\boldsymbol{v}} = \sqrt{1 - \frac{4cC^t}{\|\nabla_{\boldsymbol{v}} J(\boldsymbol{v}^t, w^t)\|_\infty^2}}$.*

Proof We first prove the following:

$$J(\boldsymbol{v}^{t+1}, w) - J(\boldsymbol{v}^t, w) \geq \alpha \left\|\nabla_{\boldsymbol{v}} J(\boldsymbol{v}^t, w)\right\|_\infty^2 - \alpha^2 c \left\|\nabla_{\boldsymbol{v}} J(\boldsymbol{v}^t, w)\right\|_\infty^2, \tag{7}$$

by adapting Theorem 3.3 in (Papini et al., 2017). To do so, we just set $\boldsymbol{\theta} = \boldsymbol{v}$ and:

$$\Lambda_{kk} = \alpha \quad \text{if} \quad k = \arg\max_i |\nabla_{\boldsymbol{v}_i} J(\boldsymbol{v})| \quad \text{else} \quad 0.$$

The value of the optimal step size α^* and the corresponding guaranteed improvement are then obtained from Corollary 3.4 in (Papini et al., 2017). The largest safe step size $\overline{\alpha}$ is derived from (7) by requiring:

$$\alpha \left\|\nabla_{\boldsymbol{v}} J(\boldsymbol{v}^t)\right\|_\infty^2 - \alpha^2 c \left\|\nabla_{\boldsymbol{v}} J(\boldsymbol{v}^t)\right\|_\infty^2 \geq C^t,$$

solving the inequality for α and taking the largest solution. ■

Theorem 3, instead, does not trivially descend from previous results. To prove it, we first need the following lemmas:

Lemma 8 *The second derivative w.r.t. the standard-deviation parameter w of a policy parametrized as in (1) is bounded as follows:*

$$\left|\frac{\partial^2 \pi_{\boldsymbol{\theta}}(a|s)}{\partial w^2}\right| \leq \frac{\psi}{\sigma_w},$$

where $\psi = \frac{4(\sqrt{7}-2)e^{\frac{\sqrt{7}}{2}-2}}{\sqrt{2\pi}}$.

Proof Let $\chi = \left(\frac{a-\mu_{\boldsymbol{v}}}{\sigma_w}\right)^2$. A first useful fact is:

$$\frac{\partial \chi}{\partial w} = (a-\mu_{\boldsymbol{v}})^2 \frac{\partial \chi}{\partial w} e^{-2w} = -2\chi. \tag{8}$$

Writing the policy as:

$$\pi_{\boldsymbol{\theta}}(a|s) = \frac{1}{\sqrt{2\pi}\sigma_w} e^{-\chi/2},$$

derivatives w.r.t. w can be easily computed as:

$$\frac{\partial \pi_{\boldsymbol{\theta}}(a|s)}{\partial w} = -\pi_{\boldsymbol{\theta}}(a|s) + \pi_{\boldsymbol{\theta}}(a|s)(-\frac{1}{2})(-2\chi) = (\chi-1)\pi_{\boldsymbol{\theta}}(a|s), \tag{9}$$

$$\frac{\partial^2 \pi_{\boldsymbol{\theta}}(a|s)}{\partial w^2} = -2\chi\pi_{\boldsymbol{\theta}}(a|s) + (\chi-1)(\chi-1)\pi_{\boldsymbol{\theta}}(a|s) = (\chi^2-4\chi+1)\pi_{\boldsymbol{\theta}}(a|s). \tag{10}$$

Next, we study the continuous function:

$$f(\chi) \coloneqq (\chi^2-4\chi+1)e^{-\frac{\chi}{2}} = \sqrt{2\pi}\sigma_w \frac{\partial^2 \pi_{\boldsymbol{\theta}}(a|s)}{\partial w^2},$$

constrained, of course, to $\chi \geq 0$. We find that $f(\chi)$ has two stationary points:

- $\chi_1 = 4+\sqrt{7}$, with $f(\chi_1) = 4(\sqrt{7}+2)e^{-\frac{\sqrt{7}}{2}-2} \simeq 0.67$;
- $\chi_2 = 4-\sqrt{7}$, with $f(\chi_2) = -4(\sqrt{7}-2)e^{\frac{\sqrt{7}}{2}-2} \simeq -1.31$, which is also the global minimum.

Moreover, $f(0) = 1$ and $f(\chi) \to 0$ as $\chi \to +\infty$, so $\chi_0 = 0$ is the global maximum in the region of interest. We can then state that $|f(\chi)| \leq |f(\chi_2)| = 4(\sqrt{7}-2)e^{\frac{\sqrt{7}}{2}-2}$, the maximum absolute value that $f(\chi)$ takes in $[0,+\infty]$. Finally:

$$\left|\frac{\partial^2 \pi_{\boldsymbol{\theta}}(a|s)}{\partial w^2}\right| = \frac{|f(\chi)|}{\sqrt{2\pi}\sigma_w} \leq \frac{4(\sqrt{7}-2)e^{\frac{\sqrt{7}}{2}-2}}{\sqrt{2\pi}\sigma_w} \coloneqq \frac{\psi}{\sigma_w}.$$

■

Lemma 9 *Let $w' = w + \Delta w$. Then, by neglecting $o(|\Delta w|^3)$ terms, the difference of the corresponding policies can be bounded as follows:*

$$\pi_{\boldsymbol{v},w'}(a|s) - \pi_{\boldsymbol{v},w}(a|s) \geq \nabla_w \pi_{\boldsymbol{\theta}}(a|s)\Delta w - \frac{\psi \Delta w^2}{2\sigma_w},$$

where ψ is defined as in Lemma 8.

Proof The Taylor expansion of $\pi_{\boldsymbol{v},w'}$ is:

$$\pi_{\boldsymbol{v},w'}(a|s) = \pi_{\boldsymbol{v},w}(a|s) + \nabla_w \pi_{\boldsymbol{\theta}}(a|s)\Delta w + R_1(\Delta w), \tag{11}$$

where $R_1(\Delta w)$ is the remainder of the series, given by:

$$R_1(\Delta w) = \frac{\partial^2 \pi_{\boldsymbol{\theta}}(a|s)}{\partial w^2}\bigg|_{w+\epsilon\Delta w} \frac{\Delta w^2}{2} \qquad \text{for some } \epsilon \in (0,1).$$

We can bound the remainder using Lemma 8:

$$\begin{aligned} R_1(\Delta w) &\geq -\sup_{\epsilon} \left| \frac{\partial^2 \pi_{\boldsymbol{\theta}}(a|s)}{\partial w^2}\bigg|_{w+\epsilon\Delta w} \frac{\Delta w^2}{2} \right| \\ &\geq -\frac{\psi \Delta w^2}{\inf_\epsilon e^{w+\epsilon\Delta w}} \geq -\frac{\psi \Delta w^2}{e^w \inf_\epsilon e^{\epsilon \Delta w}} \\ &\geq -\frac{\psi \Delta w^2}{e^w e^{-|\Delta w|}} \geq -\frac{\psi \Delta w^2 e^{|\Delta w|}}{e^w} \\ &= -\frac{\psi \Delta w^2}{e^w} + o(|\Delta w|^3), \end{aligned} \tag{12}$$

where the final equality is obtained by expanding $e^{|\Delta w|}$ in Taylor's series. The lemma follows from (11) and (12) by rearranging terms. ■

Lemma 10 *Let $w' = w + \Delta w$. Then, by neglecting $o(\Delta w^3)$ terms, the squared Chebyshev distance among the corresponding policies can be bounded as follows:*

$$\left\| \pi_{\boldsymbol{v},w'} - \pi_{\boldsymbol{v},w} \right\|_\infty^2 \leq 2\Delta w^2.$$

Proof We have that:

$$\begin{aligned} &\left\| \pi_{\boldsymbol{v},w'} - \pi_{\boldsymbol{v},w} \right\|_\infty^2 \leq \sup_{s\in\mathcal{S}} \left\| \pi_{\boldsymbol{v},w'}(\cdot|s) - \pi_{\boldsymbol{v},w}(\cdot|s) \right\|_1^2 \\ &\leq \sup_{s\in\mathcal{S}} \left(2H(\pi_{\boldsymbol{v},w'}(\cdot|s) \| \pi_{\boldsymbol{v},w}(\cdot|s)) \right) && (13) \\ &= 2\left(\log \frac{\sigma_w}{\sigma_{w'}} + \frac{\sigma_{w'}^2}{2\sigma_w^2 - \frac{1}{2}} \right) && (14) \\ &= -2\Delta w + e^{2\Delta w} - 1 && (15) \\ &= 2\Delta w^2 + o(\Delta w^3), \end{aligned}$$

where $H(\cdot\|\cdot)$ is the Kullback-Leibler (KL) divergence, (13) is from Pinsker's inequality (Pinsker, 1960), (14) is from the expression of the KL divergence for Gaussian distributions with equal mean (Gil et al., 2013), and (15) is from expanding $e^{\Delta w}$ in Taylor's series up to the second order. Note how the sup on states can be dropped since σ is state-independent. ■

Theorem 3 *Assuming $\boldsymbol{v}$ is fixed and $o(|\Delta w|^3)$ terms can be neglected,*[3] *the guaranteed performance improvement yielded by (4) is maximized by step size $\beta^* = \frac{1}{2d}$, where $d =$*

3. This approximation is not critical since the steps produced by safe algorithms tend to be very small.

$\frac{R}{(1-\gamma)^2}\left(\frac{\psi|\mathcal{A}|}{2\sigma_w}+\frac{\gamma}{1-\gamma}\right)$ *and* $\psi = 4(\sqrt{7}-2)\exp(\sqrt{7}/2-2)/\sqrt{2\pi} \simeq 0.524$. *Moreover, the largest step size guaranteeing* $J(\boldsymbol{v}, w^{t+1}) - J(\boldsymbol{v}, w^t) \geq C^t$ *is* $\overline{\beta} = \beta^*(1+\lambda_w)$, *where* $\lambda_w = \sqrt{1 - \frac{4dC^t}{\|\nabla_w J(\boldsymbol{v}^t, w^t)\|_\infty^2}}$.

Proof We first prove the following:

$$J(\boldsymbol{v}, w^{t+1}) - J(\boldsymbol{v}, w^t) \geq \beta \nabla_w J(\boldsymbol{v}, w^t)^2 - \beta^2 d \nabla_w J(\boldsymbol{v}, w^t)^2. \tag{16}$$

To do so, we start from Lemma 3.1 in (Pirotta et al., 2013a):

$$\begin{aligned} J(\boldsymbol{\theta}') - J(\boldsymbol{\theta}) \geq \frac{1}{1-\gamma} \int_{\mathcal{S}} d_\rho^{\boldsymbol{\theta}}(s) \int_{\mathcal{A}} (\pi_{\boldsymbol{\theta}'}(a|s) - \pi_{\boldsymbol{\theta}}(a|s)) Q^{\boldsymbol{\theta}}(s,a) \,\mathrm{d}a \,\mathrm{d}s \\ - \frac{\gamma}{2(1-\gamma)^2} \|\pi_{\boldsymbol{\theta}'} - \pi_{\boldsymbol{\theta}}\|_\infty^2 \left\|Q^{\boldsymbol{\theta}}\right\|_\infty, \end{aligned} \tag{17}$$

where $d_\rho^{\boldsymbol{\theta}}(s) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \Pr(s_t = s|\pi_{\boldsymbol{\theta}}, \rho)$ is the discounted future-state distribution and $Q^{\boldsymbol{\theta}}(s,a) = \mathcal{R}(s,a) + \gamma \int_{\mathcal{S}} \mathcal{P}(s'|s,a) \int_{\mathcal{A}} \pi_{\boldsymbol{\theta}}(a|s) Q^{\boldsymbol{\theta}}(s',a') \,\mathrm{d}a' \,\mathrm{d}s'$ is the action-value function defined recursively by Bellman's equation. In our case of interest, $\boldsymbol{\theta} = [\boldsymbol{v} \mid w]$ and $\boldsymbol{\theta}' = [\boldsymbol{v}|w']$. We also need the Policy Gradient Theorem (PGT) (Sutton et al., 2000), which we specialize to $\nabla_w J$:

$$\nabla_w J(\boldsymbol{\theta}) = \frac{1}{1-\gamma} \int_{\mathcal{S}} d_\rho^{\boldsymbol{\theta}}(s) \int_{\mathcal{A}} \nabla_w \pi_{\boldsymbol{\theta}}(a|s) Q^{\boldsymbol{\theta}}(s,a) \,\mathrm{d}a \,\mathrm{d}s.$$

Plugging the results of Lemmas 9 and 10 into (17) we get:

$$\begin{aligned} J(\boldsymbol{\theta}') - J(\boldsymbol{\theta}) \geq \frac{1}{1-\gamma} \int_{\mathcal{S}} d_\rho^{\boldsymbol{\theta}}(s) \int_{\mathcal{A}} \left(\nabla_w \pi_{\boldsymbol{\theta}}(a|s) \Delta w - \frac{\psi \Delta w^2}{2\sigma_w} \right) Q^{\boldsymbol{\theta}}(s,a) \,\mathrm{d}a \,\mathrm{d}s \\ - \frac{\gamma}{2(1-\gamma)^2} 2\Delta w^2 \left\|Q^{\boldsymbol{\theta}}\right\|_\infty, \end{aligned} \tag{18}$$

where $o(\Delta w^3)$ terms, as stated, are neglected. By applying the PGT:

$$\begin{aligned} J(\boldsymbol{\theta}') - J(\boldsymbol{\theta}) &\geq \Delta w \nabla_w J(\boldsymbol{\theta}) - \frac{1}{1-\gamma} \int_{\mathcal{S}} d_\rho^{\boldsymbol{\theta}}(s) \int_{\mathcal{A}} \frac{\psi \Delta w^2}{2\sigma_w} Q^{\boldsymbol{\theta}}(s,a) \,\mathrm{d}a \,\mathrm{d}s \\ &\quad - \frac{\gamma}{2(1-\gamma)^2} 2\Delta w^2 \left\|Q^{\boldsymbol{\theta}}\right\|_\infty \\ &\geq \Delta w \nabla_w J(\boldsymbol{\theta}) - \frac{\psi \Delta w^2}{2\sigma_w (1-\gamma)} \left| \int_{\mathcal{A}} Q^{\boldsymbol{\theta}}(s,a) \,\mathrm{d}a \right| \\ &\quad - \frac{\gamma}{2(1-\gamma)^2} 2\Delta w^2 \left\|Q^{\boldsymbol{\theta}}\right\|_\infty. \end{aligned}$$

Next, following (Pirotta et al., 2013a), we upper bound the integral and the infinite norm of $Q^{\boldsymbol{\theta}}$ with $\frac{|\mathcal{A}|R}{1-\gamma}$ and $\frac{R}{1-\gamma}$, respectively, obtaining:

$$J(\boldsymbol{\theta}') - J(\boldsymbol{\theta}) \geq \Delta w \nabla_w J(\boldsymbol{\theta}) - \Delta w^2 \frac{R}{(1-\gamma)^2} \left(\frac{\psi|\mathcal{A}|}{2\sigma_w} + \frac{\gamma}{1-\gamma} \right). \tag{19}$$

Now, since $\Delta w = \beta \nabla_w J(\boldsymbol{\theta})$,

$$J(\boldsymbol{\theta}') - J(\boldsymbol{\theta}) \geq \beta \nabla_w J(\boldsymbol{\theta})^2 - \beta^2 \nabla_w J(\boldsymbol{\theta})^2 \frac{R}{(1-\gamma)^2} \left(\frac{\psi|\mathcal{A}|}{2\sigma_w} + \frac{\gamma}{1-\gamma} \right).$$

The proof is completed by renaming $w \leftarrow w^t$ and $w' \leftarrow w^{t+1}$.

We can now compute the optimal step size β^*. The right-hand side of (16) is a degree-2 polynomial in β. By nullifying its first derivative:

$$\nabla_w J(\boldsymbol{v}, w^t)^2 - 2\beta^* d \nabla_w J(\boldsymbol{v}, w^t)^2 = 0,$$

we get $\beta^* = \frac{1}{2d}$. Second derivative is $(-\nabla_w J(\boldsymbol{\theta})^2) < 0$, showing that we found a maximum. In fact, we are taking the vertex of a concave parabola. The corresponding performance improvement is obtained by substituting β^* back into (16). Finally, to obtain the largest safe step size $\overline{\beta}$, given (16), it is enough to require:

$$\beta \nabla_w J(\boldsymbol{v}, w^t)^2 - \beta^2 d \nabla_w J(\boldsymbol{v}, w^t)^2 \geq C^t,$$

solve the inequality for β, and take the largest solution. ■

Corollary 11 *The optimal step size α^* from Theorem 2, guarantees:*

$$J(\boldsymbol{v}^{t+1}, w) - J(\boldsymbol{v}^t, w) \geq \frac{\left\|\nabla_{\boldsymbol{v}} J(\boldsymbol{v}^t, w)\right\|_2^2}{4mc} \coloneqq \mathcal{L}(\boldsymbol{v}^t, w),$$

where c is from Theorem 2 and m is the size of $\boldsymbol{v}$.

Proof Plugging the optimal step size α^* from Theorem 2 into Equation (7) we obtain:

$$J(\boldsymbol{v}^{t+1}, w) - J(\boldsymbol{v}^t, w) \geq \frac{\left\|\nabla_{\boldsymbol{\theta}} J(\boldsymbol{v}^t, w)\right\|_\infty^2}{4c}. \tag{20}$$

We obtain the looser, but differentiable bound of the corollary by applying the following norm inequality, true for any m-dimensional vector $\mathbf{x}$:

$$\|\mathbf{x}\|_\infty^2 = \left(\max_i\{|x_i|\}\right)^2 = \frac{1}{m}\sum_{j=1}^m \left(\max_i\{|x_i|\}\right)^2 \geq \frac{1}{m}\sum_{j=1}^m x_j^2 = \frac{1}{m}\|\mathbf{x}\|_2^2 .$$

■

Theorem 5 *Assuming $\boldsymbol{v}$ is fixed and $o(|\Delta w|^3)$ terms can be neglected, the guaranteed performance improvement yielded by $w^{t+1} \leftarrow w^t + \eta \nabla_w \mathcal{L}(\boldsymbol{v}, w^t)$ is maximized by step size $\eta^* = \frac{\nabla_w J(\boldsymbol{v}, w^t)}{2d \nabla_w \mathcal{L}(\boldsymbol{v}, w^t)}$. Moreover, the largest step size guaranteeing $J(\boldsymbol{v}, w^{t+1}) - J(\boldsymbol{v}, w^t) \geq C^t$ is $\overline{\eta} = \eta^* + |\eta^*|(1 + \lambda_w)$, where d and λ_w are from Theorem 3.*

Proof Simply replace η with $\frac{\nabla_w \mathcal{L}(\boldsymbol{v}, w^t)}{\nabla_w J(\boldsymbol{v}, w^t)}\beta$ in all the statements to obtain Theorem 3. More intuitively, it suffices to consider steps in the direction of $\nabla_w \mathcal{L}$ instead of $\nabla_w J$. Each step in $\nabla_w \mathcal{L}$ has a corresponding (possibly negative) step in $\nabla_w J$, for which Theorem 3 applies. Geometrically, we are performing a projection between one-dimensional vector spaces (note that $\nabla_w J$ and $\nabla_w \mathcal{L}$ are scalars). The absolute value in the expression of $\overline{\eta}$ is necessary

because η^* may be negative. Again, we neglect the zero-gradient case since it coincides with convergence. ■

Figure 3 provides a geometric intuition of Theorem 5.

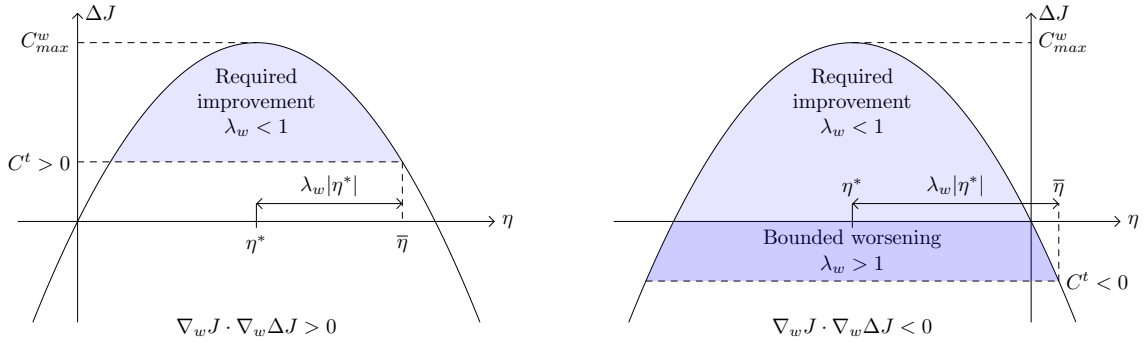


Figure 3: Guaranteed improvement of an exploratory variance-parameter update, i.e., $w \leftarrow w \oplus \eta \nabla_w \mathcal{L}$. The guaranteed improvement is quadratic in step size η. A safety requirement $C^t > 0$ corresponds to a required improvement, a negative one to a bounded worsening. A requirement $C^t > C^w_{\max}$ can never be satisfied. The requirement determines the acceptable range of step sizes. When $\nabla_w J$ and $\nabla_w J$ have equal sign, we can safely use a positive step size, whatever the safety requirement. When $\nabla_w J$ and $\nabla_w J$ have opposite sign, we can only increase $\mathcal{L}$ at the price of reducing performance. We can still use a positive step size only in the bounded worsening case, i.e., if the safety requirement allows performance losses. In the other case, a negative step size is enough to guarantee some improvement.

Theorem 12 *Constraint (5) is equivalent to the following:*

$$J(\boldsymbol{\theta}^{t+1}) - J(\boldsymbol{\theta}^t) \geq -B^t \quad \textit{for all } t.$$

Proof From (6), the above constraint is equivalent to:

$$B^{t+1} \geq 0 \quad \text{for all } t. \tag{21}$$

By expanding Definition 6:

$$B^{t+1} = \sum_{i=0}^{t} \Big(J(\boldsymbol{\theta}^{i+1}) - J(\boldsymbol{\theta}^i) \Big) \tag{22}$$

$$= J(\boldsymbol{\theta}^{t+1}) - J(\boldsymbol{\theta}^0), \tag{23}$$

since the sum in (22) telescopes. Now, (21) and (23) imply (5). ■

Appendix B. Gradient Estimation

In this section, we show a way to estimate the gradients used in our algorithms.

B.1 Policy gradient

In practice, $\nabla_{\boldsymbol{\theta}} J$ can be estimated from a batch of trajectories $\mathcal{D}_N = \{\tau_1, \dots, \tau_N\}$. The REINFORCE [4] (Williams, 1992; Baxter and Bartlett, 2001) algorithm provides an unbiased estimator:

$$\widehat{\nabla}^N_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \frac{1}{N} \sum_{n=1}^{N} \sum_{h=0}^{H-1} \left(\sum_{i=0}^{h} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_i^n | s_i^n) \right) \left(\gamma^h \mathcal{R}(s_h^n, a_h^n) - b \right), \tag{24}$$

where b is a baseline used to reduce variance. Any baseline that does not depend on actions preserves the unbiasedness of the estimator.[5] We adopt the variance-minimizing baselines provided by (Peters and Schaal, 2008a).

B.2 Gradient of the surrogate objective

We first note that, from the chain rule:

$$\nabla_w \mathcal{L}(\boldsymbol{v}, w) = e^w \nabla_\sigma \mathcal{L}(\boldsymbol{v}, e^w). \tag{25}$$

By definition:

$$\mathcal{L}(\boldsymbol{\theta}) \coloneqq \frac{\|\nabla_{\boldsymbol{v}} J(\boldsymbol{\theta})\|_2^2}{4mc} \tag{26}$$

$$= \frac{1}{2m} \alpha^* \|\nabla_{\boldsymbol{v}} J(\boldsymbol{\theta})\|_2^2, \tag{27}$$

where α^* is the optimal step size from (Pirotta et al., 2013a):

$$\alpha^* = \frac{c_1 \sigma^3}{m(c_2 \sigma + c_3)}, \tag{28}$$

$$c_1 = (1-\gamma)^3 \sqrt{2\pi}, \tag{29}$$

$$c_2 = \gamma \sqrt{2\pi} R M_\phi^2, \tag{30}$$

$$c_3 = 2(1-\gamma)|\mathcal{A}| R M_\phi^2, \tag{31}$$

which is also a function of σ. The gradient w.r.t. σ is then:

$$\nabla_\sigma \mathcal{L}(\boldsymbol{\theta}) = \nabla_\sigma \left(\frac{1}{2m} \alpha^* \|\nabla_{\boldsymbol{v}} J(\boldsymbol{\theta})\|_2^2 \right) = \frac{1}{2m} \nabla_\sigma \alpha^* \|\nabla_{\boldsymbol{v}} J(\boldsymbol{\theta})\|_2^2 + \frac{1}{2m} \alpha^* \nabla_\sigma \|\nabla_{\boldsymbol{v}} J(\boldsymbol{\theta})\|_2^2. \tag{32}$$

Policy gradient $\nabla_{\boldsymbol{v}} J$ can be estimated with REINFORCE as usual. The terms that need further development are:

$$\nabla_\sigma \alpha^* = \frac{2c_1 c_2 \sigma + 3 c_1 c_3}{m(c_2 \sigma + c_3)^2} \sigma^2, \tag{33}$$

$$\nabla_\sigma \|\nabla_{\boldsymbol{v}} J(\boldsymbol{\theta})\|_2^2 = 2 \nabla_{\boldsymbol{v}} J(\boldsymbol{\theta})^T \nabla_\sigma \nabla_{\boldsymbol{v}} J(\boldsymbol{\theta}). \tag{34}$$

4. The algorithm we present here is actually a refinement of REINFORCE, originally called G(PO)MDP (Baxter and Bartlett, 2001).
5. Action-dependent baselines are possible, but their usefulness is still controversial (Tucker et al., 2018).

To compute (34), we still need $\nabla_\sigma \nabla_{\boldsymbol{v}} J$. First note that:

$$\begin{aligned}\nabla_\sigma \nabla_{\boldsymbol{v}} \log p_{\boldsymbol{\theta}}(\tau) &= \sum_{t=0}^{H} \nabla_\sigma \nabla_{\boldsymbol{v}} \log \pi_{\boldsymbol{\theta}}(a \mid s) = \sum_{t=0}^{H} \nabla_\sigma \frac{a - \mu_{\boldsymbol{\theta}}(s)}{\sigma_{\boldsymbol{\theta}}^2} = -\frac{2}{\sigma} \sum_{t=0}^{H} \frac{a - \mu_{\boldsymbol{\theta}}}{\sigma_{\boldsymbol{\theta}}^2} \\ &= -\frac{2}{\sigma} \nabla_{\boldsymbol{v}} \log p_{\boldsymbol{\theta}}(\tau), \end{aligned} \tag{35}$$

Using the log-trick:

$$\begin{aligned}\nabla_\sigma \nabla_{\boldsymbol{v}} J(\boldsymbol{\theta}) &= \nabla_\sigma \mathop{\mathbb{E}}_{\tau \sim p_{\boldsymbol{\theta}}} [\nabla_{\boldsymbol{v}} \log p_{\boldsymbol{\theta}}(\tau) R(\tau)] \\ &= \mathop{\mathbb{E}}_{\tau \sim p_{\boldsymbol{\theta}}} [\nabla_\sigma \log p_{\boldsymbol{\theta}}(\tau) \nabla_{\boldsymbol{v}} \log p_{\boldsymbol{\theta}}(\tau) R(\tau)] + \mathop{\mathbb{E}}_{\tau \sim p_{\boldsymbol{\theta}}} [\nabla_\sigma \nabla_{\boldsymbol{v}} \log p_{\boldsymbol{\theta}}(\tau) R(\tau)] \\ &= \mathop{\mathbb{E}}_{\tau \sim p_{\boldsymbol{\theta}}} [\nabla_\sigma \log p_{\boldsymbol{\theta}}(\tau) \nabla_{\boldsymbol{v}} \log p_{\boldsymbol{\theta}}(\tau) R(\tau)] - \frac{2}{\sigma} \nabla_{\boldsymbol{v}} J(\boldsymbol{\theta}) \\ &= h(\boldsymbol{\theta}) - \frac{2}{\sigma} \nabla_{\boldsymbol{v}} J(\boldsymbol{\theta}), \end{aligned} \tag{36}$$

where $h(\boldsymbol{\theta}) \coloneqq \mathbb{E}_{\tau \sim p_{\boldsymbol{\theta}}} [\nabla_\sigma \log p_{\boldsymbol{\theta}}(\tau) \nabla_{\boldsymbol{v}} \log p_{\boldsymbol{\theta}}(\tau) R(\tau)]$. We have reduced the problem of estimating $\nabla_w \mathcal{L}(\boldsymbol{\theta})$ to the one of estimating $h(\boldsymbol{\theta})$.

B.3 Estimating $h(\boldsymbol{\theta})$

We now propose an unbiased estimator for $h(\boldsymbol{\theta}) = \mathbb{E}_{\tau \sim p_{\boldsymbol{\theta}}} [\nabla_\sigma \log p_{\boldsymbol{\theta}}(\tau) \nabla_{\boldsymbol{v}} \log p_{\boldsymbol{\theta}}(\tau) R(\tau)]$:

Theorem 13 *An unbiased estimator for $h(\boldsymbol{\theta})$ is:*

$$\hat{h}(\boldsymbol{\theta}) = \sum_{t=0}^{H} \int_T p_{\boldsymbol{\theta}}(\tau_{0:H}) \gamma^t r_t \left(\sum_{k'=0}^{t} \nabla_\sigma \log \pi_{k'} \right) \left(\sum_{h'=0}^{t} \nabla_{\boldsymbol{v}} \log \pi_{h'} \right) \mathrm{d}\tau. \tag{37}$$

Proof Let's abbreviate $\pi_k = \pi_{\boldsymbol{\theta}}(a_k \mid s_k)$. We can split $h(\boldsymbol{\theta})$ into the sum of four components:

$$\begin{aligned} h(\boldsymbol{\theta}) &= \int_T p_{\boldsymbol{\theta}}(\tau_{0:H}) \nabla_\sigma \log p_{\boldsymbol{\theta}}(\tau) \nabla_{\boldsymbol{v}} \log p_{\boldsymbol{\theta}}(\tau) R(\tau) \, \mathrm{d}\tau = \\ &= \int_T p_{\boldsymbol{\theta}}(\tau_{0:H}) \left(\sum_{t=0}^{H} \gamma^t r_t \right) \left(\sum_{k=0}^{H} \nabla_\sigma \log \pi_k \right) \left(\sum_{h=0}^{H} \nabla_{\boldsymbol{v}} \log \pi_h \right) \mathrm{d}\tau = \\ &= \sum_{t=0}^{H} \int_T p_{\boldsymbol{\theta}}(\tau_{0:H}) \gamma^t r_t \left(\sum_{k'=0}^{t} \nabla_\sigma \log \pi_{k'} \right) \left(\sum_{h'=0}^{t} \nabla_{\boldsymbol{v}} \log \pi_{h'} \right) \mathrm{d}\tau \qquad (38) \\ &+ \sum_{t=0}^{H} \int_T p_{\boldsymbol{\theta}}(\tau_{0:H}) \gamma^t r_t \left(\sum_{k'=0}^{t} \nabla_\sigma \log \pi_{k'} \right) \left(\sum_{h''=t+1}^{H} \nabla_{\boldsymbol{v}} \log \pi_{h''} \right) \mathrm{d}\tau \qquad (39) \\ &+ \sum_{t=0}^{H} \int_T p_{\boldsymbol{\theta}}(\tau_{0:H}) \gamma^t r_t \left(\sum_{k''=t+1}^{H} \nabla_\sigma \log \pi_{k''} \right) \left(\sum_{h'=0}^{t} \nabla_{\boldsymbol{v}} \log \pi_{h'} \right) \mathrm{d}\tau \qquad (40) \\ &+ \sum_{t=0}^{H} \int_T p_{\boldsymbol{\theta}}(\tau_{0:H}) \gamma^t r_t \left(\sum_{k''=t+1}^{H} \nabla_\sigma \log \pi_{k''} \right) \left(\sum_{h''=t+1}^{H} \nabla_{\boldsymbol{v}} \log \pi_{h''} \right) \mathrm{d}\tau. \qquad (41) \end{aligned}$$

Next, we show that (39), (40) and (41) all evaluate to 0:

$$
\begin{aligned}
(39) &= \sum_{t=0}^{H} \int_T p_{\boldsymbol{\theta}}(\tau_{0:H})\, \gamma^t r_t \left(\sum_{k'=0}^{t} \nabla_\sigma \log \pi_{k'}\right) \left(\sum_{h''=t+1}^{H} \nabla_{\boldsymbol{v}} \log \pi_{h''}\right) \mathrm{d}\tau \\
&= \sum_{t=0}^{H} \int_T p_{\boldsymbol{\theta}}(\tau_{0:t})\, \gamma^t r_t \left(\sum_{k'=0}^{t} \nabla_\sigma \log \pi_{k'}\right) \mathrm{d}\tau \int_T p_{\boldsymbol{\theta}}(\tau_{t+1:H}) \left(\sum_{h''=t+1}^{H} \nabla_{\boldsymbol{v}} \log \pi_{h''}\right) \mathrm{d}\tau \\
&= \sum_{t=0}^{H} \int_T p_{\boldsymbol{\theta}}(\tau_{0:t})\, \gamma^t r_t \left(\sum_{k'=0}^{t} \nabla_\sigma \log \pi_{k'}\right) \mathrm{d}\tau \int_T p_{\boldsymbol{\theta}}(\tau_{t+1:H})\, \nabla_{\boldsymbol{v}} \log p_{\boldsymbol{\theta}}(\tau_{t+1:H})\, \mathrm{d}\tau \\
&= \sum_{t=0}^{H} \int_T p_{\boldsymbol{\theta}}(\tau_{0:t})\, \gamma^t r_t \left(\sum_{k'=0}^{t} \nabla_\sigma \log \pi_{k'}\right) \mathrm{d}\tau \int_T \nabla_{\boldsymbol{v}} p_{\boldsymbol{\theta}}(\tau_{t+1:H})\, \mathrm{d}\tau \\
&= \sum_{t=0}^{H} \int_T p_{\boldsymbol{\theta}}(\tau_{0:t})\, \gamma^t r_t \left(\sum_{k'=0}^{t} \nabla_\sigma \log \pi_{k'}\right) \mathrm{d}\tau \nabla_{\boldsymbol{v}} \int_T p_{\boldsymbol{\theta}}(\tau_{t+1:H})\, \mathrm{d}\tau \\
&= 0.
\end{aligned}
$$

Analogously, we can say that $(40) = 0$. Finally:

$$
\begin{aligned}
(41) &= \sum_{t=0}^{H} \int_T p_{\boldsymbol{\theta}}(\tau_{0:H})\, \gamma^t r_t \left(\sum_{k''=t+1}^{H} \nabla_\sigma \log \pi_{k''}\right) \left(\sum_{h''=t+1}^{H} \nabla_{\boldsymbol{v}} \log \pi_{h''}\right) \mathrm{d}\tau \\
&= \sum_{t=0}^{H} \int_T p_{\boldsymbol{\theta}}(\tau_{0:t})\, \gamma^t r_t\, \mathrm{d}\tau \int_T p_{\boldsymbol{\theta}}(\tau_{t+1:H})\, \nabla_\sigma \log p_{\boldsymbol{\theta}}(\tau_{t+1:H})\, \nabla_{\boldsymbol{v}} \log p_{\boldsymbol{\theta}}(\tau_{t+1:H})\, \mathrm{d}\tau \\
&= \sum_{t=0}^{H} \int_T p_{\boldsymbol{\theta}}(\tau_{0:t})\, \gamma^t r_t\, \mathrm{d}\tau \int_T \left(\nabla_\sigma \nabla_{\boldsymbol{v}} p_{\boldsymbol{\theta}}(\tau_{t+1:H}) - \nabla_\sigma \nabla_{\boldsymbol{v}} \log p_{\boldsymbol{\theta}}(\tau_{t+1:H})\right) \mathrm{d}\tau \\
&= \sum_{t=0}^{H} \int_T p_{\boldsymbol{\theta}}(\tau_{0:t})\, \gamma^t r_t\, \mathrm{d}\tau \int_T \nabla_\sigma \nabla_{\boldsymbol{v}} p_{\boldsymbol{\theta}}(\tau_{t+1:H})\, \mathrm{d}\tau \\
&\qquad - \sum_{t=0}^{H} \int_T p_{\boldsymbol{\theta}}(\tau_{0:t})\, \gamma^t r_t\, \mathrm{d}\tau \int_T \nabla_\sigma \nabla_{\boldsymbol{v}} \log p_{\boldsymbol{\theta}}(\tau_{t+1:H})\, \mathrm{d}\tau \\
&= \sum_{t=0}^{H} \int_T p_{\boldsymbol{\theta}}(\tau_{0:t})\, \gamma^t r_t d\tau \nabla_\sigma \nabla_{\boldsymbol{v}} \int_T p_{\boldsymbol{\theta}}(\tau_{t+1:H})\, \mathrm{d}\tau \\
&\qquad - \sum_{t=0}^{H} \int_T p_{\boldsymbol{\theta}}(\tau_{0:t})\, \gamma^t r_t\, \mathrm{d}\tau \int_T -\frac{2}{\sigma} p_{\boldsymbol{\theta}}(\tau_{t+1:H}) \nabla_{\boldsymbol{v}} \log p_{\boldsymbol{\theta}}(\tau_{t+1:H})\, \mathrm{d}\tau \\
&= \frac{2}{\sigma} \sum_{t=0}^{H} \int_T p_{\boldsymbol{\theta}}(\tau_{0:t})\, \gamma^t r_t\, \mathrm{d}\tau \int_T p_{\boldsymbol{\theta}}(\tau) \nabla_{\boldsymbol{v}} \log p_{\boldsymbol{\theta}}(\tau_{t+1:H})\, \mathrm{d}\tau \\
&= \frac{2}{\sigma} \sum_{t=0}^{H} \int_T p_{\boldsymbol{\theta}}(\tau_{0:t})\, \gamma^t r_t\, \mathrm{d}\tau \int_T \nabla_{\boldsymbol{v}} p_{\boldsymbol{\theta}}(\tau_{t+1:H})\, \mathrm{d}\tau = 0.
\end{aligned}
$$

■

B.4 A baseline for $\hat{h}(\boldsymbol{\theta})$

As in REINFORCE, we can introduce a baseline to reduce the variance of the estimator. Let

$$\hat{h}_t(\boldsymbol{\theta}) = \mathbb{E}\left[\underbrace{\left(\sum_{k=0}^{t}\nabla_\theta \log \pi_k\right)}_{G_t}\underbrace{\left(\sum_{k=0}^{t}\nabla_\sigma \log \pi_k\right)}_{H_t}\left(\underbrace{\gamma^t r_t}_{F_t} - b_t\right)\right], \tag{42}$$

where b_t is a generic baseline that is independent from actions a_k. Any baseline $b_t = \tilde{b}_t\left(\frac{G_t+H_t}{G_tH_t}\right)$ will keep the estimator unbiased for any value of $\tilde{b}_t$:

$$\begin{aligned}
E\left[G_tH_t\left(F_t - b_t\right)\right] &= E\left[G_tH_tF_t\right] - E\left[G_tH_tb_t\right] \\
&= E\left[G_tH_tF_t\right] - E\left[G_tH_t\tilde{b}_t\left(\frac{G_t+H_t}{G_tH_t}\right)\right] \\
&= E\left[G_tH_tF_t\right] - \tilde{b}_tE\left[G_t\right] - \tilde{b}_t\left[H_t\right] \\
&= E\left[G_tH_tF_t\right].
\end{aligned}$$

We choose $\tilde{b}_t$ as to minimize the variance of $\hat{h}_t$:

$$\begin{aligned}
Var[\hat{h}_t] &= E\left[\hat{h}_t^{\,2}\right] - E\left[\hat{h}_t\right]^2 \\
&= E\left[G_t^2H_t^2\left(F_t^2 - 2F_t\tilde{b}_t\frac{G_t+H_t}{G_tH_t} + \tilde{b}_t^{\,2}\frac{(G_t+H_t)^2}{G_t^2H_t^2}\right)\right] - E\left[G_tH_tF_t\right]^2 \\
&= E\left[G_t^2H_t^2F_t^2\right] - 2\tilde{b}_tE\left[G_tH_tF_t\left(G_t+H_t\right)\right] + \tilde{b}_t^{\,2}E\left[\left(G_t+H_t\right)^2\right] - E\left[G_tH_tF_t\right]^2.
\end{aligned}$$

Setting the gradient to zero yields:

$$b_t^* = \arg\min_{\tilde{b}_t} Var\left[\hat{h}_t\right] = \frac{E\left[G_tH_tF_t\left(G_t+H_t\right)\right]}{E\left[\left(G_t+H_t\right)^2\right]}.$$

Hence the estimator has minimum variance with baseline:

$$b_t = b_t^*\frac{G_t+H_t}{G_tH_t} = \frac{G_tH_tF_t\left(G_t+H_t\right)}{\left(G_t+H_t\right)^2}\frac{G_t+H_t}{G_tH_t},$$

which can be estimated from samples as in (Peters and Schaal, 2008b).

Appendix C. Approximate Framework

In practice, the exact gradients needed to perform policy updates are typically not available and must be estimated from batches of trajectories. In this section, we show how gradient approximation affects safety. When the gradients are estimated from samples, we can derive high-probability variants of the performance improvement bounds of Section 3, as done in (Pirotta et al., 2013a; Papini et al., 2017). To do so, we need to bound the gradient estimation error in high probability, and this must be done for each gradient that we estimate $(\nabla_{\boldsymbol{v}}J, \nabla_w J, \nabla_w \mathcal{L})$. To keep things simple, we make the following assumption: For each

gradient of interest $\nabla_{\boldsymbol{\theta}} F$ there is an estimator $\widehat{\nabla}^N_{\boldsymbol{\theta}} F$ using batches of N samples and an $\epsilon > 0$ such that, for $i = 1, \ldots, m$:

$$\begin{aligned} &\left|\nabla_{\theta_i} F - \widehat{\nabla}_{\theta_i} F\right| \leq \epsilon \quad \text{with probability at least } (1-\delta), \\ &\left\|\widehat{\nabla}_{\boldsymbol{\theta}} F\right\|_\infty > \epsilon \quad \text{always.} \end{aligned} \tag{43}$$

Statement (43) can be used as a stopping condition in iterative algorithms. In fact, when no component of the (estimated) gradient is greater than noise ϵ, no reliable coordinate-ascent policy update can be performed.

It remains to characterize the estimation error ϵ in a meaningful way. In this work, we choose to employ t-based confidence intervals:

$$\epsilon \coloneqq \max_i \left\{ t_{\delta/2, N-1} \sqrt{\frac{S^2_N \left[\widehat{\nabla}_{\theta_i} F\right]}{N}} \right\}, \tag{44}$$

where S^2_N is the sample variance and $t_{\delta/2,N-1}$ is the $(1 - \delta/2)$-quantile of a Student's t-distribution with $N-1$ degrees of freedom. This requires a (false) Gaussianity assumption. We do not think this is critical, since the bounds of Section 3 are made very conservative by problem-dependent constants anyway. More rigorous (and conservative) error characterizations can be found in (Papini et al., 2017). In this section, we discuss approximation issues in more detail. Once the gradient estimation error has been characterized as in (44), we can use Assumption C to define the following lower-corrected and upper-corrected gradient norms:

$$\left\|\widehat{\nabla}^N_{\boldsymbol{\theta}} F\right\|_- = \left\|\widehat{\nabla}^N_{\boldsymbol{\theta}} F\right\|_\infty - \epsilon, \tag{45}$$

$$\left\|\widehat{\nabla}^N_{\boldsymbol{\theta}} F\right\|_+ = \left\|\widehat{\nabla}^N_{\boldsymbol{\theta}} F\right\|_\infty + \epsilon. \tag{46}$$

To obtain high-probability variants of the Theorems from Section 3, one must replace the true gradient norms with the corrected norms. The choice of lower-corrected versus upper-corrected must be made in order to preserve the inequalities. Note that Assumption C also allows us to safely take squares.

Theorem 14 *Assuming w is fixed, under Assumption C, update rule (4) guarantees, with probability $1-\delta$:*

$$\begin{aligned} J(\boldsymbol{v}^{t+1}, w) - J(\boldsymbol{v}^t, w) \geq \alpha \left\|\nabla_{\boldsymbol{v}} J(\boldsymbol{v}^t, w)\right\|^2_- - \\ \alpha^2 c \left\|\nabla_{\boldsymbol{v}} J(\boldsymbol{v}^t, w)\right\|^2_+, \end{aligned} \tag{47}$$

where $c = \frac{RM^2_\phi}{(1-\gamma)^2\sigma^2}\left(\frac{|\mathcal{A}|}{\sqrt{2\pi}\sigma} + \frac{\gamma}{2(1-\gamma)}\right)$ and $|\mathcal{A}|$ is the volume of the action space. Guaranteed improvement is maximized by using a step size $\alpha^* = \frac{\left\|\widehat{\nabla}^N_{\boldsymbol{v}} J(\boldsymbol{\theta})\right\|^2_-}{2\left\|\widehat{\nabla}^N_{\boldsymbol{v}} J(\boldsymbol{\theta})\right\|^2_+}$, yielding, with probability

$1-\delta$:

$$J(\boldsymbol{v}^{t+1}, w) - J(\boldsymbol{v}^t, w) \geq \frac{\left\|\widehat{\nabla}^N_{\boldsymbol{v}} J(\boldsymbol{\theta})\right\|^4_-}{4c\left\|\widehat{\nabla}^N_{\boldsymbol{v}} J(\boldsymbol{\theta})\right\|^2_+} \coloneqq C^{\boldsymbol{v}}_{max}. \tag{48}$$

For any $C^t \leq C^{\boldsymbol{v}}_{max}$*, the largest step-size guaranteeing* $J(\boldsymbol{v}^{t+1}, w) - J(\boldsymbol{v}^t, w) \geq C^t$ *with probability* $1-\delta$ *is* $\overline{\alpha} = \alpha^*(1+\lambda_{\boldsymbol{v}})$*, where* $\lambda_{\boldsymbol{v}} = \sqrt{1 - C^t/C^{\boldsymbol{v}}_{max}}$.

Similarly, we can obtain high-probability versions of Theorems 3 and 5 and the corresponding corrected step sizes. Note that, differently from the exact framework, the value of the optimal step size α^* is not constant and depends on the current gradient estimate.

Finally, since performance must also be estimated from a finite number of trajectories, the budget employed in our algorithms is also an estimate. This kind of uncertainty is intrinsic in RL and is neglected in this work.

Appendix D. Multi-Dimensional Actions

In this section we extend the results of the paper to the more general case of multi-dimensional action spaces. A common policy class for the case $\mathcal{A} \in \mathbb{R}^l$ is the factored Gaussian, i.e., a multi-variate Gaussian distribution having a diagonal covariance matrix $\Sigma_{\boldsymbol{\theta}}$. We denote with $\boldsymbol{\sigma_\theta}$ the vector of the diagonal elements, i.e., $\Sigma_{\boldsymbol{\theta}} = diag(\boldsymbol{\sigma_\theta})$. So, with a little abuse of notation[6], we can write the factored Gaussian policy as:

$$\pi_{\boldsymbol{\theta}}(a|s) = \frac{1}{\sqrt{2\pi}\boldsymbol{\sigma_\theta}} \exp\left\{-\frac{1}{2}\left(\frac{a-\mu_{\boldsymbol{\theta}}(s)}{\boldsymbol{\sigma_\theta}}\right)^2\right\},$$

where all vector operations are component-wise. The result is, of course, a multi-dimensional action. The natural generalization of Parametrization (1) is:

$$\mu_{\boldsymbol{\theta}}(s) = \boldsymbol{v}^T\boldsymbol{\phi}(s), \qquad \boldsymbol{\sigma_\theta} = e^{\boldsymbol{w}}, \qquad \boldsymbol{\theta} = [\boldsymbol{v}|\boldsymbol{w}], \tag{49}$$

where $\boldsymbol{w}$ is an l-dimensional vector. Following what (Papini et al., 2017) do for the mean parameter, we update $\boldsymbol{w}$ by greedy coordinate descent as well. All the results on $\boldsymbol{v}$ naturally extends to $\boldsymbol{w}$ since the bounds in Theorems 3 and 5 differ from the one in 2 only by a constant.

An even further generalization would be to consider a non-diagonal covariance matrix. This is interesting, but out of the scope of this work: in this paper we study the effects of the variance on exploration, while a full covariance matrix also models correlations among action dimensions that may be useful to learn in some tasks. Another promising generalization, left to future work, is represented by a state-dependent policy variance, which would allow a more powerful kind of exploration.

6. This allows us to avoid the much more cumbersome matrix notation, where even $\boldsymbol{v}$ is a matrix.

Appendix E. Task Specifications

E.1 LQG

The LQG problem is defined by transition model $s_{t+1} \sim \mathcal{N}\left(s_t + a_t, \eta_0^2\right)$, Gaussian policy $a_t \sim \mathcal{N}\left(\boldsymbol{v}^{\mathrm{T}} s, e^{2w}\right)$ and reward $r_t = -0.5\left(s_t^2 + a_t^2\right)$. Both action and state variables are bounded to the interval $[-2, 2]$ and the initial state is drawn uniformly at random. This is a well-known problem and admits a closed-form solution for a Gaussian policy linear in state. We use a discount factor of $\gamma = 0.99$, which gives an optimal parameter $\boldsymbol{v}^* \approx -0.615$ corresponding to expected deterministic performance $J(\boldsymbol{v}^*, -\infty) \approx -7.753$. We use a total batch size $N = 300$. We do not provide any initial budget in this task, i.e., $B^0 = 0$.

E.2 Mountain Car

We use the *openai/gym* implementation of Mountain Car, with $\gamma = 0.99$, $\boldsymbol{v}^0 = 0$, $\sigma_E^0 = 0.2$, $N = 300$ and $B^0 = 1$.