Generalizing Gaussian Smoothing for Random Search

Katelyn Gao [1] **Ozan Sener** [2]

Abstract

Gaussian smoothing (GS) is a derivative-free optimization (DFO) algorithm that estimates the gradient of an objective using perturbations of the current parameters sampled from a standard normal distribution. We generalize it to sampling perturbations from a larger family of distributions. Based on an analysis of DFO for non-convex functions, we propose to choose a distribution for perturbations that minimizes the mean squared error (MSE) of the gradient estimate. We derive three such distributions with provably smaller MSE than Gaussian smoothing. We conduct evaluations of the three sampling distributions on linear regression, reinforcement learning, and DFO benchmarks in order to validate our claims. Our proposal improves on GS with the same computational complexity, and are competitive with and usually outperform Guided ES (Maheswaranathan et al., 2019) and Orthogonal ES (Choromanski et al., 2018), two computationally more expensive algorithms that adapt the covariance matrix of normally distributed perturbations.

1. Introduction

In many practical applications, machine learning is complicated by the lack of analytical gradients of the objective with respect to the parameters of the predictor. For example, a search and rescue robot could have complex mechanics that may be impossible to accurately model even with full knowledge of the terrain, and without a model of the system dynamics, the analytical gradients of the success rate with respect to the policy parameters would not be available. On the other hand, noisy evaluations of the objective, such as Booleans indicating success, are inexpensive to obtain. The problem of optimizing a function with only zeroth-order evaluations is called derivative-free optimization (DFO).

[1]Intel Labs, Santa Clara, CA, USA [2]Intel Labs, Munich, Germany. Correspondence to: Katelyn Gao <katelyn.gao@intel.com>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

Gaussian smoothing (GS) (Matyas, 1965; Nesterov & Spokoiny, 2017) is a DFO algorithm that estimates the gradient using evaluations at perturbations of the parameters, randomly sampled from the standard normal distribution and computing finite differences. Current extensions of GS add post-processing. Polyak (1987) and Flaxman et al. (2005) normalize the perturbations; Orthogonal ES (Choromanski et al., 2018) orthogonalizes them. Guided ES (Maheswaranathan et al., 2019) rotates the perturbations to be better aligned with recent gradient estimates; LMRS (Sener & Koltun, 2019) rotates them to a learned subspace. The last three approaches increase the computational complexity as they require the Gram-Schmidt process.

Although Gaussian smoothing is shown to be effective, the choice of standard normal distribution is rather arbitrary. In this paper, we generalize GS to sample perturbations from arbitrary distributions. We can choose distributions that optimize any desired property thanks to the proposed generalization. Specifically, we show that a convergence bound for stochastic gradient descent for smooth non-convex functions is proportional to the mean squared error (MSE) of the gradients. Therefore, we select a distribution with reduced MSE of the gradient estimate, computing the MSE under the assumption that the entries of the perturbations are IID with mean zero.

Our first algorithm to reduce the MSE is *Bernoulli Smoothing* (BeS), which replaces the standard normal distribution with a standardized Bernoulli distribution with probability 0.5. After fixing the distributional family of perturbations to be Gaussian or Bernoulli, we obtain distributions that approximately minimize the MSE. The distributions have simple analytical forms and do not require any information about the objective. In fact, they are scaled versions of GS and BeS, with smaller variance, and so we call the resulting algorithms *GS-shrinkage* and *BeS-shrinkage*. Due to the IID assumption of the entries of the perturbations, BeS, GS-shrinkage, and BeS-shrinkage have the same computational complexity as GS.

To validate the theory, we empirically evaluate our proposed methods for derivative-free optimization on linear regression since the analytical gradients are available to compute various statistics. Results confirm that GS-shrinkage and BeS-shrinkage lead to gradient estimates with smaller MSE. We

further conduct an empirical evaluation on high-dimensional reinforcement learning (RL) benchmarks, based on locomotion or manipulation, with various budgets for trajectory simulation in the environment and linear policies. Generally, BeS learns a superior policy to GS. For the locomotion environments, GS-shrinkage and BeS-shrinkage usually outperform BeS; which one is better depends on the environment and the budget. Lastly, we evaluate on noisy DFO benchmarks, and observe that when the number of perturbations sampled at each iteration is smaller than the problem dimension, BeS outperforms GS at the start of optimization. However, BeS and GS outperform GS-shrinkage and BeS-shrinkage. Overall, our algorithms are computationally more efficient and competitive with and often outperform Guided ES and Orthogonal ES. These conclusions remain when a neural network replaces the linear policy in RL.

2. Related work

Derivative-free optimization is a research field that includes Bayesian optimization, genetic algorithms, and random search; see Conn et al. (2009) and Custódio et al. (2017) for surveys. We review in more detail literature most related to our work, Gaussian smoothing and evolutionary strategies.

Gaussian smoothing GS (Matyas, 1965; Nesterov & Spokoiny, 2017) is a random search algorithm that estimates the gradient of an objective using its values at random perturbations of the parameters sampled from the standard normal distribution. Many variants exist. Polyak (1987) and Flaxman et al. (2005) normalize the perturbations, obtaining samples from the uniform distribution on the unit sphere instead of the standard normal. More recently, methods have been proposed to improve GS by modifying the distribution of the perturbations, at the expense of greater computational complexity, which we have included as baselines in the experiments. Choromanski et al. (2018) orthogonalizes the perturbations, by either the Gram-Schmidt process or constructing random Hadamard-Rademacher matrices, which decreases the MSE of the gradient estimate. Maheswaranathan et al. (2019) changes the covariance matrix of the perturbations to be aligned with the subspace spanned by recent gradient estimates, again requiring the Gram-Schmidt process, and Sener & Koltun (2019) proposes a similar algorithm for the special case where the objective lies on a learned low-rank manifold.

Our methods do not increase the computational complexity compared to GS, as we still assume that the entries of the perturbations are IID. We explicitly minimize the MSE of the gradient estimate with respect to the distribution of the perturbations. Chen & Wild (2015) adopts a similar strategy to learn the optimal spacing for the finite difference in GS; unlike ours, their algorithms depend on characteristics of the objective.

Evolutionary strategies Evolutionary strategies (ES), a class of genetic algorithm, mathematically looks similar to Gaussian smoothing but is orthogonally motivated. ES minimizes the expected objective of a distribution over the parameter space, which is equivalent to minimizing the objective if the distribution is allowed to degenerate to a delta distribution. More concretely, if the distribution were Gaussian optimization would be done with respect to both the mean and variance. On the other hand, Gaussian smoothing and its relatives optimize only the mean; the variance is utilized purely to estimate the gradient. Due to the difference in the algorithmic structure, we decided not to include ES algorithms as baselines in the experiments. Popular ES algorithms are CMA-ES (Hansen et al., 2003), where the distribution is anisotrophic Gaussian, and NES (Wierstra et al., 2014), which performs natural gradient descent for arbitrary distributions.

GS for policy search Several of the previous works evaluate on reinforcement learning benchmarks. Salimans et al. (2017) applies GS to MuJoCo locomotion and Atari environments with MLP policies, showing performance competitive with policy gradient algorithms. However, it requires objective shaping, which Choromanski et al. (2018) and subsequent works were able to remove. For linear policies, ARS (Mania et al., 2018) showed that the MuJoCo lomotion benchmarks can be solved by GS after adding observation and reward standardization; Sener & Koltun (2019) substantially speeds up learning on the more difficult environments. We remark that the ARS gradient estimator is mathematically similar to GS-shrinkage. However, it treats the variance of the perturbation distribution as a hyperparameter to be tuned through grid search, and so we do not include it as a baseline. In contrast, we propose to set it to a value that approximately minimizes the MSE of the gradient estimate, without any knowledge of the objective or optimization needed.

3. Preliminaries

We first present the notation used in the paper and provide some background on Gaussian smoothing. Then, we show that the convergence bound for stochastic gradient descent (SGD) is proportional to the MSE of the gradient estimate, thereby providing the motivation behind the algorithms proposed in Section 4.

3.1. Notation

We are interested in an unconstrained scalar minimization objective $F(\theta) : \mathbb{R}^d \to \mathbb{R}$. Suppose that it can only be accessed via a random evaluation $f(\theta, \xi)$ satisfying $\mathbb{E}_\xi f(\theta, \xi) = F(\theta)$. For example, in supervised learning, the random evaluation could be the loss at a data point, and in reinforcement learning the negative of a trajectory reward.

First-order optimization methods estimate the gradient of the objective using samples from a gradient oracle $\nabla_\theta f(\theta, \xi)$. Alternatively, as described next, the gradient of the objective may be estimated using only random evaluations, which are generally inexpensive.

3.2. Gaussian smoothing

Gaussian smoothing (GS) (Nesterov & Spokoiny, 2017) estimates the gradient of a function by generating a direction from a standard normal distribution, computing the directional derivative along the direction using function evaluations, and then multiplying the directional derivative with the direction. The estimate can be plugged into any gradient-based optimization method, making Gaussian smoothing a widely applicable zero-order approach.

Specifically, the Gaussian smoothing gradient estimator is

$$\nabla_\theta F^{GS}(\theta) = \frac{1}{c} F(\theta + c\epsilon)\epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}). \tag{1}$$

It may be interpreted as a Monte Carlo estimate of the gradient of the following modified objective, after smoothing by a standard normal random variable:

$$F_c(\theta) \triangleq \mathbb{E}_{\epsilon \sim \mathcal{N}(0,\mathbf{I})}[F(\theta + c\epsilon)], \quad c > 0$$

The gradient of the modified objective is given by

$$\nabla_\theta F_c(\theta) = \mathbb{E}_{\epsilon \sim \mathcal{N}(0,\mathbf{I})}\left[\frac{1}{c} F(\theta + c\epsilon)\epsilon\right] \tag{2}$$

Because $\nabla_\theta F^{GS}(\theta)$ often has high variance in practice, popular alternatives are the forward-difference (FD) estimator and the antithetic (AT) estimator, which incorporate control variates: for $\epsilon \sim \mathcal{N}(0, \mathbf{I})$,

$$\begin{aligned} \nabla_\theta F^{FD}(\theta) &= \frac{1}{c}\left[F(\theta + c\epsilon) - F(\theta)\right]\epsilon \\ \nabla_\theta F^{AT}(\theta) &= \frac{1}{2c}\left[F(\theta + c\epsilon) - F(\theta - c\epsilon)\right]\epsilon \end{aligned} \tag{3}$$

The variance of each estimator can be reduced by averaging over multiple directions. We focus on the FD estimator due to its simplicity and lower computational burden.

When $F(\theta)$ enjoys some mild regularity conditions and $c \to 0$, Nesterov & Spokoiny (2017) shows that iterative optimization using gradients estimated via (3) converges to a stationary point for non-convex objectives and the optimal point for convex ones.

3.3. Convergence of biased SGD

The estimators in (3) are unbiased for the gradient of the modified objective $F_c(\theta)$, but are biased for the gradient of the objective of interest $F(\theta)$. Therefore, the usual convergence guarantees for iterative optimization, which generally assume unbiased gradient estimates, do not directly hold. Recent works have analyzed the convergence properties of SGD with biased gradient estimates, for convex (Hu et al., 2016) and smooth (Chen & Luss, 2018; Ajalloeian & Stich, 2020) objectives. In Theorem 3.1, we provide a convergence guarantee on SGD with biased gradient estimates that, in contrast to those works, depends on the MSE of the gradient estimates. The proof is given in Appendix A.

Theorem 3.1. *Assume that i) $F(\theta)$ is differentiable, μ-smooth [1], and is bounded by Δ ii) the bias and the MSE of the gradient estimates g^t satisfy $\|\mathbb{E}[g^t] - \nabla_\theta F(\theta^t)\|_2 \leq B\|\nabla_\theta F(\theta^t)\|_2$ and $\mathbb{E}[\|g^t - \nabla_\theta F(\theta^t)\|_2^2] \leq M$, respectively iii) iterative updates are applied via $\theta^{t+1} = \theta^t - \eta g(\theta^t)$ for T steps. Then, if $B < 0.5$, letting $\eta = 1/\mu\sqrt{T}$,*

$$\frac{1}{T}\sum_t \|\nabla_\theta F(\theta^t)\|_2^2 \leq \frac{M + 4\Delta\mu}{(1 - 2B)\sqrt{T}}.$$

This guarantee suggests that we may improve convergence by reducing the MSE of the gradient estimates. Ideally, we would minimize the entire bound, but doing so is impractical as it requires knowledge of μ and Δ.

The assumptions in Theorem 3.1 are standard in the analysis of convergence of iterative optimization for non-convex functions, but are fairly strong. In particular, the objective is not bounded for linear regression. However, we only use that assumption to bound the difference between the initial function value $F(\theta^0)$ and the optimal one $F(\theta^\star)$. Hence, it may be replaced with an assumption that the initialization has bounded distance from the optimal solution.

4. Generalized smoothing

Our main idea is that by generalizing GS to be able to sample from arbitrary distributions, not just the standard normal, we may select the distribution to optimize any criterion. In this paper, we propose to select the distribution that minimizes the MSE of the gradient estimates (3), aiming to improve the convergence of SGD using those gradient estimates.

In this section, we present our proposed algorithms, with all proofs in Appendix B. We focus on the forward difference gradient estimator where the objective is estimated via Monte Carlo sampling random evaluations; derivations for the antithetic gradient estimator are the same (see Appendix B.5). Mathematically, given N random evaluations ξ_i and L IID sampled directions ϵ_l, our estimator of interest is

$$\nabla_\theta \hat{F}^{FD}(\theta) = \frac{1}{cLN}\sum_{l,i}(f(\theta + c\epsilon_l, \xi_i) - f(\theta, \xi_i))\epsilon_l. \tag{4}$$

[1] $F(\theta)$ is μ-smooth if $\|\nabla_\theta F(\theta_1) - \nabla_\theta F(\theta_2)\|_2 \leq \mu\|\theta_1 - \theta_2\|_2$ for any θ_1 and θ_2 in the domain of $F(\theta)$.

We also make the following assumption:

Assumption 4.1. The entries of ϵ_l, $\{\epsilon_{lj}\}_{j=1}^d$, are IID samples from a distribution with expectation 0.

4.1. MSE of the FD estimator

We start by computing the MSE of $\nabla_\theta \hat{F}^{FD}(\theta)$.

Lemma 4.2. *Suppose that i) the first-order Taylor expansions of $F(\theta)$ and $f(\theta, \cdot)$ satisfy a regularity condition* [2] *ii) assumption 4.1 holds. Then, as $c \to 0$, the MSE of $\nabla_\theta \hat{F}^{FD}(\theta)$ approaches*

$$\left((\sigma^2 - 1)^2 + \frac{\sigma^4}{L}(d + k - 2)\right) \|\nabla_\theta F(\theta)\|_2^2 \tag{5}$$

$$+ \frac{\sigma^4}{LN}(d + k - 1)\,\mathrm{tr}(\mathrm{Var}_\xi[\nabla_\theta f(\theta, \xi)]), \tag{6}$$

where σ^2 and k are the variance and kurtosis of ϵ_{lj}. The bias of $\nabla_\theta \hat{F}^{FD}(\theta)$ approaches $(\sigma^2 - 1)\nabla_\theta F(\theta)$.

Notice that GS makes the same assumptions as Lemma 4.2, except for (i). For the rest of this section, we operate in the setting where $c \to 0$.

4.2. Bernoulli Smoothing

There is a very simple choice of ϵ_{lj} that reduces the MSE of $\nabla_\theta \hat{F}^{FD}(\theta)$. For GS, $\epsilon_{lj} \sim \mathcal{N}(0, 1)$, so $\sigma^2 = 1$ and $k = 3$ (Weisstein, b), and the MSE of $\nabla_\theta \hat{F}^{FD}(\theta)$ equals

$$\frac{d+1}{L}\|\nabla_\theta F(\theta)\|_2^2 + \frac{d+2}{LN}\,\mathrm{tr}(\mathrm{Var}_\xi[\nabla_\theta f(\theta, \xi)]).$$

Observe that if $\sigma^2 = 1$ is fixed (and the gradient estimate remains asymptotically unbiased), the MSE of $\nabla_\theta \hat{F}^{FD}(\theta)$ decreases with smaller kurtosis k. Since the Bernoulli distribution has the smallest kurtosis of any distribution (DeCarlo, 1997), a natural proposal to reduce the MSE is $\epsilon_{lj} \sim (B_{0.5} - 0.5)/0.5$, where $B_{0.5}$ follows the Bernoulli distribution with probability 0.5. In other words, ϵ_{lj} is a standardized fair Bernoulli random variable with expectation 0, $\sigma^2 = 1$, and $k = 1$ (Weisstein, a), and the MSE of $\nabla_\theta \hat{F}^{FD}(\theta)$ would equal

$$\frac{d-1}{L}\|\nabla_\theta F(\theta)\|_2^2 + \frac{d}{LN}\,\mathrm{tr}(\mathrm{Var}_\xi[\nabla_\theta f(\theta, \xi)]),$$

smaller than its MSE for GS.

We call this proposal *Bernoulli Smoothing* (BeS). It remains unknown whether there is a direct correspondence to the gradient of a smoothed objective, like for GS. Note that the bias B remains zero, so the corresponding convergence bound in Theorem 3.1 is smaller than that of GS; experimental results in Section 5 confirm that BeS is always at least competitive with, and usually outperforms, GS.

[2] See Appendix B for details.

4.3. Shrinkage gradient estimators

We next take a more principled approach to reduce the MSE of $\nabla_\theta \hat{F}^{FD}(\theta)$ and find the distribution of ϵ_{lj} that minimizes it. For mathematical tractability, we restrict to a certain distribution type (Gaussian or Bernoulli). This method can easily be extended to other distribution types.

Gaussian Since a Gaussian distribution is determined by its expectation and variance, has kurtosis 3, and we assume that ϵ_{lj} has expectation 0, we search over the variance σ^2. In particular, we minimize only the larger term of the MSE, (5), since minimizing the entire MSE requires the gradients of $F(\theta)$ and $f(\theta, \xi)$; then, the problem reduces to solving

$$\min_{\sigma^2 > 0} \quad (\sigma^2 - 1)^2 + \frac{\sigma^4}{L}(d + 1) \tag{7}$$

Doing so is reasonable for learning from mini-batch samples because (6) is $\mathcal{O}(1/N)$ smaller than (5), where N is the batch size. Moreover, the problem is simplified as (7) is quadratic in σ^2 and thus has an analytic solution.

Theorem 4.3. *The solution to (7) is $\sigma^{2*} = {}^{L}/_{L+d+1}$. When $\epsilon_{lj} \sim \mathcal{N}(0, \sigma^{2*})$, the MSE of $\nabla_\theta \hat{F}^{FD}(\theta)$ is smaller than when $\epsilon_{lj} \sim \mathcal{N}(0, 1)$.*

Notice that the variance of ϵ_{lj} has been shrunk towards zero, and the shrinkage increases as the data dimension d increases. We call setting $\epsilon_{lj} \sim \mathcal{N}(0, \sigma^{2*})$ *GS-shrinkage*.

Bernoulli Extending BeS, we consider $\epsilon_{lj} \sim (B_p - p)/m$, where B_p follows the Bernoulli distribution with probability p, and search over p and m. ϵ_{lj} is centered, with variance $p(1-p)/m^2$ and kurtosis $3 + {}^{1-6p(1-p)}/_{p(1-p)}$. As with the Gaussian case, we minimize only (5), and the problem reduces to solving

$$\min_{p \in (0,1), m > 0} \left(\frac{p(1-p)}{m^2} - 1\right)^2 + \frac{p^2(1-p)^2}{Lm^4}\left(d + 1 + \frac{1 - 6p(1-p)}{p(1-p)}\right) \tag{8}$$

Because (8) is quadratic in $p(1-p)$ for fixed m, it can be solved analytically.

Theorem 4.4. *Assume $L + d > 5$. The solution to (8) is $p^* = 0.5$ and $m^* = \sqrt{{}^{L+d-1}/_{4L}}$. When $\epsilon_{lj} \sim (B_p^* - p^*)/m^*$, the MSE of $\nabla_\theta \hat{F}^{FD}(\theta)$ is smaller than when $\epsilon_{lj} \sim (B_{0.5} - 0.5)/0.5$.*

Notice that the variance of ϵ_{lj} has been shrunk towards zero, but the kurtosis is unchanged; again the amount of shrinkage increases as the data dimension d increases. Therefore, we call setting $\epsilon_{lj} \sim (B_p^* - p^*)/m^*$ *BeS-shrinkage*.

Table 1 summarizes our three proposed algorithms and GS, and compare the corresponding MSEs of $\nabla_\theta \hat{F}^{FD}(\theta)$. All

three have the advantage that the distributions of ϵ_{lj} are objective-independent. Assumption 4.1 ensures that BeS, GS-shrinkage, and BeS-shrinkage have the same computational complexity to sample the directions as GS, $\mathcal{O}(Ld)$.

While the variance of ϵ_{lj} are similar for GS-shrinkage and BeS-shrinkage, the difference in the MSEs depends on L, N, d and the magnitude of the gradients of $F(\theta)$ and $f(\theta, \xi)$ and may be substantial. In particular, for high-dimensional problems like those considered in Section 5, BeS-shrinkage has smaller MSE at the beginning of optimization when $\nabla_\theta F(\theta)$ is large, while GS-shrinkage has smaller MSE at the end of optimization (see Appendix B.4 for further details). This suggests that BeS-shrinkage could lead to better sample efficiency for online reinforcement learning, where new trajectories are generated at each optimization iteration.

5. Experiments

We conduct experiments to i) validate the theoretical claims in Sections 3 and 4 ii) compare the three proposed algorithms to GS and two previous algorithms, guided ES (Maheswaranathan et al., 2019) and orthogonal ES (Choromanski et al., 2018), and iii) investigate the impact of increasing the dimension d or of using the antithetic gradient estimator instead of the forward difference gradient estimator. The optimizer is SGD using the gradient estimator (4) (or in (iii), its antithetic version). The learning rate and spacing c are chosen by grid search, to maximize the test performance at the end of optimization [3]. Details are found in Appendix C.

5.1. Validating theory

To validate the theoretical claims, we evaluate GS, BeS, GS-shrinkage, and BeS-shrinkage on linear regression with squared error loss, where analytical formulas for the gradient of the objective are available. Our data model is from Gao & Sener (2020):

$$\begin{aligned} &y = \gamma^\intercal x + \epsilon \quad \epsilon \sim \mathcal{N}(0, \sigma^2) \quad x \sim \mathcal{N}(0, \mathbf{Q}) \\ &\gamma \sim U([0,2]^d) \quad \sigma^2 \sim U([0,2]) \\ &\mathbf{Q} = \mathbf{V}\,\mathrm{diag}(\gamma)\mathbf{V}^\intercal \quad \mathbf{V} \sim U(\mathbb{SO}(d)) \end{aligned} \tag{9}$$

We show results for $d = 100$ in the online setting, where N new data points are sampled from the model at each optimization iteration. The first row of Figure 1 plots the MSEs of the gradient estimator (4) as optimization progresses, for the four algorithms over a range of values for L and N. The second row plots the corresponding losses on a test set, generated from (9). Appendix C.1 contains similar plots for more values of L and N. We control for L and N since they directly affect the MSE, as seen from Lemma 4.2; the average is taken over five randomly generated seeds and the bands indicate one standard deviation.

The MSE of the gradient for GS-shrinkage and BeS-shrinkage is always substantially smaller than for GS and BeS; the differences between GS and BeS and between GS-shrinkage and BeS-shrinkage are not statistically significant. In terms of the test loss, the story is mixed, but the main difference is between GS/BeS and GS-shrinkage/BeS-shrinkage. For $L = 2, N = 5$, standard errors are large and GS & BeS appear to outperform GS-shrinkage & BeS-shrinkage. However, for $L = 2$ & $N = 15$ and $L = 6$ & $N = 15$, GS-shrinkage & BeS-shrinkage statistically significantly outperform GS & BeS; BeS-shrinkage appears to be slightly better in the first case, while the two are competitive in the second case. This suggests that the lower MSE of BeS-shrinkage compared to GS-shrinkage at the beginning of optimization may indeed translate to better test performance.

5.2. Online reinforcement learning

The remaining experiments compare BeS, GS-shrinkage, and BeS-shrinkage to GS and two algorithms from the literature that also aim to improve GS by choosing the distribution from which the directions are sampled to satisfy some criterion. In order to speed up convergence, Guided ES (Maheswaranathan et al., 2019) samples from a Gaussian distribution whose covariance matrix incorporates previous gradient estimates during optimization. Orthogonal ES (Choromanski et al., 2018) samples from a standard normal distribution and then orthogonalizes the directions, which reduces the MSE of the gradient estimate compared to GS.

We experiment on four RL benchmarks based on the MuoJoCo physics simulator (Todorov et al., 2012). Two environments are classic locomotion environments, where the goal is to learn a policy that successfully walks: *i) Ant* and *ii) Walker2d* from OpenAI Gym (Brockman et al., 2016). The other two environments are from meta-RL, where the goal is to learn a policy that succeeds over a distribution of tasks: *iii) ML1-Reach*: Introduced in Yu et al. (2019), tasks correspond to moving a robot arm to random locations. *iv) HalfCheetahRandVel* (Finn et al., 2017): tasks correspond to HalfCheetah locomotion robots with random target velocities. We use the version provided in the repository for Rothfuss et al. (2018). Experiments on additional environments are in Appendix C.2.

Concretely, the objective is the episodic reward of a linear policy. Following Mania et al. (2018), during optimization we standardize the observations, divide the rewards at each iteration by their standard deviation, and remove the survival bonus of Ant and Walker2d. Dividing the rewards at each iteration by their standard deviation stabilizes the optimization and removes the need for a tuned learning rate

[3] The learning rate suggested by Theorem 3.1 is not practical to compute since it includes the smoothness of the objective.

Table 1. Comparison of our proposed algorithms and GS for gradient estimation via (4). Recall that ϵ_{lj} is the random variable that each entry of the direction is sampled from, L is the number of sampled directions, d is the dimension of the parameter space, and B_p is a Bernoulli random variable with probability p.

ALGORITHM	DISTRIBUTION OF ϵ_{lj}	SMALLER MSE THAN
GS	$\mathcal{N}(0,1)$	
BES	$(B_{0.5}-0.5)/0.5$	GS
GS-SHRINKAGE	$\mathcal{N}(0, {}^{L}/_{L+d+1})$	GS
BES-SHRINKAGE	$(B_{0.5}-0.5)/m^*$, $m^*=\sqrt{{}^{L+d-1}/_{4L}}$	BES

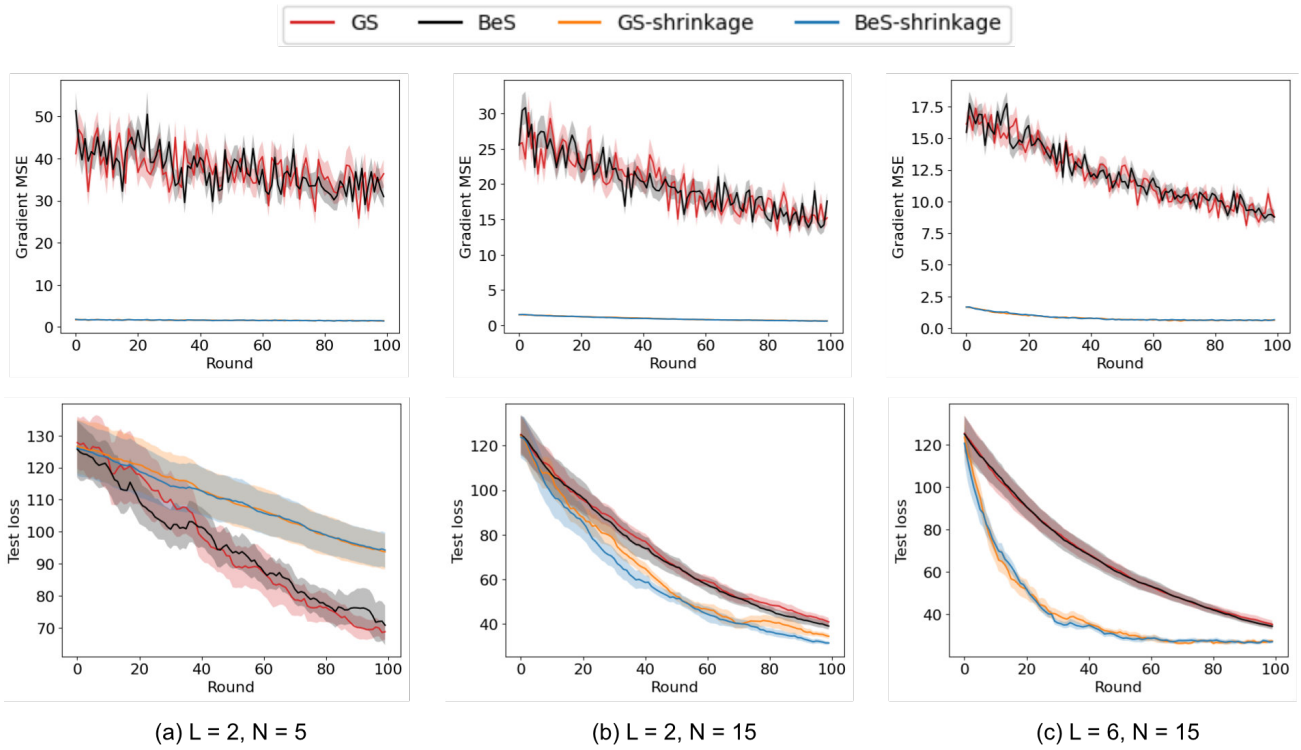


Figure 1. For linear regression with various L and N: MSE of the gradient (row 1) and test loss (row 2).

schedule, but may also diminish the benefit conferred by a decrease in gradient estimate MSE; therefore, GS has an advantage here. Figure 2 plots the episodic reward of the learned policy, tested on reinitializations of the environment (which includes the task for meta-RL); the horizontal axis is the number of trajectories generated in the simulator. Since the total number of evaluations to compute the gradient estimate is $2LN$, we see from Lemma 4.2 that given a budget of evaluations that we can obtain at one time, having L be as large as possible minimizes the MSE. Thus, we show results for a range of values of L and $N = 1$, averaging over five randomly generated seeds; in this setting L would correspond to the number of robots available to collect data in the real world.

Table 2 displays the average computation time required to sample the directions in each optimization iteration for each algorithm on HalfCheetahRandVel, using L Intel Xeon E7-8890 v3 CPUs; these numbers only depend on the parameter dimension d. We do not include the time required to generate the trajectories in the simulator, as it would be the same for all algorithms and vary between different simulators. BeS, GS-shrinkage, and BeS-shrinkage have the same computational complexity as GS, but Guided ES and Orthogonal ES have higher complexity because they require Gram-Schmidt orthonormalization. As expected, BeS, GS-shrinkage, and BeS-shrinkage have similar direction sampling time to GS, while Orthogonal ES takes at least $3\times$ more time and Guided ES at least $10\times$ more time.

In the majority of cases in Figure 2, BeS learns more quickly and achieves higher reward than GS. GS-shrinkage and BeS-shrinkage are even better in the three locomotion environments, in particular when fewer trajectories are generated at each iteration. For Ant, GS-shrinkage and BeS-shrinkage outperform the other algorithms by a large margin, albeit

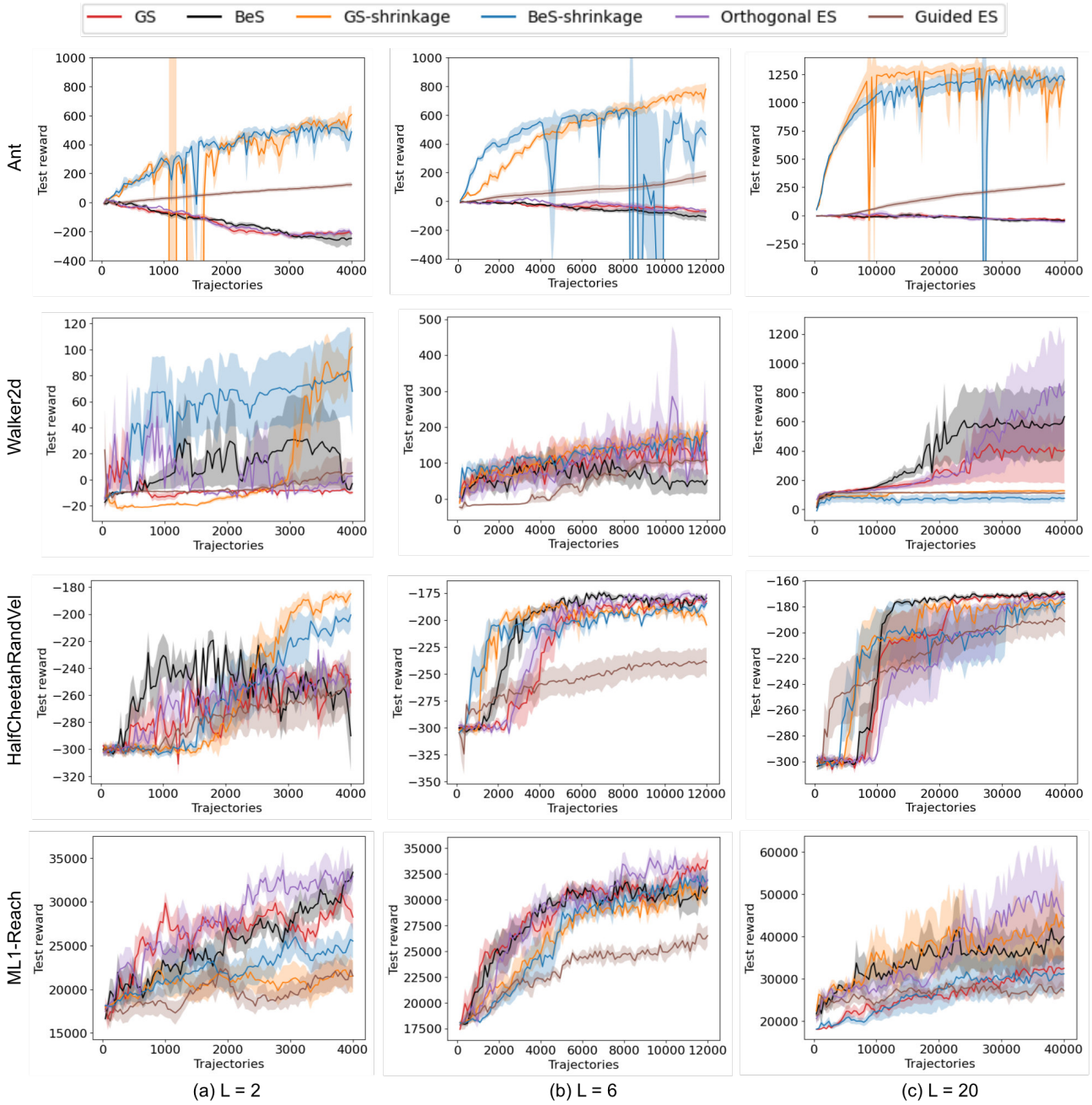


Figure 2. For RL with various L: test episodic reward

with some instability, which may be reduced by adding momentum to the optimization. Guided ES is the only other algorithm that learns, but achieves lower reward. For Walker2d $L = 2$ and $L = 6$, BeS-shrinkage outperforms all other algorithms, closely followed by GS-shrinkage; when $L = 20$, BeS and Orthogonal ES are the best. For HalfCheetahRandVel, BeS-shrinkage and GS-shrinkage learn a successful policy the fastest, but BeS and GS are able to catch up by the end of optimization for $L = 6$ and $L = 20$. However, for ML1-Reach, a manipulation environment, BeS and Orthogonal ES are the best algorithms, outperforming the others in two out of three cases. Overall, Guided ES is nearly always beaten by BeS, GS-shrinkage, or BeS-shrinkage. Likewise, Orthogonal ES is at most competitive with those three algorithms, with the exception of ML1-Reach and Walker2d $L = 20$ at the end of optimization.

Table 2. Time (10^{-5} s) to sample directions in each optimization iteration for each algorithm, on HalfCheetahRandVel.

ALGORITHM	$L = 2$	$L = 6$	$L = 20$
GES	7.5 ± 0.1	13.1 ± 0.1	19.9 ± 0.1
BES	8.9 ± 0.1	13.5 ± 0.1	21.6 ± 0.2
GES-SHRINKAGE	8.5 ± 0.1	13.6 ± 0.1	24.0 ± 0.2
BES-SHRINKAGE	8.7 ± 0.1	14.2 ± 0.1	19.1 ± 0.1
ORTHOGONAL ES	22.1 ± 0.1	50.8 ± 0.2	96.4 ± 0.5
GUIDED ES	193.0 ± 1.9	228.0 ± 1.8	222.0 ± 2.2

5.3. Ablations

Our next experiment studies whether the conclusions in the previous subsection generalize to using i) a neural network policy or ii) the antithetic gradient estimator. We repeat the set-up of Section 5.2, but only on Ant with $L = 20$. Figure 3 plots the test episodic reward of the learned policy against the number of trajectories generated during optimization.

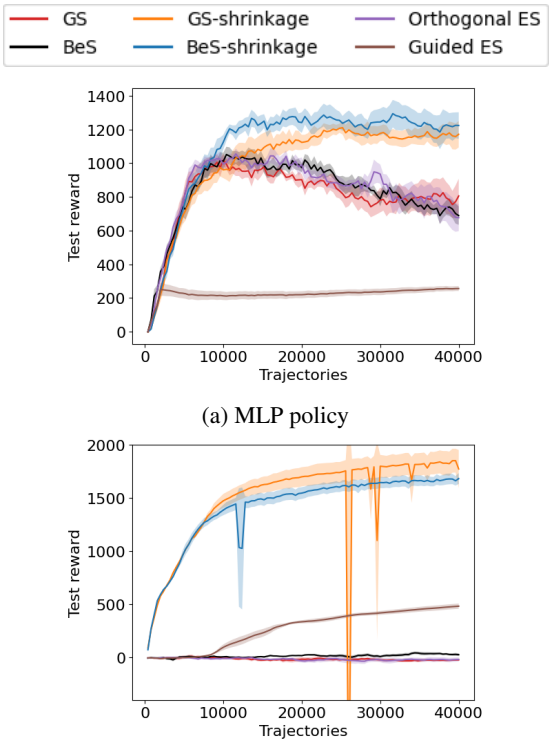
Neural network policy Instead of a linear policy, we use a MLP policy with one hidden layer of 32 nodes and tanh activations, which is popular in the policy gradient RL literature (Finn et al., 2017). GS-shrinkage and BeS-shrinkage outperform the other algorithms, with BeS-shrinkage learning statistically significantly faster. Guided ES falls behind GS, BeS, and Orthogonal ES, which start out strong but deteriorates, indicating difficulty with tuning learning rates.

Antithetic gradient estimator Instead of the forward difference gradient estimator (4), we use the antithetic gradient estimator (10). We show in Appendix B.5 that doing so does not affect the validity of BeS, GS-shrinkage, and BeS-shrinkage, but may be helpful depending on characteristics of the objective (Choromanski et al., 2018). We see that this is indeed true, the performance of all algorithms improve. However, their qualitative behavior remains the same; GS-shrinkage and BeS-shrinkage achieve the highest reward by far, with Guided ES the only other algorithm that learns.

5.4. DFO benchmarks

Finally, we experiment on the *noisy* benchmark from Nevergrad (Rapin & Teytaud, 2018), a DFO library. It consists of four classical minimization objectives, *sphere*, *rosenbrock*, *cigar*, *hm*, with only noisy evaluations available during optimization. We consider data dimensions $d = 10$ or $d = 100$ and a range of values of L for computing the gradient estimate at each iteration. As in Section 5.2, we show results for $N = 1$ averaged over five randomly generated seeds.

Figure 4 plots the objective as the optimization progresses for the six algorithms. In most cases, none of the algorithms were able to minimize *cigar*, possibly because it is too ill-conditioned to find good perturbation directions without very large L. Therefore, we do not include it in our plots.



(a) MLP policy

(b) Antithetic gradient estimator

Figure 3. For Ant with $L = 20$

The qualitative results are similar for all three objectives; GS, BeS, and Orthogonal ES are the best algorithms. When $L < d$ (first two columns), BeS often outperforms GS and Orthogonal ES at the beginning of optimization, but is overtaken by them at the end of optimization. When $L = d$ (last column), GS and BeS are not statistically significantly different, and outperform Orthogonal ES, which now behaves similarly to GS-shrinkage and BeS-shrinkage.

5.5. Discussion

The experiments show that by designing the distribution of the directions to minimize the MSE of the gradient, we are often able to obtain superior test performance to GS while remaining as computationally efficient, especially when there are few data points available at each iteration. Moreover, we are competitive with, and in many cases outperform, previously proposed algorithms to improve GS that are more computationally expensive.

One limitation of our work is that GS-shrinkage and BeS-shrinkage do not always outperform GS and BeS, although they have smaller gradient estimate MSE. We hypothesize that this is due to a form of the bias-variance trade-off. Theorem 3.1 shows that the convergence bound includes the bias

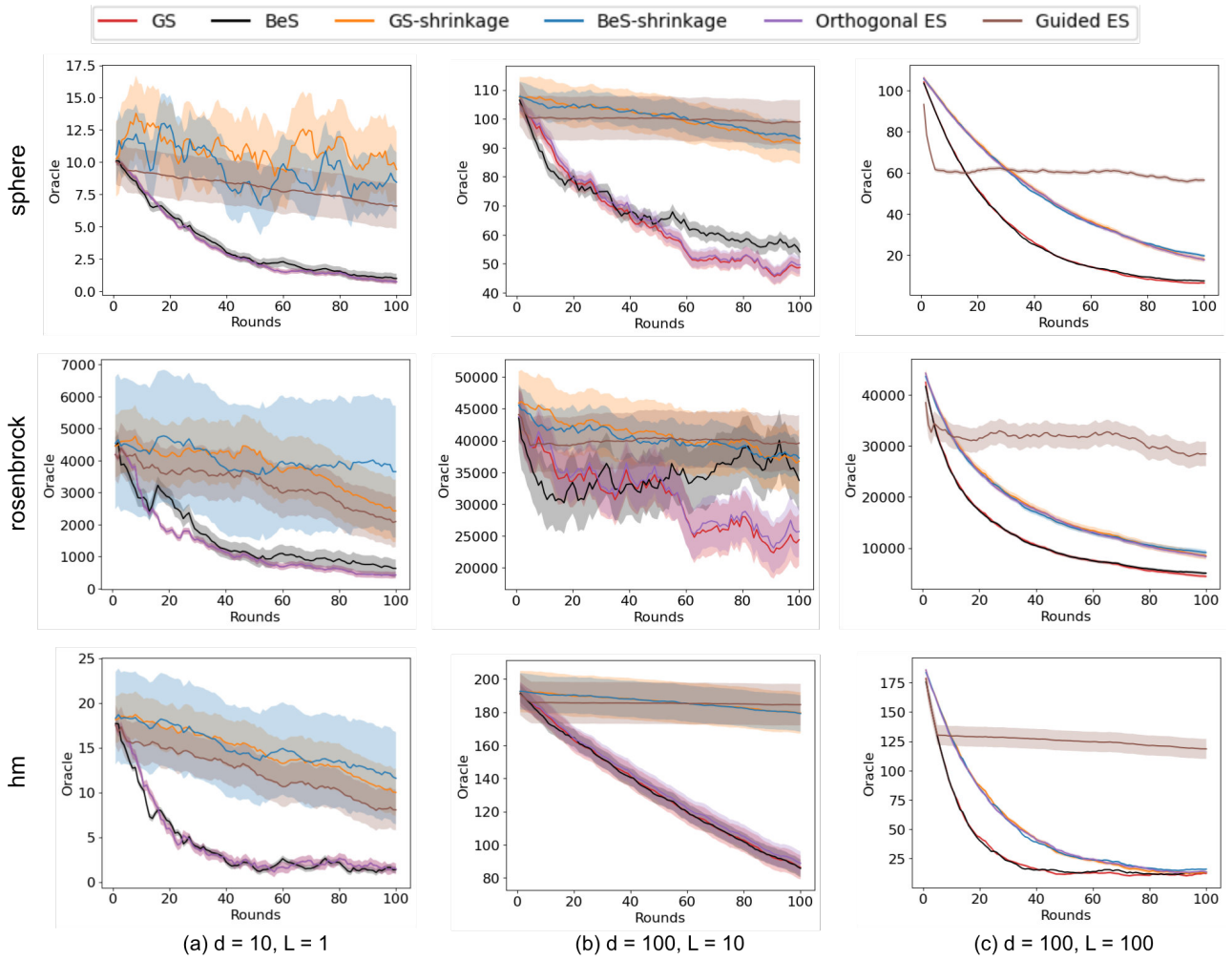


Figure 4. For *noisy* DFO benchmark: objective values. When $L = 1$, GS is the same as Orthogonal ES.

of the gradient estimate as well as the MSE in a non-linear and problem-dependent way. GS and BeS have smaller bias than GS-shrinkage and BeS-shrinkage but larger variance, so it is not surprising that their relative effectiveness is problem-dependent. Exploring how to adaptively balance the trade-off is an exciting avenue for future work.

The assumptions made highlight other potential directions for future work. We may allow the directions to have dependent entries, learn the distribution type instead of fixing it, or consider how different optimization algorithms may affect the nature of the optimal distribution. We could also design algorithms that, instead of minimizing only the largest term of the MSE (5), apply Follow the Regularized Leader (Borsos et al., 2018) to minimize the entire MSE over all optimization iterations using estimates of the gradients of the objective and the random evaluation.

6. Conclusion

In this paper, we generalize Gaussian smoothing to sample directions from arbitrary distributions. Doing us enables us to choose distributions that minimize the MSE of the gradient estimates and speed up optimization convergence. We construct three distributions that lead to lower MSE than the standard normal without needing any information about the objective. Experiments on linear regression confirm our theoretical results and experiments on reinforcement learning and DFO benchmarks show that the derived algorithms often improve over GS.

Acknowledgements

We would like to thank the other members of Emergent AI at Intel Labs for providing helpful feedback on the submission.

References

Ajalloeian, A. and Stich, S. U. Analysis of SGD with biased gradient estimators. *arXiv preprint arXiv:2008.00051*, 2020.

Billingsley, P. *Probability and Measure*. Wiley Series in Probability and Statistics. Wiley, 1995.

Borsos, Z., Krause, A., and Levy, K. Y. Online variance reduction for stochastic optimization. In *Conference On Learning Theory*, pp. 324–357. PMLR, 2018.

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

Chen, J. and Luss, R. Stochastic gradient descent with biased but consistent gradient estimators. *arXiv preprint arXiv:1807.11880*, 2018.

Chen, R. and Wild, S. Randomized derivative-free optimization of noisy convex functions. *arXiv preprint arXiv:1507.03332*, 2015.

Choromanski, K., Rowland, M., Sindhwani, V., Turner, R., and Weller, A. Structured evolution with compact architectures for scalable policy optimization. In *International Conference on Machine Learning*, pp. 970–978. PMLR, 2018.

Conn, A. R., Scheinberg, K., and Vicente, L. N. *Introduction to derivative-free optimization*. SIAM, 2009.

Custódio, A., Scheinberg, K., and Vicente, L. *Advances and Trends in Optimization with Engineering Applications*, chapter 37: Methodologies and Software for Derivative-Free Optimization, pp. 495–506. SIAM, 2017.

DeCarlo, L. T. On the meaning and use of kurtosis. *Psychological Methods*, 2:292–307, 1997.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1126–1135. JMLR, 2017.

Flaxman, A. D., Kalai, A. T., and McMahan, H. B. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 385–394, 2005.

Gao, K. and Sener, O. Modeling and optimization trade-off in meta-learning. *Advances in Neural Information Processing Systems*, 33, 2020.

Hansen, N., Müller, S. D., and Koumoutsakos, P. Reducing the time complexity of the derandomized evolution strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary Computation*, 11:1–18, 2003.

Hu, X., Prashanth, L., György, A., and Szepesvari, C. (bandit) convex optimization with biased noisy gradient oracles. In *Artificial Intelligence and Statistics*, pp. 819–828. PMLR, 2016.

Lebanon, G. Bias, variance, and mse of estimators. http://theanalysisofdata.com/notes/estimators1.pdf, 2010. Accessed: 2022-01-02.

Maheswaranathan, N., Metz, L., Tucker, G., Choi, D., and Sohl-Dickstein, J. Guided evolutionary strategies: Augmenting random search with surrogate gradients. In *International Conference on Machine Learning*, pp. 4264–4273. PMLR, 2019.

Mania, H., Guy, A., and Recht, B. Simple random search of static linear policies is competitive for reinforcement learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 1805–1814, 2018.

Matyas, J. Random optimization. *Automation and Remote Control*, 26(2):246–253, 1965.

Nesterov, Y. and Spokoiny, V. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.

Polyak, B. *Introduction to optimization*. Optimization Software, New York, 1987.

Rapin, J. and Teytaud, O. Nevergrad - A gradient-free optimization platform. https://GitHub.com/FacebookResearch/Nevergrad, 2018.

Rothfuss, J., Lee, D., Clavera, I., Asfour, T., and Abbeel, P. ProMP: Proximal meta-policy search. In *International Conference on Learning Representations*, 2018.

Salimans, T., Ho, J., Chen, X., Sidor, S., and Sutskever, I. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

Sener, O. and Koltun, V. Learning to guide random search. In *International Conference on Learning Representations*, 2019.

Todorov, E., Erez, T., and Tassa, Y. MuJoCo: A physics engine for model-based control. In *Intelligent Robots and Systems (IROS)*, 2012.

Weisstein, E. W. "Bernoulli distribution". from Mathworld - a Wolfram web resource, a. URL https://mathworld.wolfram.com/BernoulliDistribution.html. Accessed 2021-05-17.

Weisstein, E. W. "Normal distribution". from Mathworld - a Wolfram web resource, b. URL https://mathworld.wolfram.com/NormalDistribution.html. Accessed 2021-05-17.

Wierstra, D., Schaul, T., Glasmachers, T., Sun, Y., Peters, J., and Schmidhuber, J. Natural evolution strategies. *The Journal of Machine Learning Research*, 15(1):949–980, 2014.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., and Levine, S. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2019.

Zeidler, E. *Nonlinear functional analysis and its applications. I: Fixed-point theorems*, volume 12. Springer, 1986.

A. Proofs for Section 3

A.1. Proof of Theorem 3.1

Since $F(\theta)$ is differentiable and μ-smooth,

$$F(\theta^{t+1}) \leq F(\theta^t) - \eta \nabla_\theta F(\theta^t)^\intercal g(\theta^t) + \frac{\mu\eta^2}{2}\|g(\theta^t)\|_2^2.$$

For convenience, let b^t and v^t be the bias and variance of g^t, respectively. Taking the expectation with respect to the randomness in g^t,

$$\begin{aligned}\mathbb{E}[F(\theta^{t+1})] &\leq F(\theta^t) - \eta \nabla_\theta F(\theta^t)^\intercal (\nabla_\theta F(\theta^t) + b^t) + \frac{\mu\eta^2}{2}(\|\nabla_\theta F(\theta^t) + b^t\|_2^2 + \operatorname{tr}(v^t)) \\ &= F(\theta^t) - \eta\|\nabla_\theta F(\theta^t)\|_2^2 - \eta \nabla_\theta F(\theta^t)^\intercal b^t + \frac{\mu\eta^2}{2}\|\nabla_\theta F(\theta^t)\|_2^2 + \frac{\mu\eta^2}{2}\|b^t\|_2^2 + \mu\eta^2 \nabla_\theta F(\theta^t)^\intercal b^t + \frac{\mu\eta^2}{2}\operatorname{tr}(v^t)\end{aligned}$$

The first inequality uses the definition of variance.

Since $\eta \leq 1/\mu$, we have

$$\begin{aligned}\mathbb{E}[F(\theta^{t+1})] &\leq F(\theta^t) - \frac{\eta}{2}\|\nabla_\theta F(\theta^t)\|_2^2 + \frac{\mu\eta^2}{2}M + \eta(\mu\eta - 1)\nabla_\theta F(\theta^t)^\intercal b^t \\ &\leq F(\theta^t) - \frac{\eta}{2}\|\nabla_\theta F(\theta^t)\|_2^2 + \frac{\mu\eta^2}{2}M + \eta B\|\nabla_\theta F(\theta^t)\|_2^2\end{aligned}$$

where the first line follows from the fact that the MSE of an estimate can be decomposed into the sum of the trace of the variance and the squared norm of the bias (Lebanon, 2010) and the second line follows from the assumption on the norm of the bias of g^t.

Rearranging, we obtain for $B < \frac{1}{2}$

$$\begin{aligned}\eta(\frac{1}{2} - B)\|\nabla_\theta F(\theta^t)\|_2^2 &\leq F(\theta^t) - \mathbb{E}[F(\theta^{t+1})] + \frac{\mu\eta^2}{2}M \\ \frac{1}{T}\sum_t \|\nabla_\theta F(\theta^t)\|_2^2 &\leq \frac{4\Delta}{\eta(1-2B)T} + \frac{\mu\eta M}{1-2B} \\ &= \frac{M + 4\Delta\mu}{(1-2B)\sqrt{T}} \quad \text{for } \eta = \frac{1}{\mu\sqrt{T}}.\end{aligned}$$

B. Proofs for Section 4

We first present a lemma that will be useful for subsequent proofs.

Lemma B.1. *Suppose that the d entries of a vector ϵ are IID samples from a distribution with expectation 0, variance σ^2, and kurtosis k. For any matrix A, $\operatorname{tr}(\mathbb{E}_\epsilon(\epsilon\epsilon^\intercal A \epsilon\epsilon^\intercal)) = \sigma^4(d + k - 1)\operatorname{tr}(A)$.*

Proof. It suffices to sum the diagonal entries of $\mathbb{E}_\epsilon(\epsilon\epsilon^\intercal A\epsilon\epsilon^\intercal)$. The j^{th} entry is

$$\begin{aligned}\mathbb{E}_\epsilon(\epsilon\epsilon^\intercal A\epsilon\epsilon^\intercal)_{jj} &= \mathbb{E}_\epsilon(\epsilon_j^2 \sum A_{ab}\epsilon_a\epsilon_b) \\ &= \mathbb{E}_\epsilon(\epsilon_j^4)A_{jj} + \mathbb{E}_\epsilon(\epsilon_j^2)\mathbb{E}_\epsilon(\epsilon_b^2)\sum_{b\neq j} A_{bb} \\ &= k\sigma^4 A_{jj} + \sigma^4(\operatorname{tr}(A) - A_{jj}) = \sigma^4(\operatorname{tr}(A) + (k-1)A_{jj})\end{aligned}$$

Summing up over j,

$$\operatorname{tr}(\mathbb{E}_\epsilon(\epsilon\epsilon^\intercal A\epsilon\epsilon^\intercal)) = \sigma^4(d\operatorname{tr}(A) + (k-1)\operatorname{tr}(A)) = \sigma^4(d + k - 1)\operatorname{tr}(A)$$

□

B.1. Proof of Lemma 4.2

The MSE of an estimator can be decomposed as the sum of the squared norm of its bias and the trace of its variance (Lebanon, 2010). Thus, we compute the bias and the variance of $\nabla_\theta \hat{F}^{FD}(\theta)$ separately.

Bias

$$\begin{aligned}
\mathbb{E}[\nabla_\theta \hat{F}^{FD}(\theta)] &= \mathbb{E}_{\epsilon,\xi}\left[\frac{1}{cLN}\sum_{l,i}(f(\theta + c\epsilon_l, \xi_i) - f(\theta, \xi_i))\epsilon_l\right] \\
&= \mathbb{E}_\epsilon\left[\frac{1}{cL}\sum_{l}(F(\theta + c\epsilon_l) - F(\theta))\epsilon_l\right] \\
&= \mathbb{E}_\epsilon\left[\frac{1}{c}(F(\theta + c\epsilon) - F(\theta))\epsilon\right] \\
&= \mathbb{E}_\epsilon\left[\frac{1}{c}(c\nabla_\theta F(\theta)^\intercal \epsilon + c^2\epsilon^\intercal h(\theta + c\epsilon)\epsilon)\epsilon\right]
\end{aligned}$$

where the last equality follows from Taylor's Theorem (Zeidler, 1986) and h is some scalar-valued function such that $h(y) \to 0$ as $y \to \theta$.

$$\begin{aligned}
\mathbb{E}[\nabla_\theta \hat{F}^{FD}(\theta)] &= \mathbb{E}_\epsilon\left[\epsilon(\epsilon^\intercal \nabla_\theta F(\theta) + c\epsilon^\intercal h(\theta + c\epsilon)\epsilon)\right] \\
&= \sigma^2 \nabla_\theta F(\theta) + c\mathbb{E}_\epsilon[\epsilon\epsilon^\intercal h(\theta + c\epsilon)\epsilon]
\end{aligned}$$

where σ^2 is the variance of each entry of ϵ. If the Dominated Convergence Theorem (Billingsley, 1995) holds, i.e. $|\epsilon\epsilon^\intercal h(\theta + c\epsilon)\epsilon|$ is upper bounded by some integrable function of ϵ, then as $c \to 0$,

$$\mathbb{E}[\nabla_\theta \hat{F}^{FD}(\theta)] \to \sigma^2 \nabla_\theta F(\theta),$$

and the squared norm of the bias of $\nabla_\theta \hat{F}^{FD}(\theta)$ is $(\sigma^2 - 1)^2 \|\nabla_\theta F(\theta)\|_2^2$.

Variance Using the law of total variance (Billingsley, 1995),

$$\begin{aligned}
\mathrm{Var}[\nabla_\theta \hat{F}^{FD}(\theta)] &= \mathrm{Var}_\epsilon(\mathbb{E}_\xi[\nabla_\theta \hat{F}^{FD}(\theta) \mid \epsilon]) + \mathbb{E}_\epsilon(\mathrm{Var}_\xi[\nabla_\theta \hat{F}^{FD}(\theta) \mid \epsilon]) \\
&= \mathrm{Var}_\epsilon\left[\frac{1}{cL}\sum_{l}(F(\theta + c\epsilon_l) - F(\theta))\epsilon_l\right] + \mathbb{E}_\epsilon\left(\frac{1}{c^2 L}\,\mathrm{Var}_\xi\left[\frac{1}{N}\sum_{i}(f(\theta + c\epsilon, \xi_i) - f(\theta, \xi_i))\epsilon \mid \epsilon\right]\right) \\
&= \frac{1}{c^2 L}\,\mathrm{Var}_\epsilon\left[(F(\theta + c\epsilon) - F(\theta))\epsilon\right] + \frac{1}{c^2 LN}\mathbb{E}_\epsilon\left(\mathrm{Var}_\xi\left[(f(\theta + c\epsilon, \xi) - f(\theta, \xi))\epsilon \mid \epsilon\right]\right) \\
&= \frac{1}{c^2 L}\,\mathrm{Var}_\epsilon\left[\epsilon(c\epsilon^\intercal \nabla_\theta F(\theta) + c^2\epsilon^\intercal h(\theta + c\epsilon)\epsilon)\right] \\
&\quad + \frac{1}{c^2 LN}\mathbb{E}_\epsilon\left(\mathrm{Var}_\xi\left[\epsilon(c\epsilon^\intercal \nabla_\theta f(\theta, \xi) + c^2\epsilon^\intercal h'(\theta + c\epsilon, \xi)\epsilon)\right]\right) \\
&= \frac{1}{L}\,\mathrm{Var}_\epsilon\left[\epsilon(\epsilon^\intercal \nabla_\theta F(\theta) + c\epsilon^\intercal h(\theta + c\epsilon)\epsilon)\right] + \frac{1}{LN}\mathbb{E}_\epsilon\left(\mathrm{Var}_\xi\left[\epsilon(\epsilon^\intercal \nabla_\theta f(\theta, \xi) + c\epsilon^\intercal h'(\theta + c\epsilon, \xi)\epsilon)\right]\right)
\end{aligned}$$

where the next-to-last equality follows from Taylor's Theorem and h' is some scalar-valued function with the same condition as h. Assuming that the Dominated Convergence Theorem holds again, as $c \to 0$,

$$\begin{aligned}
\mathrm{Var}[\nabla_\theta \hat{F}^{FD}(\theta)] &\to \frac{1}{L}\,\mathrm{Var}_\epsilon\left[\epsilon\epsilon^\intercal \nabla_\theta F(\theta)\right] + \frac{1}{LN}\mathbb{E}_\epsilon\left(\mathrm{Var}_\xi\left[\epsilon\epsilon^\intercal \nabla_\theta f(\theta, \xi)\right]\right) \\
&= \frac{1}{L}\mathbb{E}_\epsilon\left[\epsilon\epsilon^\intercal \nabla_\theta F(\theta)\nabla_\theta F(\theta)^\intercal \epsilon\epsilon^\intercal\right] - \frac{1}{L}\mathbb{E}_\epsilon\left[\epsilon\epsilon^\intercal \nabla_\theta F(\theta)\right]\mathbb{E}_\epsilon\left[\epsilon\epsilon^\intercal \nabla_\theta F(\theta)\right]^\intercal \\
&\quad + \frac{1}{LN}\mathbb{E}_\epsilon\left(\epsilon\epsilon^\intercal\,\mathrm{Var}_\xi\left[\nabla_\theta f(\theta, \xi)\right]\epsilon\epsilon^\intercal\right)
\end{aligned}$$

Using Lemma B.1, as $c \to 0$,

$$\begin{aligned}\operatorname{tr}\left(\operatorname{Var}[\nabla_\theta \hat{F}^{FD}(\theta)]\right) &\to \frac{1}{L}\sigma^4(d+k-1)\operatorname{tr}(\nabla_\theta F(\theta)\nabla_\theta F(\theta)^\intercal) - \frac{1}{L}\operatorname{tr}(\sigma^4 \nabla_\theta F(\theta)\nabla_\theta F(\theta)^\intercal) \\ &\quad + \frac{1}{LN}\sigma^4(d+k-1)\operatorname{tr}(\operatorname{Var}_\xi[\nabla_\theta f(\theta,\xi)]) \\ &= \frac{\sigma^4}{L}(d+k-2)\|\nabla_\theta F(\theta)\|_2^2 + \frac{\sigma^4}{LN}(d+k-1)\operatorname{tr}(\operatorname{Var}_\xi[\nabla_\theta f(\theta,\xi)])\end{aligned}$$

Thus, the MSE of $\nabla_\theta \hat{F}^{FD}(\theta)$ is $((\sigma^2-1)^2 + \frac{\sigma^4}{L}(d+k-2))\|\nabla_\theta F(\theta)\|_2^2 + \frac{\sigma^4}{LN}(d+k-1)\operatorname{tr}(\operatorname{Var}_\xi[\nabla_\theta f(\theta,\xi)])$.

B.2. Proof of Theorem 4.3

We first make a change of variable. Let $x = \sigma^2$. Then, (7) becomes

$$\min_{x>0} O(x) \triangleq (x-1)^2 + \frac{d+1}{L}x^2.$$

Simplifying, $O(x) = (1+\frac{d+1}{L})x^2 - 2x + 1$. It is clear that $O(x)$ is a convex quadratic function, with minimum at $x^* = \frac{L}{L+d+1} > 0$. Thus, $\sigma^{2*} = \frac{L}{L+d+1}$.

By definition, σ^{2*} minimizes (5). Since $\sigma^{2*} < 1$ and $k = 3$ for both $\mathcal{N}(0,1)$ and $\mathcal{N}(0,\sigma^{2*})$, it follows that the value of (6) is smaller for $\epsilon_{lj} \sim \mathcal{N}(0,\sigma^{2*})$ than for $\epsilon_{lj} \sim \mathcal{N}(0,1)$. Hence, the MSE of $\nabla_\theta \hat{F}^{FD}(\theta)$ is smaller for $\epsilon_{lj} \sim \mathcal{N}(0,\sigma^{2*})$ than for $\epsilon_{lj} \sim \mathcal{N}(0,1)$.

B.3. Proof of Theorem 4.4

We first make a change of variable. Let $x = p(1-p)$ and $y = m^2$. Then, (8) becomes

$$\min_{x\in(0,0.25],y>0} O(x,y) \triangleq \left(\frac{x}{y} - 1\right)^2 + \frac{x^2}{y^2 L}\left(d + 1 + \frac{1-6x}{x}\right).$$

Simplifying,

$$\begin{aligned}O(x,y) &= \frac{x^2}{y^2} - \frac{2x}{y} + 1 + \frac{x^2}{y^2}\frac{d-5}{L} + \frac{x}{y^2 L} \\ &= \frac{x^2}{y^2}\frac{L+d-5}{L} + x\frac{1-2yL}{y^2 L} + 1\end{aligned}$$

which is convex in x for fixed y if $L + d > 5$. Next, we solve for x in terms of y, obtaining $x^* = \frac{2yL-1}{2(L+d-5)}$ if x were unconstrained.

However, since x is constrained to $(0, 0.25]$, there are three cases:

1. If $y \le \frac{1}{2L}$: The lower constraint is active. $x^* = 0$ and $O(x^*, y) = 1$.
2. If $y > \frac{L+d-3}{4L}$: The upper constraint is active. $x^* = 0.25$ and

$$\begin{aligned}O(x^*,y) &= 1 + \frac{L+d-5}{16y^2 L} + \frac{1-2yL}{4y^2 L} = 1 - \frac{1}{2y} + \frac{L+d-1}{16y^2 L} \\ &< 1 - \frac{1}{2y} + \frac{4yL+2}{16y^2 L} = 1 - \frac{1}{4y} + \frac{1}{8y^2 L}\end{aligned}$$

   where the inequality in the second line follows from the condition on y.

3. If $y > \frac{1}{2L}$ and $y \leq \frac{L+d-3}{4L}$: Neither constraint is active. $x^* = \frac{2yL-1}{2(L+d-5)}$ and

$$O(x^*, y) = 1 - \frac{(1-2yL)^2}{y^4L^2} \Big/ \frac{4(L+d-5)}{y^2L} = 1 - \frac{(1-2yL)^2}{4y^2L(L+d-5)}$$
$$\geq 1 - \frac{(2yL-1)^2}{4y^2L(4yL-2)} = 1 - \frac{2yL-1}{8y^2L} = 1 - \frac{1}{4y} + \frac{1}{8y^2L}$$

where the inequality in the second line follows from $y \leq \frac{L+d-3}{4L}$.

In cases 2 and 3, $y > \frac{1}{2L}$, so $1 - \frac{1}{4y} + \frac{1}{8y^2L} < 1$. Therefore, the smallest possible value of $O(x^*, y)$ occurs in case 2, with $x^* = 0.25$, $y > \frac{L+d-3}{4L}$, and $Q(y) = O(x^*, y) = 1 - \frac{1}{2y} + \frac{L+d-1}{16y^2L}$.

Finally, we minimize $Q(y)$ restricted to $y > \frac{L+d-3}{4L}$. Observe that $Q(y)$ is a convex quadratic function of $\frac{1}{y}$, so $\frac{1}{y^*} = \frac{4L}{L+d-1}$, or $y^* = \frac{L+d-1}{4L}$.

Thus, $x^* = 0.25$ and $y^* = \frac{L+d-1}{4L}$, corresponding to $p^* = 0.5$ and $m^* = \sqrt{\frac{L+d-1}{4L}}$.

By definition, $p^* = 0.5$ and $m^* = \sqrt{\frac{L+d-1}{4L}}$ minimizes (5). The kurtosis of $\frac{B_p^* - p^*}{m^*}$ is the same as that of $\frac{B_{0.5}^* - 0.5}{0.5}$; however, it has smaller variance since $m^* > 0.5$. Hence, the MSE of $\nabla_\theta \hat{F}^{FD}(\theta)$ is smaller for $\epsilon_{lj} \sim \frac{B_p^* - p^*}{m^*}$ than for $\epsilon_{lj} \sim \frac{B_{0.5}^* - 0.5}{0.5}$.

B.4. GS-shrinkage or BeS-shrinkage?

To analyze whether GS-shrinkage or BeS-shrinkage is superior, we compare their MSEs for the gradient estimator (4), using Lemma 4.2. The MSE for GS-shrinkage is

$$MSE(GSs) = \left(\frac{(d+1)^2}{(L+d+1)^2} + \frac{L^2(d+1)}{L(L+d+1)^2}\right)\|\nabla_\theta F(\theta)\|_2^2 + \frac{L^2(d+2)}{LN(L+d+1)^2}\operatorname{tr}(\operatorname{Var}_\xi[\nabla_\theta f(\theta,\xi)])$$
$$= \frac{(d+1)^2 + L(d+1)}{(L+d+1)^2}\|\nabla_\theta F(\theta)\|_2^2 + \frac{L(d+2)}{N(L+d+1)^2}\operatorname{tr}(\operatorname{Var}_\xi[\nabla_\theta f(\theta,\xi)])$$

The MSE for BeS-shrinkage is

$$MSE(BeSs) = \left(\left(1 - \frac{4L}{4(L+d-1)}\right)^2 + \frac{16L^2(d-1)}{16L(L+d-1)^2}\right)\|\nabla_\theta F(\theta)\|_2^2$$
$$+ \frac{d(16L^2)}{16LN(L+d-1)^2}\operatorname{tr}(\operatorname{Var}_\xi[\nabla_\theta f(\theta,\xi)])$$
$$= \left(\frac{(d-1)^2}{(L+d-1)^2} + \frac{L(d-1)}{(L+d-1)^2}\right)\|\nabla_\theta F(\theta)\|_2^2 + \frac{dL}{N(L+d-1)^2}\operatorname{tr}(\operatorname{Var}_\xi[\nabla_\theta f(\theta,\xi)])$$

Therefore,

$$MSE(GSs) - MSE(BeSs)$$
$$= \|\nabla_\theta F(\theta)\|_2^2\left(\frac{d+1}{L+d+1} - \frac{d-1}{L+d-1}\right) + \frac{\operatorname{tr}(\operatorname{Var}_\xi[\nabla_\theta f(\theta,\xi)])L}{N}\left(\frac{d+2}{(L+d+1)^2} - \frac{d}{(L+d-1)^2}\right)$$

$$
\begin{aligned}
&= \|\nabla_\theta F(\theta)\|_2^2 \frac{(d+1)(L+d-1)-(d-1)(L+d+1)}{(L+d+1)(L+d-1)} \\
&\quad + \operatorname{tr}(\operatorname{Var}_\xi[\nabla_\theta f(\theta,\xi)]) \frac{L}{N} \frac{(d+2)(L+d-1)^2 - d(L+d+1)^2}{(L+d+1)^2(L+d-1)^2} \\
&= \|\nabla_\theta F(\theta)\|_2^2 \frac{2L}{(L+d+1)(L+d-1)} + \operatorname{tr}(\operatorname{Var}_\xi[\nabla_\theta f(\theta,\xi)]) \frac{2L}{N} \frac{L^2-2L+2-(d+1)^2}{(L+d+1)^2(L+d-1)^2} \\
&= \frac{2L}{(L+d-1)(L+d+1)} \left(\|\nabla_\theta F(\theta)\|_2^2 + \frac{L^2-2L+2-(d+1)^2}{N(L+d-1)(L+d+1)} \operatorname{tr}(\operatorname{Var}_\xi[\nabla_\theta f(\theta,\xi)]) \right)
\end{aligned}
$$

At the beginning of optimization, $\|\nabla_\theta F(\theta)\|_2^2$ is large. If it is large compared to $\operatorname{tr}(\operatorname{Var}_\xi[\nabla_\theta f(\theta,\xi)])/N$, since $L^2-2L+2-(d+1)^2<0$ for high-dimensional problems, $MSE(GSs) > MSE(BeSs)$.

At the end of optimization, $\|\nabla_\theta F(\theta)\|_2^2 \approx 0$. Then, since $L^2-2L+2-(d+1)^2<0$ for high-dimensional problems, $MSE(GSs) < MSE(BeSs)$.

B.5. Algorithms for the antithetic gradient estimator

Suppose that instead of the forward difference gradient estimator, we use the antithetic gradient estimator. Mathematically, given N samples from the oracle ξ_i and L IID sampled directions ϵ_l, let

$$
\nabla_\theta \hat{F}^{AT}(\theta) = \frac{1}{2cLN} \sum_{l,i} (f(\theta + c\epsilon_l, \xi_i) - f(\theta - c\epsilon_l, \xi_i))\epsilon_l \tag{10}
$$

Under Assumption 4.1, we compute the MSE of $\nabla_\theta \hat{F}^{AT}(\theta)$, with the same strategy as for $\nabla_\theta \hat{F}^{FD}(\theta)$ (Lemma 4.2).

Bias

$$
\begin{aligned}
\mathbb{E}[\nabla_\theta \hat{F}^{AT}(\theta)] &= \mathbb{E}_{\epsilon,\xi}\left[\frac{1}{2cLN} \sum_{l,i} (f(\theta + c\epsilon_l, \xi_i) - f(\theta - c\epsilon_l, \xi_i))\epsilon_l \right] \\
&= \mathbb{E}_{\epsilon}\left[\frac{1}{2cL} \sum_{l} (F(\theta + c\epsilon_l) - F(\theta - c\epsilon_l))\epsilon_l \right] \\
&= \mathbb{E}_{\epsilon}\left[\frac{1}{2c} (F(\theta + c\epsilon) - F(\theta - c\epsilon_l))\epsilon \right] \\
&= \mathbb{E}_{\epsilon}\left[\frac{1}{2c} (2c\nabla_\theta F(\theta)^\intercal \epsilon + c^3 \sum_{|\alpha|_1=3} h_\alpha(\theta + c\epsilon)\epsilon^\alpha)\epsilon \right]
\end{aligned}
$$

where the last equality follows from Taylor's Theorem, α is a d-dimensional vector of non-negative integers, ϵ^α indicates element-wise power, and h_α is some scalar-valued function such that $h_\alpha(y) \to 0$ as $y \to \theta$.

$$
\begin{aligned}
\mathbb{E}[\nabla_\theta \hat{F}^{AT}(\theta)] &= \mathbb{E}_{\epsilon}\left[\epsilon(\epsilon^\intercal \nabla_\theta F(\theta) + c^2 \sum_{|\alpha|_1=3} h_\alpha(\theta + c\epsilon)\epsilon^\alpha) \right] \\
&= \sigma^2 \nabla_\theta F(\theta) + c^2 \mathbb{E}_\epsilon[\epsilon \sum_{|\alpha|_1=3} h_\alpha(\theta + c\epsilon)\epsilon^\alpha]
\end{aligned}
$$

where σ^2 is the variance of each entry of ϵ. If the Dominated Convergence Theorem (Billingsley, 1995) holds, then as $c \to 0$,

$$
\mathbb{E}[\nabla_\theta \hat{F}^{AT}(\theta)] \to \sigma^2 \nabla_\theta F(\theta),
$$

and the squared norm of the bias of $\nabla_\theta \hat{F}^{AT}(\theta)$ is the same as that of $\nabla_\theta \hat{F}^{FD}(\theta)$.

Variance Using the law of total variance (Billingsley, 1995),

$$\begin{aligned}
\mathrm{Var}[\nabla_\theta \hat{F}^{AT}(\theta)] &= \mathrm{Var}_\epsilon(\mathbb{E}_\xi[\nabla_\theta \hat{F}^{AT}(\theta) \mid \epsilon]) + \mathbb{E}_\epsilon(\mathrm{Var}_\xi[\nabla_\theta \hat{F}^{AT}(\theta) \mid \epsilon]) \\
&= \mathrm{Var}_\epsilon\left[\frac{1}{2cL}\sum_l (F(\theta + c\epsilon_l) - F(\theta - c\epsilon_l))\epsilon_l\right] \\
&\quad + \mathbb{E}_\epsilon\left(\frac{1}{4c^2 L}\,\mathrm{Var}_\xi\left[\frac{1}{N}\sum_i (f(\theta + c\epsilon, \xi_i) - f(\theta - c\epsilon, \xi_i))\epsilon \mid \epsilon\right]\right) \\
&= \frac{1}{4c^2 L}\,\mathrm{Var}_\epsilon\left[(F(\theta + c\epsilon) - F(\theta - c\epsilon))\epsilon\right] + \frac{1}{4c^2 LN}\mathbb{E}_\epsilon\left(\mathrm{Var}_\xi\left[(f(\theta + c\epsilon, \xi) - f(\theta - c\epsilon, \xi))\epsilon \mid \epsilon\right]\right) \\
&= \frac{1}{4c^2 L}\,\mathrm{Var}_\epsilon\left[\epsilon(2c\epsilon^\intercal \nabla_\theta F(\theta) + c^2 \sum_{|\alpha|_1=3} h_\alpha(\theta + c\epsilon)\epsilon^\alpha)\right] \\
&\quad + \frac{1}{4c^2 LN}\mathbb{E}_\epsilon\left(\mathrm{Var}_\xi\left[\epsilon(2c\epsilon^\intercal \nabla_\theta f(\theta, \xi) + c^2 \sum_{|\alpha|_1=3} h'_\alpha(\theta + c\epsilon)\epsilon^\alpha)\right]\right) \\
&= \frac{1}{L}\,\mathrm{Var}_\epsilon\left[\epsilon(\epsilon^\intercal \nabla_\theta F(\theta) + c \sum_{|\alpha|_1=3} h_\alpha(\theta + c\epsilon)\epsilon^\alpha)\right] \\
&\quad + \frac{1}{LN}\mathbb{E}_\epsilon\left(\mathrm{Var}_\xi\left[\epsilon(\epsilon^\intercal \nabla_\theta f(\theta, \xi) + c \sum_{|\alpha|_1=3} h'_\alpha(\theta + c\epsilon)\epsilon^\alpha)\right]\right)
\end{aligned}$$

where the next-to-last equality follows from Taylor's Theorem and h' is some scalar-valued function with the same condition as h. Assuming that the Dominated Convergence Theorem holds again, as $c \to 0$, the limit of $\mathrm{Var}[\nabla_\theta \hat{F}^{AT}(\theta)]$ is the same as that of $\mathrm{Var}[\nabla_\theta \hat{F}^{FD}(\theta)]$.

Thus, the MSE of $\nabla_\theta \hat{F}^{AT}(\theta)$ is the same as that of $\mathrm{Var}[\nabla_\theta \hat{F}^{FD}(\theta)]$. It follows that any algorithm to minimize the MSE of $\nabla_\theta \hat{F}^{AT}(\theta)$ must be the same as an algorithm to minimize the MSE of $\nabla_\theta \hat{F}^{FD}(\theta)$.

C. Experimental details

C.1. Validating theory on linear regression

Computing the gradient of the objective The objective is the squared error loss. For our data model (9), it is

$$\begin{aligned}
F(\theta) &= \mathbb{E}[(y - \theta^\intercal x)^2/2] = \mathbb{E}[(\gamma^\intercal x - \theta^\intercal x + \epsilon)^2/2] \\
&= \mathbb{E}[\gamma^\intercal x x^\intercal \gamma/2 + \theta^\intercal x x^\intercal \theta/2 + \epsilon^2/2 - \gamma^\intercal x x^\intercal \theta - \theta^\intercal x\epsilon + \gamma^\intercal x\epsilon] \\
&\equiv \mathbb{E}[\theta^\intercal x x^\intercal \theta/2 - \gamma^\intercal x x^\intercal \theta - \theta^\intercal x\epsilon] \\
&= \mathbb{E}[\theta^\intercal \mathbf{Q}\theta/2 - \gamma^\intercal \mathbf{Q}\theta] = \theta^\intercal \mathbb{E}[\mathbf{Q}]\theta/2 - \mathbb{E}[\gamma^\intercal \mathbf{Q}]\theta
\end{aligned}$$

where in the third line we have ignored terms that do not include θ.

The gradient of the objective is $\nabla_\theta F(\theta) = \mathbb{E}(\mathbf{Q})\theta - \mathbb{E}(\mathbf{Q}\gamma) = \theta - \mathbb{E}(\mathbf{Q}\gamma)$, since

$$\begin{aligned}
\mathbb{E}(\mathbf{Q}) &= \mathbb{E}_\mathbf{V}[\mathbb{E}_\gamma(\mathbf{V}\,\mathrm{diag}(\gamma)\mathbf{V}^\intercal \mid \mathbf{V})] = \mathbb{E}_\mathbf{V}[\mathbf{V}\mathbb{E}_\gamma(\mathrm{diag}(\gamma))\mathbf{V}^\intercal] \\
&= \mathbb{E}_\mathbf{V}[\mathbf{V}\mathbf{V}^\intercal] = \mathbf{I} \quad \text{since } \mathbb{E}(\gamma) \text{ is a vector of ones } 1_d \text{ and } \mathbf{V} \text{ is orthogonal}
\end{aligned}$$

Prior to the start of optimization, $\mathbb{E}(\mathbf{Q}\gamma)$ is estimated via Monte Carlo with 1000 samples. The estimate is plugged into $\nabla_\theta F(\theta)$ to obtain the gradient of the objective.

Optimization and testing We first sample 1000 data points from (9) to serve as the test set and initialize the parameters θ by sampling from $\mathcal{N}(0, \mathbf{I})$. There are 100 rounds. Each round consists of i) 10 optimization iterations of SGD with the gradient estimated from (4) on N newly sampled data points from (9) and L newly sampled directions ϵ_l ii) computation of

Table 3. Hyperparameters for GS and BeS in linear regression.

(L, N)	c	LEARNING RATE
$(2, 5)$	0.01	0.001
$(6, 5)$	0.01	0.001
$(20, 5)$	0.01	0.001
$(2, 15)$	0.01	0.001
$(6, 15)$	0.01	0.001
$(20, 15)$	0.01	0.01
$(2, 50)$	0.01	0.001
$(6, 50)$	0.01	0.01
$(20, 50)$	0.01	0.01

(L, N)	c	LEARNING RATE
$(2, 5)$	0.01	0.001
$(6, 5)$	0.1	0.001
$(20, 5)$	0.01	0.001
$(2, 15)$	0.01	0.001
$(6, 15)$	0.1	0.001
$(20, 15)$	0.01	0.01
$(2, 50)$	0.01	0.001
$(6, 50)$	0.01	0.01
$(20, 50)$	0.01	0.01

Table 4. Hyperparameters for GS-shrinkage and BeS-shrinkage in linear regression.

(L, N)	c	LEARNING RATE
$(2, 5)$	0.01	0.01
$(6, 5)$	0.1	0.01
$(20, 5)$	0.01	0.01
$(2, 15)$	0.01	0.1
$(6, 15)$	0.01	0.1
$(20, 15)$	0.1	0.01
$(2, 50)$	0.01	0.1
$(6, 50)$	0.1	0.1
$(20, 50)$	0.01	0.1

(L, N)	c	LEARNING RATE
$(2, 5)$	0.01	0.01
$(6, 5)$	0.1	0.01
$(20, 5)$	0.01	0.01
$(2, 15)$	0.01	0.1
$(6, 15)$	0.1	0.1
$(20, 15)$	0.1	0.01
$(2, 50)$	0.01	0.1
$(6, 50)$	0.01	0.1
$(20, 50)$	0.01	0.1

the squared error loss over the test set. The MSE of the gradient estimate is computed at each iteration and the average is taken over the 10 iterations per round. Note that $f(\theta, \xi_i)$ is the squared error loss on data point i.

Hyperparameter search We ran this experiment for $L = \{2, 6, 20\}$ and $N = \{5, 15, 50\}$, and a selection was shown in the main paper due to space constraints. Hyperparameters are the spacing c, chosen from $\{0.01, 0.1\}$, and the SGD learning rate η, chosen from $\{0.001, 0.01, 0.1\}$. The values chosen are the ones that minimize the test loss at the end of the 100 rounds, averaged over 3 randomly generated seeds different from those used in Figure 1. Tables 3 and 4 show the chosen hyperparameters for each algorithm and combination of L and N.

Full results Figures 5 and 6 contain the complete results for the MSE of the gradient estimate and test loss, respectively. We see that in all cases, the MSE of the gradient is substantially smaller for GS-shrinkage and BeS-shrinkage than GS and BeS. The story is less clear for the test loss, but usually the test loss of GS-shrinkage and BeS-shrinkage is lower than that of GS and BeS. See Section 5.1 for further discussion.

C.2. Comparing to baselines on RL

Baselines Orthogonal ES is the same as GS, but with an application of the Gram-Schmidt process to the directions after they are sampled. Guided ES samples directions from the distribution $\mathcal{N}(0, \mathbf{\Sigma})$, where $\mathbf{\Sigma} = \frac{\alpha}{d}\mathbf{I} + \frac{1-\alpha}{k}\mathbf{U}\mathbf{U}^\intercal$ and $\mathbf{U}$ is an orthonormal basis for the k previous gradient estimates; computing the basis also requires the Gram-Schmidt process. Following recommendations in Maheswaranathan et al. (2019) and Sener & Koltun (2019), we set $\alpha = 0.5$ and $k = 50$ and let $\alpha = 1$ for the first k iterations.

Optimization and testing Our code roughly follows the same structure as Mania et al. (2018), parallelizing trajectory generation and standardizing the observations. The parameters of the linear policy θ is initialized at zero. There are 100 rounds, each consisting of 10 optimization iterations and one test step. In more detail, every optimization iteration has the following steps:

1. Sample L directions ϵ_l.
2. For each direction, reinitialize the environment, generate one trajectory using the parameters $\theta + c\epsilon_l$ and another using the parameters θ. For those environments with a survival bonus, remove it.

Table 5. c and learning rate for Ant.

ALGORITHM	$L = 2$	$L = 6$	$L = 20$
GS	0.1	0.1	0.1
BES	0.1	0.1	0.1
GS-SHRINKAGE	0.1	0.1	0.01
BES-SHRINKAGE	0.1	0.01	0.01
ORTHOGONAL ES	0.1	0.1	0.1
GUIDED ES	0.1	0.1	0.1

ALGORITHM	$L = 2$	$L = 6$	$L = 20$
GS	0.0001	0.0001	0.0001
BES	0.0001	0.0001	0.0001
GS-SHRINKAGE	0.001	0.001	0.0001
BES-SHRINKAGE	0.001	0.0001	0.0001
ORTHOGONAL ES	0.0001	0.0001	0.0001
GUIDED ES	0.0001	0.0001	0.0001

Table 6. c and learning rate for Walker2d.

ALGORITHM	$L = 2$	$L = 6$	$L = 20$
GS	0.01	0.01	0.01
BES	0.1	0.01	0.01
GS-SHRINKAGE	0.1	0.01	0.01
BES-SHRINKAGE	0.1	0.01	0.1
ORTHOGONAL ES	0.01	0.01	0.01
GUIDED ES	0.1	0.1	0.01

ALGORITHM	$L = 2$	$L = 6$	$L = 20$
GS	0.0001	0.0001	0.0001
BES	0.001	0.0001	0.0001
GS-SHRINKAGE	0.0001	0.0001	0.0001
BES-SHRINKAGE	0.001	0.0001	0.01
ORTHOGONAL ES	0.0001	0.0001	0.0001
GUIDED ES	0.001	0.0001	0.0001

3. Using the $2L$ rewards, compute the gradient estimate (4) with $N = 1$, dividing by the standard deviation of the rewards.
4. Take a gradient ascent step on θ with learning rate η.

and each test step has the following steps:

1. For 1000 trials: Reinitialize the environment and generate a trajectory. Record the total reward.
2. Compute the average and standard deviation of the reward over the trials.

Hyperparameter search Ant and Walker2d have horizon 1000, ML1-Reach 150, and HalfCheetahRandVel 200. We ran this experiment for $L = \{2, 6, 20\}$. Hyperparameters are the spacing c, chosen from $\{0.01, 0.1\}$, and the learning rate η, chosen from $\{0.0001, 0.001, 0.01\}$. The values chosen are the ones that maximize the test reward at the end of the 100 rounds, averaged over 3 randomly generated seeds different from those used in Figure 2. Tables 5 – 8 show the chosen hyperparameters for each algorithm in the four environments discussed in the main paper.

Additional environments We conducted the above experiment on two additional environments, Hopper and ML1-Push. Hopper is another locomotion environment, similar to Ant and Walker2d, and ML1-Push is a meta-RL manipulation environment similar to ML1-Reach, where the goal is to push an object to some location. The selected hyperparameters are given in Tables 9 and 10 and the plots of the test reward against the number of generated trajectories during optimization are given in Figure 7. For $L = 2$ and $L = 6$, the qualitative results are similar to ML1-Reach; overall BeS is the best algorithm, although the standard errors are very large. For $L = 20$, GS outperforms the other algorithms. We suspect that this change may be due to the fact that standardizing the rewards during optimization and searching over a grid of learning rates compensates for errors in the magnitude of the gradient estimate, and thus gives a bigger advantage to GS.

C.3. Ablations

The experimental setup is the same as described for the main online RL experiments, in Appendix C.2. For brevity, we restrict to the Ant environment and set $L = 20$. The selected hyperparameters are given in the two parts of Table 11, for i) a MLP policy instead of linear policy and ii) antithetic gradient estimator instead of forward difference gradient estimator.

C.4. Comparing to baselines on DFO benchmarks

Optimization and testing The parameters are initialized by sampling from $\mathcal{N}(0, \mathbf{I})$. There are 100 rounds, each consisting of 10 optimization iterations and one test step. In more detail, every optimization iteration has the following steps:

Table 7. c and learning rate for HalfCheetahRandVel.

ALGORITHM	$L = 2$	$L = 6$	$L = 20$
GS	0.1	0.01	0.01
BES	0.01	0.01	0.01
GS-SHRINKAGE	0.1	0.1	0.01
BES-SHRINKAGE	0.1	0.1	0.01
ORTHOGONAL ES	0.1	0.01	0.01
GUIDED ES	0.1	0.1	0.1

ALGORITHM	$L = 2$	$L = 6$	$L = 20$
GS	0.001	0.0001	0.0001
BES	0.0001	0.0001	0.0001
GS-SHRINKAGE	0.001	0.01	0.001
BES-SHRINKAGE	0.001	0.01	0.001
ORTHOGONAL ES	0.001	0.0001	0.0001
GUIDED ES	0.001	0.01	0.01

Table 8. c and learning rate for ML1-Reach.

ALGORITHM	$L = 2$	$L = 6$	$L = 20$
GS	0.1	0.1	0.01
BES	0.1	0.1	0.1
GS-SHRINKAGE	0.1	0.1	0.1
BES-SHRINKAGE	0.1	0.1	0.01
ORTHOGONAL ES	0.1	0.1	0.1
GUIDED ES	0.1	0.1	0.1

ALGORITHM	$L = 2$	$L = 6$	$L = 20$
GS	0.001	0.001	0.0001
BES	0.001	0.001	0.01
GS-SHRINKAGE	0.001	0.001	0.01
BES-SHRINKAGE	0.001	0.001	0.0001
ORTHOGONAL ES	0.001	0.001	0.01
GUIDED ES	0.001	0.001	0.01

1. Sample L directions ϵ_l.
2. For each direction, obtain a noisy evaluation at the parameters $\theta + c\epsilon_l$ and another at the parameters θ.
3. Using those $2L$ numbers, compute the gradient estimate (4) with $N = 1$.
4. Take a gradient descent step on θ with learning rate η.

and at each test step, compute the objective at the current parameters.

Hyperparameter search We ran this experiment for $L = \{10, 100\}$ and $N = 1$, setting the noise level in Nevergrad to 0.1. Hyperparameters are the spacing c, chosen from $\{0.01, 0.1\}$, and the SGD learning rate η, chosen from $\{0.000001, 0.00001, 0.0001, 0.001, 0.01\}$. The values chosen are the ones that minimize the objective at the end of the 100 rounds, averaged over 3 randomly generated seeds different from those used in Figure 4. Tables 12 – 14 show the chosen hyperparameters for each algorithm in the three objectives discussed in the main paper.

Table 9. c and learning rate for Hopper.

ALGORITHM	$L = 2$	$L = 6$	$L = 20$
GS	0.1	0.01	0.01
BES	0.1	0.01	0.01
GS-SHRINKAGE	0.1	0.1	0.01
BES-SHRINKAGE	0.1	0.1	0.01
ORTHOGONAL ES	0.1	0.01	0.01
GUIDED ES	0.1	0.1	0.1

ALGORITHM	$L = 2$	$L = 6$	$L = 20$
GS	0.001	0.0001	0.0001
BES	0.001	0.0001	0.0001
GS-SHRINKAGE	0.001	0.01	0.0001
BES-SHRINKAGE	0.001	0.001	0.0001
ORTHOGONAL ES	0.001	0.0001	0.0001
GUIDED ES	0.001	0.01	0.001

Table 10. c and learning rate for ML1-Push.

ALGORITHM	$L = 2$	$L = 6$	$L = 20$
GS	0.1	0.1	0.1
BES	0.1	0.1	0.1
GS-SHRINKAGE	0.1	0.1	0.1
BES-SHRINKAGE	0.1	0.1	0.1
ORTHOGONAL ES	0.1	0.1	0.1
GUIDED ES	0.1	0.1	0.1

ALGORITHM	$L = 2$	$L = 6$	$L = 20$
GS	0.001	0.001	0.01
BES	0.001	0.001	0.01
GS-SHRINKAGE	0.001	0.001	0.01
BES-SHRINKAGE	0.001	0.001	0.01
ORTHOGONAL ES	0.001	0.001	0.01
GUIDED ES	0.001	0.001	0.01

Table 11. Selected hyperparameters for Ant, $L = 20$, with MLP policy (left) or antithetic gradient estimator (right).

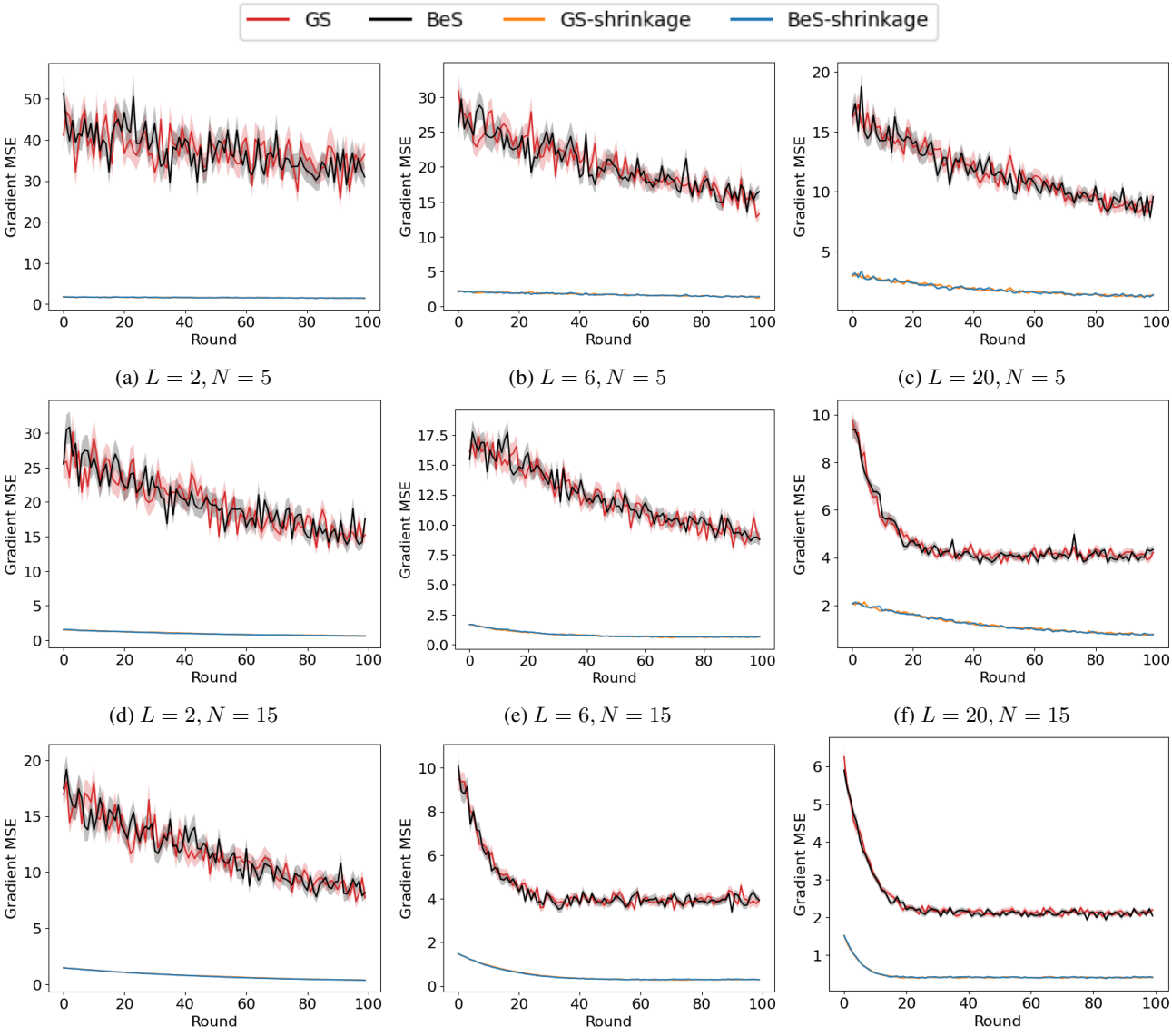
ALGORITHM	c	LEARNING RATE
GS	0.01	0.0001
BES	0.01	0.0001
GS-SHRINKAGE	0.1	0.01
BES-SHRINKAGE	0.1	0.01
ORTHOGONAL ES	0.01	0.0001
GUIDED ES	0.01	0.0001

ALGORITHM	c	LEARNING RATE
GS	0.1	0.0001
BES	0.1	0.0001
GS-SHRINKAGE	0.01	0.0001
BES-SHRINKAGE	0.01	0.0001
ORTHOGONAL ES	0.1	0.0001
GUIDED ES	0.1	0.0001

Table 12. c and learning rate for *sphere*.

ALGORITHM	$d = 10, L = 1$	$d = 100, L = 10$	$d = 100, L = 100$
GS	0.1	0.1	0.1
BES	0.1	0.1	0.1
GS-SHRINKAGE	0.1	0.1	0.1
BES-SHRINKAGE	0.1	0.1	0.1
ORTHOGONAL ES	0.1	0.1	0.1
GUIDED ES	0.1	0.1	0.1

ALGORITHM	$d = 10, L = 1$	$d = 100, L = 10$	$d = 100, L = 100$
GS	0.001	0.001	0.001
BES	0.001	0.001	0.001
GS-SHRINKAGE	0.01	0.001	0.001
BES-SHRINKAGE	0.01	0.001	0.001
ORTHOGONAL ES	0.001	0.001	0.001
GUIDED ES	0.001	0.001	0.01

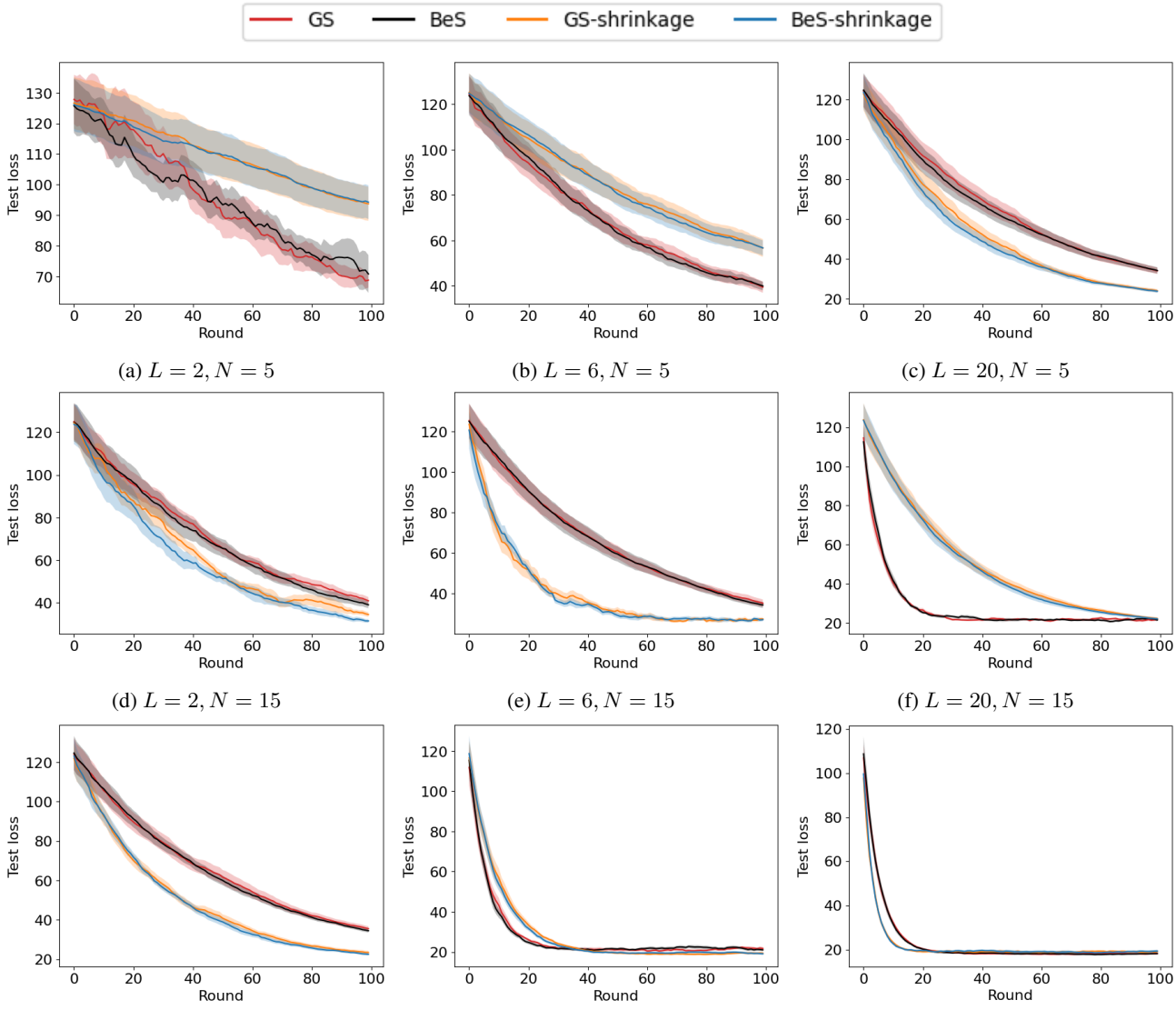


(a) $L = 2, N = 5$ (b) $L = 6, N = 5$ (c) $L = 20, N = 5$

(d) $L = 2, N = 15$ (e) $L = 6, N = 15$ (f) $L = 20, N = 15$

(g) $L = 2, N = 50$ (h) $L = 6, N = 50$ (i) $L = 20, N = 50$

Figure 5. For linear regression with various L and N: MSE of the gradient at each round averaged over the 10 iterations.

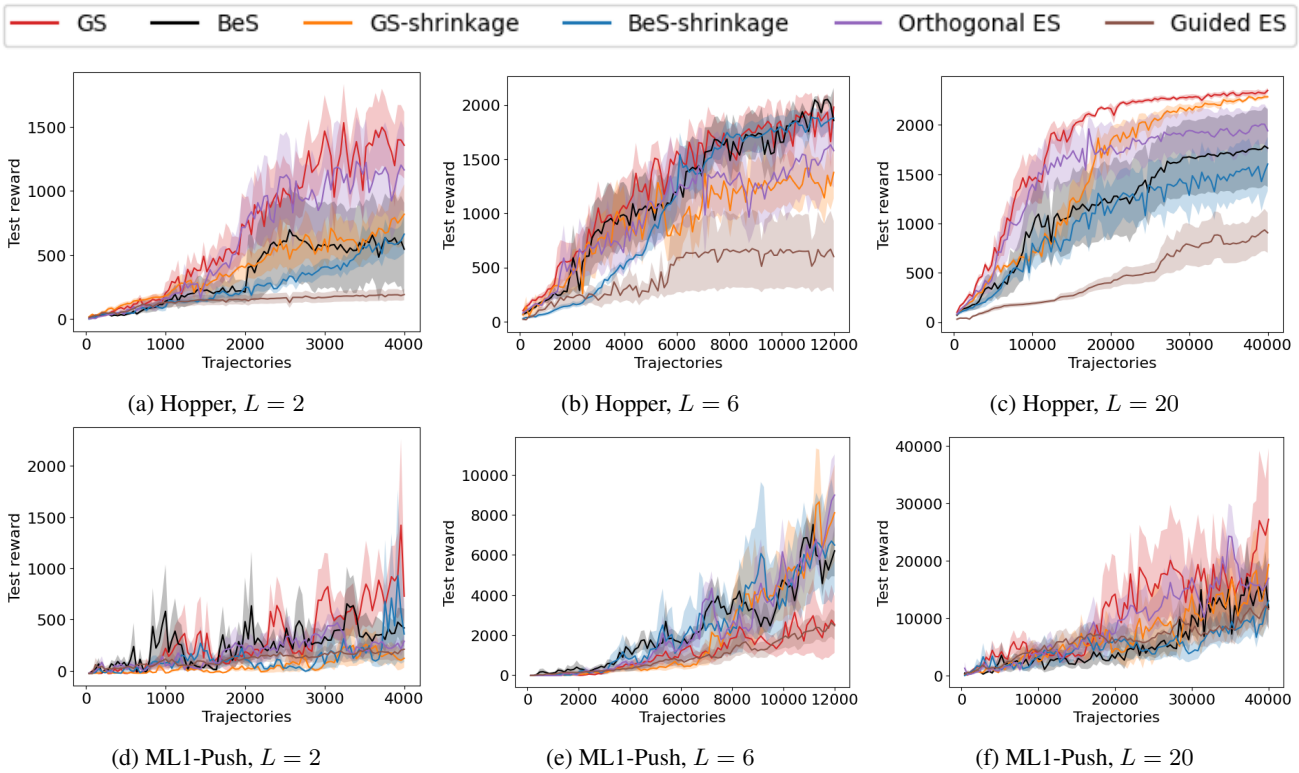


(a) $L = 2, N = 5$ (b) $L = 6, N = 5$ (c) $L = 20, N = 5$

(d) $L = 2, N = 15$ (e) $L = 6, N = 15$ (f) $L = 20, N = 15$

(g) $L = 2, N = 50$ (h) $L = 6, N = 50$ (i) $L = 20, N = 50$

Figure 6. For linear regression with various L and N: Test loss at each round.



(a) Hopper, $L = 2$ (b) Hopper, $L = 6$ (c) Hopper, $L = 20$

(d) ML1-Push, $L = 2$ (e) ML1-Push, $L = 6$ (f) ML1-Push, $L = 20$

Figure 7. RL with various L, additional environments

Table 13. c and learning rate for *rosenbrock*.

ALGORITHM	$d = 10, L = 1$	$d = 100, L = 10$	$d = 100, L = 100$
GS	0.1	0.1	0.1
BES	0.1	0.1	0.1
GS-SHRINKAGE	0.1	0.1	0.1
BES-SHRINKAGE	0.1	0.1	0.1
ORTHOGONAL ES	0.1	0.1	0.1
GUIDED ES	0.1	0.1	0.1

ALGORITHM	$d = 10, L = 1$	$d = 100, L = 10$	$d = 100, L = 100$
GS	0.000001	0.000001	0.000001
BES	0.000001	0.000001	0.000001
GS-SHRINKAGE	0.000001	0.000001	0.000001
BES-SHRINKAGE	0.000001	0.000001	0.000001
ORTHOGONAL ES	0.000001	0.000001	0.000001
GUIDED ES	0.000001	0.000001	0.00001

Table 14. c and learning rate for *hm*.

ALGORITHM	$d = 10, L = 1$	$d = 100, L = 10$	$d = 100, L = 100$
GS	0.1	0.1	0.1
BES	0.1	0.1	0.1
GS-SHRINKAGE	0.1	0.1	0.1
BES-SHRINKAGE	0.1	0.1	0.1
ORTHOGONAL ES	0.1	0.1	0.1
GUIDED ES	0.1	0.1	0.1
ALGORITHM	$d = 10, L = 1$	$d = 100, L = 10$	$d = 100, L = 10$
GS	0.001	0.0001	0.001
BES	0.001	0.0001	0.001
GS-SHRINKAGE	0.001	0.0001	0.001
BES-SHRINKAGE	0.001	0.0001	0.001
ORTHOGONAL ES	0.001	0.0001	0.001
GUIDED ES	0.001	0.0001	0.001