Provably Robust Blackbox Optimization for Reinforcement Learning

Krzysztof Choromanski[*]
Google Brain Robotics
kchoro@google.com

Aldo Pacchiano[*]
UC Berkeley
pacchiano@berkeley.edu

Jack Parker-Holder[*]
Columbia University
jh3764@columbia.edu

Yunhao Tang
Columbia University

Deepali Jain
Google Brain Robotics

Yuxiang Yang
Google Brain Robotics

Atil Iscen
Google Brain Robotics

Jasmine Hsu
Google Brain Robotics

Vikas Sindhwani
Google Brain Robotics

Abstract: Interest in derivative-free optimization (DFO) and "evolutionary strategies" (ES) has recently surged in the Reinforcement Learning (RL) community, with growing evidence that they can match state of the art methods for policy optimization problems in Robotics. However, it is well known that DFO methods suffer from prohibitively high sampling complexity. They can also be very sensitive to noisy rewards and stochastic dynamics. In this paper, we propose a new class of algorithms, called Robust Blackbox Optimization (RBO). Remarkably, even if up to 23% of all the measurements are arbitrarily corrupted, RBO can provably recover gradients to high accuracy. RBO relies on learning gradient flows using robust regression methods to enable off-policy updates. On several MuJoCo robot control tasks, when all other RL approaches collapse in the presence of adversarial noise, RBO is able to train policies effectively. We also show that RBO can be applied to legged locomotion tasks including path tracking for quadruped robots.

Keywords: Derivative-free Optimization, Reinforcement Learning, Evolutionary Strategies, Policy Search

1 Introduction

It is appealing to reduce policy learning tasks arising in Robotics to instances of blackbox optimization problems of the form,

$$\max_{\theta \in \mathbb{R}^d} F(\theta). \tag{1}$$

Above, θ encodes a policy $\pi_\theta : \mathcal{S} \rightarrow \mathcal{A}$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, and the function F maps θ to the total expected reward when the robot applies π_θ recursively in a given environment. In this context, the "blackbox" is an opaque physics simulator or even the robot hardware interacting with a real environment with unknown dynamics. As a consequence, the function F only admits point evaluations with no explicit analytical gradients available for an optimizer to exploit.

Blackbox methods and the so-called "evolutionary strategies" (ES) are instances of derivative-free optimization (DFO) [1, 2, 3, 4, 5] that aim to maximize F by applying various random search [6] techniques, while avoiding explicit gradient computation. Typically, in each epoch the policy parameter vector θ is updated using a gradient ascent rule that has the following general flavor [1, 7]:

$$\theta \leftarrow \theta + \eta \widehat{\nabla F}(\theta), \quad \text{where} \quad \widehat{\nabla F}(\theta) \approx \frac{1}{l} \sum_{j=1}^{l} w(\theta, \mathbf{g}_i)\mathbf{g}_i, \tag{2}$$

[*] Equal contribution.

3rd Conference on Robot Learning (CoRL 2019), Osaka, Japan.

and the gradient of F at θ is estimated by evaluating F at $\theta \pm \mathbf{g}_i$ for a certain choice of perturbation directions $\{\mathbf{g}_1, ..., \mathbf{g}_l\}$. Above, the function $w : \mathbb{R} \rightarrow \mathbb{R}$ translates rewards obtained by the perturbed policies to some weights and $\eta > 0$ is a step size. For example, the setting $w(\theta, \mathbf{g}) = \frac{1}{h}[F(\theta + h\mathbf{g}) - F(\theta)]$ where $\mathbf{g}$'s are the canonical directions, corresponds to the ubiquitous finite difference gradient estimator.

Surprisingly, despite not exploiting the internal structure of the RL problem, blackbox methods can be highly competitive with state of the art policy gradient approaches [8, 9, 10, 11], while admitting much simpler and embarrassingly parallelizable implementations. They can also handle complex, non-differentiable policy parameterizations, non-markovian reward structures and non-smooth hybrid dynamics. Particularly in simulation settings, they remain a serious alternative to classical model-free RL methods despite being among the simplest and oldest policy search techniques [12, 13, 14].

On the flip side, blackbox methods are notorious for requiring a prohibitively large number of rollouts. This is because these methods are exclusively on-policy and extract a relatively small amount of information from samples, compared to other model-free RL algorithms. The latter make use of the underlying structure (e.g. Markovian property) to derive updates, in particular off-policy methods which maintain and re-use previously collected data [15]. Indeed, the ES approach of Salimans et al. [1] required millions of rollouts on thousands of CPUs to get competitive results.

Furthermore, several theoretical results expose a fundamental and unavoidable gap between the performance of optimizers with access to gradients and those with access to only function evaluations, particularly in the presence of noise [16]. Even when the blackbox is a convex function, blackbox methods usually need more iterations than the standard gradient methods to converge, at a price that scales with problem dimensionality [17]. Without considerable care, they are also brittle in the face of noise and can breakdown when rewards are noisy or there is considerable stochasticity in the underlying system dynamics.

Starting from a natural regularized regression perspective on gradient estimation, we propose two simple enhancements to blackbox/ES techniques. First, we inject off-policy learning by reusing past samples to estimate an entire continuous local gradient field in the neighborhood of the current iterate. Secondly, by drawing on results from compressed sensing and error correcting codes [18, 19], we propose a robust regression LP-decoding framework that is guaranteed to provide provable accurate gradient estimates in the face of up to 23% arbitrary noise, including adversarial corruption, in function evaluations. The resulting method (abbreviated as RBO) shows dramatic resilience to massive measurement corruptions on a suite of 8 MuJoCo robot control tasks when competing algorithms appear to fall apart. We also observe favorable comparisons on walking and path tracking tasks on quadruped robots.

We start this paper with a simple regression perspective on blackbox optimization, introduce our algorithm and its off-policy elements with a striking theoretical result on its robustness, followed by a comprehensive empirical analysis on a variety of policy search problems in Robotics.

2 A Regularized Regression Perspective on Gradient Estimation

We begin by presenting a natural regression perspective on gradient estimation [20] for derivative-free optimization. First, recall the notion of a Gaussian smoothing F_σ of a given blackbox function F defined as,

$$F_\sigma(\theta) = \mathbb{E}_{\mathbf{g} \in \mathcal{N}(0, \mathbf{I}_d)}[F(\theta + \sigma\mathbf{g})] = (2\pi)^{-\frac{d}{2}} \int_{\mathbb{R}^d} F(\theta + \sigma\mathbf{g}) e^{-\frac{\|\mathbf{g}\|^2}{2}} d\mathbf{g}. \qquad (3)$$

It turns out that the updates proposed in the Evolutionary Strategies approach of Salimans et. al.[1] can be written as:

$$\theta \leftarrow \theta + \eta \widehat{\nabla}_{\mathrm{MC}} F_\sigma(\theta), \qquad (4)$$

where $\widehat{\nabla}_{\mathrm{MC}} F_\sigma(\theta)$ is the Monte Carlo (MC) estimator of the gradient $\nabla_{\mathrm{MC}} F_\sigma(\theta)$ of F_σ at θ.

Since the formula for the gradient $\nabla_{\mathrm{MC}} F_\sigma(\theta)$ is itself given as an expectation over Gaussian distribution, namely: $\nabla F_\sigma(\theta) = \frac{1}{\sigma}\mathbb{E}_{\mathbf{g} \sim \mathcal{N}(0, \mathbf{I}_d)}[F(\theta + \sigma\mathbf{g})\mathbf{g}]$, MC estimators can be easily constructed, simply by sampling k independent Gaussian perturbations $\sigma\mathbf{g}_i$ for $i = 1, ..., k$ and evaluating F at points determined by these perturbations. There exist several such unbiased MC estimators which apply different variance reduction techniques [7, 21, 22, 23]. Without loss of generality, we take an estimator using *forward finite difference* expressions [7] which is of the form:

$$\widehat{\nabla}_{\mathrm{MC}}^{\mathrm{AT}} F_{\sigma}(\theta) = \frac{1}{k\sigma} \sum_{i=1}^{k} (F(\theta + \sigma \mathbf{g}_i) - F(\theta)) \mathbf{g}_i. \tag{5}$$

One can notice by analyzing the Taylor expansion of F at θ that forward finite difference expressions $\frac{F(\theta+\sigma\mathbf{g}_i)-F(\theta)}{\sigma}$ in the formula above can be reinterpreted as estimations of the dot-products $\nabla F(\theta)^T \mathbf{g}_i$. In other words, by querying blackbox RL function F at θ with perturbations $\sigma\mathbf{g}_i$, one effectively collects lots of noisy estimates of $\nabla F(\theta)^T \mathbf{g}_i$.

The task then is to recover the unknown gradient from these estimates. This observation is the key to formulating blackbox function gradient estimation as a regression problem. This approach has two potential benefits. Firstly, it opens blackbox optimization to the wide class of regularized regression-based methods capable of recovering gradients more accurately than standard MC approaches in the presence of substantial noise. Secondly, it relaxes the independence condition regarding samples chosen in different iterations of the optimization, allowing for samples from previous iterations to be re-used.

As we will see later, the latter will eventually lead to more sample efficient methods. Interestingly, we will show that the recent orthogonal method for variance reduction in ES[7] can be interpreted as a particular instantiation of the regression-based approach.

2.1 A simple regression-based algorithm

Given scalars $\{F(\theta + \mathbf{z}_i)\}_{i=1}^{k}$ (corresponding to rewards obtained by different perturbations $\mathbf{z}_i$ of the policy encoded by θ), we formulate the regression problem by considering $\{\mathbf{z}_1, ..., \mathbf{z}_k\}$ as input vectors with target values $y_i = F(\theta + \mathbf{z}_i) - F(\theta)$ for $i = 1, ..., k$. We propose to produce a gradient estimator by solving the following regression problem:

$$\widehat{\nabla}_{\mathrm{RBO}} F(\theta) = \arg\min_{\mathbf{v} \in \mathbb{R}^d} \frac{1}{2k} \|\mathbf{y} - \mathbf{Z}\mathbf{v}\|_p^p + \alpha \|\mathbf{v}\|_q^q, \tag{6}$$

where $p, q \geq 1$, $\mathbf{Z} \in \mathbb{R}^{k \times d}$ is a matrix with the ith row encoding perturbations $\mathbf{z}_i$. The sequences of rows in $\mathbf{Z}$ are sampled from some given joint multivariate distribution $\mathbb{P} \in \mathcal{P}(\mathbb{R}^d \times ...\mathbb{R}^d)$ and $\alpha > 0$ is a regularization parameter.

As already mentioned, perturbations $\mathbf{z}_i$ do not need to be taken from the Gaussian multivariate distribution and they do not even need to be independent. Note that various known regression methods arise by instantiating the above optimization problems with different values of p, q and α. In particular, $p = q = 2$ leads to the ridge regression [24], $p = 2$, $q = 1$ to Lasso [25], $p = 1$, $q = 2$ to robust regression with least absolute deviations loss, and and $p = 1$, $\alpha = 0$ to LP decoding [18].

2.2 Using off-policy samples

Gradients reconstructed by the above regression problem (Equation 6) can be given to the ES optimizer. Furthermore, at any given iteration t the ES optimizer can reuse evaluations of these points $\theta_{t-1} + \sigma\mathbf{g}_i^{(t-1)}$ that are closest to current point θ_t, .e.g. top τ-percentage for the fixed hyperparameter $0 < \tau < 1$. Thus the regression interpretation enables us to go beyond the rigid framework of independent sets of samples. This algorithm, described in more detail in Algorithm 1 box (l.8 in the algorithm is a simple projection step restricting each parameter vector to be within a domain of allowable policies), plays the role of our base RBO variant and the backbone of our algorithmic approach. It already outperforms state-of-the-art ES methods on benchmark RL tasks. In the next section we will explain how it can be further refined to achieve even better performance.

2.3 Regression versus ES with orthogonal MC estimators

Here we show that ES methods based on orthogonal Monte Carlo estimators [7, 21], that were recently demonstrated to improve standard ES algorithms for RL, can be thought of as special cases of the regression approach.

Orthogonal MC estimators rely on pairwise orthogonal perturbations $\sigma\mathbf{g}_i$ that can be further renormalized to have length $l = \sigma\sqrt{d}$. The renormalization ensures that the marginal distributions of the orthogonal samples are the same as the unstructured ones, which render the orthogonal MC estimators unbiased. Further, the coupling induces correlation between perturbations for provable variance reduction.

Algorithm 1 Robust Blackbox Optimization Algorithm via Regression

```
Input: $F : \Theta \rightarrow \mathbb{R}$, scaling parameter sequence $\{\sigma\}_t$, initial $\theta_0 = \mathbf{u}_0 \in \Theta$, number of perturbations $k$, step size sequence $\{\eta_t\}_t$, sampling distribution $\mathbb{P} \in \mathcal{P}(\mathbb{R}^d)$, parameters $p, q, \alpha, \tau$, number of epochs $T$.
Output: Vector of parameters $\theta_T$.
1. Initialize $\Theta_{\mathrm{old}}^{\mathrm{pert}} = \emptyset$, $R_{\mathrm{old}} = \emptyset$ ($|\Theta_{\mathrm{old}}^{\mathrm{pert}}| = |R_{\mathrm{old}}|$).
for $t = 0, 1, \ldots, T-1$ do
    1. Compute all distances from $\mathbf{u}_t$ to $\theta_{\mathrm{old}}^{\mathrm{pert}} \in \Theta_{\mathrm{old}}^{\mathrm{pert}}$.
    2. Find the closest $\tau$-percentage of vectors from $\Theta_{\mathrm{old}}^{\mathrm{pert}}$ and call this set $\Theta_\tau^{\mathrm{near}}$. Call the corresponding subset of $R_{\mathrm{old}}$ as $R_\tau^{\mathrm{near}}$.
    3. Sample $\mathbf{g}_1^{(t)}, \cdots, \mathbf{g}_{k-|\Theta_\tau^{\mathrm{near}}|}^{(t)}$ from $\mathbb{P}$.
    4. Compute $F(\theta_t)$ and $F(\theta_t + \sigma_t\mathbf{g}_j^{(t)})$ for all $j$.
    5. Let $\mathbf{Z}_t \in \mathbb{R}^{k \times d}$ be a matrix obtained by concatenating rows given by $\sigma_t \times \mathbf{g}_i^{(t)}$ and those of the form: $\mathbf{p}_i - \theta_t$, where $\mathbf{p}_i \in \Theta_\tau^{\mathrm{near}}$.
    6. Let $\mathbf{y}_t \in \mathbb{R}^k$ be the vector obtained by concatenating values $F(\theta_t + \sigma_t\mathbf{g}_j^{(t)}) - F(\theta_t)$ with those of the form: $r_i - F(\theta_t)$, where $r_i \in R_\tau^{\mathrm{near}}$.
    7. Let $\widehat{\nabla}_{\mathrm{RBO}}F(\theta_t)$ be the resulting vector after solving the following optimization problem:
       $$\widehat{\nabla}_{\mathrm{RBO}}F(\theta_t) = \arg\min_{\mathbf{v}\in\mathbb{R}^d} \frac{1}{2k}\|\mathbf{y}_t - \mathbf{Z}_t\mathbf{v}\|_p^p + \alpha\|\mathbf{v}\|_q^q,$$
    8. Take $\mathbf{u}_{t+1} = \theta_t + \eta_t\widehat{\nabla}_{\mathrm{RBO}}F(\theta_t)$
    9. Take $\theta_{t+1} = \arg\max_{\theta\in\Theta}\langle\theta, \mathbf{u}_{t+1}\rangle - \frac{1}{2}\|\theta\|_2^2$.
    10. Update $\Theta_{\mathrm{old}}^{\mathrm{pert}}$ to be the set of the form $\theta_t + \mathbf{z}_i$, where $\mathbf{z}_i$s are rows of $\mathbf{Z}_t$ and $\theta_t$, and $R_{\mathrm{old}}$ to be the set of the corresponding values $F(\theta_t + \mathbf{z}_i)$ and $F(\theta_t)$.
```

Orthogonal MC estimators can be easily constructed via Gram-Schmidt orthogonalization process from the ensembles of unstructured independent samples [26]. The following is true:

Lemma 1. *The class of the orthogonal Monte Carlo estimators using renormalization with $k = d$ orthogonal samples is equivalent to particular sub-classes of* RBO *estimators with $p = q = 2$.*

Proof. The solution to the ridge regression problem for gradient estimation ($p = q = 2$) is of the form

$$\widehat{\nabla}_{\mathrm{RBO}}F_{\mathrm{ridge}}(\theta) = (\mathbf{Z}_t^\top\mathbf{Z}_t + 2d\alpha\mathbf{I}_d)^{-1}\mathbf{Z}_t^\top\mathbf{y}_t \tag{7}$$

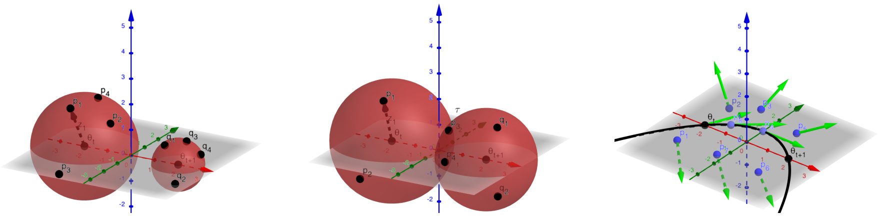
By the assumptions of the lemma we get: $\mathbf{Z}_t\mathbf{Z}_t^\top = \sigma^2 d\mathbf{I}_d$, thus $\mathbf{Z}_t^\top = \sigma^2 d\mathbf{Z}_t^{-1}$, and we obtain:

$$\widehat{\nabla}_{\mathrm{RBO}}F_{\mathrm{ridge}}(\theta) = \frac{1}{d\sigma}\mathbf{G}_{\mathrm{ort}}^\top\mathbf{y}_t \cdot \frac{\sigma^2}{\sigma^2 + 2\alpha}, \tag{8}$$

where $\mathbf{G}_{\mathrm{ort}}^\top$ is a matrix with rows given by orthogonal Gaussian vectors $\mathbf{g}_i^{\mathrm{ort}}$. Thus, if we take $\sigma = \sigma_{\mathrm{MC}}$, where σ_{MC} stands for the smoothing parameter in the MC estimator and furthermore, $\eta = \eta_{\mathrm{MC}}\frac{\sigma^2+2\alpha}{\sigma^2}$, where η_{MC} stands for the steps size in the algorithm using that MC estimator, then the RBO estimator is equivalent to the orthogonal MC and the proof is completed. □

3 Learning Gradient Flows for off-policy sample reuse

Algorithm 1 reconstructs the gradient of F only at θ. To refine this algorithm, consider reconstructing the gradient of F at an entire continuous neighborhood of θ instead of θ alone. The idea is to use values of F computed in the neighborhood $\mathcal{N}(\theta_t)$ of θ_t to approximate the gradient field $\mathcal{F}_{\mathrm{grad}}$ of F in the entire neighborhood $\mathcal{N}(\theta_t)$ rather than just at θ_t. This method utilizes past function evaluations F to an even bigger extent. In this approach k function values from iteration t of the algorithm are used to reconstruct several gradients in the neighborhood $\mathcal{N}(\theta_t)$ of θ_t as opposed to the baseline ES algorithm, where each value is used for only one gradient or Algorithm 1, where some values (from the closest τ-percentage of the new point θ_{t+1}) are reused.



(a) Vanilla ES (b) RBO (c) RBO + gradient flows

Figure 1: Comparison of different ES optimization methods: (a) Vanilla ES: to update current point θ_t, independent perturbations $\mathbf{p}_i$ are chosen. Since perturbations are not reused, all perturbations $\mathbf{q}_i$ used in θ_{t+1} are different from the previous ones. (b) Base RBO: perturbations no longer need to be independent, the τ-percentage of old perturbations closest to the new point θ_{t+1} are reused. Gradient in θ_t is reconstructed via regression. (c) RBO with gradient flows: gradients are recovered in several point of the neighborhood of θ_t via regression. An approximation of the gradient field in $\mathcal{N}(\theta_t)$ is computed via matrix-valued kernel interpolation and the update of θ_t is conducted via gradient flow.

The gradient at point $\theta_t + \sigma\mathbf{g}_i^t$ is reconstructed in the analogous way as in Algorithm 1, with the use of the estimator $\widehat{\nabla}_{\mathrm{RBO}}F(\theta_t + \sigma\mathbf{g}_i^t)$, where data for the regressor consists of the same $k+1$ points as in Algorithm 1, namely: vector θ_t and vectors $\theta_t + \sigma\mathbf{g}_j^t$ for $j = 1, ..., k$. The only difference is that now $\theta_t + \sigma\mathbf{g}_i^t$ plays the role of the base vector and other k vectors are interpreted as its perturbed versions. All the reconstructed gradients form a set $\widehat{\mathcal{F}}_{\mathrm{grad}}^{\mathrm{sparse}}$, which can be thought of as a sparse approximation of the gradient field $\mathcal{F}_{\mathrm{grad}}$ of F in $\mathcal{N}(\theta_t)$.

3.1 Interpolating gradient field via matrix-valued kernels

The set of gradients $\widehat{\mathcal{F}}_{\mathrm{grad}}^{\mathrm{sparse}}$ is used to create an interpolation $\widehat{\mathcal{F}}_{\mathrm{grad}}$ of the true gradient field $\mathcal{F}_{\mathrm{grad}}$ in $\mathcal{N}(\theta_t)$ via matrix-valued kernels. Below we give a short overview over the theory of matrix-valued kernels [27, 28, 29], which suffices to explain how they can be applied for interpolation.

Definition 1 (matrix-valued kernels)**.** *A function $K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^m \times \mathbb{R}^m$ is a matrix-valued kernel if for every $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ the following holds:*

$$K(\mathbf{x}, \mathbf{y}) = K(\mathbf{y}, \mathbf{x})^\top. \tag{9}$$

We call K a positive definite kernel if furthermore the following holds. For every set $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_l\} \subseteq \mathbb{R}^d$ the block matrix $K(\mathcal{X}, \mathcal{X}) = (K(\mathbf{x}_i, \mathbf{x}_j))_{i,j \in \{1,...,l\}}$ is positive definite.

As we see, matrix-valued kernels are extensions of their scalar counterparts. As standard scalar-valued kernels can be used to approximate scalar fields via functions from reproducing kernel Hilbert spaces (RKHS) corresponding to these kernels, matrix-valued kernels can be utilized to interpolate vector-valued fields. The interpolation problem can be formulated as:

$$\mathcal{F}_* = \mathrm{argmin} \sum_{j=1}^{m} \frac{1}{N} \sum_{i=1}^{N} (\mathcal{F}_j(\mathbf{x}_i) - \mathbf{y}_i^j)^2 + \lambda \|\mathcal{F}\|_{\mathbf{K}}^2, \tag{10}$$

where $\mathcal{F} : \mathbb{R}^d \rightarrow \mathbb{R}^m$ is a vector-valued function from the RKHS corresponding to $\mathbf{K}$, scalar $\mathbf{y}_i^j$ is the j^{th} component of the i^{th} vector-valued observations $\mathbf{y}_i \in \mathbb{R}^d$ for the i^{th} sample $\mathbf{x}_i$ from the set $\mathcal{X}$ of N input datapoints and $\|\cdot\|_{\mathbf{K}}$ stands for the norm that the above RKHS is equipped with. The solution to the optimization problem (Equation 10) is given by the formula: $\mathcal{F}(\mathbf{x}) = \sum_{i=1}^{N} K(\mathbf{x}_i, \mathbf{x})\mathbf{c}_i$, where vectors $\mathbf{c}_i \in \mathbb{R}^d$ are given by:

$$\mathbf{c} = (K(\mathcal{X}, \mathcal{X}) + \lambda N \mathbf{I}_{Nd \times Nd})^{-1}\mathbf{y}. \tag{11}$$

For separable matrix-valued kernels [30], such a problem can be solved at a complexity that scales no worse than standard scalar kernel methods. Above, $\mathbf{c} \in \mathbb{R}^{Nd}$ is a concatenations of vectors $\mathbf{c}_1, ..., \mathbf{c}_N$ and $\mathbf{y}$ is a concatenation of the observations $\mathbf{y}_1, ..., \mathbf{y}_N$.

Gradient Ascent Flow: The RBO casts gradient field reconstruction problem as the above interpolation problem, where: $\mathcal{X} = \{\theta_t, \theta_t + \sigma \mathbf{g}_1^t, ..., \theta_t + \sigma \mathbf{g}_k^t\}$ and for $\mathbf{x}_i \in \mathcal{X}$ we have: $\mathbf{y}_i = \widehat{\nabla}_{\mathrm{RBO}} F(\mathbf{x}_i)$. After obtaining the solution $\widehat{\mathcal{F}}_{\mathrm{grad}} = \mathcal{F}_*$, the update of the current point θ_t is obtained via the standard gradient ascent flow in the neighborhood of θ_t, which is the solution to the following differential equation:

$$d\theta/dt = \widehat{\mathcal{F}}_{\mathrm{grad}}(\theta), \tag{12}$$

whose solution can be numerically obtained using such methods as Euler integration. The comparison of the presented RBO algorithms with baseline ES is schematically presented in Fig. 1.

3.2 Time Complexity and Distributed Implementation

Our RBO implementation relies on distributed computations, where different workers evaluate F in different subsets of perturbations. The time needed to construct all approximate gradients $\mathbf{y}_i$ in a given iteration of the algorithm is negligible compared to the time needed for querying F (because calculations of $\mathbf{y}_i$ can be also easily parallelized). Computations of $\mathbf{c}$ from Equation 11 can be efficiently conducted using separability and random feature maps[31]. We also noted that in practice the gradient-flow extension of the RBO requires many fewer perturbations per iteration than baseline ES and one can use state-of-the-art compact neural networks from [7], which encode RL policies with a few hundred parameters. This further reduces the cost of computing the gradient field.

4 Provably Robust Gradient Recovery

It turns out that the RBO with LP decoding ($p = 1$, $\alpha = 0$) is particularly resilient to noisy measurements. This is surprising at first glance, since we empirically tested (see: Section 5) that it is true even when a substantial number of measurements are very inaccurate and when the assumption that noise for each measurement is independent clearly does not hold (e.g. for noisy dynamics or when measurements are spread into simulator calls and real hardware experiments).

In this section we explain why it is the case. We leverage the results from a completely different field: adversarial attacks for database systems [18] and explain why RBO with LP decoding can create an accurate sparse approximation $\widehat{\mathcal{F}}_{\mathrm{grad}}$ to the true gradient field $\mathcal{F}_{\mathrm{grad}}$ with loglinear number of measurements per point even if up to $\rho^* = 0.2390318914495168...$ of all the measurements are arbitrarily corrupted. Interestingly, we do not require any assumptions regarding gradient sparsity.

We also present convergence results for RBO under certain regularity assumptions regarding functions F. These can be translated to the results on convergence to local maxima for less regular mappings F. All proofs as well as standard definitions of L-Lipschitz, λ-smooth and μ-strongly concave functions are given in the Appendix.

Definition 2 (coefficient ρ^*). *Let $X \sim \mathcal{N}(0,1)$ and denote: $Y = |X|$. Let f be the* pdf *of Y and F be its* cdf *function. Define $g(x) = \int_x^\infty yf(y)dy$. Function g is continuous and decreasing in the interval $[0, \infty]$ and furthermore $g(0) = \mathbb{E}[Y]$. Since $\lim_{x\to\infty} g(x) = 0$, there exists x^* such that $g(x^*) = \frac{\mathbb{E}[Y]}{2}$. We define ρ^* as:*

$$\rho^* = 1 - F^{-1}(x*). \tag{13}$$

Its exact numerical value is $\rho^* = 0.2390318914495168...$.

The following result shows the robustness of the RBO gradients under substantial noise:

Lemma 2. *There exist universal constants $C, c > 0$ such that the following holds. Let $F : \Theta \to \mathbb{R}$ be a $\lambda-$smooth function. Assume that at most the ρ^*-fraction of all the measurements are arbitrarily corrupted and the other ones have error at most ϵ. If $\sigma_t = \sqrt{\frac{\epsilon}{d\lambda}}$, $k \geq Cd$ and RBO uses LP decoding, with probability $p = 1 - \exp(-cd)$ the following holds:*

$$\|\nabla_{\mathrm{RBO}} F(\theta_t) - \nabla F(\theta)\|_2 \leq 2C\sqrt{\epsilon d \lambda}. \tag{14}$$

We are ready to state our main theoretical result.

Theorem 1. *Consider a blackbox function $F : \Theta \rightarrow \mathbb{R}$. Assume that F is concave, Lipschitz with parameter L and smooth with smoothness parameter λ. Assume furthermore that domain $\Theta \subset \mathbb{R}^d$ is convex and has l_2 diameter $\mathcal{B} < \infty$. Consider* Algorithm 1 *with $p = 1, \alpha = 0, \tau = 0, \sigma_t \leq \frac{L}{d\lambda\sqrt{t+1}}, \eta_t = \frac{\mathcal{B}}{L\sqrt{t+1}}$ and the noisy setting in which at each step a fraction of at most ρ^* of all measurements $F(\theta_t + \sigma_t \mathbf{g}_j^t)$ are arbitrarily corrupted for $j = 1, 2, ..., k$. There exists a universal constant $c_1 > 0$ such that for any $\gamma \in (0, 1)$ and $T \leq \gamma \exp(c_1 d)$, the following holds with probability at least $1 - \gamma$:*

$$F(\theta^*) - \left[\frac{1}{T}\sum_{t=0}^{T-1} F(\theta_t)\right] \leq \frac{13}{2}\mathcal{B}L\frac{1}{\sqrt{T}},$$

where $\theta^* = \arg\max_{\theta \in \Theta} F(\theta)$. If F presents extra curvature properties such as being strongly concave, we can get a linear convergence rate. The following theorem holds:

Theorem 2. *Assume conditions from Theorem 1 and furthermore that F is strongly concave with parameter μ. Take* Algorithm 1 *with $p = 1, \alpha = 0, \tau = 0, \sigma_t \leq \frac{L^2}{d\mathcal{B}\mu\lambda(t+1)}, \eta_t = \frac{1}{\mu(t+1)}$ acting in the noisy environment in which at each step a fraction of at most ρ^* of all measurements $F(\theta_t + \sigma_t \mathbf{g}_j^t)$ are arbitrarily corrupted for $j = 1, 2, ..., k$. There exists a universal constant $c_1 > 0$ such that for any $\gamma \in (0, 1)$ and $T \leq \gamma \exp(c_1 d)$, with probability at least $1 - \gamma$:*

$$F(\theta^*) - \left[\frac{1}{T}\sum_{t=0}^{T-1} F(\theta_t)\right] \leq \frac{6L^2}{\mu}\frac{(1 + \log(T))}{T}.$$

To summarize, even if up to 23% of all the measurements are arbitrarily corrupted, RBO can provably recover gradients to high accuracy! We provide empirical evidence of this result next.

5 Empirical Analysis

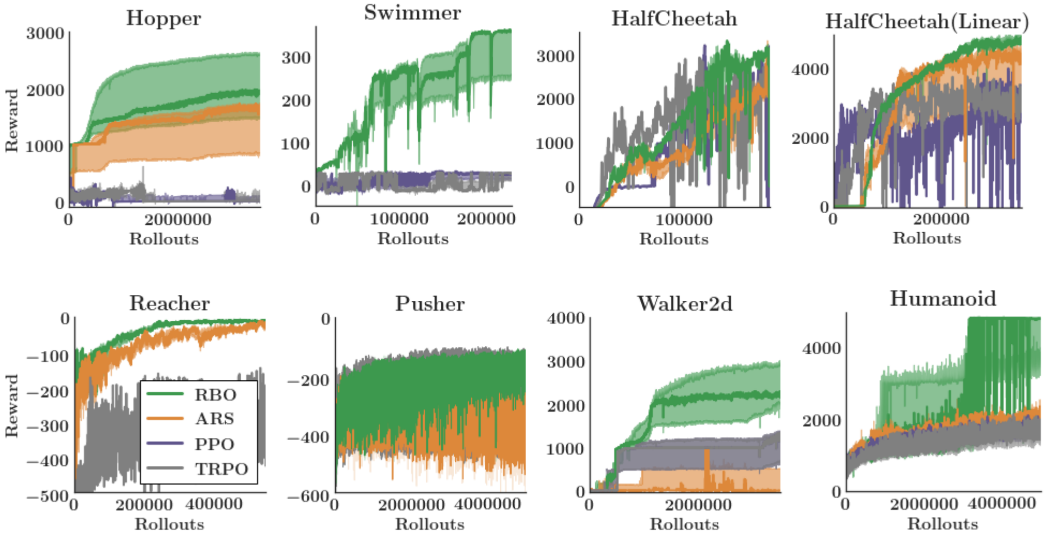


Figure 2: Performance of RBO, ARS, PPO and TRPO on seven benchmark RL environments. Results presented are the median from 3 seeds, with the min and max shaded. In each case, 20% of the rewards are significantly corrupted. In some cases, this noise led to drastically worse performance for non-RBO methods, which we omit from the plots, but include in the tabular data in Table 1.

The real world is often far noisier than environments typically used for benchmarking RL algorithms. With this in mind, the primary goal of our experiments is to demonstrate that RBO is able to efficiently learn good policies in the presence of noise, where other approaches fail. To investigate this, we consider two settings:

1. OpenAI Gym [32] MuJoCo environments, where 20% of the measurements are significantly corrupted (we show that adding noise presents a challenge to baseline algorithms).
2. Real-world quadruped locomotion tasks, where sim-to-real transfer is non-trivial.

We run RBO with LP-decoding to obtain provably noise-robust reconstruction of ES gradients.

OpenAI Gym: We conducted an exhaustive analysis of the proposed class of RBO algorithms on the following OpenAI Gym [32] benchmark RL tasks: Swimmer, HalfCheetah, Hopper, Walker2d, Humanoid, Pusher and Reacher. All but HalfCheetah Linear experiments are for a policy encoded by feedforward neural networks with two hidden layers of size $h = 41$ each and tanh nonlinearities. HalfCheetah Linear is for the linear architecture. We compare RBO to state-of-the-art ES algorithm ARS [3], as well as two state-of-the art policy gradient algorithms: TRPO [9] and PPO [8]. In all cases, we corrupt 20% of the measurements. As we show in Fig. 2, the noise often renders the other algorithms unable to learn optimal policies, yet RBO remains unscathed and consistently learns good policies for all tasks. Under the presence of substantial noise the other methods often drastically underperform RBO, as we show in Table 1.

		Median reward after # rollouts			
Environment	**Rollouts**	RBO	ARS	TRPO	PPO
HalfCheetah (Linear)	2.10^5	**4220**	4205	**-Inf**	**-Inf**
HalfCheetah (Toeplitz)	2.10^5	**3299**	3163	**-Inf**	**-Inf**
Swimmer	2.10^5	**360**	**-Inf**	32	30
Walker2d	2.10^6	**2230**	172	996	312
Hopper	1.10^6	**1503**	1408	427	256
Humanoid	5.10^6	**4865**	2355	2028	2129
Pusher	1.10^6	**-155**	-199	**-Inf**	**-Inf**
Reacher	5.10^5	**-7**	-19	**-Inf**	**-Inf**

Table 1: Median rewards obtained across $k = 5$ seeds for seven RL environments. Bold represents the best performing algorithm in each environment, red indicates failure to learn.

Quadruped Locomotion: We tested RBO on quadruped locomotion tasks for a Minitaur robot [33] (see: Fig. 3), with different reward functions for different locomotion tasks. Minitaur has 4 legs and 8 degrees of freedom, where each leg has the ability to swing and extend to a certain degree using the PD controller provided with the robot. We train our policies in simulation using the pybullet environment modeled after the robot [34]. To learn walking for quadrupeds, we use architectures called *Policies Modulating Trajectory Generators* (PMTGs) that have been recently proposed in [35]. They incorporate basic cyclic characteristics of the locomotion and leg movement primitives by using trajectory generators, a parameterized function that provides cyclic leg positions. The policy is responsible for modulating and adjusting leg trajectories. The results fully support our previous findings. While without noise RBO and ARS (used as state-of-the-art method for these tasks [35]) perform similarly, in the presence of noise RBO is superior to ARS.

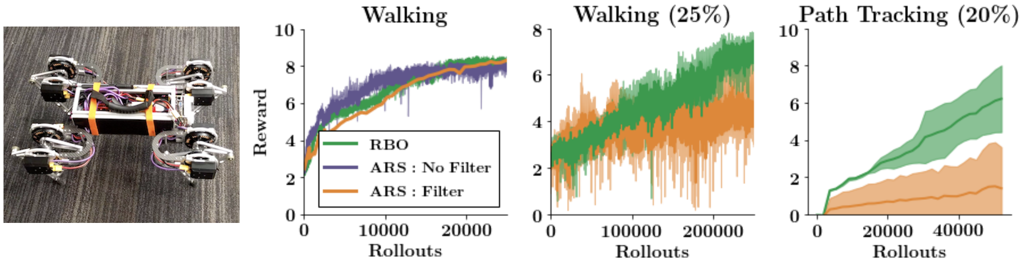


Figure 3: Left: The Minitaur robot. Right: Performance of RBO and ARS for two Minitaur simulated environments: forward walking (without noise and with 25% noisy measurements) and path tracking with 20% noisy measurements.

6 Conclusion

We proposed a new class of ES algorithms, called RBO, for optimizing RL policies that utilize gradient flows induced by vector field interpolation via matrix-valued kernels. The interpolators rely on general regularized regression methods that provide sample complexity reduction through sample reuse. We show empirically and theoretically that RBO is much less sensitive to noisy measurements, which are notoriously ubiquitous in robotics applications, than state-of-the-art baseline algorithms.

References

[1] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. 2017.

[2] D. Wierstra, T. Schaul, T. Glasmachers, Y. Sun, J. Peters, and J. Schmidhuber. Natural evolution strategies. *Journal of Machine Learning Research*, 15:949–980, 2014.

[3] H. Mania, A. Guy, and B. Recht. Simple random search provides a competitive approach to reinforcement learning. *CoRR*, abs/1803.07055, 2018. URL http://arxiv.org/abs/1803.07055.

[4] J. Lehman, J. Chen, J. Clune, and K. O. Stanley. ES is more than just a traditional finite-difference approximator. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2018, Kyoto, Japan, July 15-19, 2018*, pages 450–457, 2018. doi:10.1145/3205455.3205474. URL https://doi.org/10.1145/3205455.3205474.

[5] S. Ha and C. K. Liu. Evolutionary optimization for parameterized whole-body dynamic motor skills. In *2016 IEEE International Conference on Robotics and Automation, ICRA 2016, Stockholm, Sweden, May 16-21, 2016*, pages 1390–1397, 2016. doi:10.1109/ICRA.2016.7487273. URL https://doi.org/10.1109/ICRA.2016.7487273.

[6] T. G. Kolda, R. M. Lewis, and V. Torczon. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM review*, 45(3):385–482, 2003.

[7] K. Choromanski, M. Rowland, V. Sindhwani, R. E. Turner, and A. Weller. Structured evolution with compact architectures for scalable policy optimization. In *Proc. of the 35th Int. Conf. on Machine Learning, ICML 2018, Stockholm, Sweden, July 10-15, 2018*, pages 969–977.

[8] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2016-2018.

[9] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In *International Conference on Machine Learning (ICML)*, 2015.

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

[11] P. Hämäläinen, A. Babadi, X. Ma, and J. Lehtinen. PPO-CMA: proximal policy optimization with covariance matrix adaptation. *CoRR*, abs/1810.02541, 2018. URL http://arxiv.org/abs/1810.02541.

[12] C. Shu, H. Ding, and N. Zhao. Numerical comparison of least square-based finite-difference (lsfd) and radial basis function-based finite-difference (rbffd) methods. *Computers & Mathematics with Applications*, 51(8):1297–1310, 2006.

[13] N. Kohl and P. Stone. Policy gradient reinforcement learning for fast quadrupedal locomotion. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 3, pages 2619–2624. IEEE, 2004.

[14] J. Peters and S. Schaal. Policy gradient methods for robotics. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2219–2225. IEEE, 2006.

[15] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller. Playing atari with deep reinforcement learning. *ArXiv*, abs/1312.5602, 2013.

[16] K. G. Jamieson, R. Nowak, and B. Recht. Query complexity of derivative-free optimization. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2012.

[17] Y. Nesterov and V. Spokoiny. Random gradient-free minimization of convex functions. *Found. Comput. Math.*, 17(2):527–566, Apr. 2017. ISSN 1615-3375.

[18] C. Dwork, F. McSherry, and K. Talwar. The price of privacy and the limits of lp decoding. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 85–94. ACM, 2007.

[19] E. Candes and T. Tao. Decoding by linear programming. *arXiv preprint math/0502327*, 2005.

[20] A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to derivative-free optimization*, volume 8. Siam, 2009.

[21] M. Rowland, K. Choromanski, F. Chalus, A. Pacchiano, T. Sarlos, R. E. Turner, and A. Weller. Geometrically coupled monte carlo sampling. In *accepted to NIPS'18*, 2018.

[22] K. Choromanski, M. Rowland, W. Chen, and A. Weller. Unifying orthogonal monte carlo methods. In *International Conference on Machine Learning*, pages 1203–1212, 2019.

[23] Y. Tang, K. Choromanski, and A. Kucukelbir. Variance reduction for evolution strategies via structured control variates. *arXiv preprint arXiv:1906.08868*, 2019.

[24] H. Avron, K. L. Clarkson, and D. P. Woodruff. Sharper bounds for regression and low-rank approximation with regularization. *CoRR*, abs/1611.03225, 2016. URL http://arxiv.org/abs/1611.03225.

[25] F. Santosa and W. W. Symes. Linear inversion of band-limited reflection seismograms. In *SIAM Journal on Scientific and Statistical Computing*, pages 1307–1330, 1986.

[26] F. X. Yu, A. T. Suresh, K. M. Choromanski, D. N. Holtmann-Rice, and S. Kumar. Orthogonal random features. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 1975–1983, 2016. URL http://papers.nips.cc/paper/6246-orthogonal-random-features.

[27] M. A. Álvarez, L. Rosasco, and N. D. Lawrence. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2012. doi:10.1561/2200000036. URL https://doi.org/10.1561/2200000036.

[28] C. A. Micchelli and M. Pontil. On learning vector-valued functions. *Neural Computation*, 17(1):177–204, 2005. doi:10.1162/0899766052530802. URL https://doi.org/10.1162/0899766052530802.

[29] M. Reisert and H. Burkhardt. Learning equivariant functions with matrix valued kernels. *Journal of Machine Learning Research*, 8:385–408, 2007. URL http://dl.acm.org/citation.cfm?id=1314513.

[30] V. Sindhwani, H. Q. Minh, and A. C. Lozano. Scalable matrix-valued kernel learning for high-dimensional nonlinear multivariate regression and granger causality. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 586–595. AUAI Press, 2013.

[31] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2008.

[32] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. OpenAI Gym, 2016.

[33] G. Kenneally, A. De, and D. E. Koditschek. Design principles for a family of direct-drive legged robots. *IEEE Robotics and Automation Letters*, 1(2):900–907, 2016.

[34] E. Coumans and Y. Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. *http://pybullet.org*, 2017.

[35] A. Iscen, K. Caluwaerts, J. Tan, T. Zhang, E. Coumans, V. Sindhwani, and V. Vanhoucke. Policies modulating trajectory generators. In *2nd Annual Conference on Robot Learning, CoRL 2018, Zürich, Switzerland, 29-31 October 2018, Proceedings*, pages 916–926, 2018. URL http://proceedings.mlr.press/v87/iscen18a.html.

[36] P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, Y. Wu, and P. Zhokhov. Openai baselines. https://github.com/openai/baselines, 2017.

[37] E. Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

7 Appendix

7.1 Ablation Studies

We include an ablation study for three key hyper-parameters, the level of noise in th environment, the regression method, and the amount of sample re-use (τ).

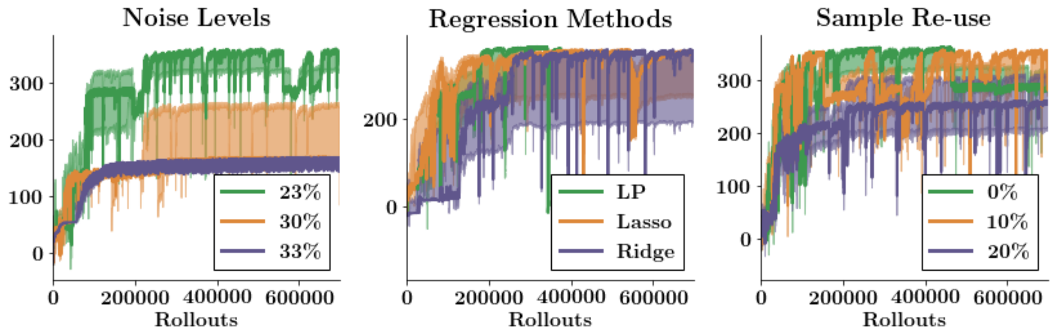


Figure 4: Ablation studies on the Swimmer environment. In each case, we use the same experiment set-up as in the main paper, only changing one element.

Interestingly, while RBO is only provably robust to 23% noise in the LP case, we see that it is still possible to learn good policies with LP for 30% noise, while Lasso and Ridge also learn good policies.

7.2 Further experimental details

In all experiments we used learning rate $\eta = 0.01$. ES algorithms (RBO and ARS) were applying smoothing parameter $\sigma = 0.1$. Furthermore, ARS used both state and reward renormalization, as described in [3]. In RBO experiments with gradient flows we used matrix-valued kernels based on the class of radial basis function scalar kernels (RBFs). Measurement noise was added by corrupting certain percentages of the computed rewards at each iteration of the algorithm.

We used implementation of the Trust Region Policy Optimization (TRPO) [9] algorithm from [36]. We applied default hyper-parameters. Similarly, we used Proximal Policy Optimization (PPO) [8] implementation from [36] and applied default hyper-parameters.

7.3 Definitions

Here we introduce the definitions of λ-smoothenss and L-lipshitz. These are standard definitions in the optimization literature which we reproduce here for clarity (see for example [37]).

Definition 3. *(λ-smoothness): A differentiable concave function $F : \Theta \to \mathbb{R}$ is smooth with parameter $\lambda > 0$ if for every pair of points $x, y \in \Theta$:*

$$\|\nabla F(y) - \nabla F(x)\|_2 \leq \lambda \|y - x\|_2$$

If F is twice differentiable it is equivalent to $-\lambda I \preceq \nabla^2 F(x) \preceq 0$ for all $x \in \Theta$.

Definition 4. *(L-Lipschitz): We say that $F : \theta \to \mathbb{R}$ is Lipschitz with parameter $L > 0$ if for all $x, y \in \Theta$ it satisfies $|F(x) - F(y)| \leq L\|x - y\|_2$.*

Definition 5. *(μ-Strong Concavity): A function $F : \Theta \to \mathbb{R}$ is strongly concave with parameter μ if:*

$$F(y) \leq F(x) + \langle \nabla F(x), y - x \rangle - \frac{\mu}{2}\|y - x\|_2^2$$

7.4 Proof of Lemma 2

In this section we prove Lemma 2 which we reproduce below for readability. This result is concerned with the case when a constant proportion of the measurements are arbitrarily corrupted, while the remaining ones are corrupted by a small amount ϵ. We quantify the degree of corruption experienced by our gradient estimator. Let $\nabla F(\theta_t)$ be the real gradient of F at θ_t. Before proving the main Lemma of this section, let's show that whenever σ_t is very small, and the evaluation of F is noiseless, a difference of function evaluations is close to a dot product between the gradient of F and the displacement vector.

Lemma 3. $F(\theta_t + \sigma_t v_j^{(t)}) - F(\theta_t) = \langle \nabla F(\theta_t), \sigma_t \mathbf{g}_j^{(t)} \rangle + \xi_t$ *with* $|\xi_t| \leq \sigma_t^2 d\lambda$*.*

This follows immediately from a Taylor expansion and the smoothness assumption on F.

Lemma 4. *There exist universal constants* $C, c > 0$ *such that the following holds. Let* $F : \Theta \to \mathbb{R}$ *be a* $\lambda-$*smooth function. Assume that at most the* ρ^**-fraction of all the measurements are arbitrarily corrupted and the other ones have error at most* ϵ*. If* $\sigma_t = \sqrt{\frac{\epsilon}{d\lambda}}$*,* $k \geq Cd$ *and RBO uses LP decoding, the following holds with probability* $p = 1 - \exp(-cd)$*:*

$$\|\nabla_{\mathrm{RBO}} F(\theta_t) - \nabla F(\theta)\|_2 \leq 2C\sqrt{\epsilon d\lambda}. \tag{15}$$

Proof. We use $F(\theta_t + \sigma_t \mathbf{g}_j^{(t)})$ as proxy measurements for $\langle \nabla F(\theta_t), \sigma_t \mathbf{g}_j^{(t)} \rangle$. Since $F(\theta_t + \sigma_t v_j^{(t)}) - F(\theta_t) = \langle \nabla F(\theta_t), \sigma_t \mathbf{g}_j^{(t)} \rangle + \xi_t$ with $|\xi_t| \leq \sigma_t^2 d\lambda$, and we assume the measurements of $F(\theta_t + \sigma_t \mathbf{g}_j^{(t)}) - F(\theta)$ are either completely corrupted or corrupted by a noise of magnitude at most ϵ, the total displacement for the normalized dot product component of the objective equals $\sigma_t d\lambda + \frac{\epsilon}{\sigma_t}$. Setting $\sigma_t = \sqrt{\frac{\epsilon}{d\lambda}}$ means the total error can be driven down to $2\sqrt{\epsilon d\lambda}$ by this choice of σ_t. After these observations, a direct application of Theorem 1 in [18] yields the result. □

Notably, the error in Equation 15 cannot be driven to zero unless the errors of magnitude ϵ are driven to zero themselves. This is in direct contrast with the supporting lemma we prove in the next section as a stepping stone towards proving Thoerem 1. Nonsurprisingly our result has dependence on the irreducible error ϵ. There is no way to get around the dependence on this error.

7.5 Proof of Theorem 1

We start with a result that is similar in spirit to Lemma 2. The assumptions behind Theorem 1 differ from those underlying Lemma 2 in that we only assume the presence of a constant fraction of arbitrary perturbations on the measurements. All the remaining measurements are assumed to be exact. We show the recovered gradient $\hat{\nabla}_{RBO}$ is close to the true gradient. This distance is controlled by the smoothing parameters $\{\sigma_t\}$.

Lemma 5. *There exist universal constants* c_1, c_2 *such that if for any* t *if up to* ρ^* *fraction of the entries of* y_t *are arbitrarily corrupted, the gradient recovery optimization problem with input* θ_t *satisfies:*

$$\|\hat{\nabla}_{RBO} F(\theta_t) - \nabla F(\theta_t)\| \leq \sigma_t d\lambda \tag{16}$$

Whenever $k \geq c_1 d$ *and with probability* $1 - \exp(-c_2 d)$

The proof of Lemma 5 and the constants c_1, c_2 are a result of a direct application of Theorem 1 in [18].

Assume from now on the domain $\Theta \subset \mathbb{R}^d$ is convex and has l_2 diameter $\mathcal{B} < \infty$. We can now show the first order Taylor approximation of F around θ_t that uses the true gradient and the one using the RBO gradient are uniformly close:

Lemma 6. *The following bound holds: For all* $\theta_t \in \Theta$*:*

$$\sup_{\theta \in \Theta} |\langle \theta - \theta_t, \hat{\nabla}_{RBO} F(\theta_t) \rangle - \langle \theta - \theta_t, \nabla F(\theta_t) \rangle| \leq \mathcal{B}\sigma_t d\lambda$$

The next lemma provides us with the first step in our convergence bound:

Lemma 7. *For any* θ^* *in* Θ*, it holds that:*

$$\begin{aligned} 2\left(F(\theta^*) - F(\theta_t)\right) \leq & \frac{\|\theta_t - \theta^*\|_2^2 - \|\theta_{t+1} - \theta^*\|_2^2}{\eta_t} \\ & + \eta_t \left(\|\nabla F(\theta_t)\| + \sigma_t d\lambda\right)^2 \\ & + 2\mathcal{B}\sigma_t d\lambda \end{aligned}$$

Proof. Recall that θ_{t+1} is the projection of u_{t+1} to a convex set Θ. And that $u_{t+1} = \theta_t + \eta_t \hat{\nabla}_{RBO} F(\theta_t)$. As a consequence:

$$\begin{aligned} \|\theta_{t+1} - \theta^*\|^2 &\leq \|\theta_t + \eta_t \hat{\nabla}_{RBO} F(\theta_t) - \theta^*\|^2 \\ &= \|\theta_t - \theta^*\|^2 + \eta_t^2 \|\hat{\nabla}_{RBO} F(\theta_t)\|^2 \\ &\quad - 2\eta_t \langle \hat{\nabla}_{RBO} F(\theta_t), \theta^* - \theta_t \rangle \end{aligned} \tag{17}$$

Lemma 5 and the triangle inequality imply:

$$\|\hat{\nabla}_{RBO} F(\theta_t)\|^2 \leq \left(\|\nabla F(\theta_t)\| + \sigma_t d\lambda\right)^2$$

This observation plus Lemma 6 applied to Equation 19 implies:

$$\begin{aligned} 2\langle \nabla F(\theta_t), \theta^* - \theta_t \rangle \leq & \frac{\|\theta_t - \theta^*\|_2^2 - \|\theta_{t+1} - \theta^*\|_2^2}{\eta_t} \\ & + \eta_t \left(\|\nabla F(\theta_t)\| + \sigma_t d\lambda\right)^2 \\ & + 2\mathcal{B}\sigma_t d\lambda \end{aligned}$$

Since concavity of F implies $F(\theta^*) - F(\theta_t) \leq \langle \nabla F(\theta_t), \theta^* - \theta_t \rangle$, the result follows. □

We proceed with the proof of Theorem 1:

$$\begin{aligned} 2\sum_{t=0}^{T-1} \left(F(\theta^*) - F(\theta_t)\right) \leq & \sum_{t=0}^{T-1} \frac{\|\theta_t - \theta^*\|_2^2 - \|\theta_{t+1} - \theta^*\|_2^2}{\eta_t} \\ & + \sum_{t=0}^{T-1} \eta_t \left(\|\nabla F(\theta_t)\| + \sigma_t d\lambda\right)^2 \quad + 2\mathcal{B}\sigma_t d\lambda \\ \leq & \sum_{t=0}^{T-1} \|\theta_t - \theta * \|^2 \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right) \\ & + \sum_{t=0}^{T-1} \eta_t (L + \sigma_t d\lambda)^2 + 2\mathcal{B}\sigma_t d\lambda, \end{aligned} \tag{18}$$

where we set $\frac{1}{\eta_{-1}} = 0$. The first inequality is a direct consequence of Lemma 7. The second inequality follows because $\|\theta_T - \theta^*\| \geq 0$ and $\|\nabla F(\theta)\| \leq L$ for all $\theta \in \Theta$.

As long as, $\sigma_t \leq \frac{L}{d\lambda\sqrt{t+1}}$ and $\eta_t = \frac{\mathcal{B}}{L\sqrt{t+1}}$ we have:

$$\sum_{t=0}^{T-1} F(\theta^*) - F(\theta_t) \leq \frac{13}{2}\mathcal{B}L\sqrt{T}$$

Since $\sum_{t=1}^{T} \frac{1}{\sqrt{t}} \leq 2\sqrt{T}$, Theorem 1 follows.

7.6 Proof of Theorem 2

In this section we flesh out the convergence results for robust gradient descent when F is assumed to be Lipschitz with parameter L, smooth with parameter λ and strongly concave with parameter μ.

Lemma 8. *For any θ^* in Θ, it holds that:*

$$\begin{gathered} 2\left(F(\theta^*) - F(\theta_t)\right) \leq \frac{\|\theta_t - \theta^*\|_2^2 - \|\theta_{t+1} - \theta^*\|_2^2}{\eta_t} - \mu\|\theta_t - \theta^*\|^2 + \\ \eta_t \left(\|\nabla F(\theta_t)\| + \sigma_t d\lambda\right)^2 + 2\mathcal{B}\sigma_t d\lambda \end{gathered}$$

Proof. Recall that θ_{t+1} is the projection of u_{t+1} to a convex set Θ. And that $u_{t+1} = \theta_t + \eta_t \hat{\nabla}_{RBO} F(\theta_t)$. As a consequence:

$$\begin{aligned} \|\theta_{t+1} - \theta^*\|^2 &\leq \|\theta_t + \eta_t \hat{\nabla}_{RBO} F(\theta_t) - \theta^*\|^2 \\ &= \|\theta_t - \theta^*\|^2 + \eta_t^2 \|\hat{\nabla}_{RBO} F(\theta_t)\|^2 \\ &\quad - 2\eta_t \langle \hat{\nabla}_{RBO} F(\theta_t), \theta^* - \theta_t \rangle \end{aligned} \tag{19}$$

Lemma 5 and the triangle inequality imply:

$$\|\hat{\nabla}_{RBO} F(\theta_t)\|^2 \leq \left(\|\nabla F(\theta_t)\| + \sigma_t d\lambda\right)^2$$

This observation plus Lemma 6 applied to Equation 19 implies:

$$\begin{aligned} 2\langle \nabla F(\theta_t), \theta^* - \theta_t \rangle \leq & \frac{\|\theta_t - \theta^*\|_2^2 - \|\theta_{t+1} - \theta^*\|_2^2}{\eta_t} \\ & + \eta_t \left(\|\nabla F(\theta_t)\| + \sigma_t d\lambda\right)^2 \\ & + 2\mathcal{B}\sigma_t d\lambda \end{aligned}$$

Since strong concavity of F implies $F(\theta^*) - F(\theta_t) \leq \langle \nabla F(\theta_t), \theta^* - \theta_t \rangle - \frac{\mu}{2}\|\theta_t - \theta^*\|^2$ the result follows. □

The proof of Theorem 2 follows from the combination of the last few lemmas. Indeed, we have:

$$\begin{aligned} 2\sum_{t=0}^{T-1} \left(F(\theta^*) - F(\theta_t)\right) \leq & \sum_{t=0}^{T-1} \frac{\|\theta_t - \theta^*\|_2^2 - \|\theta_{t+1} - \theta^*\|_2^2}{\eta_t} - \\ & \mu\|\theta_t - \theta^*\|^2 + \\ & \sum_{t=0}^{T-1} \eta_t \left(\|\nabla F(\theta_t)\| + \sigma_t d\lambda\right)^2 + 2\mathcal{B}\sigma_t d\lambda \\ & \leq \underbrace{\sum_{t=0}^{T-1} \|\theta_t - \theta *\|^2 \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} - \mu\right)}_{I} + \\ & \sum_{t=0}^{T-1} \eta_t (L + \sigma_t d\lambda)^2 + 2\mathcal{B}\sigma_t d\lambda, \end{aligned}$$

where we set $\frac{1}{\eta_{-1}} = 0$. the first inequality is a direct consequence of Lemma 8. The second inequality follows because $\|\theta_T - \theta^*\| \geq 0$ and $\|\nabla F(\theta)\| \leq L$ for all $\theta \in \Theta$. Since $\eta_t = \frac{1}{\mu *(t+1)}$, $\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} = \mu$ for all $t = 0, \cdots, T-1$ the term labeled I in the inequality above vanishes.

As long as $\sigma_t \leq \frac{L^2}{d\mathcal{B}\mu\lambda(t+1)}$, we have:

$$\sum_{t=0}^{T-1} F(\theta^*) - F(\theta_t) \leq \frac{6L^2}{\mu}(1 + \log(T))$$

Since $\sum_{t=1}^{T} \frac{1}{t} \leq 1 + \log(T)$, Theorem 2 follows.